SIAM J. OPTIM.
Vol. 28, No. 1, pp. 596–619

LINE SEARCH ALGORITHMS FOR LOCALLY LIPSCHITZ FUNCTIONS ON RIEMANNIAN MANIFOLDS*

S. HOSSEINI†, W. HUANG‡, AND R. YOUSEFPOUR§

Abstract. This paper presents line search algorithms for finding extrema of locally Lipschitz functions defined on Riemannian manifolds. To this end we generalize the so-called Wolfe conditions for nonsmooth functions on Riemannian manifolds. Using ε-subgradient-oriented descent directions and the Wolfe conditions, we propose a nonsmooth Riemannian line search algorithm and establish the convergence of our algorithm to a stationary point. Moreover, we extend the classical BFGS algorithm to nonsmooth functions on Riemannian manifolds. Numerical experiments illustrate the effectiveness and efficiency of the proposed algorithm.

Key words. Riemannian manifolds, Lipschitz functions, descent directions, Clarke subdifferential

AMS subject classifications. 49J52, 65K05, 58C05

DOI. 10.1137/16M1108145

1. Introduction.

This paper is concerned with the numerical solution of optimization problems defined on Riemannian manifolds where the objective function may be nonsmooth. Such problems arise in a variety of applications, e.g., in computer vision, signal processing, motion and structure estimation, and numerical linear algebra; see, for instance, [1, 2, 20, 30].

It is well known that the line search strategy is one of the basic iterative approaches to find a local minimum of an objective function defined on a linear space. For smooth functions defined on linear spaces, each iteration of a line search method computes a search direction and then shows how far to move along that direction. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a smooth function and the direction p be given, and define

$$\phi(\alpha) = f(x + \alpha p).$$

The problem that finds a step size in the direction p such that $\phi(\alpha) \leq \phi(0)$ is just line search about α. If we find α such that the objective function in the direction p is minimized, such a line search is called an exact line search. If we choose α such that the objective function has an acceptable descent amount, such a line search is called an inexact line search. Theoretically, an exact line search may not accelerate a line search algorithm due to, for example, the hemstitching phenomenon. Practically, exact optimal step sizes generally cannot be found, and it is also expensive to find almost exact step sizes. Therefore the inexact line search with less computation load is highly popular.

*Received by the editors December 15, 2016; accepted for publication (in revised form) January 17, 2018; published electronically March 6, 2018.
http://www.siam.org/journals/siopt/28-1/M110814.html
Funding: This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office. This work was supported by FNRS under grant PDR T.0173.13.

†Hausdorff Center for Mathematics and Institute for Numerical Simulation, University of Bonn, 53115 Bonn, Germany (hosseini@ins.uni-bonn.de).
‡Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005 (huwst08@gmail.com).
§Department of Mathematical Sciences, University of Mazandaran, Babolsar, Iran (yousefpour@umz.ac.ir).

A popular inexact line search condition stipulates that α should first of all give sufficient decrease in the objective function f, as usually measured by the following inequality, called the Armijo condition:

$$f(x+\alpha p) - f(x) \leq c_1 \alpha \langle \operatorname{grad} f(x), p\rangle_2 \tag{1.1}$$

for some $c_1 \in (0,1)$, where $\operatorname{grad} f(x)$ denotes the gradient of f at x and $\langle u, v\rangle_2$ denotes the Euclidean inner product $u^T v$. To rule out unacceptably short steps, a second requirement called the curvature condition is used, which requires α to satisfy

$$\langle p, \operatorname{grad} f(x+\alpha p)\rangle_2 \geq c_2 \langle \operatorname{grad} f(x), p\rangle_2$$

for some $c_2 \in (c_1, 1)$, where c_1 is the constant in (1.1). If α satisfies the Armijo and curvature conditions, then we say α satisfies the Wolfe conditions.

In smooth optimization algorithms on linear spaces, Zoutendijk's theorem describes how far the descent direction p at x can deviate from $-\operatorname{grad} f(x)$ to have a globally convergent iteration. In particular, if the cosine of the angle between the search direction p and negative of the gradient at the iteration x, defined as

$$\cos\theta = \frac{\langle -\operatorname{grad} f(x), p\rangle_2}{\|\operatorname{grad} f(x)\|\|p\|}, \tag{1.2}$$

is bounded away from zero and the Wolfe conditions at every iteration are satisfied, then the convergence results can be obtained; see [25]. Indeed, classical convergence results establish that accumulation points of the sequence of iterates are stationary points of the objective function f and the convergence of the whole sequence to a single limit-point is not guaranteed. The question is whether similar results are correct in nonsmooth optimization problems. In [34], the authors generalized the aforementioned Wolfe conditions for nonsmooth convex functions. They used the Clarke subdifferential instead of the gradient. But to obtain convergence, one must have not only well-chosen step lengths but also well-chosen search directions. In nonsmooth problems the angle defined in (1.2) does not propose a proper set of search directions. However, a condition on descent directions called subgradient-oriented for nonsmooth objective functions on linear spaces, to find a proper search direction, has been introduced in [26].

Euclidean spaces are not the only spaces in which optimization algorithms are used. There are many applications of optimization on Riemannian manifolds. A manifold, in general, does not have a linear structure, hence the usual techniques, which are often used to study optimization problems on linear spaces, cannot be applied and new techniques need to be developed. The common denominator of approaches in optimization methods on manifolds is that instead of conducting a linear step during the line search procedure, one uses retractions or defines the step along a geodesic via the use of the exponential map.

Contribution. Our main contributions are fourfold. First, we generalize the concept of a subgradient-oriented descent sequence from [26], to Riemannian manifolds. We define also a new notion called the ε-subgradient-oriented descent sequence. Then we present a numerical search direction algorithm to find a descent direction for a nonsmooth objective function defined on a Riemannian manifold. In this algorithm, we use a positive definite matrix P in order to define a P-norm equivalent to the usual norm induced by the inner product on our tangent space. If we use the identity matrix and, therefore, work with the usual norm on the tangent space, then the

algorithm reduces to the descent search algorithm presented in [9]. Second, we define a nonsmooth Armijo condition on Riemannian manifolds, which is a generalization of the nonsmooth Armijo condition presented in [34] to a Riemannian setting. Similar to Euclidean spaces we can add a curvature condition to the nonsmooth Armijo condition to get a nonsmooth generalization of the Wolfe conditions on Riemannian manifolds. This curvature condition is indeed a Riemannian version of the curvature condition presented in [34]. However, due to working on different tangent spaces, it is not a trivial generalization and using a notion of vector transport is needed. We present also numerical line search algorithms to find a suitable step length satisfying the Wolfe conditions for nonsmooth optimization problems on Riemannian manifolds and study the behavior of the algorithms. The idea of these algorithms is inspired by some similar algorithms from [35]. Third, we combine the search direction algorithm with the line search algorithm to define a minimization algorithm for a nonsmooth optimization problem on a Riemannian manifold. To prove the convergence results for our minimization algorithm, we need to have a sequence of ε-subgradient-oriented descent directions; hence it is important to update the sequence of positive definite matrices, which define the equivalent norms on the tangent spaces, such that the sequences of their smallest and largest eigenvalues are bounded. As our last contribution in this paper, we also plan to present a practical strategy to update the sequence of matrices to impose such a condition on the sequences of eigenvalues. This strategy can be seen as a version of the nonsmooth BFGS method on Riemannian manifolds, which is presented in this setting for the first time and can be considered as a generalization of the smooth BFGS on Riemannian manifolds in [15]. To the best of our knowledge, this version of nonsmooth BFGS has not been presented before for optimization problems on linear spaces; therefore it is new not only for Riemannian settings but also for linear spaces.

This paper is organized as follows. Section 2 presents the proposed Riemannian optimization for nonsmooth cost functions. Specifically, sections 2.1 and 2.2 respectively analyze the line search conditions and search direction for nonsmooth functions theoretically. Sections 2.3 and 2.4 respectively give a practical approach to compute a search direction and a step size. Section 2.5 combines the search direction with the line search algorithm and gives a minimization algorithm. This algorithm can be combined with the BFGS strategy and the result is presented in section 3. Finally, experiments that compare the proposed algorithm with the Riemannian BFGS and Riemannian gradient sampling (RGS) are reported in section 4.

Previous work. For the smooth optimization on Riemannian manifolds the line search algorithms have been studied in [1, 29, 32, 33]. In considering optimization problems with nonsmooth objective functions on Riemannian manifolds, it is necessary to generalize concepts of nonsmooth analysis to Riemannian manifolds. In the past few years a number of results have been obtained on numerous aspects of nonsmooth analysis on Riemannian manifolds, [3, 4, 11, 12, 13, 22]. Papers [9, 10] are among the first papers on numerical algorithms for minimization of nonsmooth functions on Riemannian manifolds.

2. Line search algorithms on Riemannian manifolds.

In this paper, we use the standard notation and known results of Riemannian manifolds; see, e.g., [19, 31]. Throughout this paper, M is an n-dimensional complete manifold endowed with a Riemannian metric $\langle .,. \rangle$ on the tangent space T_xM. We identify tangent space of M at a point x, denoted by T_xM, with the cotangent space at x (via the Riemannian metric), denoted by T_xM^*. We denote by cl$N$ the closure of the set N and by conv$N$

the convex hull of the set N. Also, let S be a nonempty closed subset of a Riemannian manifold M, we define $\mathrm{dist}_S : M \longrightarrow \mathbb{R}$ by

$$\mathrm{dist}_S(x) := \inf\{\mathrm{dist}(x, s) : s \in S\},$$

where dist is the Riemannian distance on M. We use of a class of mappings called retractions.

DEFINITION 2.1 (retraction). *A retraction on a manifold M is a smooth map $R : TM \to M$ with the following properties. Let R_x denote the restriction of R to T_xM.*

- *$R_x(0_x) = x$, where 0_x denotes the zero element of T_xM.*
- *With the canonical identification $T_{0_x}T_xM \simeq T_xM$, $DR_x(0_x) = id_{T_xM}$, where id_{T_xM} denotes the identity map on T_xM.*

By the inverse function theorem, we have that R_x is a local diffeomorphism. For example, the exponential map defined by $\exp : TM \to M$, $v \in T_xM \to \exp_x v$, $\exp_x(v) = \gamma(1)$, where γ is a geodesic starting at x with initial tangent vector v, is a retraction; see [1]. The exponential map is the most natural retraction to use on Riemannian manifolds and is used frequently in the early literature on the development of numerical algorithms on manifolds. Unfortunately, the exponential map is itself defined as the solution of a nonlinear ordinary differential equation and generally poses significant numerical challenges to be computed cheaply. Therefore, we consider alternatives in the form of retractions, which can be computed more cheaply; see [1, section 4.1] for examples of retractions on sphere and Stiefel manifolds. We define $B_R(x, \varepsilon)$ to be $\{R_x(\eta_x) | \|\eta_x\| < \varepsilon\}$. If the retraction R is the exponential function exp, then $B_R(x, \varepsilon)$ is the open ball centered at x with radius ε. By using retractions, we extend the concepts of nonsmooth analysis on Riemannian manifolds.

Recall that if G is a locally Lipschitz function defined from a Hilbert space X to $\mathbb{R}$, the Clarke generalized directional derivative of G at the point $x \in X$ in the direction $v \in X$, denoted by $G^\circ(x; v)$, is defined by

$$G^\circ(x; v) = \limsup_{y \to x,\ t \downarrow 0} \frac{G(y + tv) - G(y)}{t},$$

and the generalized subdifferential of G at x, denoted by $\partial G(x)$, is defined by

$$\partial G(x) := \{\xi \in X : \ \langle \xi, v\rangle \leq G^\circ(x; v) \text{ for all } v \in X\}.$$

Let $f : M \to \mathbb{R}$ be a locally Lipschitz function on a Riemannian manifold. For $x \in M$, we let $\hat{f}_x = f \circ R_x$ denote the restriction of the pullback $\hat{f} = f \circ R$ to T_xM. The Clarke generalized directional derivative of f at x in the direction $p \in T_xM$, denoted by $f^\circ(x; p)$, is defined by $f^\circ(x; p) = \hat{f}_x^{\,\circ}(0_x; p)$, where $\hat{f}_x^{\,\circ}(0_x; p)$ denotes the Clarke generalized directional derivative of $\hat{f}_x : T_xM \to \mathbb{R}$ at 0_x in the direction $p \in T_xM$. Therefore, the generalized subdifferential of f at x, denoted by $\partial f(x)$, is defined by $\partial f(x) = \partial \hat{f}_x(0_x)$. Note that there are other equivalent definitions for the Clarke generalized directional derivative and generalized subdifferential of functions defined on Riemannian manifolds; see [3, 6, 11]. The point x is a stationary point of f if $0 \in \partial f(x)$. A necessary condition that f achieves a local minimum at x is that x is a stationary point of f; see [9, 11]. Theorem 2.2 can be proved along the same lines as [11, Theorem 2.9].

Algorithm 1. A line search minimization algorithm on a Riemannian manifold.

Require: A Riemannian manifold M, a function $f : M \to \mathbb{R}$.
Input: $x_0 \in M, k = 0$.
Output: Sequence $\{x_k\}$.
repeat
  Choose a retraction $R_{x_k} : T_{x_k}M \to M$.
  Choose a descent direction $p_k \in T_{x_k}M$.
  Choose a step length $\alpha_k \in \mathbb{R}$.
  Set $x_{k+1} = R_{x_k}(\alpha_k p_k)$; $k = k + 1$.
until x_{k+1} sufficiently minimizes f.

THEOREM 2.2. *Let M be a Riemannian manifold, $x \in M$, and $f : M \to \mathbb{R}$ be a Lipschitz function of Lipschitz constant L near x, i.e., $|f(x) - f(y)| \leq L\text{dist}(x, y)$, for all y in a neighborhood x. Then*

(a) *$\partial f(x)$ is a nonempty, convex, compact subset of T_xM, and $\|\xi\| \leq L$ for every $\xi \in \partial f(x)$;*
(b) *for every v in T_xM, we have*

$$f^\circ(x; v) = \max\{\langle \xi, v\rangle : \quad \xi \in \partial f(x)\};$$

(c) *if $\{x_i\}$ and $\{\xi_i\}$ are sequences in M and TM such that $\xi_i \in \partial f(x_i)$ for each i, and if $\{x_i\}$ converges to x and ξ is a cluster point of the sequence $\{\xi_i\}$, then we have $\xi \in \partial f(x)$;*
(d) *∂f is upper semicontinuous at x.*

In classical optimization on linear spaces, line search methods are extensively used. They are based on updating the iterate by finding a direction and then adding a multiple of the obtained direction to the previous iterate. The extension of line search methods to manifolds is possible by the notion of retraction. We consider algorithms of the general forms stated in Algorithm 1.

Once the retraction R_{x_k} is defined, the search direction p_k and the step length α_k remain. We say p_k is a descent direction at x_k if there exists $\alpha > 0$ such that for every $t \in (0, \alpha)$, we have

$$f(R_{x_k}(tp_k)) - f(x_k) < 0.$$

It is obvious that if $f^\circ(x_k; p_k) < 0$, then p_k is a descent direction at x_k.

In order to have global convergence results, some conditions must be imposed on the descent direction p_k as well as the step length α_k.

2.1. Step length. The step length α_k has to cause a substantial reduction of the objective function f. The ideal choice would be $\alpha_k = \text{argmin}_{\alpha>0} f(R_{x_k}(\alpha p_k))$ if this exact line search can be carried out efficiently. But in general, it is too expensive to find this value. A more practical strategy to identify a step length that achieves adequate reductions in the objective function at minimal cost is an inexact line search. A popular inexact line search condition stipulates that the step length α_k should give a sufficient decrease in the objective function f, which is measured by the following condition.

DEFINITION 2.3 (Armijo condition). *Let $f : M \to \mathbb{R}$ be a locally Lipschitz function on a Riemannian manifold M with a retraction R, $x \in M$, and $p \in T_xM$. If the inequality for a step length α and a fixed constant $c_1 \in (0, 1)$*

$$f\left(R_x(\alpha p)\right) - f(x) \leq c_1 \alpha f^\circ(x; p)$$

holds, then α satisfies in the Armijo condition.

The existence of such a step size is proven later in Theorem 2.8.

2.1.1. The Wolfe conditions. As shown in the proof of Theorem 2.8, a short enough step size satisfies the Armijo condition. However, too small a step size prevents convergence of an algorithm. There are useful conditions to rule out unacceptably short step lengths. For example, one can use a requirement called the curvature condition. To present this requirement for nonsmooth functions on nonlinear spaces, some preliminaries are needed. To define the curvature condition on a Riemannian manifold, we have to translate a vector from one tangent space to another one.

DEFINITION 2.4 (vector transport). *A vector transport associated to a retraction R is defined as a continuous function $\mathcal{T} : TM \times TM \to TM$, $(\eta_x, \xi_x) \mapsto \mathcal{T}_{\eta_x}(\xi_x)$, which for all (η_x, ξ_x) satisfies the following conditions:*

(i) *$\mathcal{T}_{\eta_x} : T_xM \to T_{R(\eta_x)}M$ is a linear map,*

(ii) *$\mathcal{T}_{0_x}(\xi_x) = \xi_x$.*

In short, if $\eta_x \in T_xM$ and $R_x(\eta_x) = y$, then $\mathcal{T}_{\eta_x}$ transports vectors from the tangent space of M at x to the tangent space at y. Two additional properties are needed in this paper. First, the vector transport needs to preserve inner products, that is,

$$\langle \mathcal{T}_{\eta_x}(\xi_x), \mathcal{T}_{\eta_x}(\zeta_x) \rangle = \langle \xi_x, \zeta_x \rangle. \tag{2.1}$$

In particular, $\xi_x \mapsto \mathcal{T}_{\eta_x}(\xi_x)$ is then an isometry and possesses an isometric inverse.

Second, we will assume that $\mathcal{T}$ satisfies the following condition, called *locking condition* in [17], for transporting vectors along their own direction:

$$\mathcal{T}_{\xi_x}(\xi_x) = \beta_{\xi_x} \mathcal{T}_{R_{\xi_x}}(\xi_x), \quad \beta_{\xi_x} = \frac{\|\xi_x\|}{\|\mathcal{T}_{R_{\xi_x}} \xi_x\|}, \tag{2.2}$$

where

$$\mathcal{T}_{R_{\eta_x}}(\xi_x) = DR_x(\eta_x)(\xi_x) = \frac{d}{dt} R_x(\eta_x + t\xi_x)|_{t=0}.$$

These conditions can be difficult to verify but are in particular satisfied for the most natural choices of R and $\mathcal{T}$; for example, the exponential map as a retraction and the parallel transport as a vector transport satisfy these conditions with $\beta_{\xi_x} = 1$. For a further discussion, especially on construction of vector transports satisfying the locking condition, we refer to [17, section 4]. We introduce more intuitive notation:

$$\mathcal{T}_{x \to y}(\xi_x) = \mathcal{T}_{\eta_x}(\xi_x), \quad \mathcal{T}_{x \leftarrow y}(\xi_y) = (\mathcal{T}_{\eta_x})^{-1}(\xi_y) \quad \text{whenever } y = R_x(\eta_x).$$

Now we present the nonsmooth curvature condition for locally Lipschitz functions on Riemannian manifolds.

DEFINITION 2.5 (curvature condition). *The step length α satisfies in the curvature condition if the following inequality holds for the constant $c_2 \in (c_1, 1)$:*

$$\sup_{\xi \in \partial f(R_x(\alpha p))} \left\langle \xi, \frac{1}{\beta_{\alpha p}} \mathcal{T}_{x \to R_x(\alpha p)}(p) \right\rangle \geq c_2 f^\circ(x; p),$$

where c_1 is the constant in Definition 2.3.

Note that if there exists $\xi \in \partial f(R_x(\alpha p))$ such that

$$\left\langle \xi, \frac{1}{\beta_{\alpha p}} \mathcal{T}_{x \to R_x(\alpha p)}(p) \right\rangle \geq c_2 f^\circ(x; p),$$

then the curvature inequality holds. As in the smooth case, we can define a strong curvature condition by

$$\left| \sup_{\xi \in \partial f(R_x(\alpha p))} \left\langle \xi, \frac{1}{\beta_{\alpha p}} \mathcal{T}_{x \to R_x(\alpha p)}(p) \right\rangle \right| \leq -c_2 f^\circ(x; p).$$

The following lemma can be proved using Lemma 3.1 of [23].

LEMMA 2.6. *Let $f : M \to \mathbb{R}$ be a locally Lipschitz function on a Riemannian manifold M and the function W defined by*

$$W(\alpha) := f(R_x(\alpha p)) - f(x) - c_2 \alpha f^\circ(x; p), \tag{2.3}$$

where $c_2 \in (c_1, 1)$, $x \in M$, and $p \in T_x M$, be increasing on a neighborhood of some α_0; then α_0 satisfies the curvature condition.

Indeed, if W is increasing on a neighborhood of some α_0, then there exists ξ in

$$\partial W(\alpha_0) \subset \langle \partial f(R_x(\alpha_0 p)), DR_x(\alpha_0 p)(p) \rangle - c_2 f^\circ(x; p)$$

such that $\xi \geq 0$. Then the result will be obtained using the locking condition.

DEFINITION 2.7 (Wolfe conditions). *Let $f : M \to \mathbb{R}$ be a locally Lipschitz function and $p \in T_x M$. If α satisfies the Armijo and curvature conditions, then we say α satisfies the Wolfe conditions.*

In the following theorem the existence of step lengths satisfying the Wolfe conditions under some assumptions is proved.

THEOREM 2.8. *Assume that $f : M \to \mathbb{R}$ is a locally Lipschitz function on a Riemannian manifold M, $R_x : T_x M \to M$ is a retraction, $p \in T_x M$ is chosen such that $f^\circ(x; p) < 0$, and f is bounded below on $\{R_x(\alpha p) : \alpha > 0\}$; if $0 < c_1 < c_2 < 1$, then there exist step lengths satisfying the Wolfe conditions.*

Proof. First, we prove that the line $l(\alpha) = f(x) + \alpha c_1 f^\circ(x; p)$ intersects the graph of the function $\phi(\alpha) = f(R_x(\alpha p))$ at least once. Let us assume on the contrary that this line never intersects the graph of the function ϕ. Since $l - \phi$ is a continuous function and $l(\alpha) - \phi(\alpha) \neq 0$ for all $\alpha > 0$, we conclude that either $l(\alpha) < \phi(\alpha)$ for all $\alpha > 0$ or $l(\alpha) > \phi(\alpha)$ for all $\alpha > 0$. If $l(\alpha) < \phi(\alpha)$ for all $\alpha > 0$, then

$$f^\circ(x; p) < c_1 f^\circ(x; p) \leq \limsup_{\alpha \to 0} \frac{f(R_x(\alpha p)) - f(x)}{\alpha} \leq f^\circ(x; p),$$

which is a contradiction. It means that $l(\alpha) > \phi(\alpha)$ for all $\alpha > 0$. But since $l(\alpha)$ is not bounded below and $\phi(\alpha)$ is bounded below, this cannot be true and we get again a contradiction. Therefore, there exists $\hat{\alpha} > 0$ such that $l(\hat{\alpha}) = \phi(\hat{\alpha})$. Let α_1 be the smallest intersecting value of α. It can be shown that $\alpha_1 > 0$ as follows. Since $c_1 \in (0, 1)$ and $f^\circ(x, p) < 0$, there exists $t^* \in (0, 1)$ such that $f(R_x(tp)) - f(x) < t c_1 f^\circ(x, p)$ for all $t \in (0, t^*)$. This implies that $0 < t^* \leq \alpha_1$.

Hence

$$f(R_x(\alpha_1 p)) = f(x) + \alpha_1 c_1 f^\circ(x; p). \tag{2.4}$$

Algorithm 2. A backtracking line search on a Riemannian manifold.

1: **Require**: A Riemannian manifold M, a locally Lipschitz function $f : M \to \mathbb{R}$, a retraction R from TM to M, scalars $c_1, \rho \in (0, 1)$.
2: **Input**: $\alpha_0 > 0$.
3: **Output**: α_k.
4: $\alpha = \alpha_0$.
5: **repeat**
6: $\quad \alpha = \rho\alpha$.
7: **until** $f(R_{x_k}(\alpha p_k)) - f(x_k) \leq c_1 \alpha f^\circ(x_k; p_k)$.
8: Terminate with $\alpha_k = \alpha$.

We claim that $\phi(\alpha) < l(\alpha)$ for all $\alpha < \alpha_1$ and therefore the Armijo condition is satisfied for all $\alpha < \alpha_1$. To prove the claim, note that since α_1 is the smallest step size for which $\phi(\alpha) - l(\alpha) = 0$ holds, hence for every $\alpha < \alpha_1$, we have either $\phi(\alpha) - l(\alpha) < 0$ or $\phi(\alpha) - l(\alpha) > 0$. If $\phi(\alpha) - l(\alpha) > 0$ for all $\alpha < \alpha_1$, then

$$f^\circ(x; p) < c_1 f^\circ(x; p) \leq \limsup_{\alpha \to 0} \frac{f(R_x(\alpha p)) - f(x)}{\alpha} \leq f^\circ(x; p),$$

which is a contradiction. Hence we have $\phi(\alpha) - l(\alpha) < 0$ for all $\alpha < \alpha_1$, which proves the claim. Now by the mean value theorem for locally Lipschitz functions on Riemannian manifolds [11, Theorem 3.3], there exist $\varepsilon^* \in (0, 1)$ and $\xi \in \partial f(R_x(\varepsilon^* \alpha_1 p))$ such that

$$f(R_x(\alpha_1 p)) - f(x) = \alpha_1 \langle \xi, DR_x(\varepsilon^* \alpha_1 p)(p) \rangle. \tag{2.5}$$

By combining (2.4) and (2.5), we obtain $\langle \xi, DR_x(\varepsilon^* \alpha_1 p)(p) \rangle = c_1 f^\circ(x; p) > c_2 f^\circ(x; p)$. Using the locking condition, we conclude that $\varepsilon^* \alpha_1$ satisfies the curvature condition. ∎

Remark 2.9. There are a number of rules for choosing the step length α for problems on linear spaces; see [24, 34]. We can define their generalizations on Riemannian manifolds using the concepts of nonsmooth analysis on Riemannian manifolds and the notions of retraction and vector transport. For instance, one can use a generalization of the Mifflin condition, proposed first by Mifflin in [24]. The step length α satisfies the Mifflin condition if the following inequalities hold for the fixed constants $c_1 \in (0, 1), c_2 \in (c_1, 1)$:

$$f(R_x(\alpha p)) - f(x) \leq -c_1 \alpha \|p\|,$$

$$\sup_{\xi \in \partial f(R_x(\alpha p))} \left\langle \xi, \frac{1}{\beta_{\alpha p}} \mathcal{T}_{x \to R_x(\alpha p)}(p) \right\rangle \geq -c_2 \|p\|.$$

2.1.2. Sufficient decrease and backtracking. As explained in subsection 2.1.1, the Armijo condition does not ensure that the line search algorithm makes reasonable progress. But if the line search algorithm selects its step lengths appropriately, by using a backtracking approach, we can dispense with the curvature condition and use just the Armijo condition to terminate the line search procedure. We present here a backtracking line search algorithm, which makes adequate progress. An adequate step length will be found after a finite number of iterations, because α_k will finally become small enough that the Armijo condition holds.

2.2. Descent directions. To obtain a global convergence result for a line search method, we must have not only well-chosen step lengths but also well-chosen search directions. The following definition is equivalent to gradient-orientedness carried over nonsmooth problems; see [26]. We know that the search direction for a smooth optimization problem often has the form $p_k = -P_k \operatorname{grad} f(x_k)$, where P_k is a symmetric and nonsingular linear map. Therefore, it is not far from expectation to use elements of the subdifferential of f at x_k in Definition 2.10 and produce a subgradient-oriented descent sequence in nonsmooth problems.

DEFINITION 2.10 (subgradient-oriented descent sequence). *A sequence $\{p_k\}$ of descent directions is called subgradient-oriented if there exist a sequence of subgradients $\{g_k\}$ and a sequence of symmetric linear maps $\{P_k : T_{x_k}M \to T_{x_k}M\}$ satisfying*

$$0 < \lambda \leq \lambda_{min}(P_k) \leq \lambda_{max}(P_k) \leq \Lambda < \infty$$

for $0<\lambda<\Lambda<\infty$ and all $k \in \mathbb{N}$ such that $p_k = -P_k g_k$, where $\lambda_{min}(P_k)$ and $\lambda_{max}(P_k)$ denote respectively the smallest and largest eigenvalues of P_k.

In the next definition, we present an approximation of the subdifferential which can be computed approximately. As we aim at transporting subgradients from tangent spaces at nearby points of $x \in M$ to the tangent space at x, it is important to define a notion of injectivity radius for R_x. Let

$$\iota(x) := \sup\{\varepsilon > 0 |\ \ R_x : B(0_x, \varepsilon) \to B_R(x, \varepsilon) \text{ is injective}\}.$$

Then the *injectivity radius of M with respect to the retraction R* is defined as

$$\iota(M) := \inf_{x \in M} \iota(x).$$

When using the exponential map as a retraction, this definition coincides with the usual one.

DEFINITION 2.11 (ε-subdifferential). *Let $f : M \to \mathbb{R}$ be a locally Lipschitz function on a Riemannian manifold M and $0 < 2\varepsilon < \iota(x)$.*[1] *We define the ε-subdifferential of f at x denoted by $\partial_\varepsilon f(x)$ as follows;*

$$\partial_\varepsilon f(x) = \operatorname{clconv}\{\beta_\eta^{-1} \mathcal{T}_{x \leftarrow y}(\partial f(y)) : y \in \operatorname{cl} B_R(x, \varepsilon) \text{ and } \eta = R_x^{-1}(y)\}.$$

Every element of the ε-subdifferential is called an ε-subgradient.

DEFINITION 2.12 (ε-subgradient-oriented descent sequence). *A sequence $\{p_k\}$ of descent directions is called ε-subgradient-oriented if there exist a sequence of ε-subgradients $\{g_k\}$ and a sequence of symmetric linear maps $\{P_k : T_{x_k}M \to T_{x_k}M\}$ satisfying*

$$0 < \lambda \leq \lambda_{min}(P_k) \leq \lambda_{max}(P_k) \leq \Lambda < \infty$$

for $0 < \lambda < \Lambda < \infty$ and all $k \in \mathbb{N}$ such that $p_k = -P_k g_k$, where $\lambda_{min}(P_k)$ and $\lambda_{max}(P_k)$ denote respectively the smallest and largest eigenvalues of P_k.

From now, we assume that a basis of T_xM for all $x \in M$ is given and we denote every linear map using its matrix representation with respect to the given basis. In the following, we use a positive definite matrix P in order to define a P-norm equivalent to the usual norm induced by the inner product on our tangent space. Indeed $\|\xi\|_P = \langle P\xi, \xi\rangle^{1/2}$ and

[1] Note $y \in \operatorname{cl} B_R(x, \varepsilon)$. The coefficient 2 guarantees inverse vector transports are well defined on the boundary of $B_R(x, \varepsilon)$.

$$\lambda_{min}(P)\|.\|^2 \leq \|.\|_P^2 \leq \lambda_{max}(P)\|.\|^2. \tag{2.6}$$

THEOREM 2.13. *Assume that $f : M \to \mathbb{R}$ is a locally Lipschitz function on a Riemannian manifold M, $R_x : T_xM \to M$ is a retraction, $0 \notin \partial_\varepsilon f(x)$, and $0 < 2\varepsilon < \iota(x)$. Define*

$$g := \operatorname{argmin}_{\xi \in \partial_\varepsilon f(x)} \|\xi\|_P,$$

where P is a positive definite matrix. Assume that $p = -Pg$. Then $f_\varepsilon^\circ(x;p) = -\|g\|_P^2$ and p is a descent direction, where $f_\varepsilon^\circ(x;p) = \sup_{\xi \in \partial_\varepsilon f(x)} \langle \xi, -Pg\rangle$.

Proof. We first prove that $f_\varepsilon^\circ(x;p) = -\|g\|_P^2$. It is clear that

$$f_\varepsilon^\circ(x;p) = \sup_{\xi \in \partial_\varepsilon f(x)} \langle \xi, -Pg\rangle \geq \langle g, -Pg\rangle = -\|g\|_P^2.$$

Now we claim that $\|g\|_P^2 \leq \langle \xi, Pg\rangle$ for every $\xi \in \partial_\varepsilon f(x)$, which implies $\sup_{\xi \in \partial_\varepsilon f(x)} \mathrm{s}\langle \xi, -Pg\rangle \leq -\|g\|_P^2$. The proof of the claim is as follows: assume on the contrary; there exists $\xi \in \partial_\varepsilon f(x)$ such that $\langle \xi, Pg\rangle < \|g\|_P^2$, and consider $w := g + t(\xi - g) \in \partial_\varepsilon f(x)$, then

$$\|g\|_P^2 - \|w\|_P^2 = -t(2\langle \xi - g, Pg\rangle + t\langle \xi - g, P(\xi - g)\rangle).$$

Note that

$$2\langle \xi - g, Pg\rangle + t\langle \xi - g, P(\xi - g)\rangle = 2(\langle \xi, Pg\rangle - \|g\|_P^2) + t\|\xi - g\|_P^2;$$

therefore, if

$$0 < t \leq \min\left\{1, \frac{-2(\langle \xi, Pg\rangle - \|g\|_P^2)}{\|\xi - g\|_P^2}\right\},$$

then we have $\|g\|_P^2 > \|w\|_P^2$, which is a contradiction with the definition of g and the first part of the theorem is proved. Now we prove that p is a descent direction. Let $\alpha := \frac{\varepsilon}{\|p\|}$; then for every $t \in (0, \alpha)$, by Lebourg's mean value theorem [11, Theorem 3.3], there exist $0 < t_0 < 1$ and $\xi \in \partial f(R_x(t_0 tp))$ such that

$$f(R_x(tp)) - f(x) = \langle \xi, DR_x(t_0 tp)(tp)\rangle.$$

Using the locking condition and the isometric property of the vector transport, we have that

$$\begin{aligned} f(R_x(tp)) - f(x) &= \langle \xi, DR_x(tt_0 p)(tp)\rangle \\ &= \frac{t}{\beta_{tt_0 p}} \langle \mathcal{T}_{x \leftarrow R_x(tt_0 p)}(\xi), p\rangle. \end{aligned}$$

Since $\|tt_0 p\| \leq \varepsilon$, it follows that $\frac{1}{\beta_{tt_0 p}} \mathcal{T}_{x \leftarrow R_x(tt_0 p)}(\xi) \in \partial_\varepsilon f(x)$. Therefore, $f(R_x(tp)) - f(x) \leq t f_\varepsilon^\circ(x;p)$. ∎

2.3. A descent direction algorithm. For general nonsmooth optimization problems it may be difficult to give an explicit description of the full ε-subdifferential set. Therefore, we need an iterative procedure to approximate the ε-subdifferential at a point x. In this subsection, we assume that $0 < 2\varepsilon < \iota(x)$. We start with a subgradient of an arbitrary point nearby x and move the subgradient to the tangent space in x, and in every subsequent iteration, the subgradient of a new point nearby x is computed and moved to the tangent space in x to be added to the working set to improve the approximation of $\partial_\varepsilon f(x)$. Indeed, we do not want to provide a description of the entire ε-subdifferential set at each iteration; what we do is to approximate $\partial_\varepsilon f(x)$ by the convex hull of its elements. In this way, let P be a positive definite matrix and $W_k := \{v_1, \ldots, v_k\} \subseteq \partial_\varepsilon f(x)$; then we define

$$g_k := \underset{v\in \mathrm{conv}W_k}{\operatorname{argmin}} \|v\|_P.$$

Now we set a parameter $c \in (0,1)$ and verify the following inequality:

$$f\left(R_x\left(\frac{\varepsilon p_k}{\|p_k\|}\right)\right) - f(x) \leq \frac{-c\varepsilon\|g_k\|_P^2}{\|p_k\|}, \tag{2.7}$$

where $p_k = -Pg_k$. If (2.7) holds, then we can say $\mathrm{conv}W_k$ is an acceptable approximation for $\partial_\varepsilon f(x)$. Otherwise, using the next lemma we add a new element of $\partial_\varepsilon f(x)\backslash \mathrm{conv}W_k$ to W_k.

LEMMA 2.14. *Let* $W_k=\{v_1,\dots,v_k\}\subset\partial_\varepsilon f(x)$, $0\notin \mathrm{conv}W_k$, *and*

$$g_k = \mathrm{argmin}\{\|v\|_P : \quad v \in \mathrm{conv}W_k\}.$$

If we have

$$f\left(R_x\left(\frac{\varepsilon p_k}{\|p_k\|}\right)\right) - f(x) > \frac{-c\varepsilon\|g_k\|_P^2}{\|p_k\|},$$

where $c \in (0,1)$ *and* $p_k = -Pg_k$, *then there exist* $\theta_0 \in (0, \frac{\varepsilon}{\|p_k\|}]$ *and* $\bar{v}_{k+1} \in \partial f(R_x(\theta_0 p_k))$ *such that*

$$\langle \beta_{\theta_0 p}^{-1}\mathcal{T}_{x\leftarrow R_x(\theta_0 p)}(\bar{v}_{k+1}), p_k\rangle \geq -c\|g_k\|_P^2,$$

and $v_{k+1} := \beta_{\theta_0 p}^{-1}\mathcal{T}_{x\leftarrow R_x(\theta_0 p)}(\bar{v}_{k+1}) \notin \mathrm{conv}W_k$.

Proof. We prove this lemma using Lemma 3.1 and Proposition 3.1 in [23]. Define

$$h(t) := f(R_x(tp_k)) - f(x) + ct\|g_k\|_P^2, \ t \in \mathbb{R}, \tag{2.8}$$

and the locally Lipschitz function $\hat{f}_x : B(0_x,\varepsilon) \subset T_xM \to \mathbb{R}$ by $\hat{f}_x(g) = f(R_x(g))$; then $h(t) = \hat{f}_x(tp_k) - \hat{f}_x(0) + ct\|g_k\|_P^2$. Assume that $h(\frac{\varepsilon}{\|p_k\|}) > 0$; then by Proposition 3.1 of [23], there exists $\theta_0 \in [0, \frac{\varepsilon}{\|p_k\|}]$ such that h is increasing in a neighborhood of θ_0. Therefore, by Lemma 3.1 of [23] for every $\xi \in \partial h(\theta_0)$, one has $\xi \geq 0$. By [11, Proposition 3.1]

$$\partial h(\theta_0) \subseteq \langle \partial f(R_x(\theta_0 p_k)), DR_x(\theta_0 p_k)(p_k)\rangle + c\|g_k\|_P^2.$$

If $\bar{v}_{k+1} \in \partial f(R_x(\theta_0 p_k))$ such that

$$\langle \bar{v}_{k+1}, DR_x(\theta_0 p_k)(p_k)\rangle + c\|g_k\|_P^2 \in \partial h(\theta_0),$$

then by the locking condition

$$\langle \beta_{\theta_0 p}^{-1}\mathcal{T}_{x\leftarrow R_x(\theta_0 p)}(\bar{v}_{k+1}), p_k\rangle + c\|g_k\|_P^2 \geq 0.$$

This implies that

$$v_{k+1} := \beta_{\theta_0 p}^{-1}\mathcal{T}_{x\leftarrow R_x(\theta_0 p)}(\bar{v}_{k+1}) \notin \mathrm{conv}W_k,$$

which proves our claim. ∎

Now we present Algorithm 3 to find a vector $v_{k+1} \in \partial_\varepsilon f(x)$ which can be added to the set W_k in order to improve the approximation of $\partial_\varepsilon f(x)$. This algorithm terminates after finitely many iterations; see [9].

Then we give Algorithm 4 for finding a descent direction. Moreover, Theorem 2.15 proves that Algorithm 4 terminates after finitely many iterations.

Algorithm 3. An h-increasing point algorithm; $(v,t) = Increasing(x,p,g,a,b,P,c)$.

Require: A Riemannian manifold M, a locally Lipschitz function $f : M \to \mathbb{R}$, a retraction R from TM to M, and a vector transport $\mathcal{T}$.
Input $x \in M, g,\ p \in T_xM, a, b \in \mathbb{R}$ with $a < b$, $c, \varpi \in (0,1)$ and P a positive definite matrix such that $p = -Pg$.
Let $t \leftarrow \frac{b}{\|p\|}$, $b \leftarrow \frac{b}{\|p\|}$ and $a \leftarrow \frac{a}{\|p\|}$.
repeat
    select $v \in \partial f(R_x(tp))$ such that $\langle v, \frac{1}{\beta_{tp}}\mathcal{T}_{x\to R_x(tp)}(p)\rangle + c\|g\|_P^2 \in \partial h(t)$, where h is defined in (2.8),
    if $\langle v, \frac{1}{\beta_{tp}}\mathcal{T}_{x\to R_x(tp)}(p)\rangle + c\|g\|_P^2 < 0$ **then**
        $t = \frac{a+b}{2}$
        if $h(b) > h(t)$ **then**
            $a = t$
        else
            $b = t$
        end if
        If $b - a < \varpi$, then stop.[2]
    end if
until $\langle v, \frac{1}{\beta_{tp}}\mathcal{T}_{x\to R_x(tp)}(p)\rangle + c\|g\|_P^2 \geq 0$

Algorithm 4. A descent direction algorithm; $(g_k, p_k) = Descent(x,\delta,c,\varepsilon,P)$.

Require: A Riemannian manifold M, a locally Lipschitz function $f : M \to \mathbb{R}$, a retraction R from TM to M, the injectivity radius $\iota(M) > 0$, and a vector transport $\mathcal{T}$.
Input $x \in M$, $\delta > 0$, $c \in (0,1)$, $0 < \varepsilon < \iota(M)$ and a positive definite matrix P.
Select arbitrary $v \in \partial_\varepsilon f(x)$.
Set $W_1 = \{v\}$ and let $k = 1$.
Step 1: (Compute a descent direction)
Solve the following minimization problem and let g_k be its solution:

$$\min_{v\in \mathrm{conv} W_k} \|v\|_P.$$

if $\|g_k\|^2 \leq \delta$ **then** Stop.
else let $p_k = -Pg_k$.
end if
Step 2: (Stopping condition)
if $f\left(R_x\left(\frac{\varepsilon p_k}{\|p_k\|}\right)\right) - f(x) \leq \frac{-c\varepsilon\|g_k\|_P^2}{\|p_k\|}$, **then** Stop.
end if
Step 3: $(v,t) = Increasing(x, p_k, g_k, 0, \varepsilon, P, c)$.
Set $v_{k+1} = \beta_{tp_k}^{-1}\mathcal{T}_{x\leftarrow R_x(tp_k)}(v)$, $W_{k+1} = W_k \cup \{v_{k+1}\}$ and $k = k+1$. Go to step 1.

THEOREM 2.15. *For the point $x_1 \in M$, let the level set $N = \{x : \quad f(x) \leq f(x_1)\}$ be bounded; then for each $x \in N$, Algorithm 4 terminates after finitely many iterations.*

[2]This step is not necessary theoretically. However, it is used numerically for robustness of the algorithm.

Proof. We claim that either after a finite number of iterations the stopping condition is satisfied or for some m,

$$\|g_m\|^2 \leq \delta,$$

and the algorithm terminates. If the stopping condition is not satisfied and $\|g_k\|^2 > \delta$, then by Lemma 2.14 we find $v_{k+1} \notin \text{conv}W_k$ such that

$$\langle v_{k+1}, -p_k \rangle \leq c\|g_k\|_P^2.$$

Note that DR_x on $\text{cl}B(0_x, \varepsilon)$ is bounded by some $m_1 \geq 0$; therefore ${\beta_\eta}^{-1} \leq m_1$ for every $\eta \in \text{cl}B(0_x, \varepsilon)$. Hence by the isometry property of the vector transport and by the Lipschitzness of f of the constant L, Theorem 2.9 of [11] implies that for every $\xi \in \partial_\varepsilon f(x)$, $\|\xi\| \leq m_1 L$. Now, by definition, $g_{k+1} \in \text{conv}(\{v_{k+1}\} \cup W_k)$ has the minimum norm; therefore for all $t \in (0,1)$,

$$\begin{aligned}
\|g_{k+1}\|_P^2 &\leq \|tv_{k+1} + (1-t)g_k\|_P^2 \\
&\leq \|g_k\|_P^2 + 2t\langle Pg_k, (v_{k+1} - g_k)\rangle + t^2\|v_{k+1} - g_k\|_P^2 \\
&\leq \|g_k\|_P^2 - 2t(1-c)\|g_k\|_P^2 + 4t^2L^2m_1^2\lambda_{max}(P) \\
&\leq (1 - [(1-c)(2Lm_1)^{-1}\delta^{1/2}\lambda_{min}(P)^{1/2}\lambda_{max}(P)^{-1/2}]^2)\|g_k\|_P^2,
\end{aligned} \tag{2.9}$$

where the last inequality is obtained by assuming

$$t = (1-c)(2Lm_1)^{-2}\lambda_{max}(P)^{-1}\|g_k\|_P^2 \in (0,1),$$

$\delta^{1/2} \in (0, Lm_1)$, and $\lambda_{min}^{-1}(P)\|g_k\|_P^2 \geq \|g_k\|^2 > \delta$. Now considering

$$r = 1 - [(1-c)(2Lm_1)^{-1}\delta^{1/2}\lambda_{min}(P)^{1/2}\lambda_{max}(P)^{-1/2}]^2 \in (0,1),$$

it follows that

$$\|g_{k+1}\|_P^2 \leq r\|g_k\|_P^2 \leq \cdots \leq r^k(Lm_1)^2\lambda_{max}(P).$$

Therefore, after a finite number of iterations $\|g_{k+1}\|_P^2 \leq \delta\lambda_{min}(P)$. Using the relation between norms (2.6), we conclude that $\|g_{k+1}\|^2 \leq \delta$. ∎

2.4. Step length selection algorithms. A crucial observation is that verifying the Wolfe conditions presented in Definition 2.7 can be impractical in the case that no explicit expression for the subdifferential $\partial f(x)$ is available. Using an approximation of the Clarke subdifferential, we overcome this problem. In the last subsection, we approximated $f^\circ(x; p_k)$ by $-\|g_k\|_P^2$, where $p_k := -Pg_k$, $g_k = \text{argmin}\{\|v\|_P : v \in \text{conv}W_k\}$, and $\text{conv}W_k$ is an approximation of $\partial_\varepsilon f(x)$. Therefore, in our line search algorithm we use the approximation of $f^\circ(x; p)$ to find a suitable step length.

The task of a line search algorithm is to find a step size which decreases the objective function along the paths. The Wolfe conditions are used in the line search to enforce a sufficient decrease in the objective function and to exclude unnecessarily small step sizes. Algorithm 5 is a one-dimensional search procedure for the function $\phi(\alpha) = f(R_x(\alpha p))$ to find a step length satisfying the Armijo and curvature conditions. The procedure is a generalization of the algorithm for the well-known Wolfe conditions for smooth functions; see [25, pp. 59–60]. The algorithm has two stages. The first stage begins with a trial estimate α_1 and keeps increasing it until it finds either an acceptable step length or an interval that contains the desired step length. The parameter α_{max} is a user-supplied bound on the maximum step length allowed. The last step of Algorithm 5 performs extrapolation to find the next trial value α_{i+1}. To implement this step we can simply set α_{i+1} to some constant multiple of α_i. In the case that Algorithm 5 finds an interval $[\alpha_{i-1}, \alpha_i]$ that contains the desired step length, the second stage is invoked by Algorithm 6, called *Zoom*, which successively decreases the size of the interval.

Algorithm 5. A line search algorithm; $\alpha = Line(x, p, g, P, c_1, c_2)$.

Require: A Riemannian manifold M, a locally Lipschitz function $f : M \to \mathbb{R}$, a retraction R from TM to M, the injectivity radius $\iota(M) > 0$, and a vector transport $\mathcal{T}$.
Input $x \in M$, a descent direction p in T_xM with $p = -Pg$, where $g \in \partial_\varepsilon f(x)$ and P is a positive definite matrix and $c_1 \in (0, 1), c_2 \in (c_1, 1)$.
Set $\alpha_0 = 0$, $\alpha_{max} < \iota(M)$, $\alpha_1 = 1$ and $i = 1$.
Repeat
Evaluate $A(\alpha_i) := f(R_x(\alpha_i p)) - f(x) + c_1\alpha_i\|g\|_P^2$
if $A(\alpha_i) > 0$ **then**
    α must be obtained by $Zoom(x, p, g, P, \alpha_{i-1}, \alpha_i, c_1, c_2)$
    Stop
end if
Compute $\xi \in \partial f(R_x(\alpha_i p))$ such that $\langle \xi, \frac{1}{\beta_{\alpha_i p}}\mathcal{T}_{x \to R_x(\alpha_i p)}(p)\rangle + c_2\|g\|_P^2 \in \partial W(\alpha_i)$, where W is defined in (2.3).
if $\langle \xi, \frac{1}{\beta_{\alpha_i p}}\mathcal{T}_{x \to R_x(\alpha_i p)}(p)\rangle + c_2\|g\|_P^2 \geq 0$ **then** $\alpha = \alpha_i$
    Stop
else
    Choose $\alpha_{i+1} \in (\alpha_i, \alpha_{max})$
end if
$i = i + 1$.
End(Repeat)

Algorithm 6. $\alpha = Zoom(x, p, g, P, a, b, c_1, c_2)$.

Require: A Riemannian manifold M, a locally Lipschitz function $f : M \to \mathbb{R}$, a retraction R from TM to M, and a vector transport $\mathcal{T}$.
Input $x \in M$, a descent direction p in T_xM with $p = -Pg$, where $g \in \partial_\varepsilon f(x)$ and P is a positive definite matrix and $c_1 \in (0, 1), c_2 \in (c_1, 1)$, $a, b \in \mathbb{R}$ with $a < b$.
$i = 1$, $a_1 = a$, $b_1 = b$.
Repeat
$\alpha_i = \frac{a_i + b_i}{2}$
Evaluate $A(\alpha_i) := f(R_x(\alpha_i p)) - f(x) + c_1\alpha_i\|g\|_P^2$,
if $A(\alpha_i) > 0$ **then**
    $b_{i+1} = \alpha_i$, $a_{i+1} = a_i$.
else
    Compute $\xi \in \partial f(R_x(\alpha_i p))$ such that $\langle \xi, \frac{1}{\beta_{\alpha_i p}}\mathcal{T}_{x \to R_x(\alpha_i p)}(p)\rangle + c_2\|g\|_P^2 \in \partial W(\alpha_i)$, where W is defined in (2.3).
    if $\langle \xi, \frac{1}{\beta_{\alpha_i p}}\mathcal{T}_{x \to R_x(\alpha_i p)}(p)\rangle + c_2\|g\|_P^2 \geq 0$ **then** $\alpha = \alpha_i$
        Stop.
    else $a_{i+1} = \alpha_i$, $b_{i+1} = b_i$.
    end if
end if
$i = i + 1$.
End(Repeat)

Remark 2.16. By using Lemma 3.1 of [23], if there exists $\xi \in \partial f(R_x(\alpha_i p))$ such that $\langle \xi, DR_x(\alpha_i p)(p)\rangle + c_2\|g\|_P^2 \in \partial W(\alpha_i)$ and $\langle \xi, DR_x(\alpha_i p)(p)\rangle + c_2\|g\|_P^2 < 0$, where W is defined in (2.3), then W is decreasing on a neighborhood of α_i, which means that for every $\eta \in \partial W(\alpha_i)$, $\eta \leq 0$.

PROPOSITION 2.17. *Assume that $f : M \to \mathbb{R}$ is a locally Lipschitz function and p is the descent direction obtained by Algorithm* 4. *Then either Algorithm* 6 *terminates after finitely many iterations or it generates a sequence of intervals $[a_i, b_i]$, such that each one contains some subintervals satisfying the Wolfe conditions and a_i and b_i converge to a step length $a > 0$. Moreover, there exist $\xi_1, \xi_2, \xi_3 \in \partial f(R_x(ap))$ such that*

$$\left\langle \xi_1, \frac{1}{\beta_{ap}} \mathcal{T}_{x \to R_x(ap)}(p) \right\rangle \leq -c_2\|g\|_P^2, \left\langle \xi_2, \frac{1}{\beta_{ap}} \mathcal{T}_{x \to R_x(ap)}(p) \right\rangle \geq -c_2\|g\|_P^2,$$
$$\left\langle \xi_3, \frac{1}{\beta_{ap}} \mathcal{T}_{x \to R_x(ap)}(p) \right\rangle \geq -c_1\|g\|_P^2.$$

Proof. Suppose that the algorithm does not terminate after finitely many iterations. Since $\{a_i\}$ and $\{b_i\}$ are monotone sequences, they converge to some a and b. As we have $b_i - a_i := \frac{b_1 - a_1}{2^{i-1}}$, thus $b_i - a_i$ converges to zero. Therefore, $a = b$. We claim that $a_i > 0$ after finitely many iterations. Since p is a descent direction, then there exists $\alpha > 0$ such that $A(s) \leq 0$ for all $s \in (0, \alpha)$, where $A(s)$ is defined in Algorithm 5. Note that there exists $m > 0$ such that for every $i \geq m$, $\frac{b_1}{2^i} \leq \alpha$. If $a_{m+1} = 0$, then we must have $A(\alpha_i) > 0$ for all $i = 1, \ldots, m$. Hence, we have $b_{m+1} = \alpha_m$, $a_m = a_{m+1} = 0$, and $\alpha_{m+1} = \frac{b_{m+1}}{2} = \frac{b_1}{2^m}$. Therefore, $\alpha_{m+1} \leq \alpha$. This implies that $A(\alpha_{m+1}) \leq 0$, then $a_{m+2} = \alpha_{m+1}$. Let S be the set of all indices with $a_{i+1} = \alpha_i$. Therefore, there exists $\xi_i \in \partial f(R_x(\alpha_i p))$ such that

$$\left\langle \xi_i, \frac{1}{\beta_{\alpha_i p}} \mathcal{T}_{x \to R_x(\alpha_i p)}(p) \right\rangle + c_2\|g\|_P^2 < 0$$

for all $i \in S$. Since $\xi_i \in \partial f(R_x(\alpha_i p))$ and f is locally Lipschitz on a neighborhood of x, then by [11, Theorem 2.9] the sequence $\{\xi_i\}$ contains a convergent subsequence and without loss of generality, we can assume this sequence is convergent to some $\xi_1 \in \partial f(R_x(ap))$. Therefore,

$$\left\langle \xi_1, \frac{1}{\beta_{ap}} \mathcal{T}_{x \to R_x(ap)}(p) \right\rangle + c_2\|g\|_P^2 \leq 0.$$

Since $a_i < b_i$, $A(a_i) \leq 0$, and $A(a_i) < A(b_i)$, $A(.)$ contains a step length r_i such that $A(.)$ is increasing on its neighborhood and $A(r_i) \leq 0$. Since $c_1 < c_2$, $W(.)$ is also increasing in a neighborhood of r_i. Therefore, the Wolfe conditions are satisfied at r_i. Assume that $\langle \kappa_i, \frac{1}{\beta_{r_i p}} \mathcal{T}_{x \to R_x(r_i p)}(p)\rangle + c_2\|g\|_P^2 \in \partial W(r_i)$ for some $\kappa_i \in \partial f(R_x(r_i p))$, then $\langle \kappa_i, \frac{1}{\beta_{r_i p}} \mathcal{T}_{x \to R_x(r_i p)}(p)\rangle + c_2\|g\|_P^2 \geq 0$. Therefore, without loss of generality, we can suppose that κ_i is convergent to some $\xi_2 \in \partial f(R_x(ap))$. This implies that $\langle \xi_2, \frac{1}{\beta_{ap}} \mathcal{T}_{x \to R_x(ap)}(p)\rangle + c_2\|g\|_P^2 \geq 0$. Note that $A(.)$ is increasing on a neighborhood of r_i; therefore for all $\eta_i \in \partial f(R_x(r_i p))$ with

$$\left\langle \eta_i, \frac{1}{\beta_{r_i p}} \mathcal{T}_{x \to R_x(r_i p)}(p) \right\rangle + c_1\|g\|_P^2 \in \partial A(r_i),$$

we have $\langle \eta_i, \frac{1}{\beta_{r_i p}} \mathcal{T}_{x \to R_x(r_i p)}(p)\rangle + c_1\|g\|_P^2 \geq 0$. As before, we can say η_i is convergent to some ξ_3 in $\partial f(R_x(ap))$ and $\langle \xi_3, \frac{1}{\beta_{ap}} \mathcal{T}_{x \to R_x(ap)}(p)\rangle + c_1\|g\|_P^2 \geq 0$. ∎

In the next proposition, we prove that if Algorithm 6 does not terminate after finitely many iterations and converges to a, then the Wolfe conditions are satisfied at a.

PROPOSITION 2.18. *Assume that $f : M \to \mathbb{R}$ is a locally Lipschitz function and $p := -Pg$ is a descent direction obtained from Algorithm* 4. *If Algorithm* 6 *does not terminate after finitely many iterations and converges to a, then there exists $\xi \in \partial f(R_x(ap))$ such that*

$$\left\langle \xi, \frac{1}{\beta_{ap}} \mathcal{T}_{x \to R_x(ap)}(p) \right\rangle = -c_2 \|g\|_P^2.$$

Proof. By Proposition 2.17, there exist $\xi_1, \xi_2 \in \partial f(R_x(ap))$ such that

$$\left\langle \xi_1, \frac{1}{\beta_{ap}} \mathcal{T}_{x \to R_x(ap)}(p) \right\rangle \le -c_2 \|g\|_P^2, \left\langle \xi_2, \frac{1}{\beta_{ap}} \mathcal{T}_{x \to R_x(ap)}(p) \right\rangle \ge -c_2 \|g\|_P^2,$$

and

$$\left\langle \xi_1, \frac{1}{\beta_{ap}} \mathcal{T}_{x \to R_x(ap)}(p) \right\rangle + c_2 \|g\|_P^2, \left\langle \xi_2, \frac{1}{\beta_{ap}} \mathcal{T}_{x \to R_x(ap)}(p) \right\rangle + c_2 \|g\|_P^2 \in \partial W(a),$$

where W is defined in (2.3). Since $\partial W(a)$ is convex, $0 \in \partial W(a)$, which means there exists $\xi \in \partial f(R_x(ap))$ such that

$$\left\langle \xi, \frac{1}{\beta_{ap}} \mathcal{T}_{x \to R_x(ap)}(p) \right\rangle + c_2 \|g\|_P^2 = 0.$$

∎

In the finite precision arithmetic, if the length of the interval $[a_i, b_i]$ is too small, then two function values $f(R_x(a_i p))$ and $f(R_x(b_i p))$ are close to each other. Therefore, in practice, Algorithm 6 must be terminated after finitely many iterations; see [25]. If Algorithm 6 does not find a step length satisfying the Wolfe conditions, then we select a step length satisfying the Armijo condition.

2.5. Minimization algorithms. Finally, Algorithm 7 is the minimization algorithm for locally Lipschitz objective functions on Riemannian manifolds.

THEOREM 2.19. *If $f : M \to \mathbb{R}$ is a locally Lipschitz function on a complete Riemannian manifold M, and*

$$N = \{x : \quad f(x) \le f(x_1)\}$$

is bounded and the sequence of symmetric matrices $\{P_k^s\}$ satisfies the condition

$$0 < \lambda \le \lambda_{min}(P_k^s) \le \lambda_{max}(P_k^s) \le \Lambda < \infty, \tag{2.10}$$

for $0 < \lambda < \Lambda < \infty$ and all k, s, then either Algorithm 7 *terminates after a finite number of iterations with $\|g_k^s\| = 0$ or every accumulation point of the sequence $\{x_k\}$ belongs to the set*

$$X = \{x \in M : \quad 0 \in \partial f(x)\}.$$

Proof. If the algorithm terminates after a finite number of iterations, then x_k^s is an ε-stationary point of f. Suppose that the algorithm does not terminate after finitely many iterations. Assuming that p_k^s is a descent direction, since $\alpha \ge \frac{\varepsilon_k}{\|p_k\|}$, we have

Algorithm 7. A minimization algorithm; $x_k = Min(f, x_1, \theta_\varepsilon, \theta_\delta, \varepsilon_1, \delta_1, c_1, c_2)$.

1: **Require**: A Riemannian manifold M, a locally Lipschitz function $f : M \to \mathbb{R}$, a retraction R from TM to M, and the injectivity radius $\iota(M) > 0$.
2: **Input**: A starting point $x_1 \in M$, $c_1 \in (0,1)$, $c_2 \in (c_1, 1)$, θ_ε, $\theta_\delta \in (0,1)$, $\delta_1 > 0$, $\varepsilon_1 \in (0, \iota(M))$, $k = 1$, and $P_1 = I$.
3: Step 1 (Set new parameters) $s = 1$ and $x_k^s = x_k$, $P_k^s = P_k$.
4: Step 2. (Descent direction) $(g_k^s, p_k^s) = Descent(x_k^s, \delta_k, c_1, \varepsilon_k, P_k^s)$
5: **if** $\|g_k^s\| = 0$, **then** Stop.
6: **end if**
7: **if** $\|g_k^s\|^2 \le \delta_k$ **then** set $\varepsilon_{k+1} = \varepsilon_k\theta_\varepsilon$, $\delta_{k+1} = \delta_k\theta_\delta$, $x_{k+1} = x_k^s$, $P_{k+1} = P_k^s$, $k = k+1$. Go to step 1.
8: **else**

$$\alpha = Line(x_k^s, p_k^s, g_k^s, P_k^s, c_1, c_2)$$

and construct the next iterate $x_k^{s+1} = R_{x_k^s}(\alpha p_k^s)$ and update P_k^{s+1}. Set $s = s+1$ and go to step 2.
9: **end if**

$$f(x_k^{s+1}) - f(x_k^s) \le -\frac{c_1\varepsilon_k\|g_k^s\|^2_{P_k^s}}{\|p_k\|} < 0,$$

for $s = 1, 2, \ldots$, and therefore, $f(x_k^{s+1}) < f(x_k^s)$ for $s = 1, 2, \ldots$. Since f is Lipschitz and N is bounded, it follows that f has a minimum in N. Therefore, $f(x_k^s)$ is a bounded decreasing sequence in $\mathbb{R}$ and so is convergent. Thus $f(x_k^s) - f(x_k^{s+1})$ is convergent to zero and there exists s_k such that

$$f(x_k^s) - f(x_k^{s+1}) \le \frac{c_1\varepsilon_k\delta_k\lambda}{\|p_k\|}$$

for all $s \ge s_k$. Thus

$$\lambda\|g_k^s\|^2 \le \|g_k^s\|^2_{P_k^s} \le \left(\frac{f(x_k^s) - f(x_k^{s+1})}{c_1\varepsilon_k}\right)\|p_k\| \le \delta_k\lambda, \quad s \ge s_k. \tag{2.11}$$

Hence after finitely many iterations, there exists s_k such that

$$x_{k+1} = x_k^{s_k}.$$

Since M is a complete Riemannian manifold and $\{x_k\} \subset N$ is bounded, there exists a subsequence $\{x_{k_i}\}$ converging to a point $x^* \in M$. Since $\mathrm{conv}W_{k_i}^{s_{k_i}}$ is a subset of $\partial_{\varepsilon_{k_i}} f(x_{k_i}^{s_{k_i}})$, then

$$\|\tilde{g}_{k_i}^{s_{k_i}}\|^2_{P_{k_i}^{s_i}} := \min\left\{\|v\|^2_{P_{k_i}^{s_i}} : \ v \in \partial_{\varepsilon_{k_i}} f\left(x_{k_i}^{s_{k_i}}\right)\right\} \le \min\left\{\|v\|^2_{P_{k_i}^{s_i}} : \ v \in W_{k_i}^{s_{k_i}}\right\} \le \Lambda\delta_{k_i}.$$

Hence $\lim_{k_i\to\infty}\|g_{k_i}\| = 0$. Note that $g_{k_i} \in \partial_{\varepsilon_{k_i}} f(x_{k_i}^{s_{k_i}})$, hence $0 \in \partial f(x^*)$. ∎

3. Nonsmooth BFGS algorithms on Riemannian manifolds. In this section we discuss the nonsmooth BFGS methods on Riemannian manifolds. Let f be a smooth function defined on $\mathbb{R}^n$ and P_k be a positive definite matrix which is the approximation of the Hessian of f. We know that $p_k = -P_k^{-1}\,\mathrm{grad}\, f(x_k)$ is a descent direction. The approximation of the Hessian can be updated by the BFGS method,

when the computed step length satisfies the Wolfe conditions. Indeed we assume that $s_k = x_{k+1} - x_k$, $y_k = \operatorname{grad} f(x_{k+1}) - \operatorname{grad} f(x_k)$ and α_k satisfies the Wolfe conditions; then we have the so-called secant inequality $\langle y_k, s_k\rangle_2 > 0$. Therefore, P_k can be updated by the BFGS method as follows:

$$P_{k+1} := P_k + \frac{y_k y_k^T}{\langle s_k, y_k\rangle_2} - \frac{P_k s_k s_k^T P_k}{\langle s_k, P_k s_k\rangle_2}.$$

The structure of the smooth BFGS algorithm on Riemannian manifolds is given in several papers; see [8, 28, 29]. Note that the classical update formulas for the approximation of the Hessian have no meaning on Riemannian manifolds. First,

$$s_k := \mathcal{T}_{x_k \to R_{x_k}(\alpha_k p_k)}(\alpha_k p_k),$$
$$y_k := \frac{1}{\beta_{\alpha_k p_k}} \operatorname{grad} f(x_{k+1}) - \mathcal{T}_{x_k \to R_{x_k}(\alpha_k p_k)}(\operatorname{grad} f(x_k))$$

are vectors in the tangent space $T_{x_{k+1}}M$. The inner product on tangent spaces is then given by the chosen Riemannian metric. Furthermore, the dyadic product of a vector with the transpose of another vector, which results in a matrix in the Euclidean space, is not a naturally defined operation on a Riemannian manifold. Moreover, while in Euclidean spaces the Hessian can be expressed as a symmetric matrix, on Riemannian manifolds it can be defined as a symmetric and bilinear form. However, one can define a linear function $P_k : T_{x_k}M \to T_{x_k}M$ by

$$D^2 f(x_k)(\eta, \xi) := \langle \eta, P_k \xi\rangle, \qquad \eta, \xi \in T_{x_k}M.$$

Therefore, the approximation of the Hessian can be updated by the BFGS method as follows:

$$P_{k+1} := \tilde{P}_k + \frac{y_k y_k^\flat}{y_k^\flat s_k} - \frac{\tilde{P}_k s_k (\tilde{P}_k s_k)^\flat}{(\tilde{P}_k s_k)^\flat s_k}, \tag{3.1}$$

where $\tilde{P}_k := \mathcal{T}_{x_k \to R_{x_k}(\alpha_k p_k)} \circ P_k \circ \mathcal{T}_{x_k \leftarrow R_{x_k}(\alpha_k p_k)}$.

Now we assume that $f : M \to \mathbb{R}$ is a locally Lipschitz function and

$$g := \operatorname*{argmin}_{v \in \operatorname{conv} W_k} \|v\|_{P^{-1}}, \tag{3.2}$$

$p = -P^{-1}g$, where P is a positive definite matrix and $\operatorname{conv} W_k$ is an approximation of $\partial_\varepsilon f(x)$. The P^{-1}-norm in (3.2) approximates the Newton direction when f is smooth. Specifically, if f is twice differentiable at x_k, $\operatorname{conv} W_k = \{\operatorname{grad} f(x_k)\}$ and P is $\operatorname{Hess} f(x_k)$, then the search direction p is the Newton direction $-\operatorname{Hess} f(x_k)^{-1} \operatorname{grad} f(x_k)$. Let α be returned by Algorithm 5 and $\xi \in \partial f(R_x(\alpha p))$ be such that $\langle \xi, \frac{1}{\beta_{\alpha p}} \mathcal{T}_{x \to R_x(\alpha p)}(p)\rangle + c_2 \|g\|^2_{P^{-1}} \geq 0$. Then for all $v \in \operatorname{conv} W_k$,

$$\left\langle \xi - \beta_{\alpha p} \mathcal{T}_{x \to R_x(\alpha p)}(v), \frac{1}{\beta_{\alpha p}} \mathcal{T}_{x \to R_x(\alpha p)}(p) \right\rangle > 0.$$

This shows that if we update the approximation of the Hessian matrix by (3.1) in which $s_k := \mathcal{T}_{x_k \to R_{x_k}(\alpha_k p_k)}(\alpha_k p_k)$ and $y_k := \frac{1}{\beta_{\alpha_k p_k}} \xi_k - \mathcal{T}_{x_k \to R_{x_k}(\alpha_k p_k)}(g_k)$ are vectors provided that

$$\left\langle \xi_k, \frac{1}{\beta_{\alpha_k p_k}} \mathcal{T}_{x_k \to R_{x_k}(\alpha_k p_k)}(p_k) \right\rangle + c_2 \|g_k\|^2_{P_k^{-1}} \geq 0,$$

then the Hessian approximation P_{k+1} is symmetric positive definite.

Algorithm 8. A nonsmooth BFGS algorithm on a Riemannian manifold; $x_k = subRBFGS(f, x_1, \theta_\varepsilon, \theta_\delta, \varepsilon_1, \delta_1, c_1, c_2)$.

Require: A Riemannian manifold M, a locally Lipschitz function $f : M \to \mathbb{R}$, a retraction R from TM to M, the injectivity radius $\iota(M) > 0$, and a vector transport $\mathcal{T}$.

Input: A starting point $x_1 \in M$, $c_1 \in (0, 1)$, $c_2 \in (c_1, 1)$, θ_ε, $\theta_\delta \in (0, 1)$, $\delta_1 > 0$, $\varepsilon_1 \in (0, \iota(M))$, $k = 1$, $P_1 = I$, a bound $1/\Lambda > 0$ on $\dfrac{y_k^\flat s_k}{y_k^\flat y_k}$ and λ on $\dfrac{s_k^\flat y_k}{s_k^\flat s_k}$.

Step 1 (Set new parameters) $s = 1$, $x_k^s = x_k$ and $P_k^s = P_k$.

Step 2. (Descent direction) $(g_k^s, p_k^s) = Descent(x_k^s, \delta_k, c_1, \varepsilon_k, P_k^{s-1})$.

if $\|g_k^s\| = 0$ **then** Stop.

end if

if $\|g_k^s\|^2 \le \delta_k$, **then** set $\varepsilon_{k+1} = \varepsilon_k\theta_\varepsilon$, $\delta_{k+1} = \delta_k\theta_\delta$, $x_{k+1} = x_k^s$, $P_{k+1} = P_k^s$, $k = k + 1$. Go to step 1.

else

$$\alpha = Line(x_k^s, p_k^s, g_k^s, P_k^{s-1}, c_1, c_2)$$

and construct the next iterate $x_k^{s+1} = R_{x_k^s}(\alpha p_k^s)$ and define $s_k := \mathcal{T}_{x_k^s \to R_{x_k^s}(\alpha p_k^s)}(\alpha p_k^s)$, $y_k := \frac{1}{\beta_{\alpha p_k^s}} \xi_k - \mathcal{T}_{x_k^s \to R_{x_k^s}(\alpha p_k^s)}(g_k^s)$, $s_k := s_k + \max(0, \frac{1}{\Lambda} - \frac{s_k^\flat y_k}{y_k^\flat y_k}) y_k$.

if $\dfrac{s_k^\flat y_k}{s_k^\flat s_k} \ge \lambda$ **then**, Update

$$P_k^{s+1} := \tilde{P}_k^s + \frac{y_k y_k^\flat}{y_k^\flat s_k} - \frac{\tilde{P}_k^s s_k (\tilde{P}_k^s s_k)^\flat}{(\tilde{P}_k^s s_k)^\flat s_k}.$$

else $P_k^{s+1} := I$.

end if

Set $s = s + 1$ and go to step 2.

end if

It is worthwhile to mention that to have the global convergence of the minimization algorithm, Algorithm 7, the sequence of symmetric matrices $\{P_k^s\}$ must satisfy the condition

$$0 < \lambda \le \lambda_{min}(P_k^s) \le \lambda_{max}(P_k^s) \le \Lambda < \infty \tag{3.3}$$

for $0 < \lambda < \Lambda < \infty$ and all k, s. From a theoretical point of view it is difficult to guarantee (3.3); see [25, p. 212]. But we can translate the bounds on the spectrum of P_k^s into conditions that only involve s_k and y_k as follows:

$$\frac{s_k^\flat y_k}{s_k^\flat s_k} \ge \lambda, \qquad \frac{y_k^\flat y_k}{y_k^\flat s_k} \le \Lambda.$$

This technique is used in [25, Theorem 8.5]; see also Algorithm 1 in [34]. It is worthwhile to mention that, in practice, Algorithm 6 must be terminated after finitely many iterations. But we need to assume that even if Algorithm 6 does not find a step length satisfying the Wolfe conditions, then we can select a step length satisfying the Armijo condition and update P_k^{s+1} in Algorithm 8 by the identity matrix.

4. Experiments. In this section, we use the oriented bounding box problem [5] and the sparse vector problem [27] as applications to demonstrate the performance of Algorithm 8.

4.1. Problem statements and manifolds. The oriented bounding box problem [5] aims to find a minimum volume box containing K given points in d dimensional space. Suppose points are given by a matrix $E \in \mathbb{R}^{d\times K}$, where each column represents the coordinate of a point. A cost function of volume is given by

$$f : \mathcal{O}_d \to \mathbb{R} : O \mapsto V(OE) = \prod_{i=1}^{d} (e_{i,\max} - e_{i,\min}),$$

where $\mathcal{O}_d$ denotes the d-by-d orthogonal group, and $e_{i,\max}$ and $e_{i,\min}$ denote max and min entries, respectively, of the ith row of OE. If there exists more than one entry at any row reaching maximum or minimum values for a given O, then the cost function f is not differentiable at O. Such nondifferentiable points usually appear at minimizers; see [5]. If f is differentiable, its Riemannian gradient with respect to the Riemannian metric $\langle \eta_O, \xi_O \rangle = \operatorname{trace}(\eta_O^T \xi_O)$ is

$$\operatorname{grad} f(O) = P_O(TE^T),$$

where $T \in \mathbb{R}^{d\times K}$, $\eta_O, \xi_O \in T_O\mathcal{O}_d$, and

$$T_{ij} = \begin{cases} \frac{w}{e_{i,\max} - e_{i,\min}}, & j \text{ is the column of the largest entry in the } i\text{th row;} \\ -\frac{w}{e_{i,\max} - e_{i,\min}}, & j \text{ is the column of the smallest entry in the } i\text{th row;} \\ 0 & \text{otherwise,} \end{cases}$$

for $i = 1, \ldots, d$, $w = f(O)$, and $P_O(M) = M - O(O^T M + M^T O)/2$.

The sparse vector problem [27] finds the sparsest vector in an n-dimensional linear subspace W of $\mathbb{R}^m$. Specifically, let $Q \in \mathbb{R}^{m\times n}$ denote a matrix whose columns span the space W. The sparsest vector problem minimizes

$$\tilde{f} : \mathbb{S}^{n-1} \to \mathbb{R} : x \mapsto \|Qx\|_1,$$

where $\mathbb{S}^{n-1}$ denotes the unit sphere in $\mathbb{R}^n$. The function $\tilde{f}$ is nondifferentiable at x if and only if Qx has at least one zero entry. The Riemannian gradient at differentiable points with respect to the Riemannian metric $\langle \eta_x, \xi_x \rangle = \eta_x^T \xi_x$ is

$$\operatorname{grad} \tilde{f}(x) = (I_n - xx^T)Q^T \operatorname{sign}(Qx),$$

where sign denotes the elementwise sign function and $\eta_x, \xi_x \in T_x\mathbb{S}^{n-1}$.

For both orthogonal group $\mathcal{O}_d$ and the unit sphere $\mathbb{S}^{n-1}$, the qf retraction is used:

$$R_X(\eta_X) = \operatorname{qf}(X + \eta_X),$$

where $\operatorname{qf}(M)$ denotes the Q factor of the QR decomposition with nonnegative elements on the diagonal of R. The vector transport by parallelization [18] is isometric and essentially identity. We modify it by the approach in [17, section 4.2] and use the resulting vector transport satisfying the locking condition. To the best of our knowledge, it is unknown how large the injectivity radius for this retraction is. But in practice, the vector transport can be represented by a matrix. Therefore, we always use the inverse of the matrix as the inverse of the vector transport.

4.2. Compared methods and parameter setting. Algorithm 8 is compared with RGS (see [16, section 7.2] or [14, Algorithm 1]) and the modified Riemannian BFGS method (see [16, section 7.3]), which is a Riemannian generalization of [21].

The main difference between the RGS method, the modified Riemannian BFGS method, and Algorithm 7 is the search direction. Specifically, the search direction η_k in RGS at x_k is computed as follows: (i) randomly generate ℓ points in a small enough neighborhood of x_k; (ii) transport the gradients at those ℓ points to the tangent space at x_k; (iii) compute the shortest tangent vector in the convex hull of the resulting tangent vectors and the gradient at x_k; and (iv) set η_k to be the shortest vector. Note that the number of points, ℓ, is required to be larger than the dimension of the domain. The modified Riemannian BFGS method makes an assumption that the cost function is differentiable at all the iterates. It follows that the search direction is the same as the Riemannian BFGS method for smooth cost functions [17]. However, the stopping criterion is required to be modified for nonsmooth cost functions. Specifically, let G_k be defined as follows (j_k denotes the number of elements in G_k):

- $j_k = 1$, $G_k = \{g_k\}$ if $\|R^{-1}_{x_{k-1}}(x_k)\| > \varepsilon$ (if the x_k and x_{k-1} are not close, then reset the set G_k to be a singleton),
- $j_k = j_{k-1} + 1$, $G_k = \{g^{(k)}_{k-j_k+1}, \dots, g^{(k)}_{k-1}, g^{(k)}_k\}$ if $\|R^{-1}_{x_{k-1}}(x_k)\| \le \varepsilon$ and $j_k < J$ (if x_k and x_{k-1} are close and the number of elements in G_k is less than J, then add $g^{(k)}_k$ to G_k),
- $j_k = J$, $G_k = \{g^{(k)}_{k-J+1}, \dots, g^{(k)}_{k-1}, g^{(k)}_k\}$ if $\|R^{-1}_{x_{k-1}}(x_k)\| \le \varepsilon$ (if x_k and x_{k-1} are close and the number of elements in G_k is equal to J, then add $g^{(k)}_k$ to G_k and discard $g^{(k)}_{k-J}$),

where $g^{(j)}_i = \mathcal{T}_{x_i \to x_j}(g_i)$, $\varepsilon > 0$, and positive integer J are given parameters. The J also needs to be larger than the dimension of the domain. The modified Riemannian BFGS method stops if the shortest length vector in the convex hull of G_k is less than δ_k.

The tested algorithms stop if one of the following conditions is satisfied:

- the number of iterations reaches 5000;
- the step size is less than the machine epsilon $2.22 * 10^{-16}$;
- $\varepsilon_k \le 10^{-6}$ and $\delta_k \le 10^{-12}$.

We say that an algorithm successfully terminates if it is stopped by satisfying the last condition. Note that an unsuccessfully terminated algorithm does not imply that the last iterate must be not close to a stationary point. It may also imply that the stopping criterion is not robust.

The following parameters are used for Algorithm 8: $\varepsilon_1 = 10^{-4}$, $\delta_1 = 10^{-8}$, $\theta_\varepsilon = 10^{-2}$, $\theta_\delta = 10^{-4}$, $\lambda = 10^{-4}$, $\Lambda = 10^4$, $c_1 = 10^{-4}$ and $c_2 = 0.999$. The ε and J in the modified Riemannian BFGS method are set to be 10^{-6} and 2dim, respectively, where dim denotes the dimension of the domain, i.e., $\dim = d(d-1)/2$ for $\mathcal{O}_d$ and $\dim = n-1$ for $\mathbb{S}^{n-1}$. Multiple values of the parameter ℓ in RGS are tested. The initial iterate is given by orthonormalizing a matrix (if the domain is $\mathcal{O}_d$) or a vector (if the domain is $\mathbb{S}^{n-1}$) whose entries are drawn from the standard normal distribution. The entries in E are drawn from the uniform distribution from $[0, 1]$ and the entries of Q are drawn from the standard normal distribution.

The code is written in C++ and is available at http://www.math.fsu.edu/~whuang2/papers/LSALLFRM.htm. All experiments are performed on a 64 bit Windows platform with 3.6 GHz CPU (Intel Core i7-4790).

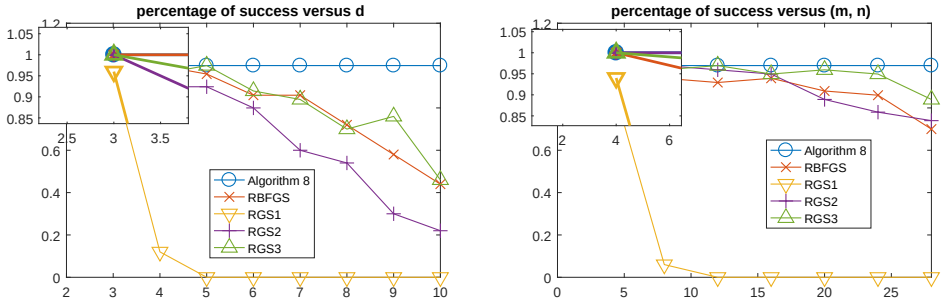

FIG. 1. *The percentage of successful runs for each algorithm versus various sizes of problems. RGS*1*, RGS*2*, and RGS*3 *denote RGS method with* $\ell = \dim + 1, 2\dim, 3\dim$*, respectively, where* $\dim = d(d-1)/2$ *for the left and* $\dim = n - 1$ *for the right.*

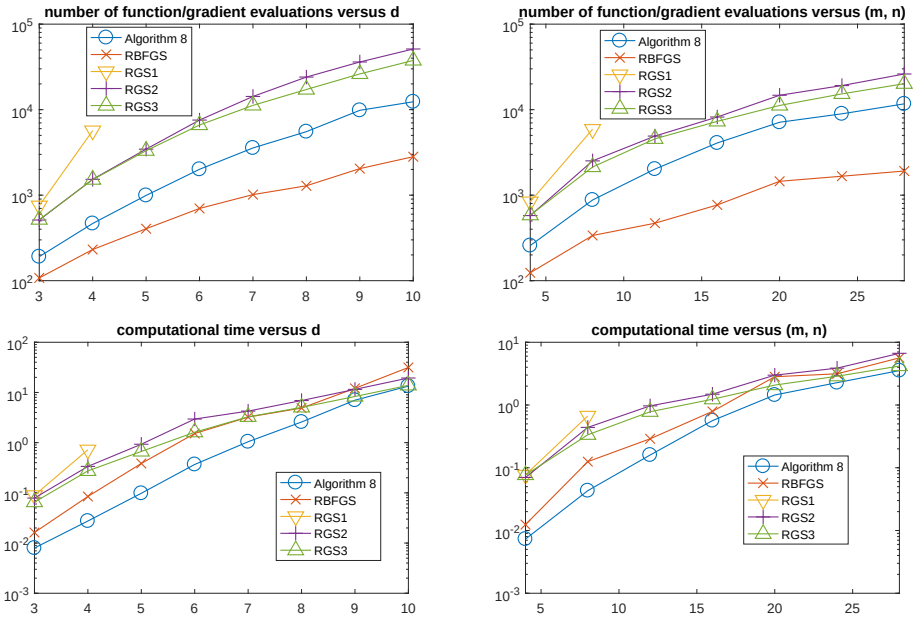

FIG. 2. *Top: an average of successful runs of the number of function evaluations versus sizes of problems. Bottom: an average of successful runs of the computational time (seconds) versus sizes of problems. RGS*1*, RGS*2*, and RGS*3 *denote RGS method with* $\ell = \dim + 1, 2\dim, 3\dim$*, respectively, where* $\dim = d(d-1)/2$ *for the left and* $\dim = n - 1$ *for the right.*

4.3. Numerical results. The three algorithms are tested with $K = 1000, d = 3, 4, \ldots 10$ for the bounding box problem and $n = 4, 8, 12, \ldots, 28, m = 10n$ for the sparse vector problem. For each setting of parameters, 50 random runs with the same 50 seeds are used. Figure 1 reports the percentage of successful runs of each algorithm. The success rate of RGS largely depends on the parameter ℓ. Specifically, the larger ℓ is, the higher the success rate is. Algorithm 8 always successfully terminates in both applications, which implies that Algorithm 8 is more robust than all the other methods.

The average number of function evaluations of successful runs and the average computational time of the successful runs are reported in Figure 2. Among the

successful tests, the RGS method is slow due to its cost in solving the quadratic programming problem and large number of gradient evaluations in each iteration. Algorithm 8 needs either the smallest or the second smallest number of function evaluations. In the bounding box problem and the sparsest vector problem, the larger the dimension of the domain is, the cheaper the function and gradient evaluations are when compared to solving the quadratic programming problem. Therefore, as shown in Figure 2, even when the number of function evaluations in Algorithm 8 is more than the modified Riemannian BFGS method, the computational time of Algorithm 8 can be smaller.

In conclusion, the experiments suggest that the proposed method, Algorithm 8, is more robust and faster than RGS and the modified Riemannian BFGS method in the sense of success rate and computational time.

Acknowledgment. We thank Pierre-Antoine Absil at Université catholique de Louvain for his helpful comments.

REFERENCES

[1] P. A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithm on Matrix Manifolds*, Princeton University Press, Princeton, NJ, 2008.

[2] R. L. Adler, J. P. Dedieu, J. Y. Margulies, M. Martens, and M. Shub, *Newton's method on Riemannian manifolds and a geometric model for the human spine*, IMA J. Numer. Anal., 22 (2002), pp. 359–390.

[3] D. Azagra, J. Ferrera, and F. López-Mesas, *Nonsmooth analysis and Hamilton-Jacobi equations on Riemannian manifolds,* J. Funct. Anal., 220 (2005), pp. 304–361.

[4] D. Azagra and J. Ferrera, *Applications of proximal calculus to fixed point theory on Riemannian manifolds*, Nonlinear. Anal., 67 (2007), pp. 154–174.

[5] P. B. Borckmans and P. A. Absil, *Oriented bounding box computation using particle swarm optimization*, in Proceedings of the 18th European Symposium on Artificial Neural Networks, 2010.

[6] G. C. Bento, O. P. Ferreira, and P. R. Oliveira, *Local convergence of the proximal point method for a special class of nonconvex functions on Hadamard manifolds,* Nonlinear Anal., (2010), pp. 564–572.

[7] G. Dirr, U. Helmke, and C. Lageman, *Nonsmooth Riemannian optimization with applications to sphere packing and grasping*, in Lagrangian and Hamiltonian Methods for Nonlinear Control 2006: Proceedings from the 3rd IFAC Workshop, Nagoya, Japan, 2006, Lecture Notes in Control and Inform. Sci., 366, Springer, New York, 2007.

[8] D. Gabay, *Minimizing a differentiable function over a differentiable manifold,* J. Optim. Theory Appl., 37 (1982), pp. 177–219.

[9] P. Grohs and S. Hosseini, *ε-subgradient algorithms for locally Lipschitz functions on Riemannian manifolds,* Adv. Comput. Math., 42 (2016), pp. 333–360.

[10] P. Grohs and S. Hosseini, *Nonsmooth trust region algorithms for locally Lipschitz functions on Riemannian manifolds,* IMA J. Numer. Anal., 36 (2016), pp. 1167–1192.

[11] S. Hosseini and M. R. Pouryayevali, *Generalized gradients and characterization of epi-Lipschitz sets in Riemannian manifolds*, Nonlinear Anal., 74 (2011), pp. 3884–3895.

[12] S. Hosseini and M. R. Pouryayevali, *Euler characterization of epi-Lipschitz subsets of Riemannian manifolds,* J. Convex. Anal., 20 (2013), pp. 67–91.

[13] S. Hosseini and M. R. Pouryayevali, *On the metric projection onto prox-regular subsets of Riemannian manifolds*, Proc. Amer. Math. Soc., 141 (2013), pp. 233–244.

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

[15] W. Huang, P.-A. Absil, and K. Gallivan, *A Riemannian BFGS Method for Nonconvex Optimization Problems*, Lect. Notes Comput. Sci. Eng. 112, Springer, New York, 2016, pp. 627–634.

[16] W. Huang, *Optimization Algorithms on Riemannian Manifolds with Applications*, Ph.D. thesis, Department of Mathematics, Florida State University, 2014.

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

[18] W. Huang, P.-A. Absil, and K. A. Gallivan, *Intrinsic representation of tangent vector and vector transport on matrix manifolds*, Numer. Math., 136 (2017), pp. 523–543, doi:10.1007/s00211-016-0848-4, 2016.
[19] S. Lang, *Fundamentals of Differential Geometry,* Grad. Texts in Math. 191, Springer, New York, 1999.
[20] P. Y. Lee, *Geometric Optimization for Computer Vision,* Ph.D. thesis, Australian National University, 2005.
[21] A. S. Lewis and M. L. Overton, *Nonsmooth optimization via quasi-Newton methods*, Math. Program., 141 (2013), pp. 135–163.
[22] C. Li, B. S. Mordukhovich, J. Wang, and J. C. Yao, *Weak sharp minima on Riemannian manifolds,* SIAM J. Optim., 21 (2011), pp. 1523–1560.
[23] N. Mahdavi-Amiri and R. Yousefpour, *An effective nonsmooth optimization algorithm for locally Lipschitz functions*, J. Optim. Theory Appl., 155 (2012), pp. 180–195.
[24] R. Mifflin, *An algorithm for constrained optimization with semismooth functions,* Math. Oper. Res., 2 (1977), pp. 191–207.
[25] J. Nocedal and S. J. Wright, *Numerical Optimization,* Springer, New York, 1999.
[26] D. Noll, *Convergence of non-smooth descent methods using the Kurdyka-Lojasiewicz inequality,* J. Optim. Theory Appl., 160 (2014), pp. 553–572.
[27] Q. Qu, J. Sun, and J. Wright, *Finding a sparse vector in a subspace: Linear sparsity using alternating directions*, IEEE Trans. Inform. Theory, 62 (2016), pp. 5855–5880.
[28] C. Qi, K. A. Gallivan, and P.-A. Absil, *Riemannian BFGS algorithm with applications*, in Recent Advances in Optimization and Its Applications in Engineering, Springer, New York, 2009, pp. 183–192.
[29] W. Ring and B. Wirth, *Optimization methods on Riemannian manifolds and their application to shape space*, SIAM J. Optim., 22 (2012), pp. 596–627.
[30] R. C. Riddell, *Minimax problems on Grassmann manifolds. Sums of eigenvalues,* Adv. Math., 54 (1984), pp. 107–199.
[31] T. Sakai, *Riemannian Geometry,* Trans. Math. Monogor. 149, AMS, Providence, RI, 1992.
[32] S. T. Smith, *Optimization techniques on Riemannian manifolds*, Fields Inst. Commun., 3 (1994), pp. 113–146.
[33] C. Udriste, *Convex Functions and Optimization Methods on Riemannian Manifolds*, Kluwer, Dordrecht, the Netherlands, 1994.
[34] J. Yu, S. V. N. Vishwanathan, S. Günter, and N. N. Schraudolph, *A quasi-Newton approach to nonsmooth convex optimization problems in machine learning*, J. Mach. Learn. Res., 11 (2010), pp. 1145–1200.
[35] R. Yousefpour, *Combination of steepest descent and BFGS methods for nonconvex nonsmooth optimization,* Numer. Algorithms., 72 (2016), pp. 57–90.