



Simple Algorithms for Optimization on Riemannian Manifolds with Constraints

Changshuo Liu^{1,2} · Nicolas Boumal²

Published online: 28 March 2019

© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

We consider optimization problems on manifolds with equality and inequality constraints. A large body of work treats constrained optimization in Euclidean spaces. In this work, we consider extensions of existing algorithms from the Euclidean case to the Riemannian case. Thus, the variable lives on a known smooth manifold and is further constrained. In doing so, we exploit the growing literature on unconstrained Riemannian optimization. For the special case where the manifold is itself described by equality constraints, one could in principle treat the whole problem as a constrained problem in a Euclidean space. The main hypothesis we test here is whether it is sometimes better to exploit the geometry of the constraints, even if only for a subset of them. Specifically, this paper extends an augmented Lagrangian method and smoothed versions of an exact penalty method to the Riemannian case, together with some fundamental convergence results. Numerical experiments indicate some gains in computational efficiency and accuracy in some regimes for minimum balanced cut, non-negative PCA and k -means, especially in high dimensions.

Keywords Riemannian optimization · Constrained optimization · Differential geometry · Augmented Lagrangian method · Exact penalty method · Nonsmooth optimization

Mathematics Subject Classification 65K05 · 90C30 · 53A99

✉ Nicolas Boumal
nboumal@math.princeton.edu

Changshuo Liu
liuchangshuo@outlook.com

¹ PACM and Mathematics Department, Princeton University, Princeton, NJ, USA

² Mathematics Department, Princeton University, Princeton, NJ, USA

1 Introduction

We consider the following problem:

$$\begin{aligned}
 & \min_x && f(x) \\
 & \text{subject to} && x \in \mathcal{M} \\
 & && g_i(x) \leq 0 \text{ for } i \in \mathcal{I} = \{1, \dots, n\}, \\
 & && h_j(x) = 0 \text{ for } j \in \mathcal{E} = \{n+1, \dots, n+m\},
 \end{aligned} \tag{1}$$

where \mathcal{M} is a Riemannian manifold and $f, \{g_i\}, \{h_j\}$ are twice continuously differentiable functions from \mathcal{M} to \mathbb{R} . The problems of this class have extra constraints in addition to the manifold constraint. Following the convention, we call problem (1) an Equality Constrained Problem (ECP) when only equality constraints exist, and an Inequality Constrained Problem (ICP) when only inequality constraints exist. If both equality and inequality constraints are present, we call it a Mixed Constrained Problem (MCP). Such problems feature naturally in applications. For instance, non-negative principal component analysis (PCA) is formulated as an optimization problem on a sphere in \mathbb{R}^n with non-negativity constraints on each entry [54]. As another example, k -means can be formulated as a constrained optimization problem on the Stiefel manifold [23]. We discuss these more in Sect. 5.

Necessary and sufficient optimality conditions for the general problem class (1) were derived in [53] and also recently in [10]—we summarize them in the next section. Some algorithmic approaches have been put forward in [14,29,39,40,55]. Nevertheless, and somewhat surprisingly, we find that there has been no systematic effort to survey and compare some of the most direct approaches to solve (1) based on prior work on the same problem class without the manifold constraint [12] and with only the manifold constraint [2].

Part of the reason may be that, in many applications, the manifold \mathcal{M} is a submanifold of a Euclidean space, itself defined by equality constraints. In such cases, the manifold constraint can be treated as an additional set of equality constraints, and the problem can be solved using the rich expertise gained over the years for constrained optimization in \mathbb{R}^n . There are also existing software packages for it, such as Lancelot, KNITRO and Algenca [16,20,27].

Yet, based on the literature for unconstrained optimization on manifolds, we see that if the manifold \mathcal{M} is nice enough, it pays to exploit its structure fully. In particular, much is now understood about optimizing over spheres, orthogonal groups, the Stiefel manifold of orthonormal matrices, the set of fixed-rank matrices, and many more. Furthermore, embracing the paradigm of optimization on manifolds also allows us to treat problems of class (1) where \mathcal{M} is an abstract manifold, such as the Grassmannian manifold of linear subspaces. Admittedly, owing to Whitney’s embedding theorem, abstract manifolds can also be embedded in a Euclidean space, and hence even those problems could in principle be treated using algorithms from the classical literature, but the mere existence of an embedding is often of little practical use.

In this paper, we survey some of the more classical methods for constrained optimization in \mathbb{R}^n and straightforwardly extend them to the more general class (1), while

preserving and exploiting the smooth geometry of \mathcal{M} . For each method, we check if some of the essential convergence results known in \mathbb{R}^n extend as well. Then, we set up a number of numerical experiments for a few applications and report performance profiles. Our purpose in doing so is to gain some perspective as to which methods are more likely to yield reliable generic software for this problem class.

1.1 Contributions

As a first contribution, we study the augmented Lagrangian method (ALM) and the exact penalty method on general Riemannian manifolds. For ALM, we study local and global convergence properties. For the exact penalty method, each iteration involves the minimization of a sum of maximum functions, which we call a mini-sum-max problem. For this nonsmooth optimization subproblem, we study and tailor two types of existing algorithms—subgradient descent and smoothing methods. In particular, we propose a robust subgradient method without gradient sampling for mini-sum-max problems, which may be of interest in itself. For smoothing methods, we study the effect of two classical smoothing functions: Log-sum-exp and ‘Linear-Quadratic + Huber loss’.

As a second contribution, we perform numerical experiments on non-negative PCA, k -means and the minimum balanced cut problems to showcase the strengths and weaknesses of each algorithm. As a baseline, we compare our approach to the traditional approach, which would simply consider the manifold as an extra set of equality constraints. For this, we use `fmincon`: a general-purpose nonlinear programming solver in Matlab. For these problems, we find that some of our methods perform better than `fmincon` in high dimensional scenarios. Our solvers are generally slower in the low-dimensional case.

We present the algorithms and convergence analyses for ALM and the exact penalty method on Riemannian manifolds in Sects. 3 and 4 respectively. Proofs are deferred to the appendix. Numerical experiments on various applications are reported in Sect. 5.

1.2 Related Literature

Dreisigmeyer [29] tackles ICPs on Euclidean submanifolds by pulling back the inequality constraints onto the tangent spaces of \mathcal{M} , and running direct search methods there. Yang et al. [53] provide necessary optimality conditions for our problem class (1). Bergmann and Herzog [10] extend a range of constraint qualifications from the Euclidean setting to the smooth manifold setting. Kovnatsky et al. [40] generalize the alternating direction method of multipliers (ADMM) to the Riemannian case—the method handles nonsmooth optimization on manifolds via variable splitting, which produces an ECP. In [14], Birgin et al. deal with MCP in the Euclidean case by splitting constraints into upper-level and lower-level, and perform ALM on upper-level constrained optimization problems in a search space confined by lower-level constraints. When \mathcal{M} is a submanifold of a Euclidean space chosen to describe the lower-level constraints, our Riemannian ALM reduces to their method. Khuzani and Li [39] propose a primal-dual method to address ICP on Riemannian manifolds with bounded

sectional curvature in a stochastic setting. Zhang et al. [55] propose an ADMM-like primal-dual method for nonconvex, nonsmooth optimization problems on submanifolds of Euclidean space coupled with linear constraints. Weber and Sra [52] recently study a Franke–Wolfe method on manifolds to design projection-free methods for constrained, geodesically convex optimization on manifolds.

2 Preliminaries and Notations

We briefly review some relevant concepts from Riemannian geometry, following the notations of [2]. Let the Riemannian manifold \mathcal{M} be endowed with a Riemannian metric $\langle \cdot, \cdot \rangle_x$ on each tangent space $T_x\mathcal{M}$, where x is in \mathcal{M} . Let $\| \cdot \|_x$ be the associated norm. We often omit the subscript x when it is clear from context. Throughout the paper, we assume that \mathcal{M} is a complete, smooth, finite-dimensional Riemannian manifold.

2.1 Gradients and Hessians on Manifolds

The gradient at x of a smooth function $f : \mathcal{M} \rightarrow \mathbb{R}$, $\text{grad } f(x)$, is defined as the unique tangent vector at x such that

$$\langle \text{grad } f(x), v \rangle_x = Df(x)[v], \quad \forall v \in T_x\mathcal{M},$$

where the right-hand side is the directional derivative of f at x along v . Let $\text{dist}(x, y)$ denote the Riemannian distance between $x, y \in \mathcal{M}$. For each $x \in \mathcal{M}$, let $\text{Exp}_x : T_x\mathcal{M} \rightarrow \mathcal{M}$ denote the exponential map at x (that is, the map such that $t \mapsto \text{Exp}_x(tv)$ is a geodesic passing through x at $t = 0$ with velocity $v \in T_x\mathcal{M}$). The injectivity radius is defined as:

$$i(\mathcal{M}) = \inf_{x \in \mathcal{M}} \sup\{\epsilon > 0 : \text{Exp}_x|_{\{\eta \in T_x\mathcal{M} : \|\eta\| < \epsilon\}} \text{ is a diffeomorphism}\}. \quad (2)$$

For any $x, y \in \mathcal{M}$ with $\text{dist}(x, y) < i(\mathcal{M})$, there is a unique minimizing geodesic connecting them, which gives rise to a parallel transport operator $\mathcal{P}_{x \rightarrow y} : T_x\mathcal{M} \rightarrow T_y\mathcal{M}$ as an isometry between two tangent spaces.

Definition 2.1 (*Hessian*, [2], *Definition 5.5.1*) Given a smooth function $f : \mathcal{M} \rightarrow \mathbb{R}$, the Riemannian Hessian of f at a point x in \mathcal{M} is the linear mapping $\text{Hess } f(x)$ of $T_x\mathcal{M}$ into itself defined by

$$\text{Hess } f(x)[\xi_x] = \nabla_{\xi_x} \text{grad } f$$

for all ξ_x in $T_x\mathcal{M}$, where ∇ is the Riemannian connection on \mathcal{M} .

Using the exponential map, one can also understand the Riemannian gradient and Hessian through the following identities [2, Eq. (4.4) and Prop. 5.5.4]:

$$\text{grad } f(x) = \text{grad}(f \circ \text{Exp}_x)(0_x), \quad \text{Hess } f(x) = \text{Hess}(f \circ \text{Exp}_x)(0_x), \quad (3)$$

where 0_x is the zero vector in $T_x\mathcal{M}$ and $f \circ \text{Exp}_x : T_x\mathcal{M} \rightarrow \mathbb{R}$ is defined on a Euclidean space with a metric, so that its gradient and Hessian are defined in the usual sense. The composition $f \circ \text{Exp}_x$ is also called the pullback of f to the tangent space at x .

2.2 Optimality Conditions

Let Ω denote the set of feasible points on \mathcal{M} satisfying the constraints in (1). At $x \in \mathcal{M}$, let \mathcal{I} denote the set of n inequality constraints, and \mathcal{E} the set of m equality constraints. Let $\mathcal{A}(x)$ denote the active set of constraints, that is,

$$\mathcal{A}(x) = \mathcal{E} \cup \{i \in \mathcal{I} | g_i(x) = 0\}. \tag{4}$$

The Lagrangian of (1) is defined similarly to the Euclidean case as

$$\mathcal{L}(x, \lambda, \gamma) = f(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x) + \sum_{j \in \mathcal{E}} \gamma_j h_j(x), \tag{5}$$

where λ_i, γ_j are vector entries of $\lambda \in \mathbb{R}^n$ and $\gamma \in \mathbb{R}^m$. Following [53], constraint qualifications and optimality conditions are generalized as follows:

Definition 2.2 (LICQ, [53], eq. (4.3)) Linear independence constraint qualifications (LICQ) are said to hold at $x \in \mathcal{M}$ if

$$\{\text{grad } g_i(x), \text{grad } h_j(x), i \in \mathcal{A}(x) \cap \mathcal{I}, j \in \mathcal{E}\} \text{ are linearly independent in } T_x\mathcal{M}. \tag{6}$$

Definition 2.3 (First-Order Necessary Conditions (KKT conditions), [53], eq. (4.8)) Given an MCP as in (1), $x^* \in \Omega$ is said to satisfy KKT conditions if there exist Lagrange multipliers λ^* and γ^* such that the following hold:

$$\begin{aligned} &\text{grad } f(x^*) + \sum_{i \in \mathcal{I}} \lambda_i^* \text{grad } g_i(x) + \sum_{j \in \mathcal{E}} \gamma_j^* \text{grad } h_j(x) = 0, \\ &h_j(x^*) = 0, \text{ for all } j \in \mathcal{E}, \text{ and} \\ &g_i(x^*) \leq 0, \lambda_i^* \geq 0, \lambda_i^* g_i(x^*) = 0 \text{ for all } i \in \mathcal{I}. \end{aligned} \tag{7}$$

For the purpose of identifying second-order optimality conditions at x^* with associated λ^* and γ^* , consider the critical cone $F(x^*, \lambda^*, \gamma^*)$ inside the tangent space at x^* defined as follows; see [53, §4.2], [47, §12.4]:

$$v \in F(x^*, \lambda^*, \gamma^*) \Leftrightarrow \begin{cases} v \in T_{x^*}\mathcal{M}, \\ \langle \text{grad } h_j(x^*), v \rangle = 0 & \text{for all } j \in \mathcal{E}, \\ \langle \text{grad } g_i(x^*), v \rangle = 0 & \text{for all } i \in \mathcal{A}(x^*) \cap \mathcal{I} \text{ with } \lambda_i^* > 0, \text{ and} \\ \langle \text{grad } g_i(x^*), v \rangle \leq 0 & \text{for all } i \in \mathcal{A}(x^*) \cap \mathcal{I} \text{ with } \lambda_i^* = 0. \end{cases} \tag{8}$$

Definition 2.4 (*Second-Order Necessary Conditions (SONC)*, [53], Theorem 4.2) Given an MCP as in (1), $x^* \in \Omega$ is said to satisfy SONCs if it satisfies KKT conditions with associated Lagrange multiplier λ^* and γ^* , and if

$$\langle v, \text{Hess}_x \mathcal{L}(x^*, \lambda^*, \gamma^*)v \rangle \geq 0, \text{ for any } v \in F(x^*, \lambda^*, \gamma^*),$$

where the Hessian is taken with respect to the first variable of \mathcal{L} , on \mathcal{M} .

Definition 2.5 (*Second-Order Sufficient Conditions (SOSC)*, [53], Theorem 4.3) Given an MCP as in (1), $x^* \in \Omega$ is said to satisfy SOSCs if it satisfies KKT conditions with associated Lagrange multiplier λ^* and γ^* , and if

$$\langle v, \text{Hess}_x \mathcal{L}(x^*, \lambda^*, \gamma^*)v \rangle > 0, \text{ for any } v \in F(x^*, \lambda^*, \gamma^*), v \neq 0.$$

Proposition 2.6 ([53], Theorem 4.1, 4.2) *If x^* is a local minimum of a given MCP and LICQ holds at x^* , then x^* satisfies KKT conditions and SONCs.*

Proposition 2.7 ([53], Theorem 4.3) *If x^* satisfies SOSCs for a given MCP, then it is a strict local minimum.*

3 Riemannian Augmented Lagrangian Methods

The augmented Lagrangian method (ALM) is a popular algorithm for constrained nonlinear programming of the form of (1) with $\mathcal{M} = \mathbb{R}^d$. At its core, ALM relies on the definition of the *augmented* Lagrangian function [16, eq. (4.3)]:

$$\mathcal{L}_\rho(x, \lambda, \gamma) = f(x) + \frac{\rho}{2} \left(\sum_{j \in \mathcal{E}} \left(h_j(x) + \frac{\gamma_j}{\rho} \right)^2 + \sum_{i \in \mathcal{I}} \max \left\{ 0, \frac{\lambda_i}{\rho} + g_i(x) \right\}^2 \right), \tag{9}$$

where $\rho > 0$ is a penalty parameter, and $\gamma \in \mathbb{R}^m, \lambda \in \mathbb{R}^n, \lambda \geq 0$. ALM alternates between updating x and updating (λ, γ) . To update x , any algorithm for unconstrained optimization may be adopted to minimize (9) with (λ, γ) fixed. We shall call the chosen solver the *subsolver*. To update (λ, γ) , a clipped gradient-type update rule is used. A vast literature covers ALM in the Euclidean case. We direct the reader in particular to the recent monograph by Birgin and Martínez [16].

The Lagrangian function as defined in (9) generalizes seamlessly to the Riemannian case simply by restricting x to live on the manifold \mathcal{M} . Importantly, \mathcal{L}_ρ is continuously differentiable in x under our assumptions. The corresponding ALM algorithm is easily extended to the Riemannian case as well: subsolvers are now optimization algorithms for unconstrained optimization on manifolds. We refer to this approach as Riemannian ALM, or RALM; see Algorithm 1, in which the clip operator is defined by

$$\text{clip}_{[a,b]}(x) = \max\{a, \min(b, x)\}.$$

Algorithm 1: Riemannian augmented Lagrangian method (RALM)

```

1 Require: Riemannian manifold  $\mathcal{M}$ , twice continuously differentiable functions  $f, \{g_i\}_{i \in \mathcal{I}}, \{h_j\}_{j \in \mathcal{E}}: \mathcal{M} \rightarrow \mathbb{R}$ .
2 Input: Starting point  $x_0 \in \mathcal{M}$ , starting Lagrangian vectors  $\lambda^0 \in \mathbb{R}^n, \gamma^0 \in \mathbb{R}^m$ , accuracy tolerance  $\epsilon_{\min}$ , starting accuracy  $\epsilon_0 > 0$ , starting penalty coefficient  $\rho_0$ , constants  $\theta_\epsilon \in (0, 1), \theta_\rho > 1$ , multiplier boundaries  $\lambda^{\max} \in \mathbb{R}^n, \gamma^{\min}, \gamma^{\max} \in \mathbb{R}^m$  with  $\gamma_i^{\min} \leq \gamma_i^{\max}$  for each  $i \in \mathcal{I}$ , ratio  $\theta_\sigma \in (0, 1)$ , minimum step size  $d_{\min}$ .
3 for  $k = 0, 1, \dots$  do
4   Compute  $x_{k+1}$ —an approximate solution to the following problem within a tolerance  $\epsilon_k$ :
      
$$\min_{x \in \mathcal{M}} \mathcal{L}_{\rho_k}(x, \lambda^k, \gamma^k). \tag{10}$$

5   if  $\text{dist}(x_k, x_{k+1}) < d_{\min}$  and  $\epsilon_k \leq \epsilon_{\min}$  then
6     | Return  $x_{k+1}$ ;
7   end
8    $\gamma_j^{k+1} = \text{clip}_{[\gamma_j^{\min}, \gamma_j^{\max}]}(\gamma_j^k + \rho_k h_j(x_{k+1}))$ , for  $j \in \mathcal{E}$ ;
9    $\lambda_i^{k+1} = \text{clip}_{[\lambda_i^{\min}, \lambda_i^{\max}]}(\lambda_i^k + \rho_k g_i(x_{k+1}))$ , for  $i \in \mathcal{I}$ ;
10   $\sigma_i^{k+1} = \max \left\{ g_i(x_{k+1}), -\frac{\lambda_i^k}{\rho_k} \right\}$ , for  $i \in \mathcal{I}$ ;
11   $\epsilon_{k+1} = \max \{ \epsilon_{\min}, \theta_\epsilon \epsilon_k \}$ ;
12  if  $k = 0$  or  $\max_{j \in \mathcal{E}, i \in \mathcal{I}} \{ |h_j(x_{k+1})|, |\sigma_i^{k+1}| \} \leq \theta_\sigma \max_{j \in \mathcal{E}, i \in \mathcal{I}} \{ |h_j(x_k)|, |\sigma_i^k| \}$  then
13    |  $\rho_{k+1} = \rho_k$ ;
14  else
15    |  $\rho_{k+1} = \theta_\rho \rho_k$ ;
16  end
17 end

```

In practice, solving (10) (approximately) involves running any standard algorithm for smooth, unconstrained optimization on manifolds with warm-start at x_k . Various kinds of tolerances for this subproblem solve will be discussed, which lead to different convergence results. We note that this approach of separating out a subset of the constraints that have special, exploitable structure in ALM is in the same spirit as the general approach in [5] and in [12, §2.4].

Notice that, in Algorithm 1, there are safeguards designed for multipliers and a conditioned update on the penalty coefficient ρ —updates are executed only when constraint violations are shrinking fast enough, which helps alleviate the effect of ill-conditioning and improves robustness; see [16,38]. As each subproblem (10) is an unconstrained problem with sufficiently smooth objective function, various Riemannian optimization methods can be used. In particular, we mention the Riemannian gradient descent, non-linear conjugate gradients and trust-regions methods, all described in [2] and available in ready-to-use toolboxes [18,35,51].

Global and local convergence properties of ALM in the Euclidean case have been studied extensively; see [5,6,13–15], among others. Some of the results available in the literature are phrased in sufficiently general terms that they readily apply to the Riemannian case, even though the Riemannian case was not necessarily explicitly acknowledged. This applies to Proposition 3.1 below. To extend certain other results,

minor modifications to the Euclidean proofs are necessary. Propositions 3.2 and 3.4 are among those. Here and in the next section, we state some of the more relevant results explicitly for the Riemannian case. For the sake of completeness, we include full proofs in the appendix, stressing again that they are either existing proofs or simple adaptations of existing proofs.

Whether or not Algorithm 1 converges depends on the tolerances for the subproblems, and the ability of the subsolver to return a point that satisfies them. In general, solving the unconstrained subproblem (9) to global optimality—even within a tolerance on the objective function value—is hard. In practice, that step is implemented by running a local optimization solver which, often, can only guarantee convergence to an approximate first- or second-order stationary point. Nevertheless, practice suggests that for many applications these local solvers perform well. An important question then becomes: assuming the subproblems are indeed solved within appropriate tolerances, does Algorithm 1 converge? In the following three propositions, we partially characterize the limit points generated by the algorithm assuming either that the subproblems are solved almost to global optimality, which is difficult to guarantee, or assuming approximate stationary points are computed, which we can guarantee [3,8,17,56].

We first consider the case when we have a global subsolver up to some tolerance on the cost for each iteration. This affords the following result:

Proposition 3.1 *In Algorithm 1 with $\epsilon_{\min} = 0$ (so that the algorithm produces an infinite sequence with $\epsilon_k \rightarrow 0$), if at each iteration k the subsolver produces a point x_{k+1} satisfying*

$$\mathcal{L}_{\rho_k}(x_{k+1}, \lambda^k, \gamma^k) \leq \mathcal{L}_{\rho_k}(z, \lambda^k, \gamma^k) + \epsilon_k, \quad (11)$$

where z is a feasible global minimizer of the original MCP, and if $\{x_k\}_{k=0}^{\infty}$ has a limit point \bar{x} , then \bar{x} is a global minimizer of the original MCP.

Proof See Theorems 1 and 2 in [15]. The proofs in that reference apply verbatim to the Riemannian case. \square

However, most Riemannian optimization methods only return approximately first- or second-order approximate stationary points. In the first-order case, we have the following:

Proposition 3.2 *In Algorithm 1 with $\epsilon_{\min} = 0$, if at each iteration k the subsolver produces a point x_{k+1} satisfying*

$$\|\text{grad}_x \mathcal{L}_{\rho_k}(x_{k+1}, \lambda^k, \gamma^k)\| \leq \epsilon_k, \quad (12)$$

and if the sequence $\{x_k\}_{k=0}^{\infty}$ has a limit point $\bar{x} \in \Omega$ where LICQ is satisfied, then \bar{x} satisfies KKT conditions of the original MCP.

Proof The proof is an easy adaptation of that of Theorem 4.2 in [5]: see Appendix A. \square

In the second-order case, we consider problems with equality constraints only, because when inequality constraints are present the augmented function (9) may not be twice differentiable. (See [12] for possible solutions to this particular issue.) We consider the notion of Weak Second-Order Necessary Conditions on manifolds, which parallels the Euclidean case definition in [7,32].

Definition 3.3 (*Weak Second-Order Necessary Conditions (WSONC)*) Given an MCP as in (1), a feasible point $x^* \in \Omega$ satisfies WSONC if it satisfies KKT conditions with multipliers $\lambda^* \in \mathbb{R}_+^n$ and $\gamma^* \in \mathbb{R}^m$ such that¹

$$\langle v, \text{Hess}_x \mathcal{L}(x^*, \lambda^*)v \rangle \geq 0, \text{ for any } v \in \mathcal{C}^W(x^*),$$

where

$$v \in \mathcal{C}^W(x^*) \Leftrightarrow \begin{cases} v \in T_{x^*} \mathcal{M}, \\ \langle \text{grad } h_j(x^*), v \rangle = 0 \text{ for all } j \in \mathcal{E}, \text{ and} \\ \langle \text{grad } g_i(x^*), v \rangle = 0 \text{ for all } i \in \mathcal{A}(x^*) \cap \mathcal{I}. \end{cases}$$

Note that the cone \mathcal{C}^W in Definition 3.3 is smaller than the cone F (8) used to define SONC. Even in Euclidean space, most algorithms do not converge to feasible SONC points in general, so we would not expect more in the Riemannian case; see [7,30]. \mathcal{C}^W is called weak critical cone in [7, Def. 2.2].

Proposition 3.4 Consider an ECP (problem (1) without inequalities) with $|\mathcal{E}| < \dim(\mathcal{M})$. In Algorithm 1 with $\epsilon_{\min} = 0$, suppose that at each iteration k the sub-solver produces a point x_{k+1} satisfying

$$\| \text{grad}_x \mathcal{L}_{\rho_k}(x_{k+1}, \lambda^k, \gamma^k) \| < \epsilon_k \tag{13}$$

and

$$\text{Hess}_x \mathcal{L}_{\rho_k}(x_{k+1}, \lambda^k, \gamma^k) \succeq -\epsilon_k I \tag{14}$$

(meaning all eigenvalues of the Hessian are at or above $-\epsilon_k$). If the sequence generated by the algorithm has a limit point $\bar{x} \in \Omega$ where LICQ is satisfied, then \bar{x} satisfies the WSONC conditions of the original ECP.

Proof The proof is adapted from Section 3 in [7]: see Appendix B. □

4 Exact Penalty Method

Another standard approach to handle constraints is the so-called exact penalty method. As a replacement for the constraints, the method supplements the cost function with a

¹ Note that this condition involves \mathcal{L} as defined in Sect. 2.2, not \mathcal{L}_ρ .

weighted L_1 penalty for violating the constraints. This leads to a nonsmooth, unconstrained optimization problem. In the Riemannian case, to solve problem (1), this approach suggests solving the following program, where $\rho > 0$ is a penalty weight:

$$\min_{x \in \mathcal{M}} f(x) + \rho \left(\sum_{i \in \mathcal{I}} \max\{0, g_i(x)\} + \sum_{j \in \mathcal{E}} |h_j(x)| \right). \tag{15}$$

In the Euclidean case, it is known that only a finite penalty weight ρ is needed for exact satisfaction of constraints, hence the name [47, Ch. 15, 17]. We have the following analogous property in the Riemannian case.

Proposition 4.1 *If x^* is a local minimum for problem (1) and it satisfies SOSC with KKT multipliers λ^*, γ^* , then for any ρ such that $\rho > \max_{i \in \mathcal{I}, j \in \mathcal{E}} \{|\lambda_i^*|, |\gamma_j^*|\}$, x^* is also a local minimum for (15).*

Proof The proof resembles that of Theorem 6.9 in [50]: see Appendix C. □

Notice that the resulting penalized cost function is nonsmooth: such problems may be challenging in general. Fortunately, the cost function in (15) is a sum of maximum functions and absolute values functions: this special structure can be exploited algorithmically. We explore two general strategies: first, we explore smoothing techniques; then, we explore a particular nonsmooth Riemannian optimization algorithm.

4.1 Smoothing Technique

We discuss two smoothing methods for (15). A first approach is to note that the absolute value function can be written as a max function: $|x| = \max\{x, -x\}$. Thus, the nonsmooth part of the cost function in (15) is a linear combination of max functions between two terms. A popular smoothing for a two-term maximum is the log-sum-exp function [25]: $\max\{a, b\} \approx u \log(e^{a/u} + e^{b/u})$, with smoothing parameter $u > 0$. This yields the following smooth, unconstrained problem on a manifold, where ρ, u are fixed constants:

$$\begin{aligned} \min_{x \in \mathcal{M}} Q^{\text{lse}}(x, \rho, u) = & f(x) + \rho \sum_{i \in \mathcal{I}} u \log(1 + e^{g_i(x)/u}) \\ & + \rho \sum_{j \in \mathcal{E}} u \log(e^{h_j(x)/u} + e^{-h_j(x)/u}). \end{aligned} \tag{16}$$

Another common approach is to smooth the absolute value and the max in (15) separately, using respectively a pseudo-Huber loss [21] and a linear-quadratic loss [49]. Here, still with smoothing parameter $u > 0$, we use $|x| \approx \sqrt{x^2 + u^2}$ and $\max\{0, x\} \approx \mathcal{P}(x, u)$ where

$$\mathcal{P}(x, u) = \begin{cases} 0 & \text{if } x \leq 0 \\ \frac{x^2}{2u} & \text{if } 0 \leq x \leq u \\ x - \frac{u}{2} & \text{if } x \geq u. \end{cases}$$

Such approximation yields, for fixed constants ρ, u , the following problem:

$$\min_{x \in \mathcal{M}} Q^{\text{lqh}}(x, \rho, u) = f(x) + \rho \sum_{i \in \mathcal{I}} \mathcal{P}(g_i(x), u) + \rho \sum_{j \in \mathcal{E}} \sqrt{h_j(x)^2 + u^2}. \quad (17)$$

Note that Q^{lsc} and Q^{lqh} are both continuously differentiable and thus we can optimize them with smooth solvers on manifolds. In view of Proposition 4.1, in the absence of smoothing, there exists a threshold such that, if ρ is set above that threshold, then optima of the penalized problem coincide with optima of the target problem. However, we often do not know this threshold, and furthermore setting ρ at a high value with a poor initial iterate can lead to poor conditioning and slow convergence. A common practice is to set a relatively low initial ρ , optimize, and iteratively increase ρ and re-optimize with a warm start. This is formalized in Algorithm 2, which we here call Riemannian Exact Penalty Method via Smoothing (REPMS). In the algorithm and in the discussion below, update_ρ is a (fixed) Boolean flag indicating whether ρ is dynamically increased or not: this allows to discuss both versions of the algorithm.

Algorithm 2: Riemannian Exact Penalty Method via Smoothing (REPMS)

```

1 Require: Riemannian manifold  $\mathcal{M}$ , twice continuously differentiable functions  $f, \{g_i\}_{i \in \mathcal{I}}, \{h_j\}_{j \in \mathcal{E}}: \mathcal{M} \rightarrow \mathbb{R}$ .
2 Input: Starting point  $x_0 \in \mathcal{M}$ , accuracy tolerance  $\epsilon_{\min}$ , starting accuracy  $\epsilon_0$ , starting penalty coefficient  $\rho_0$ , starting approx-accuracy  $u_0$ , minimum approx-accuracy  $u_{\min}$ , constants  $\theta_\epsilon, \theta_u \in (0, 1), \theta_\rho > 1, \tau \geq 0, Q \in \{Q^{\text{lsc}}, Q^{\text{lqh}}\}$ , minimum step length  $d_{\min}$ ,  $\text{update}_\rho \in \{\text{True}, \text{False}\}$ .
3 for  $k = 0, 1, 2, \dots$  do
4   To obtain  $x_{k+1}$ , choose any subsolver to approximately solve
      
$$\min_{x \in \mathcal{M}} Q(x, \rho_k, u_k) \quad (18)$$

      with warm-start at  $x_k$  and stopping criterion
      
$$\|\text{grad } Q(x, \rho_k, u_k)\| \leq \epsilon_k.$$

5   if  $\text{dist}(x_k, x_{k+1}) < d_{\min}$  and  $\epsilon_k \leq \epsilon_{\min}$  and  $u_k \leq u_{\min}$  then
6     | Return  $x_{k+1}$ ;
7   end
8    $\epsilon_{k+1} = \max\{\epsilon_{\min}, \theta_\epsilon \epsilon_k\}; u_{k+1} = \max\{u_{\min}, \theta_u u_k\};$ 
9   if  $\text{update}_\rho = \text{True}$  and  $(k = 0 \text{ or } \max_{j \in \mathcal{E}, i \in \mathcal{I}} \{|h_j(x_{k+1})|, g_i(x_{k+1})\} \geq \tau)$  then
10    |  $\rho_{k+1} = \theta_\rho \rho_k$ 
11  else
12    |  $\rho_{k+1} = \rho_k$ ;
13  end
14 end

```

The stopping criterion parameters ($d_{\min}, u_{\min}, \epsilon_{\min}$) are lower-bounds introduced in the algorithm for practical purposes. The algorithm terminates when the step size is too small while the approximation accuracy is high and the tolerance for the subsolver

is low. On the other hand, if u_k is too small, numerical difficulties may arise in these two approximation functions. Numerical concerns aside, in theory (exact arithmetic), if we set these parameters to 0, then the following convergence result holds, similar to its Euclidean counterpart of quadratic penalty method [47, §17.1].

Proposition 4.2 *In Algorithm 2, suppose we set $d_{\min} = u_{\min} = \epsilon_{\min} = 0$ and $\text{update}_\rho = \text{False}$. If the sequence $\{x_k\}$ produced by the subsolver admits a feasible limit point \bar{x} where LICQ conditions hold, then \bar{x} satisfies KKT conditions for (1).*

Proof See Appendix D. □

The conditions of this proposition are only likely to hold if ρ_0 (the initial penalty) is sufficiently large, as described in Proposition 4.1. Instead of trying to set ρ_0 above the unknown threshold, we may increase ρ in every iteration. When $\text{update}_\rho = \text{True}$, the update of ρ is a heuristic featured in [49] in the Euclidean case: it gives a conditioned increase on the penalty parameter. Intuitively, when the obtained point is far from feasible, the penalty coefficient is likely too small so that the penalty parameter is increased.

4.2 A Subgradient Method for Sums of Maximum Functions

Instead of smoothing, one may attempt to optimize the nonsmooth objective function directly. In this pursuit, the subgradient descent algorithm on Riemannian manifolds, and its variants, have received much attention in recent years; see [1,31,33]. These algorithms work for general locally Lipschitz objective functions on manifolds, and would work here as well. However, notice that (15) takes the exploitable form of minimizing a sum of maximum functions:

$$\min_{x \in \mathcal{M}} f(x) + \sum_{i \in \mathcal{I}} \max\{0, \rho g_i(x)\} + \sum_{i \in \mathcal{E}} \max\{\rho h_i(x), -\rho h_i(x)\}. \tag{19}$$

We hence propose a robust version of subgradient descent for sums of maximum functions, which may be of interest in its own right. We refer to it as Riemannian Exact Penalty Method via Subgradient Descent (REPMSD). However, due to the lengthy specification of this method and its poor performance in numerical experiments, we will only give an overview here and report its performance in Sect. 5.

In the Euclidean case, subgradient descent with line search performs two steps—finding a good descent direction and performing line search to get the next point. To find a descent direction, one often wants access to the subdifferential at the current point, because it contains a descent direction. This generalizes to the Riemannian case. We refer readers to [26,31,33] for the definition of *generalized subdifferential*, which we denote as $\partial f(x)$ for $f: \mathcal{M} \rightarrow \mathbb{R}$. Note that obtaining full information of the subdifferential is generally difficult, and thus many algorithms *sample* the gradients around the current point to approximate it. However, for a sum of maximum functions, the subdifferential is directly accessible. This class of problems, which includes (19), takes the form:

$$\min_{x \in \mathcal{M}} f(x) = \sum_{i=1}^m \max\{f_{i,j}(x) | j = 1, \dots, n_i\}, \tag{20}$$

where each function $f_{i,j} : \mathcal{M} \rightarrow \mathbb{R}$ is twice continuously differentiable. Notice that

$$f(x) = \sum_{i=1}^m \max\{f_{i,j}(x) | j = 1, \dots, n_i\} = \max_{1 \leq j_i \leq n_i \text{ for } i=1 \dots m} \sum_{i=1}^m f_{i,j_i}(x). \tag{21}$$

For a given i in $1, \dots, m$ and a given $x \in \mathcal{M}$, let \mathcal{I}_i index the set of functions $f_{i,j}$ which attain the maximum for that i :

$$\mathcal{I}_i = \{1 \leq j \leq n_i \mid f_{i,j}(x) = \max\{f_{i,j_i}(x) | j_i = 1 \dots n_i\}\}.$$

It is clear that for any choice of j_i 's (one for each i) such that $\sum_i f_{i,j_i}(x) = f(x)$, we have $j_i \in \mathcal{I}_i$ for all i . Thus, from the regularity of the maximum function as defined in [26] and Proposition 2.3.12 in that same reference, exploiting the fact that pullbacks $f \circ \text{Exp}_x$ are defined on the tangent space $T_x \mathcal{M}$ which is a linear subspace, it can be shown using standard definitions of subdifferentials that

$$\partial(f \circ \text{Exp}_x)(0_x) = \text{Conv} \left\{ \sum_{i=1}^m \text{Grad}(f_{i,j_i} \circ \text{Exp}_x)(0_x) \mid \text{all possible choices of } j_i \in \mathcal{I}_i \right\},$$

where we use the notation Grad to stress that this is a Euclidean gradient of the pullback on the tangent space, and the operator Conv returns the convex hull of given vectors. To be explicit, for each i , pick some j_i in \mathcal{I}_i : this produces one vector by summation over i as indicated; the convex hull is taken over all possible choices of the j_i 's. Then, with Proposition 2.5 of [34], analogously to (3), we see that the generalized subdifferential of f at x is given by

$$\partial f(x) = \text{Conv} \left\{ \sum_{i=1}^m \text{grad } f_{i,j_i}(x) \mid \text{all possible choices of } j_i \in \mathcal{I}_i \right\}. \tag{22}$$

This gives an explicit formula for the subdifferential of f on \mathcal{M} . Whilst the traditional approach is often to get a descent direction directly from this subdifferential or to sample around the point, numerically, it makes sense to relax the condition and to replace \mathcal{I}_i with an approximate subdifferential based on the (larger) set

$$\mathcal{I}_{i,\epsilon} = \{1 \leq j \leq n_i \mid f_{i,j}(x) \geq \max\{f_{i,j}(x) | j = 1, \dots, n_i\} - \epsilon\},$$

where $\epsilon > 0$ is a tolerance. Then, one looks for a descent direction in the subdifferential (22) extended by substituting $\mathcal{I}_{i,\epsilon}$ for \mathcal{I}_i . For the latter, the classical approach with $\epsilon = 0$ is to look for a tangent vector of minimum norm in the convex hull: this involves solving a convex quadratic program. We here compute a minimum norm vector in the extended convex hull. For the rest of the algorithm, we perform a classical Wolfe line

search and use a limited-memory version of Riemannian BFGS (LRBFGS) updates based on the now well-known observation that BFGS works surprisingly well with nonsmoothness on Euclidean spaces [44] as well as on manifolds [33]—unfortunately we could not run full Riemannian BFGS as this is expensive on high dimensional manifolds. For details on Wolfe line search and BFGS, see Chapters 3, 8 and 9 of [47].

In closing, we note that there exist other methods for nonsmooth optimization, some of which apply on manifolds as well. In particular, we mention proximal point algorithms [9,48], Douglas–Rachford-type methods [11] and iteratively reweighted least squares [24]. We did not experiment with these methods here.

5 Numerical Experiments and Discussion

5.1 Problems and Data

We describe three applications which can be modeled within the framework of (1). For each, we describe how we construct ‘baskets’ of problem instances. These instances will be used to assess the performance of various algorithms. Code and datasets to reproduce the experiments are freely available.²

5.1.1 Minimum Balanced Cut for Graph Bisection

The minimum balanced cut problem is a graph problem: given an undirected graph, one seeks to partition the vertices into two clusters (i.e., bisecting the graph) such that the number of edges between clusters is as small as possible, while simultaneously ensuring that the two clusters have about the same size. A spectral method called ratio cut tackles this problem via optimization where the number of crossing edges and the imbalance are penalized in the objective function. However, in [41], the author notes that for ‘power law graphs’—graphs whose degrees follow a power law distribution—this approach fails to enforce balance. Hence, the author proposes a method with stronger constraints to enforce balance: we sketch it below.

Let $L \in \mathbb{R}^{n \times n}$ denote the discrete Laplacian matrix of the given graph ($L = D - A$, where D is the diagonal degree matrix and A is the adjacency matrix). Furthermore, let $x \in \mathbb{R}^n$ be an indicator vector with entries in $\{-1, 1\}$ indicating which cluster each vertex belongs to. One can check that $\frac{1}{4}x^T Lx$ evaluates to the number of edges crossing the two clusters. Let e be the vector of 1’s with length n . To enforce balance, $x^T e = 0$ is imposed, i.e., the number of vertices in the two clusters is the same (this only applies to graphs with an even number of vertices). However, directly solving this integer programming problem is difficult. The authors thus transform this discrete optimization problem into a continuous one. As a first step, the constraints $x_i = \pm 1$ for all i are replaced by $\text{diag}(xx^T) = e$, and the constraint $x^T e = 0$ is replaced by $(x^T e)^2 = e^T (xx^T) e = 0$. Then, $x \in \mathbb{R}^n$ is relaxed to a matrix $X \in \mathbb{R}^{n \times k}$ for some chosen k in $1 \dots n$. Carrying over the constraints gives $\text{diag}(XX^T) = e$, so that each row of X is a unit vector, and $e^T (XX^T) e = 0$ is imposed. The problem then becomes:

² <https://github.com/losangle/Optimization-on-manifolds-with-extra-constraints>.

$$\begin{aligned}
 \min_{X \in \mathbb{R}^{n \times k}} \quad & -\frac{1}{4} \text{tr}(LXX^T) \\
 \text{subject to} \quad & \text{diag}(XX^T) = e, \quad e^T XX^T e = 0,
 \end{aligned} \tag{23}$$

For $k = 1$, the problem is equivalent to the original problem. For $k = n$, the problem is equivalent to solving a (convex) semidefinite program (SDP) by considering XX^T as a positive semidefinite matrix (X can be recovered from Cholesky factorization). One may then use a standard clustering algorithm (for example k -means) on the rows of X to obtain two clusters. By choosing $k > 1$, the problem becomes continuous (as opposed to discrete), which much extends the class of algorithms that we may apply to solve this problem. Furthermore, since for $k = n$ the problem is essentially convex (being equivalent to an SDP), we heuristically expect that for $k > 1$ the problem may be easier to solve than with $k = 1$.

To keep the dimensionality of the optimization problem low, it is of interest to consider $k > 1$ smaller than n , akin to a Burer–Monteiro approach [19]. We notice that $\text{diag}(XX^T) = e$ defines an oblique manifold (a product of n unit spheres in \mathbb{R}^k). Thus, we can view (23) as a problem on the oblique manifold with a single constraint:

$$\begin{aligned}
 \min_{X \in \text{Oblique}(n,k)} \quad & -\frac{1}{4} \text{tr}(X^T LX) \\
 \text{subject to} \quad & e^T XX^T e = 0.
 \end{aligned} \tag{24}$$

This fits our problem class (1) with $\mathcal{M} = \text{Oblique}(n, k)$ as an embedded Riemannian submanifold of $\mathbb{R}^{n \times k}$ and a single quadratic equality constraint.

Input. We generate power law graphs with the Barabási–Albert algorithm [4]. It starts off with a seed of a small connected graph, iteratively adds a vertex to the graph, and attaches l edges to the existing vertices. For each edge, it is attached to a vertex with probability proportional to the current degree of the vertex. We generate the seed with Erdős–Rényi random graphs $G(n_{\text{seed}}, p)$, that is: $G(n_{\text{seed}}, p)$ is a random graph with n_{seed} nodes such that every two nodes are connected with probability p independently of other edges. For each dimension $n \in \{50, 200, 500, 1000, 2000, 5000\}$, we collect a basket of problems by selecting a density $m \in \{0.005, 0.01, 0.02, 0.04, 0.08\}$, such that $l = mn$. For each set of parameters, we repeat the experiment four times with new graphs and new starting points. The seed is generated by $G(2\lceil mn \rceil, 0.5)$.

5.1.2 Non-Negative PCA

We follow Montanari and Richard [46] for this presentation of non-negative PCA. Let $v_0 \in \mathbb{R}^d$ be the spiked signal with $\|v_0\| = 1$. In the so-called spiked model, we are given the matrix $Z = \sqrt{SNR} v_0 v_0^T + N$, where N is a random symmetric noise matrix and SNR is the signal to noise ratio. For N , its off-diagonal entries follow a Gaussian distribution $\mathcal{N}(0, \frac{1}{n})$ and its diagonal entries follow a Gaussian distribution $\mathcal{N}(0, \frac{2}{n})$, i.i.d. up to symmetry. In classical PCA, we hope to recover the signal v_0 from finding the eigenvector that corresponds to the largest eigenvalue. That is, to solve the following problem:

$$\begin{aligned} \min_{v \in \mathbb{R}^d} \quad & -v^T Z v \\ \text{subject to} \quad & \|v\| = 1. \end{aligned} \tag{25}$$

However, it is well known that in the high dimensional regime, $n = O(d)$, this approach breaks down—the principal eigenvector of Z is asymptotically less indicative of v_0 ; see [37].

One way of dealing with this problem is by introducing structures such as sparsity on the solution space. In [46], the authors impose nonnegativity of entries as prior knowledge, and propose to solve PCA restricted to the positive orthant:

$$\begin{aligned} \min_{v \in \mathbb{R}^d} \quad & -v^T Z v \\ \text{subject to} \quad & \|v\| = 1, v \geq 0. \end{aligned} \tag{26}$$

Since $\|v\| = 1$ defines the unit sphere \mathbf{S}^{d-1} in \mathbb{R}^d , one can write the above problem as

$$\begin{aligned} \min_{v \in \mathbf{S}^{d-1}} \quad & -v^T Z v \\ \text{subject to} \quad & v \geq 0. \end{aligned} \tag{27}$$

This falls within the framework of (1), with $\mathcal{M} = \mathbf{S}^{d-1}$ as an embedded Riemannian submanifold of \mathbb{R}^d and d inequality constraints.

Input. Similar to [39], we synthetically generate data following the symmetric spiked model as described above. We consider dimensions $d = \{10, 50, 200, 500, 1000, 2000\}$, and let $n = d$. For each dimension, we consider the basket of problems parametrized by $SNR \in \{0.05, 0.1, 0.25, 0.5, 1.0, 2.0\}$ which controls the noise level and $\delta \in \{0.1, 0.3, 0.7, 0.9\}$ which controls sparsity. Furthermore, we let the support $S \subseteq \{1, \dots, n\}$ of the true principal direction v_0 be uniformly random, with cardinality $|S| = \lfloor \delta d \rfloor$, and

$$v_{0,i} = \begin{cases} \frac{1}{\sqrt{|S|}} & \text{if } i \in S, \\ 0 & \text{otherwise.} \end{cases}$$

Note that S controls how sparse v_0 is. For each set of parameters, the experiment was repeated four times with different random noise N . As there are 6 choices for d , 4 for δ , 6 for SNR , this gives 144 problems in each basket.

5.1.3 k -Means via Low-Rank SDP

k -means is a traditional clustering algorithm in machine learning. Given a set of data points $\{x_i\}_{i=1}^n, x_i \in \mathbb{R}^l$ for some l , and a desired number of clusters k , this algorithm aims to partition the data into k subsets $\{S_j\}_{j=1}^k$ such that the following is minimized:

Table 1 Details of the datasets used in k -means

Dataset name	Number of data	features	clusters
Iris	150	4	3
ecoli	336	5	8
pima	768	8	2
sonar	208	60	2
vehicle	846	18	4
wine	178	13	3
cloud	2048	10	4

$$\mathcal{D}(\{S_j\}_{j=1}^k) = \sum_{j=1}^k \sum_{x_i \in S_j} \|x_i - m_j\|^2,$$

where $m_j = \frac{1}{|S_j|} \sum_{x_i \in S_j} x_i$ is the center of mass of the points selected by S_j . Following [23], an equivalent formulation is as follows (I is the identity matrix):

$$\begin{aligned} \min_{Y \in \mathbb{R}^{n \times k}} & \quad -\text{tr}(Y^T D Y) \\ \text{subject to} & \quad Y^T Y = I, \quad Y \geq 0, \quad Y Y^T e = e, \end{aligned} \tag{28}$$

where $D_{ij} = \|x_i - x_j\|^2$ is the squared Euclidean distance matrix. The constraint $Y^T Y = I$ indicates Y has orthonormal columns. The set of such matrices is called the Stiefel manifold. Thus, we can rewrite the problem as:

$$\begin{aligned} \min_{Y \in \text{Stiefel}(n,k)} & \quad -\text{tr}(Y^T D Y) \\ \text{subject to} & \quad Y \geq 0, \quad Y Y^T e = e. \end{aligned} \tag{29}$$

This falls within the framework of (1) with \mathcal{M} the Stiefel manifold (as an embedded Riemannian submanifold of $\mathbb{R}^{n \times k}$), n equality constraints and nk inequality constraints.

Input. We take data sets from the UCI Machine Learning repository [45]. For each dataset, we clean the data by removing categorical data and normalizing each feature. This gives $X = [x_1 \dots x_N]$, and we get D via $D_{ij} = \|x_i - x_j\|^2$. The specifications of the cleaned data are described in Table 1.

5.2 Methodology

We compare the methods discussed above against each other. In addition, as a control to test the hypothesis that exploiting the geometry of the manifold constraint is beneficial, we also compare against Matlab’s built-in constrained optimization solver `fmincon`, which treats the manifold as supplementary equality constraints. We choose `fmincon` because it is a general purpose solver which combines various algorithms to enhance

performance and it has been refined over years of developments, and thus acts as a good benchmark. To summarize, the methods are:

- RALM: Riemannian Augmented Lagrangian Method, Sect. 3;
- REPMS(Q^{qh}): exact penalty method with smoothing (linear-quadratic and pseudo-Huber), Sect. 4.1;
- REPMS(Q^{lsc}): exact penalty method with smoothing (log-sum-exp), Sect. 4.1;
- REPMSD: exact penalty method via nonsmooth optimization, Sect. 4.2;
- Matlab's `fmincon`: does not exploit manifold structure.

We supply `fmincon` with the gradients of both the objective function and the constraints. We use the default settings: minimum step size is 10^{-10} and relative constraint violation tolerance is 10^{-6} .³ Furthermore, we disable the stopping criterion based on a maximum number of iterations or queries in order to allow `fmincon` to converge to good solutions. For all of our methods, we define $d_{\min} = 10^{-10}$, $\epsilon_0 = 10^{-3}$, $\epsilon_{\min} = 10^{-6}$, $\theta_{\epsilon} = (\epsilon_{\min}/\epsilon_0)^{\frac{1}{30}}$, $\rho_0 = 1$, $\theta_{\rho} = 3.3$. Specifically, for RALM, $\theta_{\sigma} = 0.8$; for exact penalty methods, $\tau = 10^{-6}$, $u_0 = 10^{-1}$, $u_{\min} = 10^{-6}$, $\theta_u = (u_{\min}/u_0)^{\frac{1}{30}}$. As subsolver for RALM and the smoothing methods REPMS, we use LRBFGS: a Riemannian, limited-memory BFGS [36] as implemented in Manopt [18]. For REPMSD, we use a minimum-norm tangent vector in the extended subgradient, then we use a type of LRBFGS inverse Hessian approximation for Hessian updates to choose the update direction [33]. For these limited-memory subsolvers, we let the memory be 30, the maximum number of iterations be 200, and the minimum step size be 10^{-10} . For each experiment, all solvers have the same starting point randomly chosen on the manifold. An experiment is also terminated if it runs over one hour. We register the last produced iterate. Maximum constraint violation (Maxvio), cost function value and computation time are recorded for each solver, where Maxvio is defined as:

$$\text{Maxvio} = \max (\{|h_j(x)| \mid j \in \mathcal{E}\} \cup \{|g_i(x)| \mid i \in \mathcal{I}\} \cup \{0\}).$$

For minimum balanced cut and non-negative PCA, we present performance profiles to compare computation time across problem instances for the various solvers. Performance profiles were popularized in optimization by Dolan and Moré [28]. Quoting from that reference, performance profiles show “(cumulative) distribution functions for a performance metric as a tool for benchmarking and comparing optimization software.” In particular, we show “the ratio of the computation time of each solver versus the best time of all of the solvers as the performance metric.” The profiles of REPMSD are only shown in lower dimensions to reduce computation time. See Fig. 1 for further details.

We consider that a point is feasible if Maxvio is smaller than 5×10^{-4} . By denoting f_{\min} as the minimum cost among all feasible solutions returned by solvers, we consider a solution to be best-in-class if it is feasible *and* the cost f satisfies $|f/f_{\min} - 1| < 2\%$.

³ When the step size is of order 10^{-10} , we believe that the current point is close to convergence. We also conducted experiments with minimum step size 10^{-7} for minimum balanced cut and non-negative PCA, and the performance profiles are visually similar to those displayed here.

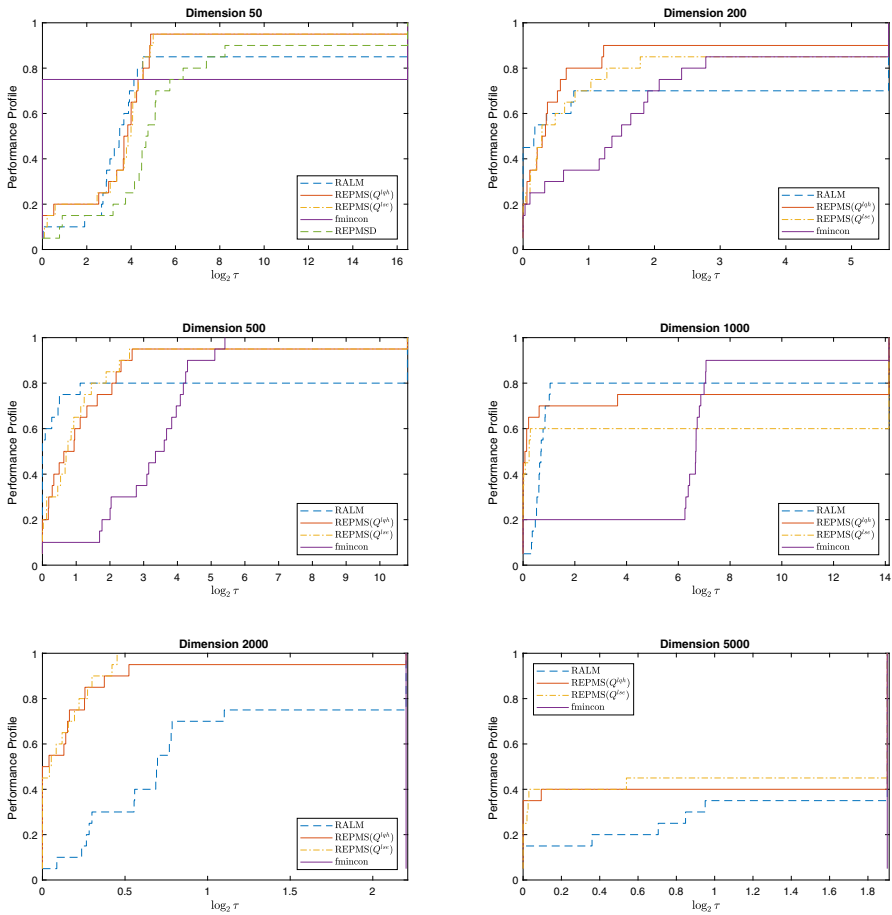


Fig. 1 Performance profiles of computation time for different dimensions of the minimum balanced cut problem (with maximum constraint violation and cost accounted for—see Sect. 5.2). For each dimension, all solvers are run on a basket of problems. For each problem that could be solved, one solver obtained the fastest computation time. The curve for a solver passes through point $(\log_2 \tau, q)$ if it solved a fraction q of the problems within a factor τ of the fastest solver (which may change for each problem). Thus, upper-left is best. In particular, for $\tau = 1$ ($\log_2 \tau = 0$), the curve of solver A passes at level q if solver A was the fastest on a fraction q of the problems. These may not sum to 100% since some problems were not solved by any solver (and there could be ties, though that is unlikely)

In view of the objective functions of these two problems, a relative error tolerance is reasonable. We consider that a solver solves the problem if the returned point is best-in-class.

For k -means, we give a table of experimental results for Maxvio, cost and time across all datasets.

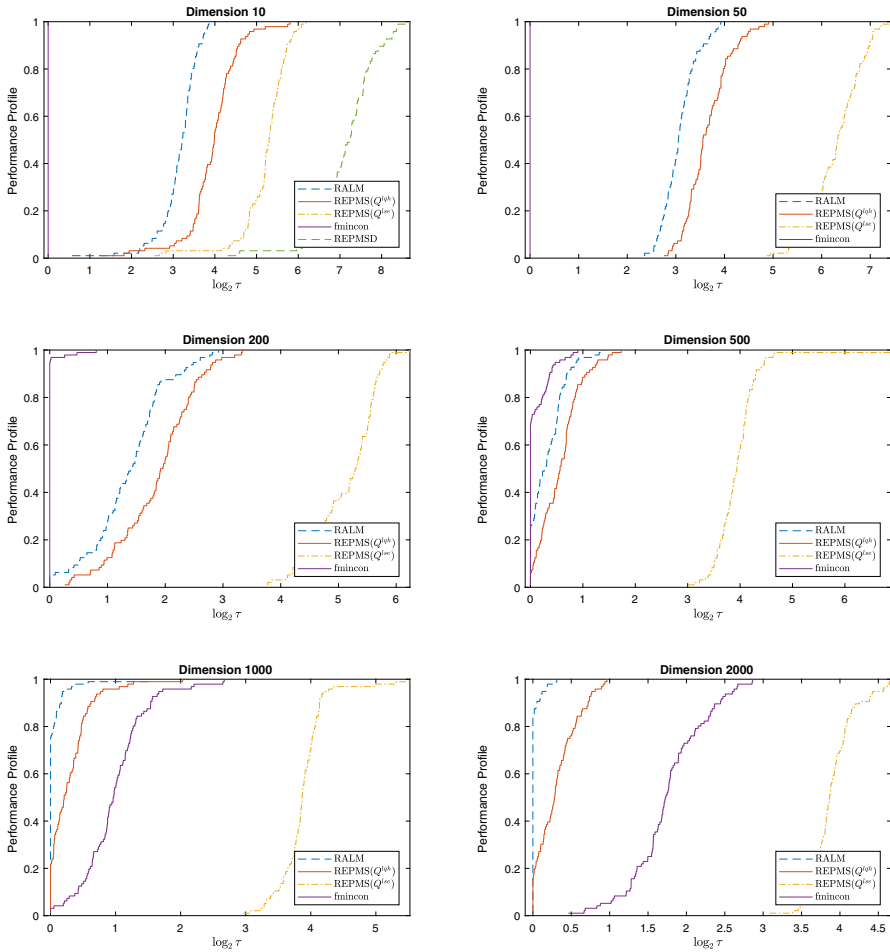


Fig. 2 Performance profile of computation time for different dimensions of non-negative PCA (with maximum constraint violation and cost accounted for—see Sect. 5.2). See caption of Fig. 1 for details on performance profiles

5.3 Results and Analysis

5.3.1 Minimum Balanced Cut for Graph Bisection

We display in Fig. 1 the performance profiles for various dimensions, corresponding to the number of vertices in the graph. We note that apart from REPMISD, all methods perform well in the low dimensional case. In the high dimensional case, fmincon fails: in all test cases it converges to points far from feasible despite having no iteration limit; exact penalty methods and RALM are still able to give satisfactory results in some cases.

Table 2 Maximum constraint violation, cost function value, and computation time results of different algorithms for k -means. For dataset ‘cloud’ (which does not come with a specified number of clusters), we set the number of clusters k to 4

		RALM	REPMS(Q^{lqh})	REPMS(Q^{lse})	fmincon	REPMSD
Iris	Maxvio	1.92×10^{-5}	6.49×10^{-6}	4.26×10^{-2}	4.44×10^{-5}	2.13×10^{-7}
	Cost	3.31×10^{-1}	4.43×10^{-2}	$8.20 \times 10^{+0}$	$1.14 \times 10^{+1}$	2.80×10^{-1}
	Time (s)	$4.72 \times 10^{+2}$	$4.89 \times 10^{+2}$	$1.70 \times 10^{+2}$	$3.13 \times 10^{+2}$	$3.86 \times 10^{+3}$
Wine	Maxvio	7.25×10^{-5}	6.62×10^{-4}	4.81×10^{-2}	1.75×10^{-5}	–
	Cost	$1.89 \times 10^{+0}$	1.68×10^{-1}	$8.43 \times 10^{+0}$	$1.19 \times 10^{+1}$	–
	Time (s)	$3.45 \times 10^{+2}$	$2.87 \times 10^{+2}$	$1.85 \times 10^{+2}$	$1.59 \times 10^{+2}$	–
Sonar	Maxvio	3.02×10^{-5}	1.42×10^{-5}	5.48×10^{-2}	6.65×10^{-7}	–
	Cost	8.99×10^{-1}	$9.50 \times 10^{+0}$	$1.50 \times 10^{+1}$	$1.80 \times 10^{+1}$	–
	Time (s)	$5.28 \times 10^{+2}$	$3.48 \times 10^{+2}$	$2.55 \times 10^{+2}$	$6.70 \times 10^{+2}$	–
Ecoli	Maxvio	8.33×10^{-4}	2.65×10^{-3}	6.73×10^{-2}	9.67×10^{-1}	–
	Cost	$4.08 \times 10^{+1}$	1.24×10^{-1}	$1.87 \times 10^{+1}$	$6.45 \times 10^{+0}$	–
	Time (s)	$1.75 \times 10^{+3}$	$1.44 \times 10^{+3}$	$1.01 \times 10^{+3}$	$3.62 \times 10^{+3}$	–
Pima	Maxvio	2.10×10^{-10}	1.82×10^{-3}	9.10×10^{-2}	2.34×10^{-5}	–
	Cost	2.60×10^{-2}	9.48×10^{-3}	$1.56 \times 10^{+0}$	$2.13 \times 10^{+1}$	–
	Time (s)	$1.37 \times 10^{+3}$	$1.25 \times 10^{+3}$	$5.79 \times 10^{+2}$	$3.62 \times 10^{+3}$	–
Vehicle	Maxvio	8.87×10^{-3}	4.35×10^{-4}	9.17×10^{-2}	$2.71 \times 10^{+2}$	–
	Cost	$3.18 \times 10^{+0}$	3.77×10^{-2}	$2.23 \times 10^{+1}$	$6.71 \times 10^{+2}$	–
	Time (s)	$2.88 \times 10^{+3}$	$2.70 \times 10^{+3}$	$2.61 \times 10^{+3}$	$3.63 \times 10^{+3}$	–
Cloud	Maxvio	1.81×10^{-2}	1.83×10^{-3}	6.79×10^{-1}	$3.70 \times 10^{+3}$	–
	Cost	1.36×10^{-1}	5.49×10^{-4}	$1.97 \times 10^{+0}$	$2.20 \times 10^{+3}$	–
	Time (s)	$4.08 \times 10^{+3}$	$3.65 \times 10^{+3}$	$3.78 \times 10^{+3}$	$3.84 \times 10^{+3}$	–

For all other datasets, we used the number of clusters recommended by the dataset documentation

5.3.2 Non-negative PCA

Table 2 displays the performance profiles for non-negative PCA for various dimensions, corresponding to the length of the sought vector. RALM and REPMS(Q^{lqh}) outperform fmincon as dimension increases. REPMS(Q^{lse}) is rather slow. Notice that in this problem there are lots of inequality constraints involved. For REPMS(Q^{lqh}), when we compute the gradient of the objective function, gradients of constraints that are not violated at the current point are not computed— they are 0. However, they are computed in REPMS(Q^{lse}), which slows down the algorithm. This observation suggests that, for problems with a large number of inequality constraints, REPMS(Q^{lse}) may not be the best choice.

5.3.3 k -means

Table 2 gives the results for k -means clustering. As the dimension gets larger, fmincon performs worse than RALM and REPMS in both Maxvio and Cost. Among

all these methods, REPMS(Q^{lqh}) and RALM are the better methods since they have relatively shorter solving time and lower Maxvio. In addition, REPMS(Q^{lqh}) achieves the lowest cost for high dimensions.

6 Conclusion

In this paper, augmented Lagrangian methods and exact penalty methods are extended to Riemannian manifolds, along with some essential convergence results. These are compared on three applications against a baseline, which is to treat the manifold as a set of general constraints. The three applications are minimum balanced cut, non-negative PCA and k -means. The conclusions from the numerical experiments are that, in high dimension, it seems to be beneficial to exploit the manifold structure; as to which Riemannian algorithm is better, the results vary depending on the application.

We consider this work to be a first systematic step towards understanding the empirical behavior of straightforward extensions of various techniques of constrained optimization from the Euclidean to the Riemannian case. One direct advantage of exploiting the Riemannian structure is that this part of the constraints is satisfied up to numerical accuracy at each iteration. While our numerical experiments indicate that some of these methods perform satisfactorily as compared to a classical algorithm which does not exploit Riemannian structure (especially in high dimension), the gains are moderate. For future research, it is of interest to pursue refined versions of some of the algorithms we studied here, possibly inspired by existing refinements in the Euclidean case but also by a more in depth study of the geometry of the problem class. Furthermore, it is interesting to pursue the study of the convergence properties of these algorithms—and the effects of the underlying Riemannian geometry—beyond the essential results covered here, in particular because the Riemannian approach extends to general abstract manifolds, including some which may not be efficiently embedded in Euclidean space through practical equality constraints.

Acknowledgements We thank an anonymous reviewer for detailed and helpful comments on the first version of this paper. NB is partially supported by NSF grant DMS-1719558.

A Proof of Proposition 3.2

We first introduce two supporting lemmas. The first lemma is a well-known fact for which we provide a proof for completeness.⁴

Lemma A.1 *Let p be a point on a Riemannian manifold \mathcal{M} , and let v be a tangent vector at p . Let \mathcal{U} be a normal neighborhood of p , that is, the exponential map maps a neighbourhood of the origin of $T_p\mathcal{M}$ diffeomorphically to \mathcal{U} . Define the following vector field on \mathcal{U} :*

$$\forall q \in \mathcal{U}, \quad V(q) = \mathcal{P}_{p \rightarrow q} v,$$

⁴ The proof follows an argument laid out by John M. Lee: <https://math.stackexchange.com/questions/2307289/parallel-transport-along-radial-geodesics-yields-a-smooth-vector-field>.

where parallel transport is done along the (unique) minimizing geodesic from p to q . Then, V is a smooth vector field on \mathcal{U} .

Proof Parallel transport from p is along geodesics passing through p . To facilitate their study, set up normal coordinates $\phi: U \subset \mathbb{R}^d \rightarrow \mathcal{U}$ around p (in particular, $\phi(0) = p$), where d is the dimension of the manifold. For a point $\phi(x_1, \dots, x_d)$, by definition of normal coordinates, the radial geodesic from p is $c(t) = \phi(tx_1, \dots, tx_d)$. Our vector field of interest is defined by $V(p) = v$ and the fact that it is parallel along every radial geodesic c as described.

For a choice of point $\phi(x)$ and corresponding geodesic, let

$$V(c(t)) = \sum_{k=1}^d v_k(t) \partial_k(c(t))$$

for some coordinate functions v_1, \dots, v_d , where ∂_k is the k th coordinate vector field. These coordinate functions satisfy the following ordinary differential equations (ODE) [22, Prop. 2.6, eq. (2)]:

$$0 = \frac{dv_k(t)}{dt} + \sum_{i,j} \Gamma_{ij}^k(tx_1, \dots, tx_d) v_j(t) x_i, \quad k = 1, \dots, d,$$

where Γ denotes Christoffel symbols. Expand $V(p)$ into the coordinate vector fields: $v = \sum_{k=1}^d w_k \partial_k(p)$. Then, the initial conditions are $v_k(0) = w_k$ for each k . Because these ODEs are smooth, solutions $v_k(t; w)$ exist, and they are smooth in both t and the initial conditions w [42, Thm. D.6]. But this is not enough for our purpose.

Crucially, we wish to show smoothness also in the choice of $x \in U$. To this end, following a classical trick, we extend the set of equations to let x be part of the variables, as follows:

$$\begin{cases} 0 = \frac{dv_k(t)}{dt} + \sum_{i,j} \Gamma_{ij}^k(tu_1(t), \dots, tu_d(t)) v_j(t) u_i(t), & k = 1, \dots, d, \\ 0 = \frac{du_k(t)}{dt}, & k = 1, \dots, d. \end{cases}$$

The extended initial conditions are:

$$v_k(0) = w_k, \quad u_k(0) = x_k, \quad k = 1, \dots, d.$$

Clearly, the functions $u_k(t)$ are constant: $u_k(t) = x_k$. These ODEs are still smooth, hence solutions $v_k(t; w, x)$ still exist and are identical to those of the previous set of ODEs, except we now see they are also smooth in the choice of x . Specifically, for every $x \in U$,

$$V(\phi(x)) = V(c(1)) = \sum_{k=1}^d v_k(1; w, x) \partial_k(\phi(x)),$$

and each $v_k(1; w, x)$ depends smoothly on x . Hence, V is smooth on $\mathcal{U} = \phi(U)$. □

Lemma A.2 Given a Riemannian manifold \mathcal{M} , a function $f : \mathcal{M} \rightarrow \mathbb{R}$ (continuously differentiable), and a point $p \in \mathcal{M}$, if p_0, p_1, p_2, \dots is a sequence of points in a normal neighborhood of p and convergent to p , then the following holds:

$$\lim_{k \rightarrow \infty} \|\mathcal{P}_{p_k \rightarrow p} \text{grad } f(p_k) - \text{grad } f(p)\|_p = 0,$$

where $\mathcal{P}_{p_k \rightarrow p}$ is the parallel transport from $T_{p_k}\mathcal{M}$ to $T_p\mathcal{M}$ along the minimizing geodesic.

Proof of Lemma A.2 As parallel transport is an isometry, it is equivalent to show

$$\lim_{k \rightarrow \infty} \|\text{grad } f(p_k) - \mathcal{P}_{p \rightarrow p_k} \text{grad } f(p)\|_{p_k} = 0. \tag{30}$$

Under our assumptions, $\text{grad } f$ is a continuous vector field. Furthermore, by Lemma A.1, in a normal neighborhood of p , the vector field $V(y) = \mathcal{P}_{p \rightarrow y} \text{grad } f(p)$ is a continuous vector field as well. Hence, $\text{grad } f - V$ is a continuous vector field around p ; since $\text{grad } f(p) - V(p) = 0$, the result is proved: $\lim_{k \rightarrow \infty} \text{grad } f(p_k) - V(p_k) = \text{grad } f(p) - V(p) = 0$. \square

Proof of Proposition 3.2 Restrict to a convergent subsequence if needed, so that $\lim_{k \rightarrow \infty} x_k = \bar{x}$. Further exclude a (finite) number of x_k 's so that all the remaining points are in a neighborhood of \bar{x} where the exponential map is a diffeomorphism. In this proof, let \mathcal{A} denote $\mathcal{A}(\bar{x})$ for ease of notation: this is the set of active constraints at the limit point. Then, there exist constants c, k_1 such that $g_i(x_k) < c < 0$ for all $k > k_1$ with $i \in \mathcal{I} \setminus \mathcal{A}$.

When $\{\rho_k\}$ is unbounded, since multipliers are bounded, there exists $k_2 > k_1$ such that $\lambda_i^k + \rho_k g_i(x_{k+1}) < 0$ for all $k \geq k_2, i \in \mathcal{I} \setminus \mathcal{A}$. Thus, by definition, $\lambda_i^{k+1} = 0$ for all $k \geq k_2, i \in \mathcal{I} \setminus \mathcal{A}$.

When instead $\{\rho_k\}$ is bounded, $\lim_{k \rightarrow \infty} |\sigma_i^k| = 0$. Thus for $i \in \mathcal{I} \setminus \mathcal{A}$, in view of $g_i(x_k) < c < 0$ for all $k > k_1$, we have $\lim_{k \rightarrow \infty} \frac{-\lambda_i^k}{\rho_k} = 0$. Then, for large enough k , $\lambda_i^k + \rho_k g_i(x_{k+1}) < 0$ and thus there exists $k_2 > k_1$ such that $\lambda_i^k = 0$ for all $k \geq k_2$. So in either case, we can find such k_2 .

As LICQ is satisfied at \bar{x} , by continuity of the gradients of $\{g_i\}$ and $\{h_j\}$, the tangent vectors $\{\text{grad } h_j(x_k)\}_{j \in \mathcal{E}} \cup \{\text{grad } g_i(x_k)\}_{i \in \mathcal{I} \cap \mathcal{A}}$ are linearly independent for all $k > k_3 > k_2$ for some k_3 . Define

$$\bar{\lambda}_i^k = \max \left\{ 0, \lambda_i^{k-1} + \rho_{k-1} g_i(x_k) \right\}, \quad \text{and} \quad \bar{\gamma}_j^k = \gamma_j^{k-1} + \rho_{k-1} h_j(x_k)$$

as the unclipped update. Define $S_k := \max\{\|\bar{\gamma}^k\|_\infty, \|\bar{\lambda}^k\|_\infty\}$. We are going to discuss separately for situations when S_k is bounded and when it is unbounded. If it is bounded, then denote a limit point of $\bar{\lambda}^k, \bar{\gamma}^k$ as $\bar{\lambda}$ and $\bar{\gamma}$. Let

$$v = \text{grad } f(\bar{x}) + \sum_{j \in \mathcal{E}} \bar{\gamma}_j \text{grad } h_j(\bar{x}) + \sum_{i \in \mathcal{I} \cap \mathcal{A}} \bar{\lambda}_i \text{grad } g_i(\bar{x}).$$

In order to prove that v is zero, we compare it to a similar vector defined at x_k , for all large k , and consider the limit $k \rightarrow \infty$. Unlike the Euclidean case in the proof in [5], we cannot directly compare tangent vectors in the tangent spaces at x_k and \bar{x} : we use parallel transport to bring all tangent vectors to the tangent space at \bar{x} :

$$\begin{aligned} \|v\| \leq & \left\| \text{grad } f(\bar{x}) - \mathcal{P}_{x_k \rightarrow \bar{x}} \text{grad } f(x_k) + \sum_{j \in \mathcal{E}} \bar{\gamma}_j (\text{grad } h_j(\bar{x}) - \mathcal{P}_{x_k \rightarrow \bar{x}} \text{grad } h_j(x_k)) \right. \\ & + \left. \sum_{i \in \mathcal{I} \cap \mathcal{A}} \bar{\lambda}_i (\text{grad } g_i(\bar{x}) - \mathcal{P}_{x_k \rightarrow \bar{x}} \text{grad } g_i(x_k)) \right\| \\ & + \left\| \mathcal{P}_{x_k \rightarrow \bar{x}} \text{grad } f(x_k) + \sum_{j \in \mathcal{E}} \bar{\gamma}_j \mathcal{P}_{x_k \rightarrow \bar{x}} \text{grad } h_j(x_k) \right. \\ & + \left. \sum_{i \in \mathcal{I} \cap \mathcal{A}} \bar{\lambda}_i \mathcal{P}_{x_k \rightarrow \bar{x}} \text{grad } g_i(x_k) \right\|. \end{aligned}$$

By Lemma A.2, the first term vanishes in the limit $k \rightarrow \infty$ since $x_k \rightarrow \bar{x}$. We can understand the second term using isometry of parallel transport and linearity:

$$\begin{aligned} & \left\| \mathcal{P}_{x_k \rightarrow \bar{x}} \text{grad } f(x_k) + \sum_{j \in \mathcal{E}} \bar{\gamma}_j \mathcal{P}_{x_k \rightarrow \bar{x}} \text{grad } h_j(x_k) + \sum_{i \in \mathcal{I} \cap \mathcal{A}} \bar{\lambda}_i \mathcal{P}_{x_k \rightarrow \bar{x}} \text{grad } g_i(x_k) \right\|_{\bar{x}} \\ & = \left\| \text{grad } f(x_k) + \sum_{j \in \mathcal{E}} \bar{\gamma}_j \text{grad } h_j(x_k) + \sum_{i \in \mathcal{I} \cap \mathcal{A}} \bar{\lambda}_i \text{grad } g_i(x_k) \right\|_{x_k} \\ & \leq \left\| \sum_{j \in \mathcal{E}} (\bar{\gamma}_j - \bar{\gamma}_j^k) \text{grad } h_j(x_k) + \sum_{i \in \mathcal{I} \cap \mathcal{A}} (\bar{\lambda}_i - \bar{\lambda}_i^k) \text{grad } g_i(x_k) \right\|_{x_k} \\ & + \left\| \text{grad } f(x_k) + \sum_{j \in \mathcal{E}} \bar{\gamma}_j^k \text{grad } h_j(x_k) + \sum_{i \in \mathcal{I}} \bar{\lambda}_i^k \text{grad } g_i(x_k) \right\|_{x_k} \\ & + \left\| \sum_{i \in \mathcal{I} \setminus \mathcal{A}} \bar{\lambda}_i^k \text{grad } g_i(x_k) \right\|_{x_k}. \end{aligned}$$

Here, the second term vanishes in the limit because it is upper bounded by ϵ_k (by assumption) and we let $\lim_{k \rightarrow \infty} \epsilon_k = 0$; the last term vanishes in the limit because of the discussion in the second paragraph; and the first term attains arbitrarily small values for large k as norms of gradients are bounded in a neighbourhood of \bar{x} and by definition of $\bar{\lambda}$ and $\bar{\gamma}$. Since v is independent of k , we conclude that $\|v\| = 0$. Therefore, \bar{x} satisfies KKT conditions.

On the other hand, if $\{S_k\}$ is unbounded, then for $k \geq k_3$, we have

$$\left\| \frac{1}{S_k} \text{grad } f(\bar{x}) + \sum_{j \in \mathcal{E}} \frac{\bar{\gamma}_j}{S_k} \text{grad } h_j(\bar{x}) + \sum_{i \in \mathcal{I}} \frac{\bar{\lambda}_i}{S_k} \text{grad } g_i(\bar{x}) \right\| \leq \frac{\epsilon_k}{S_k}.$$

As all the coefficients on the left-hand side are bounded in $[-1, 1]$, and by definition of S_k , the coefficient vector has a nonzero limit point. Denote it as $\bar{\lambda}$ and $\bar{\gamma}$. By a similar argument as above, taking the limit in k , we can obtain

$$\left\| \sum_{j \in \mathcal{E}} \bar{\gamma} \text{grad } h_j(\bar{x}) + \sum_{i \in \mathcal{I} \cap \mathcal{A}} \bar{\lambda} \text{grad } g_i(\bar{x}) \right\| = 0,$$

which contradicts the LICQ condition at \bar{x} . Hence, the situation that $\{S_k\}$ is unbounded does not take place, so we are left with the cases where it is bounded, for which we already showed that \bar{x} satisfies KKT condition. □

B Proof of Proposition 3.4

Proof The proof is adapted from Section 3 in [7]. Define $\bar{\gamma}_j^k = \gamma_j^{k-1} + \rho_{k-1} h_j(x_k)$. By Proposition 3.2, \bar{x} is a KKT point and by taking a subsequence of $\{x_k\}$ if needed, $\bar{\gamma}^k$ is bounded and converges to $\bar{\gamma}$.

For any tangent vector $d \in \mathcal{C}^W(\bar{x})$, we have $\langle d, \text{grad } h_j(\bar{x}) \rangle = 0$ for all $j \in \mathcal{E}$. Let $m = |\mathcal{E}|$, and dimension of \mathcal{M} be $n \geq m$. Let φ be a chart such that $\varphi(\bar{x}) = 0$. From [42, Prop. 8.1], the component functions of h_j with respect to this chart are smooth. Let $\partial_1 \dots \partial_n$ be the basis vectors of the given local chart. Let $d = (d_1 \partial_1, \dots, d_n \partial_n)$. Define: $\mathcal{F}: \mathbb{R}^{n+m} \rightarrow \mathbb{R}^m$, i.e. for $x \in \mathbb{R}^n, y \in \mathbb{R}^m, j \in \{1, \dots, m\}$ as

$$\mathcal{F}_j(x, y) = \langle (y_1 \partial_1, \dots, y_m \partial_m, d_{m+1} \partial_{m+1}, \dots, d_n \partial_n), \text{grad } h_j(\varphi^{-1}(x)) \rangle_{\varphi^{-1}(x)}.$$

If we denote h_j^l as the l th coordinate of vector $\text{grad } h_j$ in this system, and G_x as gram matrix for the metric where $G_{x_{p,q}} = \langle \partial_p, \partial_q \rangle_x$, then the above expression can be written as

$$\mathcal{F}_j(x, y) = [y_1, \dots, y_m, d_{m+1}, \dots, d_n] G_{\varphi^{-1}(x)} [h_j^1(\varphi^{-1}(x)), \dots, h_j^n(\varphi^{-1}(x))]^T.$$

and by abuse of notation where $[1 \dots m]$ means extracting the first m columns, we have

$$\frac{\partial \mathcal{F}_j}{\partial y} = \left([h_j^1(\varphi^{-1}(x)), \dots, h_j^n(\varphi^{-1}(x))] G_{\varphi^{-1}(x)} \right)_{[1 \dots m]}.$$

Notice that $[h^1(\varphi^{-1}(\bar{x})), \dots, h^n(\varphi^{-1}(\bar{x}))]$ has full row rank (by LICQ), so it has rank m . As $G_{\varphi^{-1}(\bar{x})}$ is invertible, $\frac{\partial \mathcal{F}}{\partial y}(\bar{x})$ must be invertible (reindex the columns from

the top of the proof for this $m \times n$ matrix if needed so that the m columns form a full rank matrix). Then, by the implicit function theorem, for a small neighbourhood U of $\varphi^{-1}(\bar{x})$, we have a continuously differentiable function $g : U \rightarrow \mathbb{R}^m$, where $g(\varphi^{-1}(\bar{x})) = [d_1, \dots, d_m]$ and

$$\mathcal{F}(x, g(\varphi^{-1}(x))) = 0.$$

For each x locally around \bar{x} , let

$$d_x = [g(\varphi^{-1}(x))_1 \partial_1, \dots, g(\varphi^{-1}(x))_m \partial_m, d_{m+1} \partial_{m+1}, \dots, d_n \partial_n] \in T_x \mathcal{M}.$$

These vectors then forms a smooth vector field such that $\langle d_x, \text{grad } h_j(x) \rangle = 0$ for all $j \in \mathcal{E}$, and $d = d_{\bar{x}}$. Then we have that

$$\begin{aligned} & \text{Hess}_x \mathcal{L}_{\rho_{k-1}}(x_k, \gamma^{k-1})(d_{x_k}, d_{x_k}) \\ &= \langle d_{x_k}, \text{Hess } f(x_k) d_{x_k} \rangle + \rho_{k-1} \sum_{j \in \mathcal{E}} \langle d_{x_k}, \nabla_{d_x} (h_j(x) + \frac{\gamma_j^{k-1}}{\rho_{k-1}}) \text{grad } h_j(x) \rangle_{x_k} \\ &= \langle d_{x_k}, \text{Hess } f(x_k) d_{x_k} \rangle + \rho_{k-1} \sum_{j \in \mathcal{E}} d_x [h_j(x) + \frac{\gamma_j^{k-1}}{\rho_{k-1}}](x_k) \langle d_x, \text{grad } h_j(x) \rangle_{x_k} \\ &\quad + \sum_{j \in \mathcal{E}} (\rho_{k-1} h_j(x_k) + \gamma_j^{k-1}) \langle d_x, \nabla_{d_x} \text{grad } h_j(x) \rangle_{x_k} \\ &= \langle d_{x_k}, \text{Hess } f(x_k) d_{x_k} \rangle + \sum_{j \in \mathcal{E}} (\rho_{k-1} h_j(x_k) + \gamma_j^{k-1}) \langle d_x, \nabla_{d_x} \text{grad } h_j(x) \rangle_{x_k} \\ &= \langle d_{x_k}, \nabla_{d_{x_k}} \text{grad } f(x_k) \rangle + \sum_{j \in \mathcal{E}} \bar{\gamma}_j^k \langle d_x, \nabla_{d_x} \text{grad } h_j(x) \rangle_{x_k} \end{aligned}$$

where the second equality is by definition of connection; the third is by orthogonality of d with $\{\text{grad } h_j\}$; the fourth is from the definition of Hessian and $\bar{\gamma}$. Therefore we have

$$\langle d_x, \nabla_{d_x} \text{grad } f(x) \rangle_{x_k} + \sum_{j \in \mathcal{E}} \bar{\gamma}_j^k \langle d_x, \nabla_{d_x} \text{grad } h_j(x) \rangle_{x_k} \geq -\epsilon_k \|d_{x_k}\|^2$$

Since the connection maps two continuously differentiable vector fields to a continuous vector field, we can take a limit and state:

$$\langle d, \nabla_d \text{grad } f(\bar{x}) \rangle + \sum_{j \in \mathcal{E}} \bar{\gamma}_j \langle d, \nabla_d \text{grad } h_j(\bar{x}) \rangle \geq 0$$

which is just $\text{Hess} \mathcal{L}(\bar{x}, \bar{\gamma})(d, d) \geq 0$. □

C Proof of Proposition 4.1

In the proof below, we use the following notation:

$$v \in F'(x^*, \lambda^*, \gamma^*) \Leftrightarrow \begin{cases} v \in T_{x^*} \mathcal{M}, \\ \langle \text{grad } h_j(x^*), v \rangle = 0 & \text{for all } j \in \mathcal{E}, \text{ and} \\ \langle \text{grad } g_i(x^*), v \rangle \leq 0 & \text{for all } i \in \mathcal{A}(x^*) \cap \mathcal{I}. \end{cases} \quad (31)$$

Proof Consider the function Q , defined by:

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

In a small enough neighbourhood of x^* , terms for inactive constraints disappear and Q is just:

$$Q(x, \rho) = f(x) + \rho \left(\sum_{i \in \mathcal{A}(x^*) \cap \mathcal{I}} \max\{0, g_i(x)\} + \sum_{j \in \mathcal{E}} |h_j(x)| \right).$$

Although Q is nonsmooth, it is easy to verify that it has directional derivative in all directions:

$$\lim_{\tau \rightarrow 0} \frac{\max\{0, g_i(\text{Exp}_{x^*}(\tau d))\} - \max\{0, g_i(x^*)\}}{\tau} = \lim_{\tau \rightarrow 0} \frac{\max\{0, g_i(\text{Exp}_{x^*}(\tau d))\}}{\tau}$$

and since $g_i \circ \text{Exp}_{x^*}$ is sufficiently smooth, discussing separately the sign of $\frac{d}{d\tau}(g_i \circ \text{Exp}_{x^*})(\tau d)$, we have the right hand side equal to $\max\{0, \frac{d}{d\tau}(g_i \circ \text{Exp}_{x^*})(\tau d)\} = \max\{0, \langle \text{grad } g_i(x^*), d \rangle\}$. Similarly, we have

$$\lim_{\tau \rightarrow 0} \frac{|h_j(\text{Exp}_{x^*}(\tau d))| - |h_j(x^*)|}{\tau} = \left| \frac{d}{d\tau}(h_j \circ \text{Exp}_{x^*})(\tau d) \right| = |\langle \text{grad } h_j(x^*), d \rangle|.$$

Hence, the directional derivative along direction d , $Q(x^*, \rho; d)$, is well defined:

$$Q(x^*, \rho; d) = \langle \text{grad } f(x^*), d \rangle + \rho \left(\sum_{i \in \mathcal{A}(x^*) \cap \mathcal{I}} \max\{0, \langle \text{grad } g_i(x^*), d \rangle\} + \sum_{j \in \mathcal{E}} |\langle \text{grad } h_j(x^*), d \rangle| \right). \quad (32)$$

As x^* is a KKT point,

$$\text{grad } f(x^*) + \sum_{i \in \mathcal{A}(x^*) \cap \mathcal{I}} \lambda_i^* \text{grad } g_i(x^*) + \sum_{j \in \mathcal{E}} \gamma_j^* \text{grad } h_j(x^*) = 0.$$

Thus,

$$\begin{aligned}
 0 &= \langle \text{grad } f(x^*), d \rangle + \sum_{i \in \mathcal{A}(x^*) \cap \mathcal{I}} \lambda_i^* \langle \text{grad } g_i(x^*), d \rangle + \sum_{j \in \mathcal{E}} \gamma_j^* \langle \text{grad } h_j(x^*), d \rangle \\
 &\leq \langle \text{grad } f(x^*), d \rangle + \sum_{i \in \mathcal{A}(x^*) \cap \mathcal{I}} \lambda_i^* \max\{0, \langle \text{grad } g_i(x^*), d \rangle\} \\
 &\quad + \sum_{j \in \mathcal{E}} \gamma_j^* |\langle \text{grad } h_j(x^*), d \rangle|.
 \end{aligned}$$

Combining with equation (32), we have

$$\begin{aligned}
 Q(x^*, \rho; d) &\geq \sum_{i \in \mathcal{A}(x^*) \cap \mathcal{I}} (\rho - \lambda_i^*) \max\{0, \langle \text{grad } g_i(x^*), d \rangle\} \\
 &\quad + \sum_{j \in \mathcal{E}} (\rho - \gamma_j^*) |\langle \text{grad } h_j(x^*), d \rangle|. \tag{33}
 \end{aligned}$$

For contradiction, suppose x^* is not a local minimum of Q . Then, there exists $\{y_k\}_{k=1}^\infty$, $\lim_{k \rightarrow \infty} y_k = x^*$ such that $Q(y_k, \rho) < Q(x^*, \rho) = f(x^*)$. By restricting to a small enough neighbourhood, there exists $\eta_k = \text{Exp}_{x^*}^{-1}(y_k)$. Considering only a subsequence if needed, we have $\lim_{k \rightarrow \infty} \frac{\eta_k}{\|\eta_k\|} = \bar{\eta}$. It is easy to see that $Q(\text{Exp}_{x^*}(\cdot), \rho)$ is locally Lipschitz continuous at 0_{x^*} , which gives

$$Q(\text{Exp}_{x^*}(\|\eta_k\|\bar{\eta}), \rho) = Q(\text{Exp}_{x^*}(\eta_k), \rho) + o(\|\eta_k\|) = Q(y_k, \rho) + o(\|\eta_k\|).$$

Subtract $Q(x^*, \rho)$ and take the limit:

$$\begin{aligned}
 \lim_{k \rightarrow \infty} \frac{Q(\text{Exp}_{x^*}(\|\eta_k\|\bar{\eta}), \rho) - Q(x^*, \rho)}{\|\eta_k\|} &= \lim_{k \rightarrow \infty} \frac{Q(y_k, \rho) - Q(x^*, \rho)}{\|\eta_k\|} \\
 + \lim_{k \rightarrow \infty} \frac{o(\|\eta_k\|)}{\|\eta_k\|} &\leq 0.
 \end{aligned}$$

Notice the left-most expression is just $Q(x^*, \rho; \bar{\eta})$. Since coefficients on the right-hand side of (33) are strictly positive, we must have $\langle \text{grad } g_i(x^*), \bar{\eta} \rangle \leq 0$ and $\langle \text{grad } h_j(x^*), \bar{\eta} \rangle = 0$. Since the exponential mapping is of second order, we have a Taylor expansion for f ,

$$f(y_k) = f(x^*) + \langle \text{grad } f(x^*), \eta_k \rangle + \frac{1}{2} \langle \eta_k, \text{Hess } f(x^*)[\eta_k] \rangle + o(\|\eta_k\|^2),$$

and similarly for g_i and h_j . Notice that

$$\begin{aligned}
 Q(y_k, \rho) &= f(y_k) + \rho \left(\sum_{i \in \mathcal{A}(x^*) \cap \mathcal{I}} \max\{0, g_i(y_k)\} + \sum_{j \in \mathcal{E}} |h_j(y_k)| \right) \\
 &\geq \left(f(y_k) + \sum_{i \in \mathcal{A}(x^*) \cap \mathcal{I}} \lambda_i^* g_i(y_k) + \sum_{j \in \mathcal{E}} \gamma_j^* h_j(y_k) \right) \\
 &\quad + \sum_{i \in \mathcal{A}(x^*) \cap \mathcal{I}} (\rho - \lambda_i^*) \max\{0, g_i(y_k)\} \\
 &\quad + \sum_{j \in \mathcal{E}} (\rho - \gamma_j^*) |h_j(y_k)| \\
 &\geq f(x^*) + \langle \text{grad } f(x^*) + \sum_{i \in \mathcal{A}(x^*) \cap \mathcal{I}} \lambda_i^* \text{grad } g_i(x^*) \\
 &\quad + \sum_{j \in \mathcal{E}} \gamma_j^* \text{grad } h_j(x^*), \eta_k \rangle \\
 &\quad + \frac{1}{2} \langle \eta_k, \text{Hess}(f(x^*) + \sum_{i \in \mathcal{A}(x^*) \cap \mathcal{I}} \lambda_i^* g_i(x^*) \\
 &\quad + \sum_{j \in \mathcal{E}} \gamma_j^* h_j(x^*))[\eta_k] \rangle + o(\|\eta_k\|^2) + P(y_k) \\
 &= f(x^*) + 0 + \frac{1}{2} \langle \eta_k, \text{Hess}(\mathcal{L}(x, \lambda^*, \gamma^*)(x^*)[\eta_k]) \rangle + o(\|\eta_k\|^2) + P(y_k)
 \end{aligned}$$

where $P(y_k) = \sum_{i \in \mathcal{A}(x^*) \cap \mathcal{I}} (\rho - \lambda_i^*) \max\{0, g_i(y_k)\} + \sum_{j \in \mathcal{E}} (\rho - \gamma_j^*) |h_j(y_k)|$. The first inequality follows from quadratic approximation of $f + \sum_{i \in \mathcal{A}(x^*) \cap \mathcal{I}} \lambda_i g_i + \sum_{j \in \mathcal{E}} \gamma_j h_j$ and bilinearity of the metric. The last equality comes from the definition of KKT points. Dividing the equation through by $\|\eta\|^2$, we obtain

$$\begin{aligned}
 \lim_{k \rightarrow \infty} \frac{Q(y_k, \rho) - f(x^*)}{\|\eta\|^2} &= \lim_{k \rightarrow \infty} \frac{1}{2} \left\langle \frac{\eta_k}{\|\eta_k\|}, \text{Hess}(\mathcal{L}(x, \lambda^*, \gamma^*)(x^*)) \left[\frac{\eta_k}{\|\eta_k\|} \right] \right\rangle \\
 &\quad + 0 + \lim_{k \rightarrow \infty} \frac{P(y_k)}{\|\eta\|^2}. \tag{34}
 \end{aligned}$$

If $\bar{\eta} \in F'$, then as $P(y_k) \geq 0$, the first term on the right hand side will be strictly larger than 0, which is a contradiction to $Q(y_k, \rho) < f(x^*)$ for all k . If $\bar{\eta} \in F - F'$, then there exists $g_{i'}$ such that $\langle \text{grad } g_{i'}(x), \bar{\eta} \rangle > 0$. Then,

$$g_{i'}(y_k) = g_{i'}(x^*) + \langle \text{grad } g_{i'}(x^*), \eta_k \rangle + o(\|\bar{\eta}\|) = \langle \text{grad } g_{i'}(x^*), \eta_k \rangle + o(\|\eta_k\|).$$

Hence, dividing the above expression by $\|\eta_k\|$ gives

$$\lim_{k \rightarrow \infty} \frac{g_i'(y_k)}{\|\eta_k\|} \geq \lim_{k \rightarrow \infty} \left\langle \text{grad } g_i'(x^*), \frac{\eta_k}{\|\eta_k\|} \right\rangle + 0 = \langle \text{grad } g_i'(x^*), \bar{\eta}_k \rangle > 0.$$

Notice that $\frac{P(y_k)}{\|\eta\|^2} \geq \frac{g_i'(y_k)}{\|\eta_k\|}$ for large enough k and a contradiction is obtained by plugging it into (34). □

D Proof of Proposition 4.2

Proof We give a proof for Q^{lsc} —it is analogous for Q^{lqh} . For each iteration k and for each $i \in \mathcal{I}$ and $j \in \mathcal{E}$, define the following coefficients:

$$\lambda_i^k = \frac{e^{g_i(x_{k+1})/u_k}}{1 + e^{g_i(x_{k+1})/u_k}}, \quad \text{and} \quad \gamma_j^k = \frac{e^{h_j(x_{k+1})/u_k} - e^{-h_j(x_{k+1})/u_k}}{e^{h_j(x_{k+1})/u_k} + e^{-h_j(x_{k+1})/u_k}}.$$

Then, a simple calculation shows that (under our assumptions, $\rho_k = \rho_0$ for all k ; we simply write ρ):

$$\begin{aligned} \text{grad } Q^{\text{lsc}}(x_{k+1}, \rho_k, u_k) &= \text{grad } f(x_{k+1}) + \rho \sum_{i \in \mathcal{I}} \lambda_i^k \text{grad } g_i(x_{k+1}) \\ &\quad + \rho \sum_{j \in \mathcal{E}} \gamma_j^k \text{grad } h_j(x_{k+1}). \end{aligned}$$

Notice that the multipliers are bounded: $\gamma_j^k \in [-1, 1]$ and $\lambda_i^k \in [0, 1]$. Hence, as sequences indexed by k , they have a limit point: we denote them by $\bar{\gamma} \in [-1, 1]$ and $\bar{\lambda} \in [0, 1]$. Furthermore, since \bar{x} is feasible, there exists k_1 such that for any $k > k_1$, $i \in \mathcal{I} \setminus \mathcal{A}(\bar{x})$, $g_i(x_k) < c$ for some constant $c < 0$. Then, as $u_k \rightarrow 0$, by definition, λ_i^k goes to 0 for $i \in \mathcal{I} \setminus \mathcal{A}(\bar{x})$. This shows $\bar{\lambda}_i = 0$ for $i \in \mathcal{I} \setminus \mathcal{A}(\bar{x})$. Considering a convergent subsequence if needed, there exists $k_2 > k_1$ such that, for all $k > k_2$, $\text{dist}(x_k, \bar{x}) < i(\bar{x})$ (the injectivity radius). Thus, parallel transport from each x_k to \bar{x} is well defined. Consider

$$v = \text{grad } f(\bar{x}) + \rho \sum_{i \in \mathcal{I} \cap \mathcal{A}(\bar{x})} \bar{\lambda}_i \text{grad } g_i(\bar{x}) + \rho \sum_{j \in \mathcal{E}} \bar{\gamma}_j \text{grad } h_j(\bar{x}).$$

Notice that its coefficients are bounded, so we can get $\|v\| = 0$ similar to the proof of Proposition 3.2. □

References

1. Absil, P.-A., Hosseini, S.: A collection of nonsmooth Riemannian optimization problems. Technical Report UCL-INMA-2017.08, Université catholique de Louvain (2017)

2. Absil, P.-A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton (2008)
3. Agarwal, N., Boumal, N., Bullins, B., Cartis, C.: Adaptive regularization with cubics on manifolds (2018). arXiv preprint [arXiv:1806.00065](https://arxiv.org/abs/1806.00065)
4. Albert, R., Barabási, A.-L.: Statistical mechanics of complex networks. *Rev. Mod. Phys.* **74**, 47–97 (2002)
5. Andreani, R., Birgin, E.G., Martínez, J.M., Schuverdt, M.L.: On augmented Lagrangian methods with general lower-level constraints. *SIAM J. Optim.* **18**(4), 1286–1309 (2007)
6. Andreani, R., Haeser, G., Martínez, J.M.: On sequential optimality conditions for smooth constrained optimization. *Optimization* **60**(5), 627–641 (2011)
7. Andreani, R., Haeser, G., Ramos, A., Silva, P.J.: A second-order sequential optimality condition associated to the convergence of optimization algorithms. *IMA J. Numer. Anal.* **37**, 1902–1929 (2017)
8. Bento, G., Ferreira, O., Melo, J.: Iteration-complexity of gradient, subgradient and proximal point methods on Riemannian manifolds. *J. Optim. Theory Appl.* **173**(2), 548–562 (2017)
9. Bento, G.C., Ferreira, O.P., Melo, J.G.: Iteration-complexity of gradient, subgradient and proximal point methods on Riemannian manifolds. *J. Optim. Theory Appl.* **173**(2), 548–562 (2017)
10. Bergmann, R., Herzog, R.: Intrinsic formulation of KKT conditions and constraint qualifications on smooth manifolds (2018). arXiv preprint [arXiv:1804.06214](https://arxiv.org/abs/1804.06214)
11. Bergmann, R., Persch, J., Steidl, G.: A parallel Douglas-Rachford algorithm for minimizing ROF-like functionals on images with values in symmetric Hadamard manifolds. *SIAM J. Imaging Sci.* **9**(3), 901–937 (2016)
12. Bertsekas, D.P.: Constrained Optimization and Lagrange Multiplier Methods. Athena Scientific, Belmont (1982)
13. Bertsekas, D.P.: Nonlinear Programming. Athena Scientific, Belmont (1999)
14. Birgin, E., Haeser, G., Ramos, A.: Augmented Lagrangians with constrained subproblems and convergence to second-order stationary points. *Optimization Online* (2016)
15. Birgin, E.G., Floudas, C.A., Martínez, J.M.: Global minimization using an augmented Lagrangian method with variable lower-level constraints. *Math. Program.* **125**(1), 139–162 (2010)
16. Birgin, E.G., Martínez, J.M.: Practical Augmented Lagrangian Methods for Constrained Optimization. SIAM (2014)
17. Boumal, N., Absil, P.-A., Cartis, C.: Global rates of convergence for nonconvex optimization on manifolds. *IMA J. Numer. Anal.* (2018)
18. Boumal, N., Mishra, B., Absil, P.-A., Sepulchre, R.: Manopt, a Matlab toolbox for optimization on manifolds. *J. Mach. Learn. Res.* **15**(1), 1455–1459 (2014)
19. Burer, S., Monteiro, R.D.: A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Program.* **95**(2), 329–357 (2003)
20. Byrd, R.H., Nocedal, J., Waltz, R.A.: Knitro: An integrated package for nonlinear optimization. In: Large-Scale Nonlinear Optimization, pp. 35–59. Springer (2006)
21. Cambier, L., Absil, P.-A.: Robust low-rank matrix completion by Riemannian optimization. *SIAM J. Sci. Comput.* **38**(5), S440–S460 (2016)
22. Carmo, M.P.: Riemannian Geometry. Birkhäuser, Boston (1992)
23. Carson, T., Mixon, D.G., Villar, S.: Manifold optimization for k-means clustering. In: Sampling Theory and Applications (SampTA), 2017 International Conference on, pp. 73–77. IEEE (2017)
24. Chatterjee, A., Madhav Govindu, V.: Efficient and robust large-scale rotation averaging. In: The IEEE International Conference on Computer Vision (ICCV) (December 2013)
25. Chen, C., Mangasarian, O.L.: Smoothing methods for convex inequalities and linear complementarity problems. *Math. Program.* **71**(1), 51–69 (1995)
26. Clarke, F.H.: Optimization and nonsmooth analysis. SIAM (1990)
27. Conn, A.R., Gould, G., Toint, P.L.: LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A), vol. 17. Springer, New York (2013)
28. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. *Math. Program.* **91**(2), 201–213 (2002)
29. Dreisigmeyer, D.W.: Equality constraints. Riemannian manifolds and direct search methods. *Optimization-Online* (2007)
30. Gould, N.I., Toint, P.L.: A note on the convergence of barrier algorithms to second-order necessary points. *Math. Program.* **85**(2), 433–438 (1999)

31. Grohs, P., Hosseini, S.: ε -subgradient algorithms for locally Lipschitz functions on Riemannian manifolds. *Adv. Comput. Math.* **42**(2), 333–360 (2016)
32. Guo, L., Lin, G.-H., Jane, J.Y.: Second-order optimality conditions for mathematical programs with equilibrium constraints. *J. Optim. Theory. Appl.* **158**(1), 33–64 (2013)
33. Hosseini, S., Huang, W., Yousefpour, R.: Line search algorithms for locally Lipschitz functions on Riemannian manifolds. *SIAM J. Optim.* **28**(1), 596–619 (2018)
34. Hosseini, S., Pouryayevali, M.: Generalized gradients and characterization of epi-Lipschitz sets in Riemannian manifolds. *Nonlinear Anal.* **74**(12), 3884–3895 (2011)
35. Huang, W., Absil, P.-A., Gallivan, K., Hand, P.: ROPTLIB: an object-oriented C++ library for optimization on Riemannian manifolds. Technical Report FSU16-14.v2, Florida State University (2016)
36. Huang, W., Gallivan, K.A., Absil, P.-A.: A Broyden class of quasi-Newton methods for Riemannian optimization. *SIAM J. Optim.* **25**(3), 1660–1685 (2015)
37. Johnstone, I.M., Lu, A.Y.: On consistency and sparsity for principal components analysis in high dimensions. *J. Am. Stat. Assoc.* **104**(486), 682–693 (2009)
38. Kanzow, C., Steck, D.: An example comparing the standard and safeguarded augmented Lagrangian methods. *Oper. Res. Lett.* **45**(6), 598–603 (2017)
39. Khuzani, M.B., Li, N.: Stochastic primal-dual method on Riemannian manifolds with bounded sectional curvature (2017). arXiv preprint [arXiv:1703.08167](https://arxiv.org/abs/1703.08167)
40. Kovnatsky, A., Glashoff, K., Bronstein, M.M.: Madmm: a generic algorithm for non-smooth optimization on manifolds. In: *European Conference on Computer Vision*, pp. 680–696. Springer (2016)
41. Lang, K.: Fixing two weaknesses of the spectral method. In: *Advances in Neural Information Processing Systems*, pp. 715–722 (2006)
42. Lee, J.: *Introduction to Smooth Manifolds*. Graduate Texts in Mathematics, vol. 218, 2nd edn. Springer, New York (2012)
43. Lee, J.M.: *Smooth manifolds*. In: *Introduction to Smooth Manifolds*, pp. 1–29. Springer (2003)
44. Lewis, A.S., Overton, M.L.: Nonsmooth optimization via BFGS. *SIAM J. Optim.* 1–35 (Submitted) (2009)
45. Lichman, M.: UCI machine learning repository (2013)
46. Montanari, A., Richard, E.: Non-negative principal component analysis: message passing algorithms and sharp asymptotics. *IEEE Trans. Inf. Theory* **62**(3), 1458–1484 (2016)
47. Nocedal, J., Wright, S.J.: *Numerical Optimization*, 2nd edn. Springer, New York (2006)
48. Parikh, N., Boyd, S.: *Proximal Algorithms*, vol. 1. Now Publishers inc., Hanover (2014)
49. Pinar, M.Ç., Zenios, S.A.: On smoothing exact penalty functions for convex constrained optimization. *SIAM J. Optim.* **4**(3), 486–511 (1994)
50. Ruszczynski, A.P.: *Nonlinear Optimization*, vol. 13. Princeton University Press, Princeton (2006)
51. Townsend, J., Koep, N., Weichwald, S.: Pymanopt: a Python toolbox for optimization on manifolds using automatic differentiation. *J. Mach. Learn. Res.* **17**, 1–5 (2016)
52. Weber, M., Sra, S.: Frank–Wolfe methods for geodesically convex optimization with application to the matrix geometric mean (2017). arXiv preprint [arXiv:1710.10770](https://arxiv.org/abs/1710.10770)
53. Yang, W.H., Zhang, L.-H., Song, R.: Optimality conditions for the nonlinear programming problems on Riemannian manifolds. *Pac. J. Optim.* **10**(2), 415–434 (2014)
54. Zass, R., Shashua, A.: Nonnegative sparse pca. In: *Advances in Neural Information Processing Systems*, pp. 1561–1568 (2007)
55. Zhang, J., Ma, S., Zhang, S.: Primal-dual optimization algorithms over Riemannian manifolds: an iteration complexity analysis (2017). arXiv preprint [arXiv:1710.02236](https://arxiv.org/abs/1710.02236)
56. Zhang, J., Zhang, S.: A cubic regularized Newton’s method over Riemannian manifolds (2018). arXiv preprint [arXiv:1805.05565](https://arxiv.org/abs/1805.05565)