IMA Journal of Numerical Analysis (2023) **43**, 1653–1684
https://doi.org/10.1093/imanum/drac018
Advance Access publication on 25 May 2022

A manifold inexact augmented Lagrangian method for nonsmooth optimization on Riemannian submanifolds in Euclidean space

Kangkang Deng
Beijing International Center for Mathematical Research, Peking University, Beijing 100871, China

and

Zheng Peng*
School of Mathematics and Computational Science, Xiangtan Univeristy, Xiangtan 411105, China, and National Center for Applied Mathematics in Hunan, Xiangtan 411105, China
*Corresponding author: pzheng@xtu.edu.cn

[Received on 28 January 2021; revised on 31 March 2022]

We develop a manifold inexact augmented Lagrangian framework to solve a family of nonsmooth optimization problem on Riemannian submanifold embedding in Euclidean space, whose objective function is the sum of a smooth function (but possibly nonconvex) and a nonsmooth convex function in Euclidean space. By utilizing the Moreau envelope, we get a smoothing Riemannian minimization subproblem at each iteration of the proposed method. Consequentially, each iteration subproblem is solved by a Riemannian Barzilai–Borwein gradient method. Theoretically, the convergence to critical point of the proposed method is established under some mild assumptions. Numerical experiments on compressed modes problems in physic and sparse principal component analysis demonstrate that the proposed method is a competitive method compared with some state-of-the-art methods.

Keywords: nonsmooth optimization; Riemannian manifold constraint; augmented Lagrangian method; Moreau envelope.

1. Introduction

Riemannian optimization is concerned with optimizing a real-value function over a nonlinear constraint endowed with a manifold structure $\mathscr{M}$. The area has recently aroused considerable research interests due to the wide applications in different fields, such as computer vision, signal processing, etc. (Absil *et al.*, 2009). In these applications, manifold $\mathscr{M}$ could be Stiefel manifold, Grassmann manifold or symmetric positive definite manifold and so on. Many classical optimization methods in Euclidean space have been extended to Riemannian optimization problem, e.g., gradient-type methods (Absil *et al.*, 2009; Zhang & Sra, 2016; Boumal *et al.*, 2018), Newton-type methods (Ferreira & Silva, 2012; Huang *et al.*, 2015b; Yuan *et al.*, 2017; Bortoloti *et al.*, 2020) and trust region methods (Absil *et al.*, 2007; Baker *et al.*, 2008; Boumal, 2015; Huang *et al.*, 2015a). However, most of the existing works focus on the case when objective function is smooth, nonsmooth Riemannian optimization problem is less explored, but has drawn increasing attention in recent years; see Absil & Hosseini (2019) for an example.

In this paper, we consider a nonconvex nonsmooth Riemannian optimization problem as follows:

$$\begin{cases} \min_{X\in\mathbb{E}} F(X) := f(X) + g(\mathcal{A}X) \\ \text{s.t. } X \in \mathcal{M}, \end{cases} \tag{1.1}$$

where $\mathscr{M}$ is a Riemannian submanifold embedding in a Euclidean space $\mathbb{E}$, $f : \mathscr{M} \to \mathbb{R}$ is a smooth but possibly nonconvex function and $g : \mathbb{R}^m \to \mathbb{R}$ is convex but nonsmooth in usual Euclidean space, $\mathscr{A} : \mathbb{E} \to \mathbb{R}^m$ is a linear operator. We assume throughout this paper that, the proximal mapping of g can be cheaply evaluated.

Many convex or nonconvex problems in real-world applications have the form of problem (1.1), e.g., sparse principal component analysis (SPCA) (Zou *et al.*, 2006), compressed modes (CMs) problem (Ozolinš *et al.*, 2013), robust low-rank matrix completion (Cambier & Absil, 2016) and multi-antenna channel communications (Zheng & Tse, 2002; Gohary & Davidson, 2009), etc. Absil & Hosseini (2019) presented many examples of manifold optimization with nonsmooth objective.

In this paper, we will propose a manifold inexact augmented Lagrangian method (MIALM) for solving problem (1.1). We first reformulate problem (1.1) to a separable form and then develop a manifold inexact augmented Lagrangian framework to the resulting separable optimization problem. By utilizing the Moreau envelope technique, the iteration subproblem owning the main computational cost in the proposed method is formulated to a smooth optimization problem on Riemannian manifold, and then a Riemannian gradient method is applied on the smooth subproblem. Our main contributions in this paper are summarized as follows:

(1) By utilizing the Moreau envelope technique, we reformulate the nonsmooth iteration subproblem in the proposed method to a smooth one, and consequentially, it can be solved by some classical Riemannian optimization methods, such as Riemannian gradient/Newton/quasi-Newton method.

(2) The convergence to critical point of the proposed manifold inexact augmented Lagrangian framework is established under some mild assumptions.

(3) Numerical experiments on CMs and SPCA problems show that the proposed MIALM is competitive compared with some existing methods.

Notations: Let $\mathscr{M}$ be a Riemannian submanifold embedded in a Euclidean space $\mathbb{E}$, $\Im_x(\mathscr{M})$ denotes the set of all real-valued functions f defined in a neighborhood of x in $\mathscr{M}$. Given a real-valued function f and a point $x \in \mathscr{M}$, we use $\nabla f(x)$ $(\nabla^2 f(x))$ and $\mathrm{grad} f(x)$ $(\mathrm{Hess} f(x))$ to denote the Euclidean and Riemannian gradient (Hessian) of f, respectively. The Euclidean and Riemannian Clarke subdifferential of f at $x \in \mathscr{M}$ are denoted by $\partial f(x)$ and $\partial_R f(x)$. As usual, f^* is the Fenchel conjugate of an arbitrary function f. We use $\|\cdot\|_1, \|\cdot\|_2, \|\cdot\|_\infty$ denote the $\ell_1, \ell_2, \ell_\infty$ norm in the usual sense. Other notations will be defined when they occur.

Organization: The rest of this paper is organized as follows. Some related works on nonsmooth manifold optimization problem are summarized in Section 2, and some preliminaries on manifold optimization are given in Section 3. In Section 4, an MIALM is proposed and the iteration subproblem solver is also presented. The convergence to critical point of the proposed manifold inexact augmented Lagrangian framework is established in Section 5. Numerical results on CMs problems in physics and SPCA are reported in Section 6. Finally, Section 7 concludes this paper with some remarks.

2. Related works

2.1 *Some existing methods for nonsmooth manifold optimization problem*

The existing methods for nonsmooth manifold optimization problem are mainly focused on three categories as follows: subgradient-oriented methods, proximal point methods and operator-splitting methods.

The subgradient-oriented methods require subgradient information for finding the search direction. Grohs & Hosseini (2016) proposed the ϵ-subgradient algorithm for minimizing a locally Lipschitz function on Riemannian manifold. By utilizing ϵ-subgradient-oriented descent direction and the generalized Wolfe line-search on Riemannian manifold, Hosseini *et al.* (2018) presented a nonsmooth Riemannian line-search algorithm and established the convergence to a stationary point. Grohs & Hosseini (2015) presented a nonsmooth trust-region algorithm for minimizing locally Lipschitz objective function on Riemannian manifold. The iteration complexity of these subgradient algorithms was also investigated by Bento *et al.* (2017) and Ferreira *et al.* (2019). Hosseini & Uschmajew (2017) and Burke *et al.* (2020) proposed the Riemannian gradient sampling algorithms. At each iteration of these Riemannian gradient sampling methods, the subdifferential of the objective function is approximated by the convex hull of transported gradients of nearby points, and the nearby points are randomly generated in the tangent space of the current iterate.

Proximal point algorithms on Riemannian manifold has attracted much research attention in the past few years. Bento *et al.* (2017) analyzed the iteration complexity of a proximal point algorithm on Hadamard manifold with nonpositive sectional curvature. Without assumption on the sign of the sectional curvature on manifold, de Carvalho Bento *et al.* (2016) established the global convergence of any bounded sequence generated by the proximal point method. The Kurdyka–Łojasiewicz property on Riemannian manifold is a powerful tool for convergence analysis of manifold optimization methods. Bento *et al.* (2011) analyzed the global convergence of a steepest descent method and a proximal point method via Kurdyka–Łojasiewicz property. Hosseini (2015) proposed a subgradient-oriented descent method, and proved that, if the objective function has the Kurdyka–Łojasiewicz property, the sequence generated by the subgradient-oriented descent method converges to a singular critical point.

If $g(\cdot)$ in problem (1.1) is a regularization, function and the corresponding proximal operator is easy to be obtained, then there exist several operator splitting methods for problem (1.1). By introducing an auxiliary variable, the manifold constrained and the nonsmooth term can be handled separately. To do so will result in two easy subproblems in the operator splitting methods. The splitting of orthogonality constraints (SOCs) method (Lai & Osher, 2014) is proposed for solving a special case of problem (1.1), in which $\mathscr{A} = \mathscr{I}$ is the identity operator and $\mathscr{M} = St_n$ is a Stiefel manifold. That is,

$$\min_X f(X) + g(X) \text{ s.t. } X \in St_n. \tag{2.1}$$

For problem (2.1), the SOC method considered the following separable reformulation:

$$\min_{X,Y,Q} f(Y) + g(Q) \text{ s.t. } X = Y, X = Q, X \in St_n. \tag{2.2}$$

The partial augmented Lagrangian function associated with problem (2.2) is

$$\mathcal{L}_\beta := f(Y) + g(Q) - \langle \Lambda_1, X - Y \rangle + \frac{\beta}{2}\|X - Y\|_F^2 - \langle \Lambda_2, X - Q \rangle + \frac{\beta}{2}\|X - Q\|_F^2,$$

where Λ_1, Λ_2 is the Lagrangian multiplier and β is a penalty parameter. The SOC method updates the iterate via

$$\begin{cases} X^{k+1} = \arg\min_{X\in St_n} \mathcal{L}_\beta(X, Y^k, Q^k; \Lambda_1^k, \Lambda_2^k), \\ Y^{k+1} = \arg\min_Y \mathcal{L}_\beta(X^{k+1}, Y, Q^k; \Lambda_1^k, \Lambda_2^k), \\ Q^{k+1} = \arg\min_Q \mathcal{L}_\beta(X^{k+1}, Y^{k+1}, Q; \Lambda_1^k, \Lambda_2^k), \\ \Lambda_1^{k+1} = \Lambda_1^k - \beta(X^{k+1} - Y^{k+1}), \Lambda_2^{k+1} = \Lambda_2^k - \beta(X^{k+1} - Q^{k+1}). \end{cases}$$

The X-subproblem is 'easy' via projection on St_n, and the Q-subproblem is often structured in real applications. The Y-subproblem is a classical smoothing optimization problem in Euclidian space and can be solved by an efficient optimization method.

Kovnatsky *et al.* (2016) proposed a manifold alternating direction method of multipliers (MADMM) for solving a separable optimization problem of the form

$$\min_{X,Y} f(X) + g(Y) \text{ s.t. } \mathcal{A}X = Y, X \in \mathcal{M}.$$

The associated partial augmented Lagrangian function is

$$\mathcal{L}_\beta(X, Y; \Lambda) := f(X) + g(Y) - \langle \Lambda, \mathcal{A}X - Y\rangle + \frac{\beta}{2}\|\mathcal{A}X - Y\|_2^2.$$

The MADMM updates the iterate as follows:

$$\begin{cases} X^{k+1} = \arg\min_{X\in\mathcal{M}} \mathcal{L}_\beta(X, Y^k, \Lambda^k), \\ Y^{k+1} = \arg\min_{Y} \mathcal{L}_\beta(X^{k+1}, Y, \Lambda^k), \\ \Lambda^{k+1} = \Lambda^k - \beta(\mathcal{A}X^{k+1} - Y^{k+1}). \end{cases}$$

To the best of our knowledge, the global convergence of both SOC and MADMM keeps open so far.

Chen *et al.* (2016) proposed an augmented Lagrangian method (ALM) to handle problem (1.1) with $\mathscr{A} = \mathscr{I}$ and $\mathscr{M} = St_n$. They also considered the separable problem (2.2). They adopted the ALM of multipliers framework for problem (2.2), which first obtains a solution for primal variable (X, Y, Q) jointed by X, Y and Q and then updates the multipliers. The iterate is produced by

$$\begin{cases} (X^{k+1}, Y^{k+1}, Q^{k+1}) = \arg\min_{X\in St_n, Y, Q} \mathcal{L}_\beta(X, Y, Q; \Lambda_1^k, \Lambda_2^k), \\ \Lambda_1^{k+1} = \Lambda_1^k - \beta(X^{k+1} - Y^{k+1}), \\ \Lambda_2^{k+1} = \Lambda_2^k - \beta(X^{k+1} - Q^{k+1}). \end{cases} \tag{2.3}$$

The subproblem on (X, Y, Q) in (2.3) is intractable; the authors handled it by a proximal alternating minimization method. Hong *et al.* (2017) considered a more general form of problem (1.1) where $\mathscr{M}$ is a generalized orthogonal manifold and proposed an approximate ALM, in which a proximal alternating linearized minimization method is employed for iteration subproblem. Chen *et al.* (2020) proposed a manifold proximal gradient (ManPG) method for problem (1.1) with $\mathscr{A} = \mathscr{I}$, i.e.,

$$\min_X f(X) + g(X), \quad \text{s.t. } X \in \mathcal{M}.$$

At the kth iteration, the search direction D^k of the ManPG is get by solving

$$\begin{cases} \min\limits_D \left\langle D, \operatorname{grad} f(X^k) \right\rangle + \dfrac{\beta}{2}\|D\|_F^2 + g(X^k + D), \\ \text{s.t. } D \in T_{X^k}\mathcal{M}. \end{cases} \tag{2.4}$$

The constraint $D \in T_{X^k}\mathscr{M}$ can be represented by a linear system $\mathscr{A}_k(D) = 0$, where $\mathscr{A}_k$ is a specified linear operator at the kth iteration. Subproblem (2.4) is solved by applying a semismooth Newton method to its KKT system. The next iterate X^{k+1} is then obtained by

$$X^{k+1} = \mathcal{R}_{X^k}(\alpha_k D^k),$$

where $\mathscr{R}$ is a retraction operator; see Definition 3.1 in the next section. In addition, Wang *et al.* (2019) consider an ℓ_1 principal component analysis problem and proposed a novel accelerated version of the proximal alternating maximization method.

2.2 *ALM for manifold optimization*

Another line of work handles a manifold constrained optimization problem via the augmented Lagrangian framework; see Sahin *et al.* (2019) and Liu & Boumal (2020). In particular, Liu & Boumal (2020) proposed an ALM to solve the optimization problem on manifold with extra constraints, which has the form

$$\begin{cases} \min\limits_x & f(x) \\ \text{s.t.} & x \in \mathcal{M} \\ & g_i(x) \le 0 \text{ for } i \in \mathcal{I} = \{1, ..., n\}. \\ & h_j(x) = 0 \text{ forj } \in \mathcal{E} = \{n+1, ..., n+m\}. \end{cases} \tag{2.5}$$

Let $\mathscr{F} = \left\{x : g_i(x) \le 0 \text{ forı } \in \mathscr{I} = \{1, ..., n\};\ h_j(x) = 0 \text{ forj } \in \mathscr{E} = \{n+1, ..., n+m\}\right\}$ and $g(x) = \delta_{\mathscr{F}}(x)$ be the indicator function of set $\mathscr{F}$, i.e., $g(x) = 1$ if $x \in \mathscr{F}$; otherwise, $g(x) = \infty$. Then, problem (2.5) can be formulated to a special case of problem (1.1), in which $g(x)$ is the nonsmooth component in the objective function. They proposed two approaches to problem (2.5): one is a Riemannian ALM and the other is an exact penalty function method on Riemannian manifold. For the first approach, the augmented Lagrangian function associated with problem (2.5) is given by

$$\mathcal{L}_\rho(x, \lambda, \gamma) = f(x) + \frac{\rho}{2}\left(\sum_{j\in\mathcal{E}}\left(h_j(x) + \frac{\gamma_j}{\rho}\right)^2 + \sum_{i\in\mathcal{I}} \max\left\{0, \frac{\lambda_i}{\rho} + g_i(x)\right\}^2\right).$$

The x-subproblem is a smooth manifold optimization problem and solved inexactly in this Riemannian ALM. Similar to this approach, we also use the augmented Lagrangian framework in this paper, but our MIALM is totally different to this Riemannian ALM proposed by Liu & Boumal (2020). The main difference is that the proximal operator and Moreau envelope technique are used to handle the nonsmooth term $g(x)$ in our MIALM, while the Riemannian ALM penalizes all constraints explicitly to the objective function to get an augmented Lagrangian function. For the second approach, the exact penalty function method proposed by Liu & Boumal (2020) firstly reformulates the constrained optimization problem on manifold to an unconstrained optimization problem on manifold, which is as follows:

$$\min_{x\in\mathcal{M}} f(x) + \rho\left(\sum_{i\in\mathcal{I}} \max\{0, g_i(x)\} + \sum_{j\in\mathcal{E}} |h_j(x)|\right).$$

Then, the authors adopted some smoothing techniques for the nonsmooth term (i.e., the second part) in the above exact penalty function. It is different from the problem and methods considered by Liu & Boumal (2020), the objective function of problem (1.1) itself has a nonsmooth component $g(\cdot)$, and in our MIALM, we do not use any smoothing technique to the nonsmooth function $g(\cdot)$. What is more noteworthy is that, by the Moreau envelope technique, our method only needs to solve a smooth manifold optimization subproblem at each iteration, which has the same complexity as the smoothing exact penalty function methods in Liu & Boumal (2020).

Sahin *et al.* (2019) proposed an inexact ALM for a nonconvex and nonsmooth problem with nonlinear constraints, which has the form

$$\min_{x\in\mathbb{R}^d} f(x) + g(x), \quad \text{s.t. } c(x) = 0, \tag{2.6}$$

where $f(\cdot)$ is a smooth nonconvex and $g(\cdot)$ is a proximal-friendly convex function (not necessarily smooth), $c : \mathbb{R}^d \to \mathbb{R}^m$ is a nonlinear operator. If $c(x) = 0$ defines a manifold (for example $c(x) := x^T x - 1 = 0$ defines a Stiefel manifold), problem (2.6) is also a special case of problem (1.1). Sahin *et al.* (2019) employed a general augmented Lagrangian function by penalizing the nonlinear equality-constraint to Lagrangian function of problem (2.6), i.e.,

$$\mathcal{L}_\beta(x, y) = f(x) + \langle c(x), y\rangle + \frac{\beta}{2}\|c(x)\|_2^2 + g(x),$$

which results an unconstrained Lagrangian saddle point problem in Euclidean space. At each iteration, they use a classical optimization method to solve the x-subproblem inexactly. In contrast with their method, we use a partial augmented Lagrangian function for problem (1.1), which keeps the resulting subproblem to a manifold constrained optimization problem. More importantly, our MIALM also keeps the iterate sequence satisfying the manifold constraint via some manifold optimization tools, such as retraction and vector transport, etc.

3. Preliminaries for Riemannian optimization

An n-dimensional manifold $\mathscr{M}$ is a Hausforff and second-countable topological space, which is homeomorphic to the n-dimensional Euclidean space locally via a family of charts. Let (U, φ) be a chart, where U is an open set with $x \in U \subset \mathscr{M}$, and φ is a homeomorphism between U and open set $\varphi(U) \subset \mathbb{R}^n$. A tangent vector ξ_x to $\mathscr{M}$ at x is a mapping such that there exists a curve γ on $\mathscr{M}$ with $\gamma(0) = x$, satisfying

$$\xi_x u := \dot{\gamma}(0)u = \frac{d(u(\gamma(t)))}{dt}\Big|_{t=0} \ \forall u \in \Im_x(\mathcal{M}).$$

The tangent space of $\mathscr{M}$ at $x \in \mathscr{M}$ is denoted by $T_x\mathscr{M}$ and defined as the set of all tangent vectors to $\mathscr{M}$ at x. A Riemannian manifold $(\mathscr{M}, \langle\cdot,\cdot\rangle)$ is a smooth manifold equipped with an inner product $\langle\cdot,\cdot\rangle_x$ at each point $x \in \mathscr{M}$. For $\xi_x \in T_x\mathscr{M}$, we use $\|\xi_x\|_x = \sqrt{\langle\xi_x, \xi_x\rangle_x}$ to denote the norm induced by the Riemannian metric. The Riemannian gradient $\mathrm{grad} f(x) \in T_x\mathscr{M}$ is denoted as the unique tangent vector satisfying $\langle \mathrm{grad} f(x), \xi\rangle_x = df(x)[\xi]$, for all $\xi \in T_x\mathscr{M}$. If $\mathscr{M}$ is a Riemannian submanifold embedded in Euclidean space, we have $\mathrm{grad} f(x) = \mathrm{Proj}_{T_x\mathscr{M}}(\nabla f(x))$, where $\nabla f(x)$ is a Euclidean gradient, $\mathrm{Proj}_{T_x\mathscr{M}}$ is the orthogonal projection operator on to the tangent space $T_x\mathscr{M}$. Let $T\mathscr{M} := \bigcup_{x\in\mathscr{M}} T_x\mathscr{M}$ be the tangent bundle of $\mathscr{M}$. The orthogonal complement of $T_x\mathscr{M}$ is called the normal space to $\mathscr{M}$ at x and denoted by $N_x\mathscr{M}$.

Definition 3.1 (Retraction, Absil *et al.*, 2009). A retraction operator on manifold $\mathscr{M}$ is a smooth mapping $\mathscr{R} : T\mathscr{M} \to \mathscr{M}$, which has the following properties: for a given $x \in \mathscr{M}$, let $\mathscr{R}_x : T_x\mathscr{M} \to \mathscr{M}$ be the retraction $\mathscr{R}$ restricted to $T_x\mathscr{M}$, then

- $\mathscr{R}_x(0_x) = x$, where 0_x is the zero element of $T_x\mathscr{M}$;
- $d\mathscr{R}_x(0_x) = id_{T_x\mathscr{M}}$, where $id_{T_x\mathscr{M}}$ is the identity mapping on $T_x\mathscr{M}$.

Definition 3.2 (Vector Transport; Absil *et al.*, 2009). A vector transport $\mathscr{T}$ is a smooth mapping

$$\mathcal{T} : T\mathcal{M} \oplus T\mathcal{M} \to T\mathcal{M} : (\eta_x, \xi_x) \mapsto \mathcal{T}_{\eta_x}(\xi_x) \in T\mathcal{M}, \forall\, x \in \mathcal{M},$$

which satisfies that

- $\mathscr{T}_{0_x}\xi_x = \xi_x$ holds for $\forall\ \xi_x \in T_x\mathscr{M}$;
- $\mathscr{T}_{\eta_x}(a\xi_x + b\zeta_x) = a\mathscr{T}_{\eta_x}(\xi_x) + b\mathscr{T}_{\eta_x}(\zeta_x)$.

We also use the notation $\mathscr{T}_{x\to y}(\xi_x)$ to indicate $\mathscr{T}_{\eta_x}(\xi_x)$, where η_x is such that $y = \mathscr{R}_x(\eta_x)$.

Definition 3.3 (The Clarke subdifferential on Riemannian manifold;Yang *et al.*, 2014). For a locally Lipschitz continuous function f on $\mathscr{M}$, the Riemannian generalized directional derivative of f at $x \in \mathscr{M}$ on direction $v \in T_x\mathscr{M}$ is given by

$$f^\circ(x; v) = \lim_{y\to x} \sup_{t\downarrow 0} \frac{f \circ \varphi^{-1}(\varphi(y) + t\mathrm{D}\varphi(y)[v]) - f \circ \varphi^{-1}(\varphi(y))}{t},$$

where (φ, U) is a coordinate chart at x. The generalized Riemannian gradient or the Clarke Riemannian subdifferential of f at $x \in \mathscr{M}$ is

$$\partial_R f(x) = \{\xi \in T_x\mathcal{M} : \langle\xi, v\rangle_x \le f^\circ(x; v), \forall v \in T_x\mathcal{M}\}. \tag{3.1}$$

Consider a Riemannian manifold minimization problem of the form

$$\min_{x\in\mathcal{M}} f(x) \text{ s.t. } \quad c_i(x)=0,\ i=1,\cdots,m. \tag{3.2}$$

Let $\varOmega := \{x \in \mathscr{M} : c_i(x) = 0, i = 1\cdots, m\}$. Given $x^* \in \varOmega$, assume that the linear independent constraint qualification (LICQ) holds at x^*, then the normal cone $\mathscr{N}_\varOmega(x^*)$ is defined by Yang *et al.* (2014):

$$\mathcal{N}_\varOmega(x^*) = \left\{ \sum_{i=1}^{m} \lambda_i \operatorname{grad} c_i(x^*) \middle| \lambda \in \mathbb{R}^m \right\}. \tag{3.3}$$

The first-order optimality condition of problem (3.2) can be elaborated via the following lemma.

Lemma 3.4 (Zhang *et al.*, 2020, Proposition 2.7) If $x^* \in \varOmega$, and

$$\partial_R f(x^*) \cap (-\mathcal{N}_\varOmega(x^*)) \neq \emptyset, \tag{3.4}$$

then x^* is a stationary solution of problem (3.2).

4. The MIALM

4.1 *The proposed method, MIALM*

Throughout this paper, we make the following assumptions.

Assumption 4.1

A: Manifold $\mathscr{M}$ is a compact Riemannian submanifold embedded in $\mathbb{E}$.

B: Function f is smooth, but not necessarily convex, g is a nonsmooth convex function and $\partial g(Y)$ is uniformly bounded for all Y, where $\partial g(Y)$ is a subdifferential of g at Y.

C: There exist constants $\alpha > 0$ and $\beta > 0$ such that, for all $X \in \mathscr{M}$ and all $U \in T_X\mathscr{M}$, we have

$$\begin{cases} \|\mathcal{R}_X(U) - X\|_X \le \alpha\|U\|_X, \\ \|\mathcal{R}_X(U) - X - U\|_X \le \beta\|U\|_X^2. \end{cases} \tag{4.1}$$

Remark 4.2 Assumption 4.1 is a standard assumption. Assumption 4.1 holds for usual nonsmooth functions, e.g., $g(X) = \|X\|_1$; see Qu *et al.* (2015). For Assumption 4.1, the constants α and β depend on the manifold and the dimensions. For instance, the polar retraction and the QR retraction on Stiefel manifold satisfy this regularity assumption; see Boumal *et al.* (2018).

By introducing an auxiliary variable $Y = \mathscr{A}X$, problem (1.1) can be reformulated to

$$\min_{X,Y} f(X) + g(Y), \text{ s.t. } \mathcal{A}X = Y,\ X \in \mathcal{M}. \tag{4.2}$$

The partial Lagrangian function associated with problem (4.2) is

$$L(X, Y; Z) := f(X) + g(Y) - \langle Z, \mathcal{A}X - Y\rangle. \tag{4.3}$$

By using Lemma 3.4, the KKT conditions of problem (4.2) can be stated as follows.

PROPOSITION 4.3 (Zhang *et al.*, 2020, Theorem 2.8) Suppose that Assumption 4.1 holds. Then, $(X^*, Y^*) \in \mathscr{M} \times \mathbb{R}^m$ satisfies the KKT conditions of problem (4.2) if there exists a Lagrange multiplier $Z^* \in \mathbb{R}^m$ such that

$$\begin{cases} 0 = & \mathrm{Proj}_{T_{X^*}\mathcal{M}}(\nabla f(X^*) - \mathcal{A}^* Z^*), \\ 0 \in & \partial g(Y^*) + Z^*, \\ 0 = & \mathcal{A}X^* - Y^*, \end{cases} \tag{4.4}$$

where $\mathscr{A}^*$ is the adjoint operator of $\mathscr{A}$.

The augmented Lagrangian function associated with (4.2) is

$$\begin{aligned} \mathcal{L}_\rho(X, Y; Z) &= L(X, Y; Z) + \frac{\rho}{2}\|\mathcal{A}X - Y\|_2^2 \\ &= f(X) + g(Y) - \langle Z, \mathcal{A}X - Y\rangle + \frac{\rho}{2}\|\mathcal{A}X - Y\|_2^2. \end{aligned} \tag{4.5}$$

For a given (X^k, Y^k, Z^k), the manifold augmented Lagrangian method (MALM) produces the next iterate via

$$\begin{cases} (X^{k+1}, Y^{k+1}) & = \arg \min_{X \in \mathcal{M}, Y \in \mathbb{R}^m} \mathcal{L}_\rho(X, Y; Z^k), \\ Z^{k+1} & = Z^k - \rho(\mathcal{A}X^{k+1} - Y^{k+1}). \end{cases} \tag{4.6}$$

We note that the (X, Y)-subproblem in iteration (4.6) is intractable due to coupling on variables X and Y. We need to handle it by some special techniques, the Moreau envelope provides a possible approach.

At the kth iteration for the first subproblem in (4.6), for fixed $\rho > 0$ and Z, we aim to solve

$$\min_{X \in \mathcal{M}, Y \in \mathbb{R}^m} \mathcal{L}_\rho(X, Y; Z). \tag{4.7}$$

Fixed X, the optimal solution $\bar{Y}$ of (4.7) is

$$\bar{Y} = \mathrm{prox}_{g/\rho}\left(\mathcal{A}X - \frac{1}{\rho}Z\right), \tag{4.8}$$

where prox_g is a proximal mapping of g defined as follows

$$\mathrm{prox}_g(v) := \arg\min_y \left\{g(y) + \frac{1}{2}\|y - v\|_2^2\right\}.$$

Denote $\psi_Z(X) := \inf_Y \mathscr{L}_\rho(X,Y;Z)$. Then the optimal solutions $(\bar{X},\bar{Y})$ for (4.7) can be computed as follows:

$$\bar{X} = \arg\min_{X\in\mathcal{M}} \psi_Z(X), \quad \bar{Y} = \mathrm{prox}_{g/\rho}\left(\mathcal{A}\bar{X} - \frac{1}{\rho}Z\right).$$

By this way, the MALM iterate (4.6) can be rewritten to

$$\begin{cases} X^{k+1} = \arg\min_{X\in\mathcal{M}} \psi_{Z^k}(X), \\ Y^{k+1} = \mathrm{prox}_{g/\rho}(\mathcal{A}X^{k+1} - \frac{1}{\rho}Z^k), \\ Z^{k+1} = Z^k - \rho(\mathcal{A}X^{k+1} - Y^{k+1}). \end{cases} \tag{4.9}$$

It is worth noting that the variable Y does not appear explicitly in the X-subproblem of iteration (4.9). However, the X-subproblem in iteration (4.9) has not closed the solution in general, and it may be very expensive to get an exact solution. Fortunately, we only need to compute an approximate solution of X-subproblem at each iteration, and we describe the resulting method as the MIALM. Algorithm 1 summarizes the proposed MIALM in details.

Algorithm 1 Manifold inexact augmented Lagrangian method, MIALM

```
Input: Let $Z_{\min} < Z_{\max}, X^0 \in \mathscr{M}, \bar{Z}^0 \in \mathbb{R}^m$. Given $\epsilon_{\min} \geq 0, \epsilon_0 > 0, \rho_0 > 1, \sigma > 1, 0 < \tau < 1$.
for $k = 0, 1, \cdots$ do
    Produce the next iterate $(X^{k+1}, Y^{k+1})$ via: Get $X^{k+1}$ by inexactly solving
        $$\min_{X\in\mathcal{M}} \psi_{\bar{Z}^k}(X) \tag{4.10}$$
    with a tolerance $\epsilon_k$, where $\{\epsilon_k\}_{k\in\mathbb{N}} \downarrow 0$. And let
        $$Y^{k+1} = \mathrm{prox}_{g/\rho_k}(\mathcal{A}X^{k+1} - \bar{Z}^k).$$
    Update the Lagrangian multiplier $Z^{k+1}$ by
        $$Z^{k+1} = \bar{Z}^k - \rho_k(\mathcal{A}X^{k+1} - Y^{k+1}). \tag{4.11}$$
    Project $Z^{k+1}$ on $\mathscr{B} = \{Z : Z_{\min} \leq Z \leq Z_{\max}\}$ to get $\bar{Z}^{k+1}$.
    Update penalty parameter by
        $$\rho_{k+1} = \begin{cases} \rho_k, & \text{if } \|\mathcal{A}X^{k+1} - Y^{k+1}\|_\infty \leq \tau\|\mathcal{A}X^k - Y^k\|_\infty \\ \sigma\rho_k, & \text{otherwise} \end{cases} \tag{4.12}$$
end for
```

REMARK 4.4

(1) The proposed MIALM is actually an ALM-type method. The complexity of X-subproblem is the same as that of MADMM. However, the proposed MIALM obtains an optimal solution on variable (X, Y) jointed by X and Y, which guarantees the convergence, but the convergence of the MADMM keeps open until now.

(2) Section 4.2 will show that ψ is continuous differentiable on manifold and its derivatives are easy to compute. Thus, all efficient Riemannian optimization methods, such as Riemannian gradient method, Riemannian Newton method, etc., are adoptable for the X-subproblem.

(3) The proposed method is also utilizable for smooth Riemannian optimization problem under set-constrained, in which $g(Y) = \delta_{\Omega}(Y)$ is the indictor function of set Ω, and Ω is the constraint-set.

(4) Only an approximate solution X^{k+1} of subproblem (4.10) in our MIALM is needed for the convergence. The stopping criteria for X-subproblem (4.10) is given by

$$\|\operatorname{grad} \psi_{\bar{Z}^k}(X^{k+1})\|_{X^{k+1}} \leq \epsilon_k,$$

where $\epsilon_k \to 0$ as $k \to \infty$.

4.2 *Riemannian optimization subproblem*

The main computational cost of Algorithm 1 is to solve the subproblem (4.10). Given $\bar{Z}$ and $\rho > 0$, we consider the following minimization problem:

$$\min_{X\in\mathcal{M}} \left\{ \psi(X) := \inf_{Y\in\mathbb{R}^m} \mathcal{L}_\rho(X, Y, \bar{Z}) \right\}. \tag{4.13}$$

By the closed form solution of Y, we reformulate $\psi(X)$ as follows:

$$\begin{aligned}\psi(X) = \ & \inf_Y \mathcal{L}_\rho(X, Y, \bar{Z}) = f(X) + g(\operatorname{prox}_{g/\rho}(\mathcal{A}X - \bar{Z}/\rho)) \\ & + \frac{\rho}{2}\left\|\mathcal{A}X - \bar{Z}/\rho - \operatorname{prox}_{g/\rho}(\mathcal{A}X - \bar{Z}/\rho)\right\|_2^2 - \frac{1}{2\rho}\|\bar{Z}\|_2^2 \\ = \ & f(X) + M_g^{1/\rho}\left(\mathcal{A}X - \frac{1}{\rho}\bar{Z}\right) - \frac{1}{2\rho}\|\bar{Z}\|_2^2. \end{aligned} \tag{4.14}$$

In the second equality, $M_g^{\mu} : \mathbb{E} \to \mathbb{R}$ is the Moreau envelope of g defined by

$$M_g^{\mu}(v) : = \min_y \left\{ g(y) + \frac{1}{2\mu}\|y - v\|_2^2 \right\}. \tag{4.15}$$

The following lemma state that M_g^{μ} is a continuously differentiable function, even if g is not.

Lemma 4.5 (Beck, 2017, Theorem 6.60) Let $g : \mathbb{E} \to \mathbb{R}$ be a proper closed and convex function, and $\mu > 0$. Then M_g^μ is $\frac{1}{\mu}$-smooth in $\mathbb{E}$, and for all $v \in \mathbb{E}$ one has

$$\nabla M_g^\mu(v) = \frac{1}{\mu}(v - \mathrm{prox}_{\mu g}(v)).$$

By the property of the proximal mapping and the Moreau envelope, one can readily conclude that $\psi(X)$ is continuously differentiable and

$$\begin{aligned} \nabla \psi(X) &= \nabla f(X) + \rho \mathcal{A}^* \left(\mathcal{A}X - \frac{1}{\rho}\bar{Z} - \mathrm{prox}_{g/\rho}\left(\mathcal{A}X - \frac{1}{\rho}\bar{Z} \right) \right) \\ &= \nabla f(X) + \rho \mathcal{A}^* \left(\mathrm{prox}_{\rho g^*}\left(\mathcal{A}X - \frac{1}{\rho}\bar{Z} \right) \right). \end{aligned} \tag{4.16}$$

The second equality is followed by the important Moreau identity:

$$\mathrm{prox}_{\mu g}(v) + \mu \mathrm{prox}_{g^*/\mu}(v/\mu) = v,$$

where $\mu > 0$ is a given parameter, g^* is the conjugate function of g defined by

$$g^*(x) := \sup_v \{\langle x, v\rangle - g(v)\}. \tag{4.17}$$

In what follows, we show that ψ is a retraction smooth function with respect to retraction $\mathscr{R}$ in the sense that, for $\forall X \in \mathscr{M}$ and $\forall U \in T_X\mathscr{M}$, one has

$$\psi(\mathcal{R}_X(U)) \leq \psi(X) + \langle \mathrm{grad}\, \psi(X), U\rangle_X + \frac{\bar{L}_\psi}{2}\|U\|_X^2, \tag{4.18}$$

where $\bar{L}_\psi$ is a constant associated with α and β in Assumption 4.1. Lemma 4.6 shows that the connection between the Lipschitz continuity in Euclidean space and the smoothness in Riemannian manifold, for the proof, the readers are referred to Lemma 2.7 in Boumal *et al.* (2018).

Lemma 4.6 Let $\mathbb{E}$ be a Euclidean space and $\mathscr{M}$ be a compact Riemannian submanifold embedding in $\mathbb{E}$. If $f : \mathbb{E} \to \mathbb{R}$ has Lipschitz continuous gradient in the convex hull of $\mathscr{M}$ then there exists a positive constant ℓ_g such that

$$f(\mathcal{R}_x(\eta)) \leq f(x) + \langle \eta, \mathrm{grad} f(x)\rangle_x + \frac{\ell_g}{2}\|\eta\|_x^2 \tag{4.19}$$

holds at $\forall\ \eta \in T_x\mathscr{M}$.

By Lemma 4.6, we get the Riemannian gradient of $\psi(X)$

$$\mathrm{grad}\, \psi(X) = \mathrm{Proj}_{T_X\mathcal{M}}(\nabla \psi(X)).$$

So problem (4.13) can be solved by some Riemannian optimization methods. In this paper, we did not adopt the second-order method because ψ has only continuous gradient: the Riemannian gradient $\operatorname{grad}\psi$ is continuous, but not differentiable. Although the Riemannian semismooth Newton method (de Oliveira & Ferreira, 2020) can be adopted, its convergence analysis is not clear due to ψ being nonconvex in Euclidean space and the monotonicity of $\operatorname{grad}\psi$ cannot be guaranteed. For first-order methods, the Barzilai–Borwein (BB) gradient method is easy to implement and effective in Euclidean space. Thus, we adopt a Riemannian Barzilai–Borwein gradient (RBB) method proposed by Iannazzo & Porcelli (2018) to solve problem (4.13). At the kth iteration, for given X^k, the RBB method get X^{k+1} by

$$X^{k+1} = \mathcal{R}_{X^k}(-\alpha_k g^k), \tag{4.20}$$

where $g^k = \operatorname{grad}\psi(X^k)$ is the Riemannian gradient of ψ at X^k and α_k is a stepsize. Similar to Euclidean case, we choose an appropriate α_k via a line search strategy. Given σ and γ, we try to find the smallest integer l such that

$$\psi(\mathcal{R}_{X^k}(-\sigma^l\alpha_k^{BB}g^k)) \leq \psi(X^k) - \gamma\sigma^l\alpha_k^{BB}\|g^k\|_{X^k}^2, \tag{4.21}$$

where α_k^{BB} is an initial step size. A good initial step size can reduce the number of line searches and improve the efficiency of the method. It is well known that an initial step size computed the BB method can speed up the convergence in Euclidean optimization. In this paper, we choose an RBB step size proposed by Iannazzo & Porcelli (2018), i.e.,

$$\tau_{k+1}^1 = \frac{\langle s^k, s^k\rangle_{X^{k+1}}}{\langle s^k, y^k\rangle_{X^{k+1}}}, \quad \text{or} \quad \tau_{k+1}^2 = \frac{\langle s^k, y^k\rangle_{X^{k+1}}}{\langle y^k, y^k\rangle_{X^{k+1}}}, \tag{4.22}$$

where $y^k := g^{k+1} - \mathscr{T}_{X^k\to X^{k+1}}(g^k)$, $s^k := \mathscr{T}_{X^k\to X^{k+1}}(-\alpha_k g^k)$ and $\mathscr{T}_{X^k\to X^{k+1}}$ denotes an appropriate vector transport. The RBB gradient method for X-subproblem (4.13) is summarized in Algorithm 2. The results on global convergence of the RBB algorithm was established by Theorem 3.1 of Iannazzo & Porcelli (2018).

5. Convergence analysis

We present the global convergence analysis of Algorithm 1 in this section. For convenience in notation, we rewrite problem (4.2) to a standard constrained optimization problem on manifold. Let $\mathcal{N} = \mathcal{M} \times \mathbb{R}^m$ be a product manifold and $W = (X, Y) \in \mathcal{N}$. Then, problem (4.2) can be rewritten to

$$\min_W \ \theta(W), \ \text{s.t.} \ h(W) = 0, \ W \in \mathcal{N}, \tag{5.1}$$

where $\theta(W) = f(X) + g(Y)$ and $h(W) = [\mathscr{A}, -\mathscr{I}]W \in \mathbb{R}^m$ is a linear operator where $\mathscr{I}$ denotes the identity operator. The partial augmented Lagrangian function associated with problem (5.1) is

$$\mathcal{L}_\rho(W; Z) = \theta(W) - \sum_{i=1}^m Z_i[h(W)]_i + \frac{\rho}{2}\sum_{i=1}^m [h(W)]_i^2.$$

Algorithm 2 Proximal point algorithm for penalty problem

```
Given: $X^0 \in \mathscr{M}$, tolerance $\epsilon > 0$, initial step size $\alpha_0^{BB}$. Let $g^0 = \operatorname{grad} \psi(X^0)$, the sufficient decrease parameter $\gamma$ and the step length contraction factor $\sigma \in (0,1)$.
Initialize: k = 0.
while $\|g^k\| \geq \epsilon$ do
  Find the smallest positive integer $h$ that satisfies (4.21) and set $\alpha_k := \sigma^h \alpha_k^{BB}$.
  Compute $X^{k+1} = \mathscr{R}_{X^k}(-\alpha_k g^k)$, and $g^{k+1} = \operatorname{grad} \psi(X^{k+1})$.
  Compute $\tau_{k+1}$ via (4.22), and compute the new step size $\alpha_{k+1}^{BB}$ by
      $\alpha_{k+1}^{BB} = \begin{cases} \min\{\alpha_{\max}, \max\{\alpha_{\min}, \tau_{k+1}\}\} & \text{if} \langle s^k, y^k \rangle_{X^{k+1}} > 0, \\ \alpha_{\max} & \text{otherwise.} \end{cases}$
  Let $k := k + 1$.
end while
```

The KKT conditions of problem (5.1) are given by: if W^* is a solution of (5.1) then there exists a $Z^* \in \mathbb{R}^m$ such that

$$0 \in \partial_R \theta(W^*) - \sum_{i=1}^m \operatorname{grad}(Z_i^*[h(W^*)]_i), \quad h(W^*) = 0, \quad W^* \in \mathcal{N}, \tag{5.2}$$

where $\partial_R \theta(W^*)$ is Riemannian subdifferential of θ at W^*. The KKT system (5.2) is identical to (4.4) because of that $\mathscr{M}$ is a Riemannian submanifold embedded in Euclidean space. Specifically, let $W^* = (X^*, Y^*)$, we obtain the equivalence between (5.2) and (4.4) by

$$\begin{aligned} 0 \in \begin{pmatrix} \operatorname{grad} f(X^*) - \sum_{i=1}^m \operatorname{grad}(Z_i^*[\mathcal{A}X^*]_i) \\ \partial_R g(Y^*) + Z^* \end{pmatrix} &= \begin{pmatrix} \operatorname{grad} f(X^*) - P_{T_{X^*}\mathcal{M}}(\mathcal{A}^* Z^*) \\ \partial_R g(Y^*) + Z^* \end{pmatrix} \\ &= \begin{pmatrix} P_{T_{X^*}\mathcal{M}}(\nabla f(X^*) - \mathcal{A}^* Z^*) \\ \partial g(Y^*) + Z^* \end{pmatrix}, \end{aligned} \tag{5.3}$$

where the latter equation use the fact that Y lies on the linear manifold $\mathbb{R}^m$ and then $\partial_R g(Y^*) = \partial g(Y^*)$. Inspired by Yang *et al.* (2014), the constraint qualifications of problem (5.1) is given by the following definition.

Definition 5.1 (LICQ). LICQs are said to hold at $W^* \in \mathcal{N}$ for problem (5.1) if

$$\left\{ \operatorname{grad}[h(W^*)]_i \,|\, i = 1, 2, \cdots, m. \right\} \text{ are linearly independent in } T_{W^*}\mathcal{N}. \tag{5.4}$$

The following lemma proves that the LICQ (5.4) always holds at $\forall W \in \mathscr{N}$ for our problem (5.1).

LEMMA 5.2 For arbitrary compact Riemannian manifold $\mathscr{M}$, the LICQ condition (5.4) of problem (5.1) always holds at $\forall W \in \mathscr{N} = \mathscr{M} \times \mathbb{R}^m$.

Proof. Let $h_1(x) = \mathscr{A}X \in \mathbb{R}^m, h_2(Y) = -Y \in \mathbb{R}^m$, then $h(W) = h_1(X) + h_2(Y)$ and

$$\operatorname{grad} h(W) = \begin{pmatrix} \operatorname{grad}_X h(W) \\ \operatorname{grad}_Y h(W) \end{pmatrix} = \begin{pmatrix} \operatorname{grad} h_1(X) \\ \operatorname{grad} h_2(Y) \end{pmatrix}.$$

It is sufficient to show that $\{\operatorname{grad}[h_2(Y)]_i, i = 1, 2, \cdots, m\}$ is linearly independent. Notice that Y lies on a linear manifold, i.e., $\operatorname{grad}[h_2(Y)]_i = \nabla[h_2(Y)]_i$. It is obvious that $\{\nabla[h_2(Y)]_i, i = 1, 2, \cdots, m\}$ is linearly independent, and the proof is completed. □

By Remark 4.4(4), step 2 of Algorithm 1 satisfies the following condition: the iterate X^{k+1} is an ϵ_k-stationary point of X-subproblem (4.10), i.e.,

$$\|\operatorname{grad} \psi_{\bar{Z}^k}(X^{k+1})\|_{X^{k+1}} \leq \epsilon_k. \tag{5.5}$$

Given $W = (X, Y)$, and $\xi = (\xi_X, \xi_Y) \in T_W\mathscr{N}$, where $\xi_X \in T_X\mathscr{M}, \xi_Y \in \mathbb{R}^m$. The norm of ξ is defined as $\|\xi\|_W = \|\xi_X\|_X + \|\xi_Y\|_2$. Before presenting our analysis of Algorithm 1, by (5.5), we have the following lemma.

LEMMA 5.3 If X^{k+1} is an approximate solution of problem (4.13) satisfying (5.5), and Y^{k+1} is updated via (4.8) and $W^{k+1} = (X^{k+1}, Y^{k+1})$. Then, there exists ν_k such that

$$\nu^k \in \partial_R \mathcal{L}_{\rho_k}(W^{k+1}; \bar{Z}^k), \text{ and } \|\nu^k\|_{W^{k+1}} \leq \epsilon_k.$$

Proof. By a slight abuse of notation, we write $\partial_{Y,R}\mathscr{L}_{\rho_k}(W^{k+1}; \bar{Z}^k)$ as the Riemannian Clarke subdifferential of $\mathscr{L}_{\rho_k}(W^{k+1}; \bar{Z}^k)$ with respect to Y, then $\partial_R\mathscr{L}$ is given by

$$\partial_R \mathcal{L}_{\rho_k}(W^{k+1}; \bar{Z}^k) = \begin{pmatrix} \operatorname{grad}_X \mathcal{L}_{\rho_k}(W^{k+1}; \bar{Z}^k) \\ \partial_{Y,R}\mathcal{L}_{\rho_k}(W^{k+1}; \bar{Z}^k) \end{pmatrix}.$$

Assume that $\delta^k = \operatorname{grad} \psi_{\bar{Z}^k}(X^{k+1})$ and $\|\delta_k\|_{X^{k+1}} \leq \epsilon_k$, then

$$\begin{aligned}
\delta_k &= P_{T_{X^{k+1}}\mathcal{M}}\left(\nabla \psi_{\bar{Z}^k}(X^{k+1})\right) \\
&= P_{T_{X^{k+1}}\mathcal{M}}\left(\nabla f(X^{k+1}) + \rho \mathcal{A}^T\left(\mathcal{A}X^{k+1} - \bar{Z}^k/\rho_k - \operatorname{prox}_{g/\rho_k}\left(\mathcal{A}X^{k+1} - \frac{1}{\rho_k}\bar{Z}^k\right)\right)\right) \\
&= P_{T_{X^{k+1}}\mathcal{M}}\left(\nabla f(X^{k+1}) + \rho_k \mathcal{A}^T\left(\mathcal{A}X^{k+1} - \bar{Z}^k/\rho_k - Y^{k+1}\right)\right) \\
&= P_{T_{X^{k+1}}\mathcal{M}}\left(\nabla_X \mathcal{L}_{\rho_k}(W^{k+1}; \bar{Z}^k)\right) = \operatorname{grad}_X \mathcal{L}_{\rho_k}(W^{k+1}; \bar{Z}^k).
\end{aligned} \tag{5.6}$$

By (4.8), $Y^{k+1} = \arg\min_{Y\in\mathbb{R}^m} \mathscr{L}_{\rho_k}(X^{k+1}, Y; \bar{Z}^k)$, we have

$$0 \in \partial_Y \mathcal{L}_{\rho_k}(W^{k+1}; \bar{Z}^k). \tag{5.7}$$

Notice that the domain of Y is $\mathbb{R}^m$, which is a linear manifold, (5.7) implies that

$$0 \in \partial_{Y,R} \mathcal{L}_{\rho_k}(W^{k+1}; \bar{Z}^k). \tag{5.8}$$

Combining (5.6) and (5.8), and let $\nu_k = (\delta_k, 0^m)$, we have that

$$\nu_k \in \partial_R \mathcal{L}_{\rho_k}(W^{k+1}; \bar{Z}^k)$$

and $\|\nu_k\|_{W^{k+1}} = \|\delta_k\|_{X^{k+1}} \le \epsilon_k$. The proof is completed. □

THEOREM 5.4 Suppose $\{W^k\}_{k\in\mathbb{N}}$ is a sequence generated by Algorithm 1, Assumption 4.1 and (5.5) hold. Then, sequence $\{W^k\}_{k\in\mathbb{N}}$ has at least one cluster point. Furthermore, if W^* is a cluster point then W^* is a KKT point of problem (5.1).

Proof. To prove the first part of Theorem 5.4, we need firstly to show that sequence $\{W^k\}_{k\in\mathbb{N}}$ is bounded. By Assumption 4.1, $\mathscr{M}$ is a compact submanifold embedded in $\mathbb{E}$, then $\{X^k\}_{k\in\mathbb{N}}$ is bounded. By $Y^{k+1} = \mathrm{prox}_{g/\rho_k}(\mathscr{A}X^{k+1} - \frac{1}{\rho_k}\bar{Z}^k)$, there exists $v^k \in \partial g(Y^{k+1})$ such that

$$v^k - \rho_k\left(\mathcal{A}X^{k+1} - \frac{1}{\rho_k}\bar{Z}^k - Y^{k+1}\right) = 0.$$

Again using Assumption 4.1, $\partial g(Y^{k+1})$ is a bounded set, sequence $\{v^k\}_{k\in\mathbb{N}}$ is bounded. It is obvious that $\bar{Z}^k \in \mathscr{B} = \{Z : Z_{min} \le Z \le Z_{max}\}$ is bounded. Since sequence $\{\rho_k\}_{k\in\mathbb{N}}$ is nondecreasing, we have $\rho_k \ge \rho_0$ $(\forall k \in \mathbb{N})$, deduces that $\{Y^k\}_{k\in\mathbb{N}}$ is bounded. In summary, sequence $\{W^k\}_{k\in\mathbb{N}}$ is bounded.

Next, we show that W^* is a feasible point of (5.1). By the updating rule of $\{W^k\}_{k\in\mathbb{N}}$ in Algorithm 1, we have $W^k \in \mathscr{N}$. If $\{\rho_k\}_{k\in\mathbb{N}}$ is bounded, by the updating rule of ρ_k, there exists a $k_0 \in \mathbb{N}$ such that

$$\|h(W^k)\|_\infty \le \tau\|h(W^{k-1})\|_\infty, \quad \forall k \ge k_0,$$

where $\tau \in (0, 1)$. Which deduces $h(W^*) = 0$. If $\{\rho_k\}$ is unbounded, by Lemma 5.3, we have

$$\nu^k \in \partial_R \mathcal{L}_{\rho_k}(W^{k+1}; \bar{Z}^k), \ \|\nu^k\|_{W^{k+1}} \le \epsilon_k,$$

where $\epsilon_k \downarrow 0$ as $k \to \infty$. There exists $U^k \in \partial_R\theta(W^{k+1})$ such that

$$\begin{aligned}\nu^k &= U^k + \sum_{i=1}^m \mathrm{grad}\left(-\bar{Z}_i^k[h(W^{k+1})]_i + \frac{\rho_k}{2}[h(W^{k+1})]_i^2\right)\\ &= U^k + \sum_{i=1}^m P_{T_{W^{k+1}}\mathcal{N}}\left(\left(-\bar{Z}_i^k + \rho_k[h(W^{k+1})]_i\right)\nabla[h(W^{k+1})]_i\right).\end{aligned} \tag{5.9}$$

Dividing both sides of (5.9) by ρ_k, we get

$$\sum_{i=1}^{m} P_{T_{W^{k+1}}\mathcal{N}}\left(\left(-\bar{Z}_i^k/\rho_k + [h(W^{k+1})]_i\right)\nabla[h(W^{k+1})]_i\right) = (\nu^k - U^k)/\rho_k,$$

where $\{\bar{Z}^k\}$ is bounded, and $\nu^k \downarrow 0$. Note that $\theta(W) = f(X) + g(Y)$ and g is a convex function on $\mathbb{E}$, and

$$\partial_R\theta(W) = \begin{pmatrix} \operatorname{grad} f(X) \\ \partial_R g(Y) \end{pmatrix} = \begin{pmatrix} \operatorname{grad} f(X) \\ \partial g(Y) \end{pmatrix}.$$

Invoked Proposition B.24(b) in Bertsekas (1997), we get the boundedness of $\bigcup_{k\in\mathbb{N}} \partial g(Y^k)$ by the boundedness of $\{Y^k\}$. In addition, $f(X)$ is a smooth function and $\mathscr{M}$ is a compact manifold, which implies Riemannian gradient sequence $\{\operatorname{grad} f(X^k)\}_{k\in\mathbb{N}}$ is bounded. We have that $\bigcup_{k\in\mathbb{N}} \partial_R\theta(W^k)$ is bounded, which deduces $\{U^k\}_{k\in\mathbb{N}}$ is also bounded.

By the boundedness of $\{(W^k, \bar{Z}^k)\}$, there exists $\mathscr{K} \subset \mathbb{N}$ such that $\lim\limits_{k\in\mathscr{K}, k\to\infty} W^k = W^*$. Taking limits as $k \in \mathscr{K}$ going to infinity on (5.9), and using the continuity and differentiability of h, we have

$$\begin{aligned}\sum_{i=1}^{m} \left([h(W^*)]_i\right) \operatorname{grad}[h(W^*)]_i &= \sum_{i=1}^{m} [h(W^*)]_i P_{T_{W^*}\mathcal{N}}(\nabla[h(W^*)]_i) \\ &= \sum_{i=1}^{m} P_{T_{W^*}\mathcal{N}}([h(W^*)]_i \nabla[h(W^*)]_i) = 0.\end{aligned}$$

Note that LICQ holds at W^* by Lemma 5.2, we conclude that $[h(W^*)]_i = 0$ for all i.

Since $\{U^k\}_{k\in\mathbb{N}}$ is bounded, there exists a subsequence $\mathscr{K}_1 \subset \mathscr{K}$ such that $\lim\limits_{k\to\infty, k\in\mathscr{K}_1} U^k = U^*$. Recall that $\lim\limits_{k\to\infty, k\in\mathscr{K}} W^k = W^*$, we get $U^* \in \partial_R\theta(W^*)$ by the closedness property of the limiting subdifferential. Together with $Z_i^{k+1} = \bar{Z}_i^k + \rho_k[h(W^{k+1})]_i$, $\forall\, i$, by (5.9), we have from Algorithm 1 that $\forall\, k \in \mathscr{K}_1$,

$$\begin{aligned}\nu^k &= U^k + \sum_{i=1}^{m} P_{T_{W^{k+1}}\mathcal{N}}\left(Z_i^{k+1}\nabla[h(W^{k+1})]_i\right) \\ &= U^k + \sum_{i=1}^{m} Z_i^{k+1} \operatorname{grad}[h(W^{k+1})]_i, \qquad (5.10)\end{aligned}$$

where $\|\nu^k\|_{W^{k+1}} \le \epsilon_k$ and $U^k \in \partial_R\theta(W^{k+1})$.

We claim that $\{Z^k\}$ is bounded. Otherwise, assume that $\{Z^k\}$ is unbounded, by (5.9), we have

$$\frac{U^k}{\|Z^{k+1}\|_\infty} + \sum_{i=1}^{m}\left(\frac{Z_i^{k+1}}{\|Z^{k+1}\|_\infty}\right)\operatorname{grad}[h(W^{k+1})]_i = \frac{\nu^k}{\|Z^{k+1}\|_\infty}.$$

Since $\frac{Z^{k+1}}{\|Z^{k+1}\|_\infty} \in [-1, 1]$ is bounded, there exists a subsequence $\mathscr{K}_2 \subset \mathscr{K}_1$ such that $\lim\limits_{k\to\infty,k\in\mathscr{K}_2} \frac{Z^{k+1}}{\|Z^{k+1}\|_\infty} = \bar{Z}$, where $\bar{Z}$ is a nonzero matrix. Taking limits as $k \in \mathscr{K}_2$ going to infinity, we obtain

$$\sum_{i=1}^{m} \bar{Z}_i \operatorname{grad}[h(W^*)]_i = 0,$$

which leads contradiction to the LICQ condition at W^*.

Using the boundedness of $\{U^k\}$ and $\{\nu^k\} \downarrow 0$, there exists a subsequence $\mathscr{K}_3 \subset \mathscr{K}_2$ such that $\lim\limits_{k\to\infty,k\in\mathscr{K}_3} U^k = U^*$ and $\lim\limits_{k\to\infty,k\in\mathscr{K}_3} Z^k = Z^*$. By the continuity of mapping grad h, and taking limits as $k \in \mathscr{K}_3$ going to infinity on both sides of (5.10), we have

$$U^* + \sum_{i=1}^{m} Z_i^* \operatorname{grad}[h(W^*)]_i = 0, \tag{5.11}$$

and complete the proof. □

6. Experiments

In this section, some numerical experiments are presented to evaluate the performance of our MIALM. We compare our algorithm with the existing methods, including SOC (Lai & Osher, 2014), MADMM (Kovnatsky *et al.*, 2016), PAMAL (Hong *et al.*, 2017), ManPG and its adaptive version (ManPG-adap) (Chen *et al.*, 2020). We also test the Riemannian subgradient method (Rsub for short) proposed by Li *et al.* (2021). At the kth iteration, the Riemannian subgradient method gets X^{k+1} via

$$X^{k+1} = \mathcal{R}_{X^k}(-\gamma_k \tilde{\nabla}_R F(X^k)), \tilde{\nabla}_R F(X^k) \in \partial_R F(X^k),$$

where $\partial_R F(X^k)$ denote the Riemannian subgradient of F at X^k and γ_k is a step size. For the MADMM, the Riemannian manifold optimization subproblem is also handled by RBB gradient method (Algorithm 2). We use the polar decomposition as the retraction mapping in our MILAM, MADMM, ManPG and the Riemannian subgradient method. For the SOC, PAMAL and ManPG methods, we use the code provided by Chen *et al.* (2020) (all codes are available[1]). All experiments are performed on a Linux server with a 12-cores Intel Xeon E5-2680 CPU and 128 GB memory. The reported time is wall-clock time in seconds. The codes of our algorithm are available at https://gitee.com/DENGKANGKANG/mialm_code.

6.1 *Stopping criteria*

In our experiments, the CMs in physics and SPCA problems are used as test problems. The following relative KKT residual of problem (1.1) is set to a stopping criterion for our MIALM and MADMM,

[1] https://github.com/chenshixiang/ManPG

both involve primal variable (X, Y) and Lagrangian multiple Z:

$$\begin{aligned}
\eta_p &:= \frac{\|\mathcal{A}X^k - Y^k\|_2}{1 + \|\mathcal{A}X^k\|_2 + \|Y^k\|_2},\\
\eta_d &:= \frac{\left\|\operatorname{Proj}_{T_{X^k}\mathcal{M}}(\nabla f(X^k) - \mathcal{A}^* Z^k)\right\|_{X^k}^2}{1 + \|\nabla f(X^k)\|_2},\\
\eta_C &:= \frac{\left\|\mathcal{A}X^k - \operatorname{prox}_g(\mathcal{A}X^k - Z^k)\right\|_2^2}{1 + \|\mathcal{A}X^k\|_2}.
\end{aligned} \tag{6.1}$$

The MIALM and MADMM methods are terminated if

$$\text{error} := \max\left\{\eta_p, \eta_d, \eta_C\right\} \le \text{tol}, \tag{6.2}$$

where ‘tol’ is a given accuracy tolerance. Both SOC and PAMAL generate iteration sequence $\{(X^k, Y^k, Q^k)\}$, which are given by (??) and (2.3), respectively, and both are terminated if

$$\frac{\|X^k - Y^k\|_2}{\max\left\{1, \|Y^k\|_2, \|X^k\|_2\right\}} + \frac{\|X^k - Q^k\|_2}{\max\left\{1, \|X^k\|_2, \|Q^k\|_2\right\}} \le \text{tol}. \tag{6.3}$$

The ManPG and ManPG-A algorithm are terminated if

$$\|E(\Lambda^k)\|_2 \le \text{tol}, \tag{6.4}$$

where $E(\Lambda^k)$ is given by Chen *et al.* (2020). Since the evaluation criterion is different for different methods, we first run our MIALM and get the objective value F_M, then for the other methods except the Riemannian subgradient method, we give three conditions as follows:

(1) the criterion listed above is hit with the given tolerance;

(2) a specified maximum number of iterations is reached;

(3) the objection value satisfies $F(X^k) \le F_M + 10^{-4}$.

To be fair, we terminate those methods when one of the three conditions is satisfied. For the Riemannian subgradient method, there is no evaluation criterion, it is terminated whenever one of the latter two conditions listed above is hit.

6.2 *Tested problem 1: CMs in physics*

In physics, the CMs problem seeks spatially localized solutions of the independent-particle Schrödinger equation:

$$\hat{H}\phi(x) = \lambda\phi(x), \quad x \in \Omega, \tag{6.5}$$

where $\hat{H} = -\frac{1}{2}\Delta$ and Δ is a Laplacian operator. Consider the one-dimensional free-electron (FE) model with $\hat{H} = -\frac{1}{2}\partial_{x^2}$ in our experiments. By a proper discretization, the CMs can be reformulated to

$$\min_{X\in\mathbb{R}^{n\times r}} tr(X^T HX) + \mu\|X\|_1, \quad \text{s.t.} \quad X^T X = I_r, \tag{6.6}$$

where H is the discretized Schrödinger operator, μ is a regularization parameter. The interesting readers are referred to Ozolinš *et al.* (2013) for more details. For problem (6.6), both SOC and PAMAL solve an equivalent form as follows:

$$\min_{X,Y,Q\in\mathbb{R}^{n\times r}} tr(X^T HX) + \mu\|Y\|_1 \ \text{ s.t.} \quad X = Y, X = Q, Q^T Q = I_r.$$

The MADMM handles a separable reformulation of the form

$$\min_{X,Y\in\mathbb{R}^{n\times r}} tr(X^T HX) + \mu\|Y\|_1 \ \text{ s.t.} \quad X = Y, X^T X = I_r.$$

In our experiments, the domain $\Omega := [0, 50]$ is discretized with n equally spaced nodes. The parameters of our MIALM are set to $\tau = 0.99, \sigma = 1.05, \rho_0 = \lambda_{\max}(H)/2, Z_{\min} = -100 \cdot 1_{d\times r}, Z_{\max} = 100 \cdot 1_{d\times r}, Z^0 = 0_{d\times r}$ and $\epsilon_k = \max(10^{-5}, 0.9^k)$ where $k \in \mathbb{N}$ is iteration counter. We terminate MIALM if the stopping criterion (6.2) is hit with tol $= 10^{-8} * n * r$ or iteration counter $k \geq 5000$. The inner iteration is terminated if $\|\text{grad}\,\psi_{\bar{Z}^k}(X^k)\|_{X^k} \leq \epsilon_k$ or inner iteration number exceeds 100. For the MADMM, the penalty parameter is set to $\rho = \lambda_{\max}(H)/2$, where $\lambda_{\max}(H)$ denote the largest eigenvalue of H, the tolerance of the stopping criterion (6.2) is set tol $= 10^{-8} * n * r$ and the maximum iterations is $k_{\max} = 5000$. The subproblem of the MADMM is terminated if the norm of Riemannian gradient of X-subproblem is less than 10^{-5} or the inner iteration number exceeds 100. The parameters of the SOC, PAMAL and ManPG are set the same as in Chen *et al.* (2020), the tolerance of the SOC and PAMAL are set tol $= 10^{-6}$, and the maximum iterations are $k_{\max} = 20000$. For ManPG and ManPG-adap, the tolerance in (6.4) is set tol $= 10^{-8} * n * r$, and the maximum iterations are $k_{\max} = 20000$.

We test the performance of all methods for solving CMs problem (6.6) with different n, r and sparsity parameter μ, where $n \in \{128, 256, 512\}; r \in \{10, 20, 30, 50\}; \mu \in \{0.05, 0.1, 0.2, 0.3\}$. Table 1 reports the computational results of MIALM and MADMM on CMs problem. Due to the space limitation, Table 1 only shows the results of $n \in \{256, 512\}$. In Table 1, 'obj' denotes the objective value, 'iter' and 'siter' are corresponding to the number of outer iterations and the average number of inner iteration per outer iteration, respectively. The parameters η_p, η_d, η_C are given by (6.1). Table 1 indicates that our MIALM is over five times faster than MADMM in most instances, the number of iterations of MIALM is significantly less than that of MADMM. In Fig. 1, we describe the relationship between two criterion,

TABLE 1 *The computational results of MIALM and MADMM on CMs*

(n, r, μ)	MIALM							MADMM						
	obj	time	iter	siter	η_p	η_d	η_C	obj	time	iter	siter	η_p	η_d	η_C
256 / 10 / 0.05	3.746e+0	1.11	107	37.7	5.30e-6	2.52e-5	9.32e-6	3.746e+0	11.76	3446	11.5	5.93e-6	2.52e-5	1.04e-5
256 / 10 / 0.10	6.272e+0	0.64	75	30.3	8.68e-6	2.15e-5	1.53e-5	6.272e+0	10.34	3085	11.3	6.17e-6	2.52e-5	1.09e-5
256 / 10 / 0.20	1.083e+1	5.22	416	46.1	2.87e-6	2.52e-5	5.05e-6	1.084e+1	18.72	5000	13.0	2.43e-5	1.26e-3	4.27e-5
256 / 10 / 0.30	1.498e+1	0.70	82	30.6	4.00e-7	2.54e-5	7.04e-7	1.498e+1	11.36	3390	11.5	4.16e-7	2.54e-5	7.31e-7
256 / 20 / 0.05	1.067e+1	0.57	54	24.1	1.34e-5	4.31e-5	2.42e-5	1.067e+1	7.86	1422	9.1	1.63e-5	5.04e-5	2.97e-5
256 / 20 / 0.10	1.522e+1	0.85	46	19.7	9.57e-6	4.68e-5	1.74e-5	1.522e+1	7.48	1374	8.3	1.75e-5	4.89e-5	3.18e-5
256 / 20 / 0.20	2.350e+1	27.97	1000	48.8	4.47e-5	5.88e-4	8.12e-5	2.350e+1	33.65	5000	11.7	4.50e-5	3.79e-4	8.18e-5
256 / 20 / 0.30	3.120e+1	1.01	65	29.1	4.79e-6	4.17e-5	8.70e-6	3.120e+1	13.58	2366	9.6	4.69e-6	5.07e-5	8.51e-6
256 / 30 / 0.05	2.509e+1	1.49	52	27.3	1.31e-5	7.16e-5	2.41e-5	2.510e+1	10.55	1484	8.9	1.21e-5	7.42e-5	2.24e-5
256 / 30 / 0.10	3.143e+1	0.95	44	20.8	1.46e-5	7.21e-5	2.70e-5	3.143e+1	10.69	1449	8.7	1.38e-5	7.64e-5	2.55e-5
256 / 30 / 0.20	4.304e+1	1.12	47	22.2	3.17e-5	6.53e-5	5.85e-5	4.305e+1	11.07	1406	8.6	2.01e-5	7.63e-5	3.71e-5
256 / 30 / 0.30	5.394e+1	1.34	52	24.7	2.81e-5	6.65e-5	5.18e-5	5.381e+1	11.33	1439	8.1	3.67e-5	5.62e-5	6.76e-5
256 / 50 / 0.05	9.161e+1	2.70	51	31.9	1.07e-5	9.10e-5	2.00e-5	9.160e+1	28.78	1554	10.6	2.71e-5	1.20e-4	5.09e-5
256 / 50 / 0.10	1.011e+2	1.73	43	24.0	2.31e-5	1.14e-4	4.33e-5	1.011e+2	18.84	1198	8.8	2.07e-5	1.21e-4	3.89e-5
256 / 50 / 0.20	1.185e+2	0.87	31	15.5	3.81e-5	1.13e-4	7.14e-5	1.185e+2	15.47	1131	7.8	3.56e-5	1.08e-4	6.68e-5
256 / 50 / 0.30	1.347e+2	0.71	31	14.3	5.26e-5	1.25e-4	9.86e-5	1.347e+2	16.47	1163	7.7	2.37e-5	1.23e-4	4.45e-5
512 / 10 / 0.05	4.826e+0	3.56	163	43.1	2.27e-6	4.31e-5	3.99e-6	4.826e+0	31.81	2955	21.0	2.92e-6	5.12e-5	5.14e-6
512 / 10 / 0.10	8.226e+0	3.42	158	42.0	1.74e-6	4.56e-5	3.07e-6	8.226e+0	49.62	3939	24.0	5.20e-6	5.09e-5	9.14e-6
512 / 10 / 0.20	1.430e+1	6.52	283	45.1	5.14e-6	4.91e-5	9.05e-6	1.430e+1	63.07	5000	23.7	5.77e-6	2.99e-4	1.02e-5
512 / 10 / 0.30	1.978e+1	2.54	123	39.3	9.78e-6	4.88e-5	1.72e-5	1.978e+1	51.08	4752	20.8	5.25e-6	5.09e-5	9.24e-6
512 / 20 / 0.05	1.262e+1	3.28	105	38.7	6.06e-6	9.46e-5	1.10e-5	1.262e+1	38.37	1952	20.6	1.41e-5	1.02e-4	2.56e-5
512 / 20 / 0.10	1.878e+1	3.90	98	38.6	9.76e-6	8.51e-5	1.77e-5	1.877e+1	36.84	2132	18.4	1.27e-5	9.33e-5	2.30e-5
512 / 20 / 0.20	2.997e+1	3.39	120	40.4	6.45e-6	8.20e-5	1.17e-5	2.997e+1	54.36	2763	20.4	7.26e-6	1.01e-4	1.32e-5
512 / 20 / 0.30	4.033e+1	6.60	153	42.5	1.94e-6	8.08e-5	3.52e-6	4.033e+1	89.87	5000	19.2	3.89e-6	1.47e-4	7.07e-6
512 / 30 / 0.05	2.789e+1	3.43	66	36.8	1.38e-5	1.28e-4	2.54e-5	2.790e+1	29.81	1256	17.6	3.16e-5	1.48e-4	5.84e-5
512 / 30 / 0.10	3.648e+1	2.27	56	31.1	1.13e-5	1.28e-4	2.09e-5	3.648e+1	23.40	1214	15.0	2.34e-5	1.43e-4	4.31e-5
512 / 30 / 0.20	5.218e+1	4.94	82	36.5	7.90e-6	1.29e-4	1.46e-5	5.218e+1	39.34	1755	17.5	1.21e-5	1.52e-4	2.24e-5
512 / 30 / 0.30	6.680e+1	5.70	106	41.3	1.49e-5	1.53e-4	2.75e-5	6.680e+1	59.73	2285	19.7	1.66e-5	1.52e-4	3.06e-5
512 / 50 / 0.05	9.682e+1	7.00	64	39.0	1.55e-5	2.47e-4	2.90e-5	9.683e+1	45.69	1017	18.1	6.98e-5	2.14e-4	1.31e-4
512 / 50 / 0.10	1.098e+2	4.27	48	34.6	1.34e-5	2.55e-4	2.51e-5	1.098e+2	44.07	1124	16.0	4.49e-5	2.38e-4	8.42e-5
512 / 50 / 0.20	1.338e+2	5.90	66	38.0	1.65e-5	2.34e-4	3.04e-5	1.338e+2	60.73	1473	17.3	3.66e-5	2.17e-4	6.77e-5
512 / 50 / 0.30	1.562e+2	2.62	37	29.3	3.60e-5	2.24e-4	6.73e-5	1.562e+2	32.44	1038	13.4	4.35e-5	2.45e-4	8.16e-5

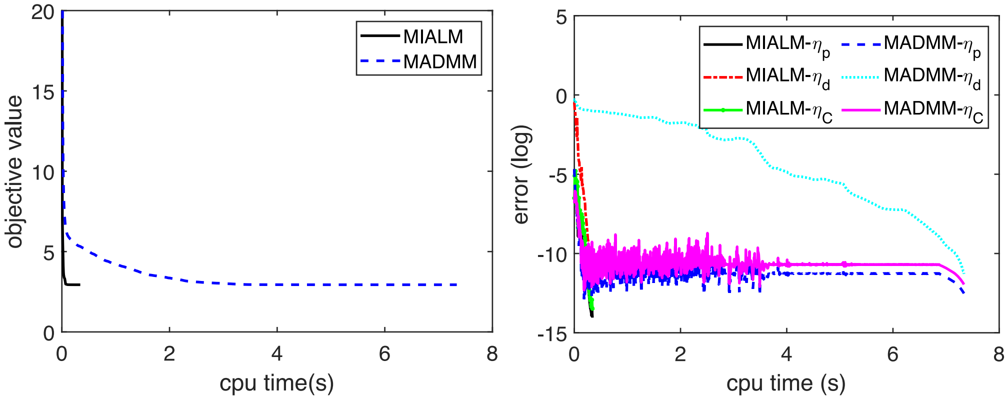

FIG. 1. The performance of MIALM and MADMM on CMs with $n = 128, r = 10, \mu = 0.05$.

with cpu-time in the iterative process of one instance with $n = 128, r = 10, \mu = 0.05$. Those two criteria are objective value and error, where error is given by (6.1). Although both MIALM and MADMM make rapid progress lowering the cost in the early stage, MIALM eventually converge considerably faster than MADMM, which is shown in Fig. 1 (left). Figure 1 (right) shows that both η_p and η_C of MADMM have similar curve as MIALM at early stage, and η_d is decreasing slowly, leading to a shock of η_p and η_C. The reason for the slow decline of η_d is that the MADMM minimizing the variables X and Y, respectively, in each iteration, while MIALM solves a joint minimization problem respect to X, Y. At the end, a sudden drop for η_d since η_d has the same precision as η_p and η_C.

We compare the accuracy and efficiency of MIALM with other algorithms using the performance profiling method proposed in Dolan & Moré (2002). Let $t_{p,s}$ be some performance quantity (e.g., time or accuracy, lower is better) associated with the s-th solver on problem p. Then, one computes the ratio $r_{p,s}$ between $t_{p,s}$ over the smallest value obtained by n_s solvers on problem p, i.e., $r_{p,s} := \frac{t_{p,s}}{\min\{t_{p,s}:1\leq s\leq n_s\}}$. For $\tau > 0$, the value

$$\pi_s(\tau) := \frac{\text{number of problems where } \log_2(r_{p,s}) \leq \tau}{\text{total number of problems}}$$

indicates that solver s is within a factor $2^\tau \geq 1$ of the performance obtained by the best solver. Then the performance plot is a curve $\pi_s(\tau)$ for each solver s as a function of τ. In Fig. 3, we show the performance profiles of two criteria: 'relative-obj' and 'CPU time', where 'relative-obj' is the relative objective value defined by $F/|F_{\max}|$, where F is the objective value obtained by each method, $F_{\max}$ denotes the largest objective value of all methods. In particular, the intercept point of the axis 'ratio of problems' and the curve in each subfigure is the percentage of the lower/faster one between the two solvers. These figures show that the objective value and the CPU time of our MIALM are better than other algorithms in most problems. Table 2 give the detail results. The results show that the MIALM compares favorably to the other methods. For the same objective function value, our MIALM takes less CPU-time. Figure 2 describes the relationship between objective value with cpu-time in the iterative process of one instance with $n = 128, r = 10, \mu = 0.1$. One can see that the MIALM and SOC, PAMAL are comparable for CMs problem.

TABLE 2 *Computational results of our MIALM compared with MADMM, ManPG, ManPG-adap, Rsub, SOC and PAMAL on CMs*

(n, r, μ)	MIALM		MADMM		ManPG		ManPG-adap		Rsub		SOC		PAMAL	
	obj	time	obj	time	obj	time	obj	time	obj	time	obj	time	obj	time
256 / 10 / 0.05	3.746e+0	1.11	3.746e+0	11.76	3.747e+0	4.71	3.747e+0	3.45	3.891e+0	4.29	3.746e+0	1.18	3.746e+0	1.33
256 / 10 / 0.10	6.272e+0	0.64	6.272e+0	10.34	6.273e+0	3.62	6.273e+0	4.11	6.372e+0	4.90	6.272e+0	1.74	6.272e+0	1.18
256 / 10 / 0.20	1.083e+1	5.22	1.084e+1	18.72	1.084e+1	7.80	1.084e+1	6.26	1.094e+1	5.09	1.083e+1	17.06	1.083e+1	84.42
256 / 10 / 0.30	1.498e+1	0.70	1.498e+1	11.36	1.498e+1	3.17	1.498e+1	2.60	1.506e+1	5.31	1.498e+1	3.12	1.498e+1	38.50
256 / 20 / 0.05	1.067e+1	0.57	1.067e+1	7.86	1.067e+1	17.60	1.067e+1	21.09	1.091e+1	8.22	1.067e+1	0.64	1.067e+1	0.33
256 / 20 / 0.10	1.522e+1	0.85	1.522e+1	7.48	1.522e+1	13.84	1.522e+1	16.99	1.526e+1	9.59	1.522e+1	0.71	1.522e+1	8.82
256 / 20 / 0.20	2.350e+1	27.97	2.350e+1	33.65	2.350e+1	20.81	2.350e+1	18.56	2.370e+1	11.04	2.350e+1	5.81	2.350e+1	132.00
256 / 20 / 0.30	3.120e+1	1.01	3.120e+1	13.58	3.120e+1	21.61	3.120e+1	24.88	3.156e+1	10.85	3.120e+1	5.10	3.120e+1	118.00
256 / 30 / 0.05	2.509e+1	1.49	2.510e+1	10.55	2.510e+1	60.39	2.510e+1	76.57	2.631e+1	10.52	2.509e+1	7.04	2.509e+1	3.39
256 / 30 / 0.10	3.143e+1	0.95	3.143e+1	10.69	3.143e+1	59.25	3.143e+1	89.89	3.167e+1	15.51	3.143e+1	2.44	3.142e+1	1.31
256 / 30 / 0.20	4.304e+1	1.12	4.305e+1	11.07	4.308e+1	51.29	4.308e+1	83.60	4.313e+1	18.59	4.305e+1	6.27	4.304e+1	153.97
256 / 30 / 0.30	5.394e+1	1.34	5.381e+1	11.33	5.394e+1	43.81	5.394e+1	67.82	5.405e+1	19.61	5.394e+1	0.87	5.394e+1	129.25
256 / 50 / 0.05	9.161e+1	2.70	9.160e+1	28.78	9.161e+1	239.85	9.161e+1	420.55	9.270e+1	28.35	9.161e+1	5.44	9.161e+1	304.30
256 / 50 / 0.10	1.011e+2	1.73	1.011e+2	18.84	1.011e+2	277.88	1.011e+2	511.96	1.025e+2	47.76	1.011e+2	5.16	1.011e+2	356.28
256 / 50 / 0.20	1.185e+2	0.87	1.185e+2	15.47	1.185e+2	342.81	1.185e+2	606.45	1.190e+2	57.04	1.185e+2	4.46	1.185e+2	358.10
256 / 50 / 0.30	1.347e+2	0.71	1.347e+2	16.47	1.347e+2	270.59	1.347e+2	337.98	1.350e+2	59.60	1.347e+2	4.18	1.347e+2	399.58
512 / 10 / 0.05	4.826e+0	3.56	4.826e+0	31.81	4.831e+0	18.99	4.827e+0	12.81	9.606e+0	3.33	4.826e+0	1.66	4.826e+0	1.57
512 / 10 / 0.10	8.226e+0	3.42	8.226e+0	49.62	8.235e+0	17.53	8.228e+0	21.03	1.533e+1	3.44	8.226e+0	4.34	8.226e+0	1.63
512 / 10 / 0.20	1.430e+1	6.52	1.430e+1	63.07	1.431e+1	18.97	1.430e+1	11.85	1.879e+1	3.46	1.430e+1	8.35	1.430e+1	93.64
512 / 10 / 0.30	1.978e+1	2.54	1.978e+1	51.08	1.978e+1	12.52	1.978e+1	14.18	2.212e+1	4.23	1.978e+1	8.44	1.978e+1	63.66
512 / 20 / 0.05	1.262e+1	3.28	1.262e+1	38.37	1.263e+1	1554.24	1.263e+1	36.34	2.379e+1	15.85	1.262e+1	2.69	1.262e+1	1.19
512 / 20 / 0.10	1.878e+1	3.90	1.877e+1	36.84	1.878e+1	1359.27	1.878e+1	38.14	3.477e+1	12.82	1.878e+1	6.66	1.878e+1	1.21
512 / 20 / 0.20	2.997e+1	3.39	2.997e+1	54.36	2.998e+1	1368.49	2.998e+1	40.71	4.745e+1	17.63	2.998e+1	14.15	2.997e+1	210.49
512 / 20 / 0.30	4.033e+1	6.60	4.033e+1	89.87	4.035e+1	45.74	4.033e+1	71.02	5.032e+1	12.20	4.035e+1	20.02	4.033e+1	217.55
512 / 30 / 0.05	2.789e+1	3.43	2.790e+1	29.81	2.790e+1	217.09	2.790e+1	57.12	4.402e+1	21.24	2.790e+1	3.58	2.789e+1	1.67
512 / 30 / 0.10	3.648e+1	2.27	3.648e+1	23.40	3.648e+1	65.07	3.648e+1	54.01	6.206e+1	22.65	3.648e+1	3.08	3.648e+1	165.59
512 / 30 / 0.20	5.218e+1	4.94	5.218e+1	39.34	5.218e+1	1169.60	5.218e+1	92.72	8.093e+1	22.88	5.222e+1	5.46	5.218e+1	205.64
512 / 30 / 0.30	6.680e+1	5.70	6.680e+1	59.73	6.681e+1	721.27	6.680e+1	90.44	8.084e+1	23.15	6.692e+1	8.80	6.680e+1	249.79
512 / 50 / 0.05	9.682e+1	7.00	9.683e+1	45.69	9.683e+1	176.29	9.683e+1	155.48	1.231e+2	30.08	9.684e+1	7.38	9.684e+1	607.41
512 / 50 / 0.10	1.098e+2	4.27	1.098e+2	44.07	1.098e+2	238.37	1.098e+2	218.45	1.578e+2	31.34	1.098e+2	6.89	1.098e+2	494.21
512 / 50 / 0.20	1.338e+2	5.90	1.338e+2	60.73	1.338e+2	325.16	1.338e+2	308.49	2.090e+2	31.79	1.341e+2	7.65	1.338e+2	511.16
512 / 50 / 0.30	1.562e+2	2.62	1.562e+2	32.44	1.562e+2	276.42	1.562e+2	259.63	2.327e+2	31.95	1.564e+2	5.55	1.562e+2	549.22

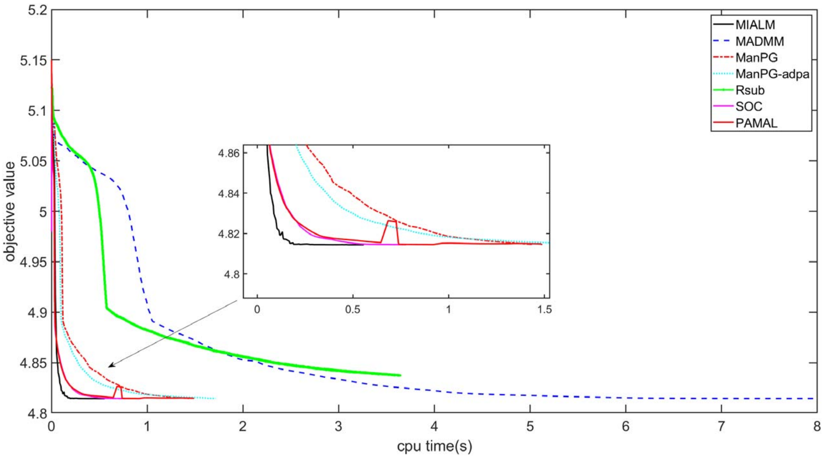

FIG. 2. The objective function value vs cputime of MIALM and the compared algorithms on CMs with $n = 128, r = 10, \mu = 0.1$.

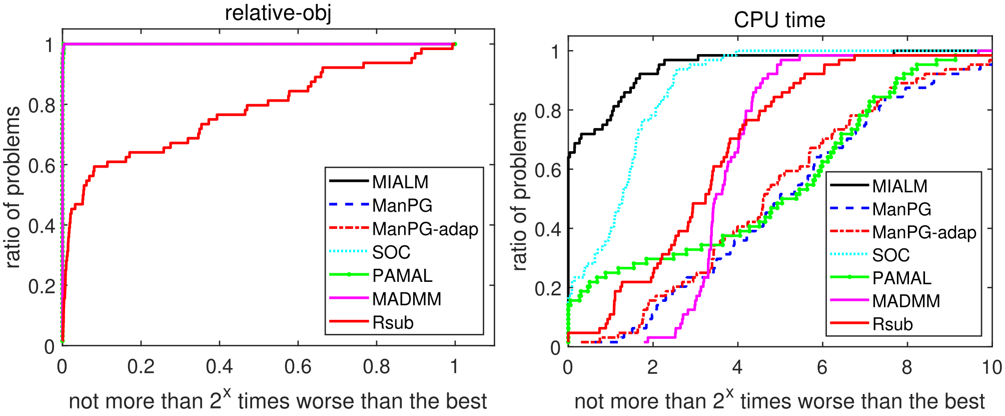

FIG. 3. The performance profiles of MIALM and other algorithms on CMs.

6.3 *Test problem 2: SPCA*

Given a data set $\{b_1, \cdots, b_m\}$ where $b_i \in \mathbb{R}^{n\times 1}$, the sparse PCA problem is

$$\min_{X\in\mathbb{R}^{n\times r}} \sum_{i=1}^{m} \|b_i - XX^T b_i\|_2^2 + \mu\|X\|_1, \quad \text{s.t.} \quad X^T X = I_r, \tag{6.7}$$

where μ is a regularization parameter. Let $B = [b_1, \cdots, b_m]^T \in \mathbb{R}^{m\times n}$, problem (6.7) has the form

$$\min_{X\in\mathbb{R}^{n\times r}} -tr(X^T B^T BX) + \mu\|X\|_1, \quad \text{s.t.} \quad X^T X = I_r. \tag{6.8}$$

In our experiments, data matrix $B \in \mathbb{R}^{m\times n}$ is generated via the following two settings.

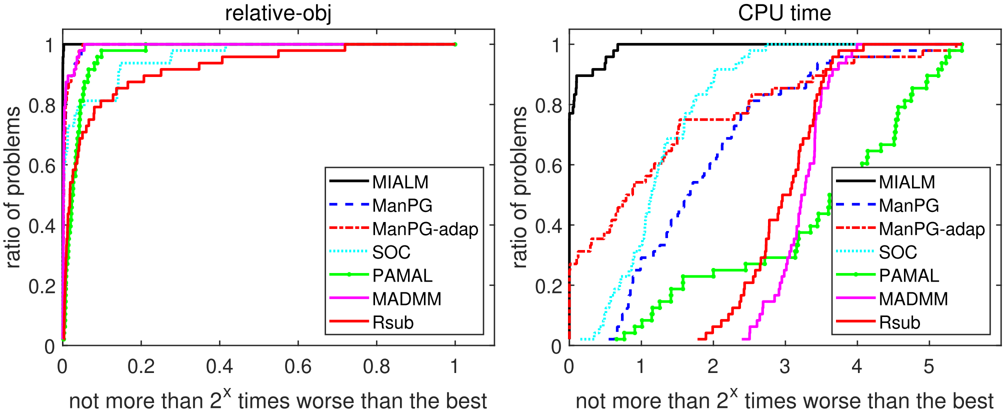

FIG. 4. The performance profiles of MIALM and other algorithms on SPCA with random data.

(1) Data matrix $B \in \mathbb{R}^{m\times n}$ is produced by MATLAB function $randn(m, n)$, which all entries of B follow the standard Gaussian distribution. We shift the columns of B such that they have 0-mean, and finally the column-vectors are normalized.

(2) Data matrix $B \in \mathbb{R}^{m\times n}$ is selected from real data. In those real data, 'Arabidopsis' and 'Leukemia' are the gene expression data selected from Li & Toh (2010), 'Staunton' and 'Ross' are the NCI 60 data selected from Culhane *et al.* (2003) and 'realEQTL' is the yeast eQTL data selected from Zhu *et al.* (2008).

All methods are terminated with the same stopping criterion as that used on the CMs. The parameters of our MIALM and MADMM are set to the same as that used in the CMs, except that the penalty parameter is re-set to $\rho_0 = \lambda^2_{\max}(B^TB)/2$. For the SOC, PAMAL and ManPG methods, the parameter settings provided in Chen *et al.* (2020) are copied, the interested readers are referred to Chen *et al.* (2020) for details. We set $m = 50$ in these experiments.

The results in setting 1. We test the performance of all methods for solving different instances of problem (6.8) induced by different n, r and sparsity parameter μ, where $n \in \{200, 300, 500, 1000\}$; $r \in \{10, 20, 30, 50\}$; $\mu \in \{0.4, 0.6, 0.8\}$. Due to the space limitation, we only report the results of $n \in \{300, 500, 1000\}$ and $r \in \{20, 30, 50\}$. The total number of line search steps and the average number of inner/outer iteration of the MIALM and MADMM are listed in Table 4. One can see that MIALM is over 10 times faster than MADMM in most instances, and has lower outer iteration and line search steps than that of the MADMM. The computational results of our MIALM and other methods on objective value and time are reported in Table 3. The performance profile of MIALM and other methods on criteria 'relative-obj' and 'CPU time' are displayed in Fig. 4, where 'relative-obj' is defined as

$$\text{relative-obj} := e^{F/|F_{\max}|}, \tag{6.9}$$

where F is the objective value obtained by each method, $F_{\max}$ denotes the largest objective value of all methods. The results show that, for these settings of SPCA, our MIALM has the better performance compared to other methods.

TABLE 3 *The computational results of our MIALM compared with MADMM, ManPG, ManPG-adap, Rsub, SOC and PAMAL on SPCA with random data*

(n, r, μ)	MIALM		MADMM		ManPG		ManPG-adap		Rsub		SOC		PAMAL	
	obj	time	obj	time	obj	time	obj	time	obj	time	obj	time	obj	time
300 / 20 / 0.40	−8.439e+1	0.89	−8.421e+1	26.84	−8.428e+1	4.38	−8.428e+1	1.75	−8.351e+1	11.46	−8.433e+1	3.34	v8.370e+1	128.56
300 / 20 / 0.60	−4.942e+1	0.75	−4.912e+1	40.49	−4.884e+1	4.86	−4.888e+1	1.64	−4.710e+1	11.48	−4.898e+1	4.33	−4.844e+1	139.54
300 / 20 / 0.80	−2.251e+1	1.39	−2.228e+1	33.62	−2.227e+1	6.37	−2.213e+1	2.44	−1.835e+1	11.48	−2.211e+1	4.38	−2.176e+1	163.23
300 / 30 / 0.40	−1.043e+2	1.61	−1.042e+2	62.14	−1.040e+2	17.42	−1.040e+2	7.19	−1.028e+2	49.95	−1.043e+2	7.92	−1.034e+2	145.98
300 / 30 / 0.60	−5.754e+1	3.35	−5.736e+1	57.89	−5.730e+1	12.93	−5.730e+1	4.55	−5.394e+1	47.89	−4.578e+1	9.55	−5.683e+1	145.04
300 / 30 / 0.80	−2.394e+1	3.44	−2.269e+1	64.14	−2.287e+1	37.04	−2.277e+1	57.53	−1.529e+1	48.67	−2.287e+1	4.01	−2.239e+1	139.28
300 / 50 / 0.40	−1.165e+2	9.77	−1.165e+2	107.08	−1.164e+2	31.40	−1.162e+2	13.08	−1.135e+2	71.48	−6.982e+1	19.98	−1.146e+2	224.91
300 / 50 / 0.60	−6.155e+1	6.23	−6.122e+1	91.79	−6.100e+1	177.42	−6.104e+1	324.76	−5.441e+1	70.99	−6.043e+1	9.43	−5.952e+1	265.55
300 / 50 / 0.80	−2.593e+1	2.18	−2.484e+1	45.09	−2.451e+1	198.13	−2.441e+1	295.34	−1.072e+1	72.07	−2.544e+1	6.61	−2.432e+1	208.40
500 / 20 / 0.40	−1.483e+2	1.41	−1.480e+2	39.86	−1.480e+2	5.81	−1.480e+2	2.70	−1.475e+2	42.62	−1.481e+2	3.43	−1.437e+2	33.79
500 / 20 / 0.60	−9.659e+1	1.49	−9.648e+1	44.71	−9.630e+1	5.85	−9.630e+1	2.42	−9.475e+1	44.78	−9.627e+1	5.41	−9.290e+1	47.58
500 / 20 / 0.80	−5.391e+1	2.46	−5.368e+1	53.98	−5.370e+1	4.01	−5.369e+1	1.48	−5.069e+1	44.83	−5.383e+1	4.89	−5.103e+1	112.31
500 / 30 / 0.40	−1.925e+2	2.00	−1.923e+2	53.65	−1.921e+2	15.38	−1.923e+2	6.22	−1.911e+2	48.33	−1.541e+2	10.96	−1.835e+2	8.23
500 / 30 / 0.60	−1.215e+2	2.70	−1.208e+2	67.74	−1.209e+2	14.47	−1.211e+2	6.65	−1.183e+2	49.34	−1.210e+2	10.53	−1.168e+2	84.36
500 / 30 / 0.80	−6.270e+1	4.54	−6.248e+1	63.75	−6.230e+1	15.90	−6.219e+1	4.53	−5.726e+1	48.21	−6.247e+1	7.25	−5.928e+1	109.50
500 / 50 / 0.40	−2.340e+2	3.46	−2.335e+2	75.24	−2.340e+2	41.02	−2.337e+2	15.47	−2.309e+2	74.55	−9.387e+1	34.96	−2.188e+2	16.77
500 / 50 / 0.60	−1.333e+2	4.68	−1.339e+2	142.28	−1.330e+2	32.83	−1.330e+2	19.37	−1.268e+2	75.11	−8.075e+1	31.51	−1.232e+2	109.72
500 / 50 / 0.80	−6.132e+1	6.66	−6.130e+1	127.35	−6.118e+1	98.93	−6.086e+1	196.14	−4.920e+1	75.31	−4.870e+1	21.67	−5.781e+1	246.98
1000 / 20 / 0.40	−3.177e+2	2.04	−3.173e+2	47.74	−3.174e+2	12.97	−3.175e+2	6.62	−3.160e+2	49.40	−3.173e+2	10.18	−3.057e+2	30.71
1000 / 20 / 0.60	−2.392e+2	2.73	−2.388e+2	37.00	−2.389e+2	10.48	−2.388e+2	4.91	−2.369e+2	49.40	−2.390e+2	7.84	−2.241e+2	8.57
1000 / 20 / 0.80	−1.659e+2	2.08	−1.656e+2	55.28	−1.656e+2	9.38	−1.656e+2	4.36	−1.630e+2	50.04	−1.656e+2	7.17	−1.465e+2	8.53
1000 / 30 / 0.40	−4.349e+2	2.14	−4.346e+2	50.41	−4.346e+2	17.88	−4.349e+2	9.76	−4.330e+2	60.43	−4.350e+2	15.36	−4.134e+2	5.32
1000 / 30 / 0.60	−3.181e+2	2.29	−3.176e+2	56.94	−3.178e+2	17.86	−3.179e+2	10.60	−3.154e+2	58.73	−3.180e+2	17.03	−2.878e+2	6.28
1000 / 30 / 0.80	−2.142e+2	2.63	−2.137e+2	59.53	−2.136e+2	22.03	−2.138e+2	10.07	−2.094e+2	60.96	−2.143e+2	19.81	−1.870e+2	12.73
1000 / 50 / 0.40	−5.831e+2	2.15	−5.823e+2	52.96	−5.830e+2	57.49	−5.830e+2	21.26	−5.787e+2	90.71	−5.837e+2	32.92	−5.401e+2	4.15
1000 / 50 / 0.60	−4.070e+2	2.41	−4.059e+2	77.34	−4.069e+2	66.35	−4.068e+2	28.91	−4.000e+2	90.82	−4.067e+2	29.36	−3.489e+2	8.45
1000 / 50 / 0.80	−2.510e+2	3.24	−2.507e+2	85.23	−2.507e+2	41.30	−2.506e+2	11.90	−2.425e+2	90.26	−2.516e+2	35.64	−1.748e+2	6.95

TABLE 4 *The detailed results of MIALM and MADMM on SPCA with random data*

(n, r, μ)	MIALM							MADMM						
	obj	time	iter	siter	η_p	η_d	η_C	obj	time	iter	siter	η_p	η_d	η_C
300 / 20 / 0.40	–8.439e+1	0.89	81	8.9	4.65e-6	5.33e-5	8.45e-6	–8.421e+1	26.84	3866	6.2	5.41e-6	5.99e-5	9.82e-6
300 / 20 / 0.60	–4.942e+1	0.75	76	8.2	6.09e-6	5.65e-5	1.11e-5	–4.912e+1	40.49	5759	6.4	5.51e-6	5.99e-5	1.00e-5
300 / 20 / 0.80	–2.251e+1	1.39	109	11.9	1.04e-5	5.56e-5	1.90e-5	–2.228e+1	33.62	4849	6.4	4.59e-6	5.98e-5	8.33e-6
300 / 30 / 0.40	–1.043e+2	1.61	86	10.4	4.29e-6	8.63e-5	7.92e-6	–1.042e+2	62.14	4882	6.8	7.98e-6	8.98e-5	1.47e-5
300 / 30 / 0.60	–5.754e+1	3.35	128	12.6	1.03e-5	8.64e-5	1.91e-5	–5.736e+1	57.89	4413	6.9	8.19e-6	8.98e-5	1.51e-5
300 / 30 / 0.80	–2.394e+1	3.44	130	14.0	7.82e-6	8.41e-5	1.44e-5	–2.269e+1	64.14	5218	6.5	3.27e-6	8.98e-5	6.04e-6
300 / 50 / 0.40	–1.165e+2	9.77	285	17.1	5.11e-6	1.46e-4	9.58e-6	–1.165e+2	107.08	7672	7.5	1.07e-5	1.50e-4	2.00e-5
300 / 50 / 0.60	–6.155e+1	6.23	195	15.4	1.32e-5	1.45e-4	2.47e-5	–6.122e+1	91.79	6814	7.2	8.15e-6	1.50e-4	1.53e-5
300 / 50 / 0.80	–2.593e+1	2.18	88	11.9	3.35e-5	1.33e-4	6.28e-5	–2.484e+1	45.09	3403	7.1	6.98e-6	1.49e-4	1.31e-5
500 / 20 / 0.40	–1.483e+2	1.41	77	10.2	5.80e-6	9.02e-5	1.05e-5	–1.480e+2	39.86	3301	6.7	9.27e-6	9.99e-5	1.68e-5
500 / 20 / 0.60	–9.659e+1	1.49	80	10.8	1.42e-5	9.26e-5	2.58e-5	–9.648e+1	44.71	3637	6.6	9.02e-6	9.98e-5	1.64e-5
500 / 20 / 0.80	–5.391e+1	2.46	105	12.3	4.52e-6	9.30e-5	8.20e-6	–5.368e+1	53.98	4535	6.8	9.19e-6	9.98e-5	1.67e-5
500 / 30 / 0.40	–1.925e+2	2.00	88	12.1	4.90e-6	1.42e-4	9.05e-6	–1.923e+2	53.65	4019	7.1	1.37e-5	1.50e-4	2.54e-5
500 / 30 / 0.60	–1.215e+2	2.70	104	12.9	1.58e-5	1.36e-4	2.91e-5	–1.208e+2	67.74	5325	6.9	1.37e-5	1.50e-4	2.53e-5
500 / 30 / 0.80	–6.270e+1	4.54	151	14.1	5.45e-6	1.37e-4	1.01e-5	–6.248e+1	63.75	4632	7.4	1.42e-5	1.50e-4	2.62e-5
500 / 50 / 0.40	–2.340e+2	3.46	101	14.1	2.57e-5	2.21e-4	4.81e-5	–2.335e+2	75.24	4245	7.6	2.26e-5	2.49e-4	4.24e-5
500 / 50 / 0.60	–1.333e+2	4.68	124	14.8	1.10e-5	2.40e-4	2.07e-5	–1.339e+2	142.28	7690	7.9	2.29e-5	2.50e-4	4.30e-5
500 / 50 / 0.80	–6.132e+1	6.66	165	16.0	1.82e-5	2.43e-4	3.41e-5	–6.130e+1	127.35	7493	7.2	1.10e-5	2.49e-4	2.06e-5
1000 / 20 / 0.40	–3.177e+2	2.04	78	13.0	3.19e-5	1.86e-4	5.79e-5	–3.173e+2	47.74	3381	7.0	1.82e-5	2.00e-4	3.30e-5
1000 / 20 / 0.60	–2.392e+2	2.73	98	13.8	2.13e-5	1.86e-4	3.87e-5	–2.388e+2	37.00	2660	6.9	1.82e-5	2.00e-4	3.31e-5
1000 / 20 / 0.80	–1.659e+2	2.08	81	13.0	1.09e-5	1.87e-4	1.98e-5	–1.656e+2	55.28	4001	6.9	1.83e-5	2.00e-4	3.32e-5
1000 / 30 / 0.40	–4.349e+2	2.14	63	12.8	2.37e-5	2.73e-4	4.37e-5	–4.346e+2	50.41	2758	7.2	2.76e-5	3.00e-4	5.09e-5
1000 / 30 / 0.60	–3.181e+2	2.29	64	13.2	1.75e-5	2.71e-4	3.22e-5	–3.176e+2	56.94	3209	7.0	2.78e-5	3.00e-4	5.12e-5
1000 / 30 / 0.80	–2.142e+2	2.63	73	13.3	1.96e-5	2.72e-4	3.62e-5	–2.137e+2	59.53	3287	7.0	2.83e-5	3.00e-4	5.22e-5
1000 / 50 / 0.40	–5.831e+2	2.15	53	13.5	3.55e-5	4.64e-4	6.64e-5	–5.823e+2	52.96	2283	7.5	4.51e-5	5.00e-4	8.46e-5
1000 / 50 / 0.60	–4.070e+2	2.41	53	13.8	4.57e-5	4.67e-4	8.54e-5	–4.059e+2	77.34	3474	7.4	4.54e-5	4.99e-4	8.51e-5
1000 / 50 / 0.80	–2.510e+2	3.24	59	13.9	5.31e-5	4.81e-4	9.91e-5	–2.507e+2	85.23	3678	7.4	4.38e-5	4.99e-4	8.22e-5

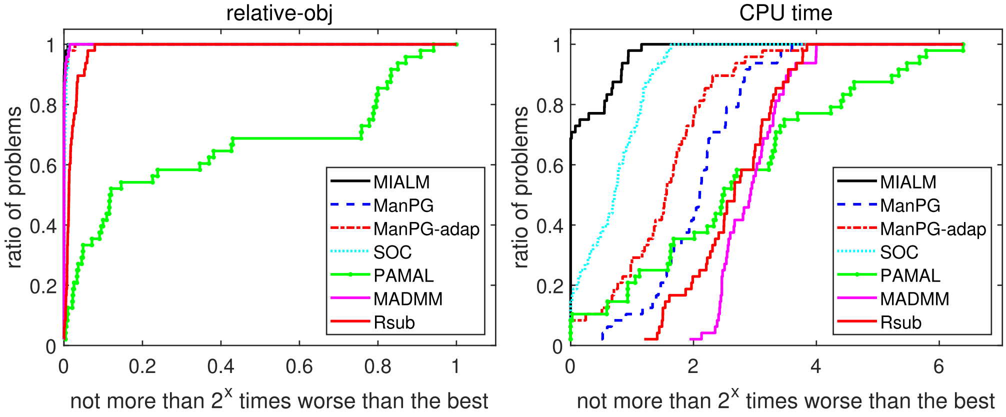

FIG. 5. The performance profiles of MIALM and other algorithms on SPCA with real data.

The results in setting 2. We test the performance of all methods for solving problem (6.8) induced by different μ and r, where $r \in \{10, 20\}$ and $\mu \in \{0.4, 0.6\}$. The computational results of our MIALM and other methods on objective value and time are reported in Table 5. Table 6 reports the detailed results of MIALM and MADMM on SPCA problem with real data. One can easily conclude from Table 6 that, for these settings of SPCA problems with real data, our MIALM performs better than MADMM on most instances. The performance profile of MIALM and other methods on criteria ‘relative-obj’ and ‘CPU time’ are displayed in Fig. 5, where ‘relative-obj’ is defined by (6.9). All the results show that, for the SPCA problem with this settings, our MIALM compares favorably to the other methods.

7. Conclusions

In this paper, we proposed an MIALM for nonsmooth composite minimization problem on Riemannian manifold. Based on the Moreau envelope, the iteration subproblem is reformulated to a smooth Riemannian manifold minimization problem and a proximal operator, and at each iteration in our method, one only needs an inexact solution of the smooth Riemannian manifold minimization problem. The convergence to critical point of the proposed method is established under some mild assumptions. Numerical experiments show that our method is competitive to some existing state-of-the-art methods.

Acknowledgements

We are extremely grateful to the anonymous referee for constructive comments, which play a key role for improving this paper.

Funding

National Natural Science Foundation of China (12071398, 11671125 and 11571074); Natural Science Foundation of Hunan Province (2020JJ4567); Key Scientific Research Found of Hunan Education Department (20A097 and 18A351).

TABLE 5 *The computational results of our MIALM compared with MADMM, ManPG, ManPG-adap, Rsub, SOC and PAMAL on SPCA with real data*

(n, r, μ)	MIALM		MADMM		ManPG		ManPG-adap		Rsub		SOC		PAMAL	
	obj	time	obj	time	obj	time	obj	time	obj	time	obj	time	obj	time
Arabidopsis														
834 / 10 / 0.40	–4.810e+2	2.91	–4.811e+2	34.55	–4.810e+2	7.17	–4.810e+2	3.73	–4.751e+2	18.94	–4.811e+2	9.48	–4.663e+2	34.05
834 / 10 / 0.60	–4.427e+2	1.37	–4.423e+2	17.26	–4.427e+2	5.93	–4.427e+2	2.77	–4.380e+2	16.56	–4.423e+2	4.11	–4.215e+2	36.70
834 / 20 / 0.40	–5.093e+2	3.95	–5.093e+2	44.60	–5.093e+2	49.21	–5.093e+2	29.01	–4.971e+2	17.73	–5.088e+2	4.62	–4.427e+2	36.53
834 / 20 / 0.60	–4.523e+2	3.90	–4.527e+2	79.26	–4.523e+2	12.58	–4.523e+2	6.50	–4.322e+2	19.46	–4.508e+2	7.25	–3.771e+2	47.52
Leukemia														
1255 / 10 / 0.40	–6.578e+2	1.24	–6.578e+2	31.67	–6.578e+2	10.43	–6.578e+2	5.46	–6.504e+2	15.11	–6.577e+2	3.51	–5.544e+2	34.94
1255 / 10 / 0.60	–6.106e+2	1.67	–6.109e+2	37.30	–6.106e+2	6.73	–6.106e+2	3.28	–5.990e+2	15.21	–6.104e+2	2.41	–5.077e+2	43.63
1255 / 20 / 0.40	–7.282e+2	4.03	–7.279e+2	49.36	–7.281e+2	61.72	–7.282e+2	32.85	–7.085e+2	16.49	–7.244e+2	5.09	–3.399e+2	20.59
1255 / 20 / 0.60	–6.487e+2	3.73	–6.483e+2	53.85	–6.487e+2	30.54	–6.487e+2	14.54	–6.157e+2	15.79	–6.444e+2	6.09	–3.246e+2	42.96
Ross.large														
5643 / 10 / 0.40	–2.402e+3	3.32	–2.402e+3	76.15	–2.402e+3	25.59	–2.402e+3	18.37	–2.395e+3	145.44	–2.400e+3	7.87	2.273e+2	16.89
5643 / 10 / 0.60	–2.288e+3	6.67	–2.288e+3	138.31	–2.288e+3	53.51	–2.288e+3	35.40	–2.254e+3	148.09	–2.285e+3	14.48	3.436e+2	16.87
5643 / 20 / 0.40	–3.255e+3	8.62	–3.250e+3	44.18	–3.255e+3	69.72	–3.255e+3	38.02	–3.231e+3	176.54	–3.240e+3	3.77	4.528e+2	20.16
5643 / 20 / 0.60	–3.038e+3	10.21	–3.032e+3	86.87	–3.038e+3	115.80	–3.038e+3	58.62	–3.004e+3	174.43	–3.014e+3	4.00	6.910e+2	19.47
Ross.small														
1375 / 10 / 0.40	–6.463e+2	1.64	–6.462e+2	44.46	–6.463e+2	15.11	–6.463e+2	9.59	–6.425e+2	17.28	–6.459e+2	3.55	–6.013e+2	41.34
1375 / 10 / 0.60	–5.942e+2	1.14	–5.942e+2	32.11	–5.942e+2	10.79	–5.942e+2	5.24	–5.860e+2	18.03	–5.942e+2	2.68	–5.332e+2	36.99
1375 / 20 / 0.40	–7.849e+2	3.59	–7.849e+2	30.21	–7.849e+2	18.41	–7.849e+2	9.55	–7.707e+2	16.73	–7.830e+2	6.56	–5.145e+2	49.05
1375 / 20 / 0.60	–6.912e+2	2.01	–6.910e+2	33.78	–6.912e+2	19.91	–6.912e+2	9.59	–6.740e+2	19.52	–6.903e+2	3.97	–4.658e+2	55.80
Staunton100														
1517 / 10 / 0.40	–6.618e+2	1.36	–6.618e+2	25.56	–6.618e+2	10.07	–6.618e+2	6.54	–6.543e+2	17.30	–6.611e+2	4.27	–5.516e+2	38.00
1517 / 10 / 0.60	–6.088e+2	1.29	–6.088e+2	41.15	–6.088e+2	10.79	–6.088e+2	5.99	–5.960e+2	18.58	–6.087e+2	3.35	–5.033e+2	51.73
1517 / 20 / 0.40	–8.297e+2	2.97	–8.299e+2	39.86	–8.297e+2	18.01	–8.297e+2	9.51	–8.101e+2	21.65	–8.288e+2	6.58	–3.147e+2	35.23
1517 / 20 / 0.60	–7.365e+2	3.90	–7.364e+2	27.01	–7.365e+2	18.05	–7.365e+2	7.25	–7.048e+2	17.48	–7.331e+2	4.33	–3.312e+2	40.02
Staunton200														
2455 / 10 / 0.40	–1.044e+3	3.66	–1.044e+3	35.82	–1.044e+3	15.28	–1.044e+3	7.16	–1.031e+3	37.81	–1.044e+3	5.51	1.426e+2	1.87
2455 / 10 / 0.60	–9.771e+2	1.59	–9.771e+2	44.03	–9.771e+2	10.95	–9.771e+2	6.32	–9.605e+2	35.11	–9.769e+2	6.03	–3.756e+2	48.17
2455 / 20 / 0.40	–1.356e+3	4.08	–1.356e+3	41.87	–1.356e+3	19.91	–1.356e+3	14.97	–1.316e+3	44.24	–1.352e+3	7.06	2.755e+2	2.26
2455 / 20 / 0.60	–1.236e+3	5.06	–1.236e+3	50.07	–1.236e+3	34.91	–1.236e+3	16.52	–1.177e+3	44.39	–1.227e+3	5.50	4.434e+2	2.22
Staunton500														
3144 / 10 / 0.40	–1.326e+3	1.40	–1.326e+3	27.51	–1.326e+3	12.83	–1.326e+3	7.50	–1.304e+3	55.39	–1.326e+3	7.16	1.620e+2	4.01
3144 / 10 / 0.60	–1.254e+3	1.62	–1.253e+3	20.92	–1.254e+3	14.92	–1.254e+3	9.11	–1.230e+3	55.80	–1.253e+3	7.74	2.519e+2	4.10
3144 / 20 / 0.40	–1.756e+3	2.49	–1.756e+3	45.26	–1.756e+3	42.20	–1.756e+3	22.18	–1.695e+3	65.56	–1.750e+3	8.16	3.312e+2	4.49
3144 / 20 / 0.60	–1.621e+3	4.91	–1.621e+3	51.75	–1.621e+3	71.99	–1.621e+3	34.59	–1.545e+3	73.25	–1.616e+3	9.36	5.048e+2	4.55
eqtl.large														
3684 / 10 / 0.40	–1.964e+3	6.44	–1.963e+3	75.52	–1.964e+3	34.40	–1.964e+3	17.48	–1.950e+3	71.75	–1.962e+3	8.54	1.816e+2	6.62
3684 / 10 / 0.60	–1.877e+3	2.75	–1.877e+3	45.45	–1.877e+3	25.69	–1.877e+3	24.11	–1.842e+3	71.14	–1.876e+3	9.31	2.773e+2	6.96
3684 / 20 / 0.40	–2.332e+3	5.87	–2.332e+3	61.88	–2.332e+3	70.24	–2.332e+3	32.62	–2.128e+3	79.17	–2.297e+3	2.29	3.603e+2	7.03
3684 / 20 / 0.60	–2.173e+3	3.86	–2.173e+3	49.21	–2.173e+3	47.69	–2.173e+3	25.28	–1.927e+3	84.40	–2.135e+3	3.95	5.576e+2	7.03
eqtl.small														
1260 / 10 / 0.40	–3.375e+2	2.30	–3.375e+2	7.61	–3.375e+2	1.41	–3.375e+2	0.72	–3.371e+2	23.02	–3.375e+2	0.82	–3.355e+2	73.15
1260 / 10 / 0.60	–3.003e+2	1.87	–3.003e+2	13.43	–3.003e+2	1.55	–3.003e+2	0.94	–3.000e+2	23.91	–3.003e+2	0.98	–2.902e+2	14.09
1260 / 20 / 0.40	–5.182e+2	2.38	–5.174e+2	19.70	–5.180e+2	2.99	–5.180e+2	1.77	–5.152e+2	25.47	–5.174e+2	2.42	–4.818e+2	13.27
1260 / 20 / 0.60	–4.571e+2	2.09	–4.571e+2	10.43	–4.571e+2	4.43	–4.571e+2	3.01	–4.568e+2	25.46	–4.571e+2	0.90	–3.914e+2	9.57

TABLE 6 *The detail results of MIALM and MADMM on SPCA with real data*

(n, r, μ)	MIALM							MADMM						
	obj	time	iter	siter	η_p	η_d	η_C	obj	time	iter	siter	η_p	η_d	η_C
Arabidopsis														
834 / 10 / 0.40	−4.810e+2	2.91	49	35.2	9.04e-6	7.00e-5	1.58e-5	−4.811e+2	34.55	2610	8.1	6.89e-6	8.28e-5	1.21e-5
834 / 10 / 0.60	−4.427e+2	1.37	31	26.6	2.62e-5	7.72e-5	4.56e-5	−4.423e+2	17.26	1281	7.7	7.93e-6	8.29e-5	1.40e-5
834 / 20 / 0.40	−5.093e+2	3.95	53	40.0	5.45e-5	1.66e-4	9.89e-5	−5.093e+2	44.60	2360	10.2	1.43e-5	1.66e-4	2.61e-5
834 / 20 / 0.60	−4.523e+2	3.90	51	39.4	2.19e-5	1.52e-4	3.97e-5	−4.527e+2	79.26	5000	8.3	7.22e-6	2.02e-4	1.31e-5
Leukemia														
1255 / 10 / 0.40	−6.578e+2	1.24	26	29.1	4.65e-6	1.23e-4	8.18e-6	−6.578e+2	31.67	1789	9.8	1.28e-5	1.25e-4	2.25e-5
1255 / 10 / 0.60	−6.106e+2	1.67	31	30.8	3.75e-5	1.24e-4	6.58e-5	−6.109e+2	37.30	2151	9.5	1.23e-5	1.25e-4	2.17e-5
1255 / 20 / 0.40	−7.282e+2	4.03	49	41.5	4.51e-5	2.46e-4	8.15e-5	−7.279e+2	49.36	2119	10.8	1.97e-5	2.50e-4	3.58e-5
1255 / 20 / 0.60	−6.487e+2	3.73	48	40.6	1.44e-5	2.45e-4	2.60e-5	−6.483e+2	53.85	2269	11.3	2.19e-5	2.50e-4	3.98e-5
Ross.large														
5643 / 10 / 0.40	−2.402e+3	3.32	9	35.2	2.46e-4	4.87e-4	4.05e-4	−2.402e+3	76.15	539	9.8	6.60e-5	5.62e-4	1.16e-4
5643 / 10 / 0.60	−2.288e+3	6.67	14	40.6	2.72e-4	4.57e-4	4.56e-4	−2.288e+3	138.31	921	9.9	5.79e-5	5.63e-4	1.02e-4
5643 / 20 / 0.40	−3.255e+3	8.62	17	42.5	1.63e-4	4.80e-4	2.89e-4	−3.250e+3	44.18	399	8.4	6.66e-5	1.13e-3	1.21e-4
5643 / 20 / 0.60	−3.038e+3	10.21	20	43.4	1.11e-4	4.82e-4	2.01e-4	−3.032e+3	86.87	666	9.5	7.90e-5	1.13e-3	1.44e-4
Ross.small														
1375 / 10 / 0.40	−6.463e+2	1.64	35	31.2	6.61e-6	1.05e-4	1.16e-5	−6.462e+2	44.46	2842	8.6	1.14e-5	1.37e-4	2.01e-5
1375 / 10 / 0.60	−5.942e+2	1.14	27	25.4	1.12e-5	1.02e-4	1.97e-5	−5.942e+2	32.11	2067	8.5	1.24e-5	1.37e-4	2.19e-5
1375 / 20 / 0.40	−7.849e+2	3.59	44	40.0	4.51e-5	2.72e-4	8.17e-5	−7.849e+2	30.21	1259	10.6	2.51e-5	2.74e-4	4.56e-5
1375 / 20 / 0.60	−6.912e+2	2.01	27	32.9	4.87e-5	2.70e-4	8.84e-5	−6.910e+2	33.78	1580	9.7	2.49e-5	2.74e-4	4.53e-5
Staunton100														
1517 / 10 / 0.40	−6.618e+2	1.36	28	29.3	2.78e-5	1.26e-4	4.88e-5	−6.618e+2	25.56	1573	9.3	1.38e-5	1.51e-4	2.44e-5
1517 / 10 / 0.60	−6.088e+2	1.29	28	27.9	6.96e-6	1.32e-4	1.22e-5	−6.088e+2	41.15	2493	9.2	1.31e-5	1.52e-4	2.30e-5
1517 / 20 / 0.40	−8.297e+2	2.97	25	33.0	2.67e-5	2.91e-4	4.82e-5	−8.299e+2	39.86	1464	10.9	2.70e-5	3.03e-4	4.90e-5
1517 / 20 / 0.60	−7.365e+2	3.90	41	39.1	1.57e-5	2.55e-4	2.83e-5	−7.364e+2	27.01	1075	10.2	2.84e-5	3.03e-4	5.17e-5
Staunton200														
2455 / 10 / 0.40	−1.044e+3	3.66	20	27.4	7.92e-6	2.14e-4	1.39e-5	−1.044e+3	35.82	980	8.8	2.28e-5	2.45e-4	4.01e-5
2455 / 10 / 0.60	−9.771e+2	1.59	18	25.8	3.37e-5	2.10e-4	5.87e-5	−9.771e+2	44.03	1069	8.7	2.08e-5	2.45e-4	3.66e-5
2455 / 20 / 0.40	−1.356e+3	4.08	29	39.6	8.90e-5	4.09e-4	1.59e-4	−1.356e+3	41.87	993	10.1	4.99e-5	4.89e-4	9.07e-5
2455 / 20 / 0.60	−1.236e+3	5.06	23	37.0	5.22e-5	3.66e-4	9.44e-5	−1.236e+3	50.07	985	10.4	4.72e-5	4.89e-4	8.58e-5
Staunton500														
3144 / 10 / 0.40	−1.326e+3	1.40	14	24.0	2.52e-5	2.70e-4	4.43e-5	−1.326e+3	27.51	608	8.7	2.75e-5	3.13e-4	4.83e-5
3144 / 10 / 0.60	−1.254e+3	1.62	14	25.8	7.35e-5	2.46e-4	1.25e-4	−1.253e+3	20.92	490	8.5	2.43e-5	3.12e-4	4.27e-5
3144 / 20 / 0.40	−1.756e+3	2.49	13	39.2	8.75e-5	5.67e-4	1.56e-4	−1.756e+3	45.26	661	10.6	5.99e-5	6.20e-4	1.09e-4
3144 / 20 / 0.60	−1.621e+3	4.91	17	39.8	9.25e-5	5.74e-4	1.67e-4	−1.621e+3	51.75	907	9.8	4.65e-5	6.25e-4	8.45e-5
eqtl.large														
3684 / 10 / 0.40	−1.964e+3	6.44	13	33.8	9.43e-5	3.37e-4	1.59e-4	−1.963e+3	75.52	883	8.8	2.23e-5	3.68e-4	3.91e-5
3684 / 10 / 0.60	−1.877e+3	2.75	13	32.8	3.66e-5	3.38e-4	6.43e-5	−1.877e+3	45.45	656	9.3	3.44e-5	3.67e-4	6.06e-5
3684 / 20 / 0.40	−2.332e+3	5.87	23	44.3	4.60e-5	5.10e-4	8.30e-5	−2.332e+3	61.88	885	11.1	6.92e-5	7.33e-4	1.26e-4
3684 / 20 / 0.60	−2.173e+3	3.86	16	41.8	5.45e-5	5.25e-4	9.84e-5	−2.173e+3	49.21	677	11.7	7.91e-5	7.36e-4	1.44e-4
eqtl.small														
1260 / 10 / 0.40	−3.375e+2	2.30	35	39.1	1.52e-5	1.17e-4	2.66e-5	−3.375e+2	7.61	605	6.3	1.28e-5	1.25e-4	2.25e-5
1260 / 10 / 0.60	−3.003e+2	1.87	33	31.6	1.26e-5	1.21e-4	2.22e-5	−3.003e+2	13.43	1069	7.0	1.12e-5	1.25e-4	1.96e-5
1260 / 20 / 0.40	−5.182e+2	2.38	31	35.0	4.00e-5	2.15e-4	7.23e-5	−5.174e+2	19.70	1155	7.6	2.31e-5	2.51e-4	4.19e-5
1260 / 20 / 0.60	−4.571e+2	2.09	31	31.1	3.17e-5	2.16e-4	5.75e-5	−4.571e+2	10.43	641	7.2	1.51e-5	2.50e-4	2.74e-5

References

Absil, P.-A., Baker, C. G. & Gallivan, K. A. (2007) Trust-region methods on Riemannian manifolds. *Found. Comput. Math.*, **7**, 303–330.

Absil, P.-A. & Hosseini, S. (2019) A collection of nonsmooth Riemannian optimization problems. *Nonsmooth Optimization and Its Applications*. Cham: Birkhäuser/Springer, pp. 1–15.

Absil, P.-A., Mahony, R. & Sepulchre, R. (2009) *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ: Princeton University Press.

Baker, C. G., Absil, P.-A. & Gallivan, K. A. (2008) An implicit trust-region method on Riemannian manifolds. *IMA J. Numer. Anal.*, **28**, 665–689.

Beck, A. (2017) *First-Order Methods in Optimization*. New Delhi: Society for Industrial and Applied Mathematics.

Bento, G. C., da Cruz Neto, J. X. & Oliveira, P. R. (2011) Convergence of inexact descent methods for nonconvex optimization on Riemannian manifolds. Preprint available at arXiv:1103.4828.

de Carvalho Bento, G., da Cruz Neto, J. X. & Oliveira, P. R. (2016) A new approach to the proximal point method: convergence on general Riemannian manifolds. *J. Optim. Theory Appl.*, **168**, 743–755.

Bento, G. C., Ferreira, O. P. & Melo, J. G. (2017) Iteration-complexity of gradient, subgradient and proximal point methods on Riemannian manifolds. *J. Optim. Theory Appl.*, **173**, 548–562.

Bertsekas, D. P. (1997) Nonlinear programming. *J. Oper. Res. Soc.*, **48**, 334–334.

Bortoloti, M. A. d. A., Fernandes, T. A., Ferreira, O. P. & Yuan, J. (2020) Damped Newton's method on Riemannian manifolds. *J. Global Optim.*, **77**, 643–660.

Boumal, N. (2015) Riemannian trust regions with finite-difference hessian approximations are globally convergent. *Geometric Science of Information*. Cham: Springer, pp. 467–475.

Boumal, N., Absil, P.-A. & Cartis, C. (2018) Global rates of convergence for nonconvex optimization on manifolds. *IMA J. Numer. Anal.*, **39**, 1–33.

Burke, J. V., Curtis, F. E., Lewis, A. S., Overton, M. L. & Simões, L. E. (2020) Gradient sampling methods for nonsmooth optimization. *Numerical Nonsmooth Optimization: State of the Art Algorithms*, Cham: Springer, pp. 201–225.

Cambier, L. & Absil, P.-A. (2016) Robust low-rank matrix completion by Riemannian optimization. *SIAM J. Sci. Comput.*, **38**, S440–S460.

Chen, S., Ma, S., Man-Cho So, A. & Zhang, T. (2020) Proximal gradient method for nonsmooth optimization over the Stiefel manifold. *SIAM J. Optim.*, **30**, 210–239.

Chen, W., Ji, H. & You, Y. (2016) An augmented Lagrangian method for 1-regularized optimization problems with orthogonality constraints. *SIAM J. Sci. Comput.*, **38**, B570–B592.

Culhane, A. C., Perrière, G. & Higgins, D. G. (2003) Cross-platform comparison and visualisation of gene expression data using co-inertia analysis. *BMC Bioinform.*, **4**, 1–15.

de Oliveira, F. R. & Ferreira, O. P. (2020) Newton method for finding a singularity of a special class of locally Lipschitz continuous vector fields on Riemannian manifolds. *J. Optim. Theory Appl.*, **185**, 522–539.

Dolan, E. D. & Moré, J. J. (2002) Benchmarking optimization software with performance profiles. *Math. Programming*, **91**, 201–213.

Ferreira, O. P., Louzeiro, M. S. & Prudente, L. (2019) Iteration-complexity of the subgradient method on Riemannian manifolds with lower bounded curvature. *Optimization*, **68**, 713–729.

Ferreira, O. P. & Silva, R. C. (2012) Local convergence of Newton's method under a majorant condition in Riemannian manifolds. *IMA J. Numer. Anal.*, **32**, 1696–1713.

Gohary, R. H. & Davidson, T. N. (2009) Noncoherent mimo communication: Grassmannian constellations and efficient detection. *IEEE Trans. Inform. Theory*, **55**, 1176–1205.

Grohs, P. & Hosseini, S. (2015) Nonsmooth trust region algorithms for locally Lipschitz functions on Riemannian manifolds. *IMA J. Numer. Anal.*, **36**, 1167–1192.

Grohs, P. & Hosseini, S. (2016) ϵ-Subgradient algorithms for locally lipschitz functions on Riemannian manifolds. *Adv. Comput. Math.*, **42**, 333–360.

Hong, Z., Zhang, X., Chu, D. & Liao, L. (2017) Nonconvex and nonsmooth optimization with generalized orthogonality constraints: an approximate augmented Lagrangian method. *J. Sci. Comput.*, **72**, 1–42.

Hosseini, S. (2015) Convergence of nonsmooth descent methods via Kurdyka–Lojasiewicz inequality on Riemannian manifolds. *Hausdorff Center for Mathematics and Institute for Numerical Simulation*. Bonn, Germany: University of Bonn.

Hosseini, S., Huang, W. & Yousefpour, R. (2018) Line search algorithms for locally Lipschitz functions on Riemannian manifolds. *SIAM J. Optim.*, **28**, 596–619.

Hosseini, S. & Uschmajew, A. (2017) A Riemannian gradient sampling algorithm for nonsmooth optimization on manifolds. *SIAM J. Optim.*, **27**, 173–189.

Huang, W., Absil, P.-A. & Gallivan, K. A. (2015a) A Riemannian symmetric rank-one trust-region method. *Math. Programming*, **150**, 179–216.

Huang, W., Gallivan, K. A. & Absil, P.-A. (2015b) A Broyden class of quasi-Newton methods for Riemannian optimization. *SIAM J. Optim.*, **25**, 1660–1685.

Iannazzo, B. & Porcelli, M. (2018) The Riemannian Barzilai–Borwein method with nonmonotone line search and the matrix geometric mean computation. *IMA J. Numer. Anal.*, **38**, 495–517.

Kovnatsky, A., Glashoff, K. & Bronstein, M. M. (2016) MADMM: a generic algorithm for non-smooth optimization on manifolds. *Computer Vision—ECCV 2016: 14th European Conference*. Amsterdam, The Netherlands, 11–14 October 2016, Proceedings, Part V. NY: Springer, pp. 680–696.

Lai, R. & Osher, S. (2014) A splitting method for orthogonality constrained problems. *J. Sci. Comput.*, **58**, 431–449.

Li, L. & Toh, K.-C. (2010) An inexact interior point method for ℓ_1-regularized sparse covariance selection. *Math. Program. Comput.*, **2**, 291–315.

Li, X., Chen, S., Deng, Z., Qu, Q., Zhu, Z. & Man-Cho So, A. (2021) Weakly convex optimization over Stiefel manifold using Riemannian subgradient-type methods. *SIAM J. Optim.*, **31**, 1605–1634.

Liu, C. & Boumal, N. (2020) Simple algorithms for optimization on Riemannian manifolds with constraints. *Appl. Math. Optim.*, **82**, 949–981.

Ozolinš, V., Lai, R., Caflisch, R. & Osher, S. (2013) Compressed modes for variational problems in mathematics and physics. *Proc. Natl. Acad. Sci.*, **110**, 18368–18373.

Qu, B., Liu, B. & Zheng, N. (2015) On the computation of the step-size for the cq-like algorithms for the split feasibility problem. *Appl. Math. Comput.*, **262**, 218–223.

Sahin, M. F., Eftekhari, A., Alacaoglu, A., Gómez, F. L. & Cevher, V. (2019) An inexact augmented Lagrangian framework for nonconvex optimization with nonlinear constraints. *Proceedings of NeurIPS 2019. Curran Associates, Inc.*

Wang, P., Liu, H. & So, A. M.-C. (2019) Globally convergent accelerated proximal alternating maximization method for ℓ_1-principal component analysis. *ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. NJ: IEEE, pp. 8147–8151.

Yang, W. H., Zhang, L. H. & Song, R. (2014) Optimality conditions for the nonlinear programming problems on Riemannian manifolds. *Pac. J. Optim.*, **10**, 415–434.

Yuan, X., Huang, W., Absil, P.-A. & Gallivan, K. A. (2017) *A Riemannian Quasi-Newton Method for Computing the Karcher Mean of Symmetric Positive Definite Matrices*. Florida State: Florida State University. (FSU17-02).

Zhang, H. & Sra, S. (2016) First-order methods for geodesically convex optimization. *Conference on Learning Theory, Curran Associates*, pp. 1617–1638.

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

Zheng, L. & Tse, D. N. C. (2002) Communication on the Grassmann manifold: a geometric approach to the noncoherent multiple-antenna channel. *IEEE Trans. Inform. Theory*, **48**, 359–383.

Zhu, J., Zhang, B., Smith, E. N., Drees, B., Brem, R. B., Kruglyak, L., Bumgarner, R. E. & Schadt, E. E. (2008) Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. *Nat. Gen.*, **40**, 854–861.

Zou, H., Hastie, T. & Tibshirani, R. (2006) Sparse principal component analysis. *J. Comput. Graph. Statist.*, **15**, 265–286.