LINE SEARCH ALGORITHMS FOR LOCALLY LIPSCHITZ FUNCTIONS ON RIEMANNIAN MANIFOLDS

SOMAYEH HOSSEINI, WEN HUANG, ROHOLLAH 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.

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, 19, 28].

In the linear case, 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. 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, 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, such as, 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.

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 named 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.

Key words and phrases. Riemannian manifolds, Lipschitz functions, Descent directions, Clarke subdifferential.

AMS Subject Classifications: 49J52, 65K05, 58C05.

Somayeh Hosseini, Hausdorff Center for Mathematics and Institute for Numerical Simulation, University of Bonn, 53115 Bonn, Germany (hosseini@ins.uni-bonn.de).

Wen Huang, Department of Computational and Applied Mathematics, Rice University, Houston, USA (huwst08@gmail.com).

Rohollah Yousefpour, Department of Mathematical Sciences, University of Mazandaran, Babolsar, Iran (yousefpour@umz.ac.ir).

In smooth optimization algorithms on linear spaces for choosing the search direction p at x, we need that the angle θ, defined below, is bounded away from 90°;

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

The convergence results are obtained by the Armijo condition along with a safeguard against too small step sizes; see [24]. 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 that whether similar results are correct in nonsmooth optimization problems? In [32], 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 not only have well-chosen step lengths but also well-chosen search directions. In nonsmooth problems the angle condition (1.2) does not propose a proper set of search directions. However, an equivalent condition carried over nonsmooth problems to obtain the convergence results has been introduced in [25].

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 [25], to Riemannian manifolds. We define also a new notion called 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 norm induced by the inner product on our tangent space. If we use the identity matrix and, therefore, work with the induced norm on the tangent space, then the algorithm reduces to the descent search algorithm presented in [8]. Second, we define a nonsmooth Armijo condition on Riemannian manifolds, which is a generalization of the nonsmooth Armijo condition presented in [32] 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 [32]. 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 are inspired by some similar algorithms from [33]. 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 have 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 nonsmooth BFGS method on Riemannian manifolds, which is presented on this setting for the first time and can be considered as a generalization of the smooth BFGS on Riemannian manifolds in [14]. 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 not only new on Riemannian settings, but also on 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 are reported in Section 4.

Previous Work. For the smooth optimization on Riemannian manifolds the line search algorithms have been studied in [1, 27, 30, 31]. 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, 10, 11, 12, 21]. Papers [8, 9] 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 notations and known results of Riemannian manifolds, see, e.g. [18, 29]. 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. Also, let S be a nonempty closed subset of a Riemannian manifold M, we define $\text{dist}_S : M \longrightarrow \mathbb{R}$ by

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

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]. 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.

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. Recall that if G is a locally Lipschitz function defined from a Banach 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

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

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 $\partial f(x) = \partial \hat{f}_x(0_x)$. A 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 [8, 10]. Theorem 2.2 can be proved along the same lines as [10, Theorem 2.9].

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

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

(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.

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 $f$.
```

Once the retraction R_{x_k} is defined, the search direction p_k and the step length α_k are remained. 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. More practical strategies 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 following inequality holds for a step length α and a fixed constant $c_1 \in (0, 1)$*

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

then α satisfies in the Armijo condition.

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

2.1.1. *Sufficient decrease and backtracking.* The Armijo condition does not ensure that the algorithm makes reasonable progress. We present here a backtracking line search algorithm, which chooses its candidates appropriately to make an 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.

Algorithm 2 A backtracking line search on a Riemannian manifold.

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

2.1.2. *The Wolfe conditions.* There are other useful conditions to rule out unacceptably short step lengths. For example, one can use a second 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 need 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 [16], 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 [16, Sec. 4]. We introduce more intuitive notations:

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

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 inequality, if the following inequality holds for constant $c_2 \in (c_1, 1)$,*

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

where c_1 is the Armijo constant.

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

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

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

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

The following lemma can be proved using Lemma 3.1 of [22].

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 $p \in T_xM$, 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 $\xi \geq 0$. 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 $p \in T_xM$. 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, $R_x : T_xM \to M$ is a retraction, $p \in T_xM$ 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. Since $\phi(\alpha) = f(R_x(\alpha p))$ is bounded below for all $\alpha > 0$ and $0 < c_1 < 1$, the line $l(\alpha) = f(x) + \alpha c_1 f^\circ(x;p)$ must intersect the graph ϕ at least once. Since if we assume $l(\alpha) < \Phi(\alpha)$ for all $\alpha > 0$, then

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

which is a contradiction. It means that there exists $\tilde{\alpha} > 0$ such that $l(\tilde{\alpha}) \ge \Phi(\tilde{\alpha})$. But since $l(\alpha)$ is not bounded below and $\Phi(\alpha)$ is bounded below, then there exists $\hat{\alpha} > 0$, $l(\hat{\alpha}) = \Phi(\hat{\alpha})$. Let $\alpha_1 > 0$ be the smallest intersecting value of α, hence

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

It follows that the Armijo condition is satisfied for all step lengths less than α_1. Now by the mean value theorem, 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 [23, 32]. 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 [23]. The step length α satisfies the Mifflin condition if the following inequalities hold for the fixed constants $c_1 \in (0,1), c_2 \in (c_1, 1)$

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

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

2.2. **Descent directions.** To obtain global convergence result for a line search method, we must not only have well-chosen step lengths but also well-chosen search directions. The following definition is equivalent to gradient-orientedness carried over nonsmooth problems; see [25]. 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*

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

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;*

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

where $\eta = R_x^{-1}(y)$. 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*

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

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 induced norm induced by the inner product on our tangent space. Indeed $\|\xi\|_P = \langle P\xi, \xi\rangle^{1/2}$ and $\lambda_{min}(P)\|.\|^2 \le \|.\|_P^2 \le \lambda_{max}(P)\|.\|^2$.

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 and $0 \notin \partial_\varepsilon f(x)$,*

$$g = argmin_{\xi\in\partial_\varepsilon f(x)}\|\xi\|_P,$$

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

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

Now we claim that $\|g\|_P^2 \le \langle \xi, Pg\rangle$ for every $\xi \in \partial_\varepsilon f(x)$, which implies $\sup_{\xi\in\partial_\varepsilon f(x)}\langle \xi, -Pg\rangle \le -\|g\|_P^2$. Proof of the claim: 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

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

we can assume that t is small enough such that $\|g\|_P^2 > \|w\|_P^2$, which is a contradiction 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, there exist $0 < t_0 < 1$ and $\xi \in \partial f(R_x(t_0tp))$ such that

$$f(R_x(tp)) - f(x) = \langle \xi, DR_x(t_0tp)(tp)\rangle.$$

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

$$\begin{aligned} f(R_x(tp)) - f(x) &= \langle \xi, DR_x(tt_0p)(tp)\rangle \\ &= \frac{t}{\beta_{tt_0p}}\langle \mathcal{T}_{x\leftarrow R_x(tt_0p)}(\xi), p\rangle. \end{aligned}$$

Since $\|tt_0p\| \le \varepsilon$, it follows that $\frac{1}{\beta_{tt_0p}}\mathcal{T}_{x\leftarrow R_x(tt_0p)}(\xi) \in \partial_\varepsilon f(x)$. Therefore, $f(R_x(tp)) - f(x) \le tf_\varepsilon^\circ(x;p)$. □

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

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. 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 $W_k := \{v_1, ..., v_k\} \subseteq \partial_\varepsilon f(x)$, then we define

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

Now if we have

$$f(R_x(\frac{\varepsilon p_k}{\|p_k\|})) - f(x) \leq \frac{-c\varepsilon \|g_k\|_P^2}{\|p_k\|}, \ c \in (0,1) \tag{2.6}$$

where $p_k = -Pg_k$, 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 $\partial_\varepsilon f(x) \setminus \mathrm{conv}W_k$ to W_k.

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

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

If we have

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

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 [22]. Define

$$h(t) := f(R_x(tp_k)) - f(x) + ct\|g_k\|_P^2, \ t \in \mathbb{R}, \tag{2.7}$$

and a new locally Lipschitz function $G : B(0_x, \varepsilon) \subset T_xM \to \mathbb{R}$ by $G(g) = f(R_x(g))$, then $h(t) = G(tp_k) - G(0) + ct\|g_k\|_P^2$. Assume that $h(\frac{\varepsilon}{\|p_k\|}) > 0$, then by Proposition 3.1 of [22], 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 [22] for every $\xi \in \partial h(\theta_0)$, one has $\xi \geq 0$. By [10, 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 [8].

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

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

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

```
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 $x \in M, g,\ p \in T_xM, a, b \in \mathbb{R}, c \in (0,1)$ 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 (2.7),
    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
    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; $(g_k, p_k) = \mathit{Descent}(x, \delta, c, \varepsilon, P)$.

```
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 $x \in M, \delta, c \in (0,1)$, $0 < \varepsilon < \iota(M)$ 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:
    $\min_{v\in \text{conv}W_k} \|v\|_P.$
if $\|g_k\|^2 \leq \delta$ then Stop.
else let $p_k = -Pg_k$.
end if
Step 2: (Stopping condition)
if $f(R_x(\frac{\varepsilon p_k}{\|p_k\|})) - f(x) \leq \frac{-c\varepsilon\|g_k\|_P^2}{\|p_k\|}$, then Stop.
end if
Step 3: $(v,t) = \mathit{Increasing}(x, p_k, g_k, 0, \varepsilon, P, c)$.
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.
```

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 isometry property of the vector transport and by the Lipschitzness of f of the constant L, Theorem 2.9 of [10] 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 $t \in (0,1)$,

$$\begin{aligned} \|g_{k+1}\|_P^2 &\leq \|tv_{k+1} + (1-t)g_k\|_P^2 \\ &\leq \|g_k\|_P^2 + 2t\langle Pg_k, (v_{k+1} - g_k)\rangle + t^2\|v_{k+1} - g_k\|_P^2 \\ &\leq \|g_k\|_P^2 - 2t(1-c)\|g_k\|_P^2 + 4t^2L^2m_1^2\lambda_{max}(P) \\ &\leq (1 - [(1-c)(2Lm_1)^{-1}\delta^{1/2}\lambda_{min}(P)^{1/2}\lambda_{max}(P)^{-1/2}]^2)\|g_k\|_P^2, \end{aligned} \tag{2.8}$$

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

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

Therefore, after a finite number of iterations $\|g_{k+1}\|_P^2 \leq \delta\lambda_{min}(P)$, which proves 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 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.

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
    $\alpha$ 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)
```

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 [24, p. 59-60]. The algorithm has two stages. The first stage begins with a trial estimate α_1 and keeps it increasing 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

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 $c_1 \in (0,1), c_2 \in (c_1, 1)$, $a, b \in \mathbb{R}$.
$i = 1,\ a_1 = a,\ b_1 = b.$
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)
```

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.

Remark 2.16. By using Lemma 3.1 of [22], 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*

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

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

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 $b_i - a_i := \dfrac{b_1 - a_1}{2^{i-1}}$, 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 $i = 1, ..., m$. Hence, we have $b_{m+1} = \alpha_m$, $a_m = a_{m+1} = 0$ and $\alpha_{m+1} = \dfrac{b_{m+1}}{2} = \dfrac{b_1}{2^m}$. 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

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

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 [10, 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,

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

Since $a_i < b_i$, $A(a_i) \leq 0$ and $A(a_i) < A(b_i)$, therefore $A(.)$ contains a step length r_i such that $A(.)$ is increasing on its neighborhood and $A(r_i) \leq 0$. Since $c_1 < c_2$, therefore $W(.)$ is also increasing in a neighborhood of r_i. Therefore, the Wolfe conditions are satisfied at r_i. Assume that $\langle \kappa_i, \frac{1}{\beta_{r_ip}} \mathcal{T}_{x\to R_x(r_ip)}(p)\rangle + c_2\|g\|_P^2 \in \partial W(r_i)$ for some $\kappa_i \in \partial f(R_x(r_ip))$, then $\langle \kappa_i, \frac{1}{\beta_{r_ip}} \mathcal{T}_{x\to R_x(r_ip)}(p)\rangle + c_2\|g\|_P^2 \geq 0$. 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 $\eta_i \in \partial f(R_x(r_ip))$ with

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

we have $\langle \eta_i, \frac{1}{\beta_{r_ip}} \mathcal{T}_{x\to R_x(r_ip)}(p)\rangle + c_1\|g\|_P^2 \geq 0$. 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 Algorithms 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*

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

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

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

and

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

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

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

□

In the finite precision arithmetic, if the length of the interval $[a_i, b_i]$ is too small, then two function values $f(R_x(a_ip))$ and $f(R_x(b_ip))$ are close to each other. Therefore, in practice, Algorithm 6 must be terminated after finitely many iterations; see [24]. 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*

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

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

$$0 < \lambda \leq \lambda_{min}(P_k^s) \leq \lambda_{max}(P_k^s) \leq \Lambda < \infty, \tag{2.9}$$

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

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

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

```
Require: A Riemannian manifold $M$, a locally Lipschitz function $f : M \to \mathbb{R}$, a retraction $R$ from $TM$ to $M$ and the injectivity radius $\iota(M) > 0$.
Input: A starting point $x_1 \in M$, $c_1 \in (0,1)$, $c_2 \in (c_1, 1)$, $\theta_\varepsilon, \theta_\delta \in (0,1)$, $\delta_1 \in (0,1)$, $\varepsilon_1 \in (0, \iota(M))$, $k = 1$ and $P_1 = I$.
Step 1 (Set new parameters) $s = 1$ and $x_k^s = x_k$, $P_k^s = P_k$.
Step 2. (Descent direction) $(g_k^s, p_k^s) = Descent(x_k^s, \delta_k, c_1, \varepsilon_k, P_k^s)$
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
    $$\alpha = Line(x_k^s, p_k^s, g_k^s, P_k^s, c_1, c_2)$$
    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.
end if
```

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

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

for $s = 1, 2, ...$, therefore, $f(x_k^{s+1}) < f(x_k^s)$ for $s = 1, 2, ...$ 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 $\mathbb{R}$, 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

$$\lambda \|g_k^s\|^2 \le \|g_k^s\|^2_{P_k^s} \le \left(\frac{f(x_k^s) - f(x_k^{s+1})}{c_1 \varepsilon_k}\right) \|p_k\| \le \delta_k \lambda, \;\; s \ge s_k. \tag{2.10}$$

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 conv$W_{k_i}^{s_{k_i}}$ is a subset of $\partial_{\varepsilon_{k_i}} f(x_{k_i}^{s_{k_i}})$, then

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

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 $p_k = -P_k^{-1} \operatorname{grad} f(x_k)$ is a descent direction. The approximation of the Hessian can be updated by the BFGS method, when the computed step length satisfies in 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 smooth BFGS algorithm on Riemannian manifolds are given in several papers; see [7, 26, 27]. 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

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

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

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

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

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

$p = -P^{-1}g$, where P is a positive definite matrix and $\operatorname{conv}W_k$ is an approximation of $\partial_\varepsilon f(x)$. 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 $v \in \operatorname{conv}W_k$,

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

This shows that if we update the approximation of the Hessian matrix by the BFGS method as:

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

where $\tilde{P}_k := \mathcal{T}_{x_k \to R_{x_k}(\alpha_k p_k)} \circ P_k \circ \mathcal{T}_{x_k \leftarrow R_{x_k}(\alpha_k p_k)}$ and $s_k := \mathcal{T}_{x_k \to R_{x_k}(\alpha_k p_k)}(\alpha_k p_k)$, $y_k := \frac{1}{\beta_{\alpha_k p_k}} \xi_k - \mathcal{T}_{x_k \to R_{x_k}(\alpha_k p_k)}(g_k)$ are vectors provided that

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

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

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

$$0 < \lambda \leq \lambda_{min}(P_k^s) \leq \lambda_{max}(P_k^s) \leq \Lambda < \infty, \tag{3.1}$$

for $0 < \lambda < \Lambda < \infty$ and all k, s. From a theoretical point of view it is difficult to guarantee (3.1); see[24, page 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 follow;

$$\frac{s_k^\flat y_k}{s_k^\flat s_k} \geq \lambda, \ \frac{y_k^\flat y_k}{y_k^\flat s_k} \leq \Lambda.$$

This technique is used in [24, Theorem 8.5]; see also Algorithm 1 in [32]. 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 identity matrix.

4. Experiments

The oriented bounding box problem [5], which aims to find a minimum volume box containing n given points in d dimensional space, is used to illustrate the performance of Algorithm 8. Suppose points are given by a matrix $E \in \mathbb{R}^{d \times n}$, where each column represents the coordinate of a point. A cost function of volume is given by

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

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 $x_1 \in M$, $c_1 \in (0,1)$, $c_2 \in (c_1, 1)$, $\theta_\varepsilon$, $\theta_\delta \in (0,1)$, $\delta_1 \in (0,1)$, $\varepsilon_1 \in (0, \iota(M))$, $k = 1$, $P_1 = I$, a bound $1/\Lambda > 0$ on $\frac{y_k^\flat s_k}{y_k^\flat y_k}$ and $\lambda$ on $\frac{s_k^\flat y_k}{s_k^\flat s_k}$.
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 $\|g_k^s\|^2 \leq \delta_k$, 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 $\frac{s_k^\flat y_k}{s_k^\flat s_k} \geq \lambda$ 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
```

where $\mathcal{O}_d$ denotes the d-by-d orthogonal group, $e_{i,\max}$ and $e_{i,\min}$ denote max and min entries, respectively, of the i-th 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. Otherwise, f is differentiable and its gradient is

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

where $T \in \mathbb{R}^{d \times n}$ and

$$i\text{-th row of } T = \begin{cases} \frac{w}{e_{i,\max} - e_{i,\min}}, & \text{the column of } e_{i,\max}; \\ -\frac{w}{e_{i,\max} - e_{i,\min}}, & \text{the column of } e_{i,\min}; \\ 0, & \text{otherwise}. \end{cases}$$

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

The qf retraction is used

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

where $\mathrm{qf}(M)$ denotes the Q factor of the QR decomposition with nonnegative elements on the diagonal of R. The vector transport by parallelization [17] is isometric and essentially identity. We modify it by the approach in [16, 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. 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.

Algorithm 8 is compared with the Riemannian gradient sampling (see [15, Section 7.2] or [13, Algorithm 1]) and the modified Riemannian BFGS method (see [15, Section 7.3]), which is a Riemannian generalization of [20].

The main difference between the Riemannian gradient sampling (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 m points in a small enough neighborhood of x_k; ii) transport the gradients at those m 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, m, 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 [16]. However, the stopping criterion is required to be modified for non-smooth cost functions. Specifically, let G_k be defined as follows:

- $j_k = 1$, $G_k = \{g_k\}$ if $\|R_{x_{k-1}}^{-1}(x_k)\| > \varepsilon$;
- $j_k = j_{k-1} + 1$, $G_k = \{g_{k-j_k+1}^{(k)}, \dots, g_{k-1}^{(k)}, g_k^{(k)}\}$ if $\|R_{x_{k-1}}^{-1}(x_k)\| \leq \varepsilon$;
- $j_k = J$, $G_k = \{g_{k-J+1}^{(k)}, \dots, g_{k-1}^{(k)}, g_k^{(k)}\}$ if $\|R_{x_{k-1}}^{-1}(x_k)\| \leq \varepsilon$;

where $g_i^{(j)} = \mathcal{T}_{x_i \to x_j}(g_i)$, $\varepsilon > 0$ 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 $2.22 * 10^{-16}$;
- $\varepsilon_k \leq 10^{-6}$ and $\delta_k \leq 10^{-12}$.

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 $dim = d(d-1)/2$ is the dimension of the domain. Multiple values of the parameter m in RGS are tested and given in the caption of Figures 1 and 2. Initial iterate is given by orthonormalizing a matrix whose entries are drawn from a 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 Ubuntu platform with 3.6 GHz CPU (Intel Core i7-4790).

The three algorithms are tested with $d = 3, 4, \dots 10$. For each value of d, we use 100 random runs. Note that the three algorithms use 100 same random seeds. Figure 1 reports the percentage of successful runs of each algorithm. The success rate of RGS largely depends on the parameter m. Specifically, the larger m is, the higher the success rate is. Algorithm 8 always successfully terminates, which means that Algorithm 8 is more robust than all the other methods.

The average number of function and gradient evaluations of successful runs and the average computational time of the successful runs are reported in Figure 2. Among the successful tests, the modified Riemannian BFGS method needs the least number of function and gradient evaluations due to its simple approach while Algorithm 8 needs the second least. For the bounding box 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, i.e., finding the shortest length vector in a convex hull of a set of vectors. Therefore, as shown in Figure 2, even though the number of function and gradient evaluations are different for big d, the computational time is not significantly different. However, Algorithm 8 is still always slightly faster than all the other methods for all the values of d.

In conclusion, the experiments show 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.

5. Acknowledgements

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

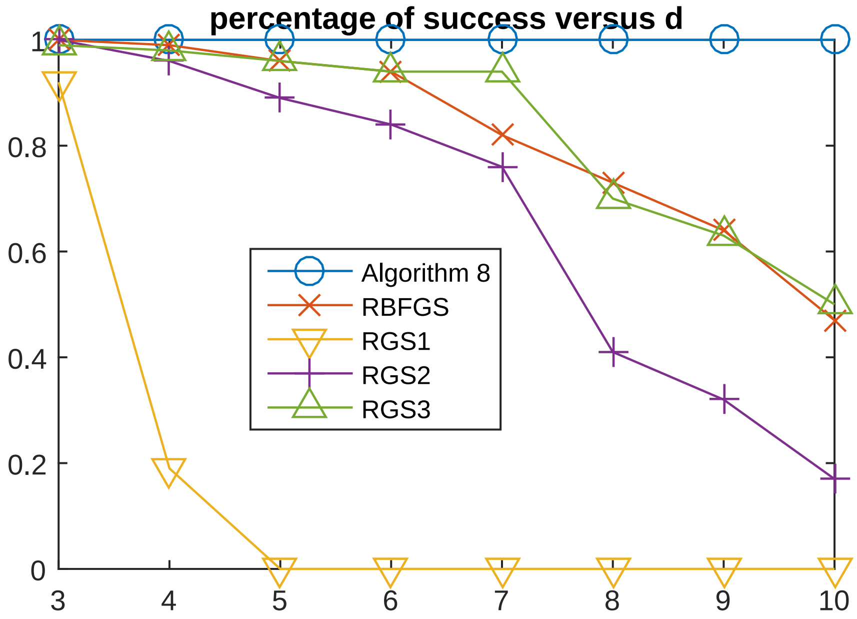

FIGURE 1. The percentage of successfully runs for each algorithms versus d. RGS1, RGS2, and RGS3 denote RGS method with $m = dim + 1, 2dim, 3dim$, respectively, where $dim = d(d-1)/2$.

REFERENCES

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

[2] R. L. Adler, J. P. Dedieu, J. Y. Margulies, M. Martens, 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, F. López-Mesas, *Nonsmooth analysis and Hamilton-Jacobi equations on Riemannian manifolds,* J. Funct. Anal., 220 (2005), pp. 304-361.

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

[5] P. B. Borckmans, P. A. Absil, *Fast oriented bounding box computation using particle swarm optimization*, In Proceedings of the 18th European Symposium on Artificial Neural Network(ESANN), 2010.

[6] G. Dirr, U. Helmke, 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 Information Sciences, Vol. 366, Springer Verlag, Berlin, 2007.

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

[8] P. Grohs, S. Hosseini, *ε-subgradient algorithms for locally Lipschitz functions on Riemannian manifolds,* Adv. Comput. Math., 42(2)(2016), pp. 333-360.

[9] P. Grohs, S. Hosseini, *Nonsmooth trust region algorithms for locally Lipschitz functions on Riemannian manifolds,* IMA J. Numer. Anal., 36(3)(2016), pp. 1167-1192.

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

[11] S. Hosseini, M. R. Pouryayevali, *Euler characterization of epi-Lipschitz subsets of Riemannian manifolds,* J. Convex. Anal., 20 (2013), No. 1, pp. 67-91.

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

[13] S. Hosseini, A. Uschmajew, *A Riemannian gradient sampling algorithm for nonsmooth optimization on manifolds*, Institut für Numerische Simulation, INS Preprint No. 1607, 2016.

[14] W. Huang, P.-A. Absil, K. Gallivan, *A Riemannian BFGS Method for Nonconvex Optimization Problems*, Lecture Notes in Computational Science and Engineering, to appear, 2016.

[15] W. Huang, *Optimization algorithms on Riemannian manifolds with applications*, Ph.D thesis, Florida State University, Department of Mathematics, 2014.

[16] W. Huang, K. A. Gallivan, and P.-A. Absil, *A Broyden class of quasi-Newton methods for Riemannian optimization,* SIAM J. Optim., 25(3)(2015), pp. 1660-1685.

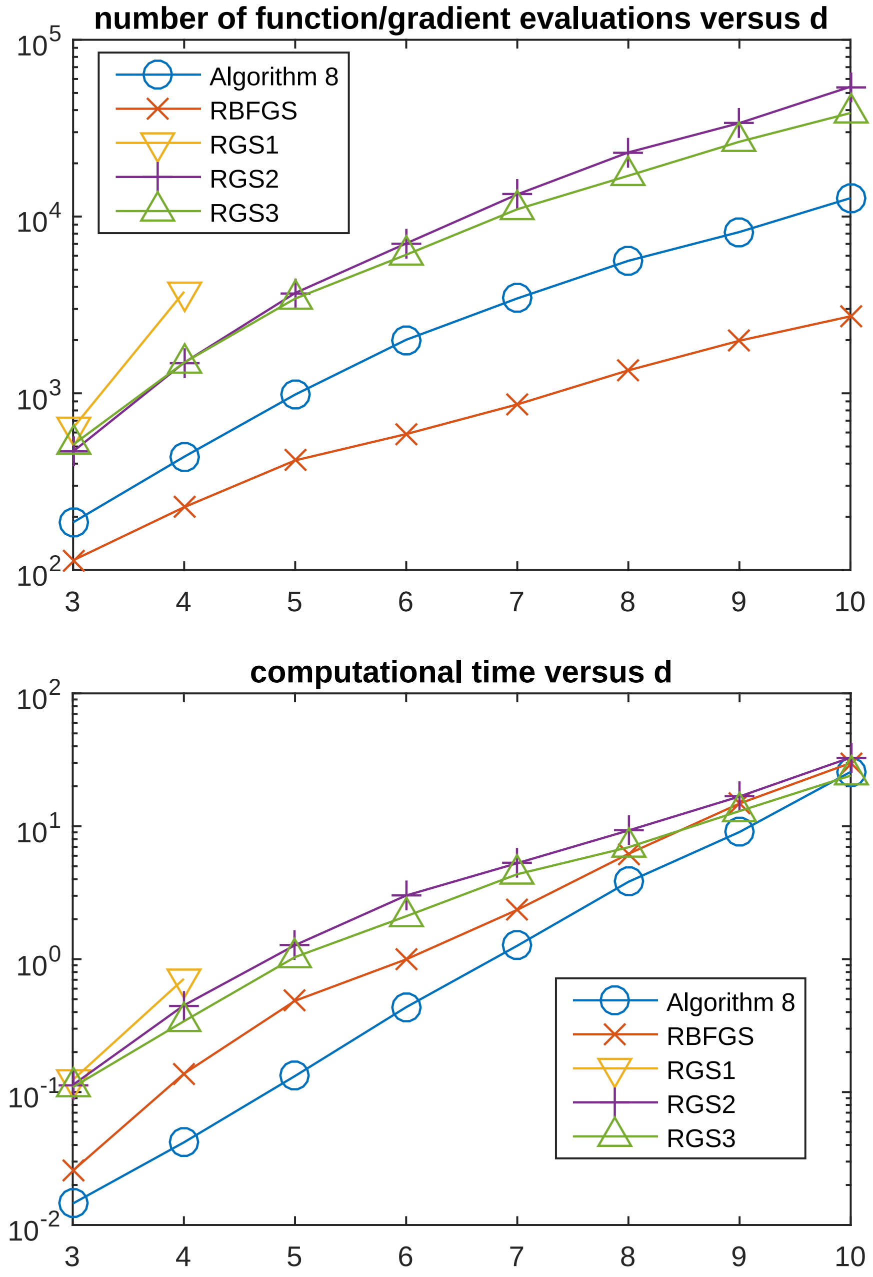

FIGURE 2. Top: an average of 100 runs of the number of function evaluations versus d. Bottom: an average of 100 runs of the computational time (second) versus d. RGS1, RGS2, and RGS3 denote RGS method with $m = dim + 1, 2dim, 3dim$, respectively, where $dim = d(d-1)/2$.

[17] W. Huang, P.-A. Absil, K. A. Gallivan, *Intrinsic Representation of Tangent Vector and Vector Transport on Matrix Manifolds*, Numerische Mathematik, DOI:0.1007/s00211-016-0848-4, 2016.

[18] S. Lang, *Fundamentals of Differential Geometry,* Graduate Texts in Mathematics, Vol. 191, Springer, New York, 1999.

[19] P. Y. Lee, *Geometric Optimization for Computer Vision,* PhD thesis, Australian National University, 2005.

[20] A. S. Lewis, M. L. Overton, *Nonsmooth optimization via quasi-Newton methods*, Mathematical Programming, 141(1-2), (2013), pp. 135-163.

[21] C. Li, B. S. Mordukhovich, J. Wang, J. C. Yao, *Weak sharp minima on Riemannian manifolds,* SIAM J. Optim., 21(4) (2011), pp. 1523-1560.

[22] N. Mahdavi-Amiri, R. Yousefpour, *An effective nonsmooth optimization algorithm for locally Lipschitz functions*, J. Optim. Theory Appl., 155 (2012), pp. 180-195.

[23] R. Mifflin, *An algorithm for constrained optimization with semismooth functions,* Math. Oper. Res., 2 (1977), pp. 191-207.

[24] J. Nocedal, S. J. Wright,*Numerical Optimization,* Springer, 1999.

[25] D. Noll, *Convergence of non-smooth descent methods using the Kurdyka-Lojasiewicz Inequality.,* J. Optim. Theory Appl., 160(2014), pp. 553 -572.

[26] C. Qi, K. A. Gallivan, P.-A. Absil , *Riemannian BFGS algorithm with applications*, Recent advances in Optimization and its Applications in Engineering, Springer, 2009.

[27] W. Ring, B. Wirth, *Optimization methods on Riemannian manifolds and their application to shape space*, SIAM J. Optim., 22(2) (2012), pp. 596-627.

[28] R. C. Riddell, *Minimax problems on Grassmann manifolds. Sums of eigenvalues,* Adv. Math., 54 (1984), pp. 107-199.

[29] T. Sakai, *Riemannian Geometry,* Trans. Math. Monogor. Vol. 149, Amer. Math. Soc. 1992.

[30] S. T. Smith, *Optimization techniques on Riemannian manifolds*, Fields Institute Communications, 3 (1994), pp. 113-146.

[31] C. Udriste, *Convex Functions and Optimization Methods on Riemannian Manifolds*, Kluwer Academic Publishers, Dordrecht, Netherlands, 1994.

[32] J. Yu, S. V. N. Vishwanathan, S. Günter, N. N. Schraudolph, *A quasi-Newton approach to nonsmooth convex optimization problems in machine learning*, J. Mach. Learn. Res., 11 (2010), pp. 1145-1200.

[33] R. Yousefpour, *Combination of steepest descent and BFGS methods for nonconvex nonsmooth optimization,* Numer. Algorithms., 72 (2016), pp. 57-90.