Journal of Optimization Theory and Applications
https://doi.org/10.1007/s10957-023-02268-3

Retraction-Based Direct Search Methods for Derivative Free Riemannian Optimization

Vyacheslav Kungurtsev[1] · **Francesco Rinaldi**[2] · **Damiano Zeffiro**[2]

Received: 7 November 2022 / Accepted: 27 June 2023

Abstract
Direct search methods represent a robust and reliable class of algorithms for solving black-box optimization problems. In this paper, the application of those strategies is exported to Riemannian optimization, wherein minimization is to be performed with respect to variables restricted to lie on a manifold. More specifically, classic and linesearch extrapolated variants of direct search are considered, and tailored strategies are devised for the minimization of both smooth and nonsmooth functions, by making use of retractions. A class of direct search algorithms for minimizing nonsmooth objectives on a Riemannian manifold without having access to (sub)derivatives is analyzed for the first time in the literature. Along with convergence guarantees, a set of numerical performance illustrations on a standard set of problems is provided.

Keywords Direct search · Derivative free optimization · Riemannian manifold · Retraction

Mathematics Subject Classification 90C06 · 90C30 · 90C56

Communicated by Sonia Cafieri.

Vyacheslav Kungurtsev: Research supported by the OP VVV project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics".

✉ Damiano Zeffiro
damiano.zeffiro@math.unipd.it

Vyacheslav Kungurtsev
kunguvya@fel.cvut.cz

Francesco Rinaldi
rinaldi@math.unipd.it

[1] Department of Computer Science, Czech Technical University, Prague, Czech Republic

[2] Dipartimento di Matematica "Tullio Levi-Civita", Università di Padova, Padua, Italy

Published online: 30 July 2023

1 Introduction

Riemannian optimization, or solving minimization problems with decision variables constrained to lie on a Riemannian manifold, is an important and active area of research since there are numerous problems in data science, robotics, and other settings wherein there is a geometric structure characterizing the allowable inputs. Derivative free optimization (DFO), or zeroth-order optimization, involves algorithms that only make use of function evaluations rather than any gradient computations in their implementation, designed for applications where accurate approximations of the gradient are unavailable due to noise or high computational cost. This paper specializes existing direct search DFO algorithms to Riemannian optimization problems. Reference of Riemannian optimization and DFO includes [1, 6, 11, 22], respectively.

Direct search methods (see, e.g., [21] and references therein) belong to the class of derivative free algorithms that do not build models of the objective or gradient approximations. Thus, they are particularly suitable for problems with function evaluations considered as a black box with little prior information that could suggest how accurate different interpolation models would be, as both differentiability and conditioning properties of the function are unknown.

To the best of our knowledge, thorough studies of derivative free optimization (DFO) on Riemannian manifolds have only been carried out recently in the literature. The closest would be direct search confined to a subspace, presented in [4]. In [23], the authors focus on a model-based method using a two-point function approximation for the gradient. The paper [31] presents a specialized Polak–Ribière–Polyak procedure for finding a zero of a tangent vector field on a Riemannian manifold. In [13], it is claimed that the convergence analysis of mesh adaptive direct search methods (MADS; see, e.g., [5, 6]) for unconstrained objectives can be extended to the case of Riemannian manifolds using the exponential map. In the subsequent work [14], the author focuses on a specific class of manifolds (reductive homogeneous spaces, including several matrix manifolds), discussing more in detail how, thanks to the properties of exponential maps, a straightforward extension of MADS is possible at least for that class. Some nonsmooth problems on Riemannian manifolds and references to derivative free optimization methods without convergence analyses can be found in [18].

Thus, our paper presents the first analysis of retraction-based direct search strategies on Riemannian manifolds, and the first analysis of a direct search algorithm for minimizing nonsmooth objectives in Riemannian optimization. In particular, a classic direct search scheme (see, e.g., [11, 21]) and a linesearch-based scheme (see, e.g., [12, 24–26] for further details on this class of methods) to deal with the minimization of a given smooth function over a manifold are adapted from analogous methods in the unconstrained settings. Then, inspired by the ideas in [15], the two proposed strategies are extended to the nonsmooth case. The introduction of the geometric constraint presents significant challenges: Namely, the stable structure of the Euclidean vector space makes it natural for a fixed set of coordinate-like directions to consistently approximate desired directions by spanning the space in a uniform way. The fact that this geometric structure can change necessitates a careful adjustment of the poll directions corresponding to the change in this structure, with minimal computational

expense. The associated convergence theory presents some novel results that could be of independent interest.

The remainder of this paper is as follows. In Sect. 2, some definitions are presented. In Sect. 3, a direct search method applicable for continuously differentiable f is presented, with a convergence proof. In Sect. 4, the case of f not being continuously differentiable but rather only Lipschitz continuous is considered. Some numerical results are presented in Sect. 5. Detailed proofs can be found in Appendix.

The codes relevant to the numerical tests are available at the following link: https://github.com/DamianoZeffiro/riemannian-ds.

2 Definitions and Notation

This section introduces some notation for the formalism used in this article. The reader is referred to, e.g., [1, 8] for an overview of the relevant background.

Let $\mathcal{M}$ be a smooth finite dimensional connected manifold.

The problem of interest here is

$$\min_{x\in\mathcal{M}} f(x), \tag{1}$$

with f being continuous and bounded below. Both the case of $f(x)$ being continuously differentiable and a more general nonsmooth case are considered. For $x \in \mathcal{M}$, let $T_x\mathcal{M}$ be the tangent vector space at x and $T\mathcal{M}$ be the tangent bundle $\cup_{x\in\mathcal{M}} T_x\mathcal{M}$. $\mathcal{M}$ is assumed to be a Riemannian manifold, so that for x in $\mathcal{M}$, there is a scalar product $\langle\cdot,\cdot\rangle_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$ and a norm $\|\cdot\|_x$ on $T_x\mathcal{M}$ smoothly depending on x. Let $\mathrm{dist}(\cdot,\cdot)$ be the distance induced by the scalar product, so that for $x, y \in \mathcal{M}$ the distance $\mathrm{dist}(x, y)$ is the length of the shortest geodesic connecting x and y. Furthermore, let $\nabla_{\mathcal{M}}$ be the Levi-Civita connection for $\mathcal{M}$ (see [8, Theorem 5.5] for a precise definition), and $\Gamma : T\mathcal{M} \times \mathcal{M} \to T\mathcal{M}$ be a parallel transport with respect to $\nabla_{\mathcal{M}}$, with $\Gamma_x^y(v) \in T_y\mathcal{M}$ transport of the vector $v \in T_x\mathcal{M}$ to one in $T_y\mathcal{M}$ along a fixed curve connecting x and y. The parallel transport Γ is assumed to operate always along a distance minimizing geodesic when it exists. Consequently, for any $x \in \mathcal{M}$ there is a neighborhood U of x such that the parallel transport $\Gamma_y^z(v)$ is well defined and depends smoothly on y, z, v for $y, z \in U$ and $v \in T_y\mathcal{M}$. Any nonuniqueness in the definition of Γ is either explicitly accounted for or inconsequential without loss of generality in the context.

When $\mathcal{M}$ is embedded in $\mathbb{R}^n$, P_x is defined as the orthogonal projection from $\mathbb{R}^n$ to $T_x\mathcal{M}$, and $S(x, r) \subset \mathbb{R}^n$ as the sphere centered at x and with radius r.

$\{a_k\}$ is used as a shorthand for $\{a_k\}_{k\in I}$ when the index set I is clear from the context. The shorthand notations $T_k\mathcal{M}$, P_k, $\langle\cdot,\cdot\rangle_k$, $\|\cdot\|_k$, Γ_i^j are also employed, in place of $T_{x_k}\mathcal{M}$, P_{x_k}, $\langle\cdot,\cdot\rangle_{x_k}$, $\|\cdot\|_{x_k}$ and $\Gamma_{x_i}^{x_j}$. For x_0 given point in $\mathcal{M}$ serving as initialization of the algorithms presented in this manuscript, the sublevel set relative to $f(x_0)$ is denoted as $\mathcal{L}_0 = \{x \in \mathcal{M} \mid f(x) \leq f(x_0)\}$. When there is no ambiguity on the value of x, $\|\cdot\|$ is used instead of $\|\cdot\|_x$.

The distance dist* is defined between vectors in different tangent spaces in a standard way using parallel transport (see, for instance, [7]): for $x, y \in \mathcal{M}$, $v \in T_x\mathcal{M}$ and $w \in T_y M$,

$$\operatorname{dist}^*(v, w) = \left\| v - \Gamma_y^x w \right\| = \left\| w - \Gamma_x^y v \right\|, \tag{2}$$

and for a sequence $\{(y_k, v_k)\}$ in $T\mathcal{M}$ the notation $v_k \to v$ means $y_k \to y$ in $\mathcal{M}$ and $\operatorname{dist}^*(v_k, v) \to 0$. On compact subsets of $\mathcal{M}$, for $\operatorname{dist}(x, y)$ small enough the minimizing geodesic between x and y is uniquely defined and consequently the parallel transport Γ and the distance dist* also are. As it is common in the Riemannian optimization literature (see, e.g., [2]), to define our tentative descent directions a retraction $R : T\mathcal{M} \to \mathcal{M}$ is used. This retraction R is assumed to be in $C^1(T\mathcal{M}, \mathcal{M})$, with

$$\operatorname{dist}(R(x, d), x) \leq L_r \|d\|, \tag{3}$$

(true in any compact subset of $T\mathcal{M}$ given the C^1 regularity of R, without any further assumptions).

For a scalar-valued function $f : \mathcal{M} \to \mathbb{R}$, the gradient $\operatorname{grad} f(x)$ is defined as the unique element of $T_x\mathcal{M}$ such that for all $v \in \mathcal{M}$, it holds that

$$Df(x)[v] = \langle v, \operatorname{grad} f(x) \rangle_x.$$

When $\mathcal{M}$ is embedded in $\mathbb{R}^n$, the (Riemannian) gradient is a simple projection onto $T_x\mathcal{M}$, i.e., $\operatorname{grad} f(x) = \mathsf{P}_x(\nabla f(x))$.

3 Smooth Optimization Problems

In this section, methods for the solution of problem (1) with the objective $f \in C^1(\mathcal{M})$ are considered. In particular, the gradient $\operatorname{grad} f(x)$ is assumed to be continuous along $\mathcal{M}$ as a function of x.

3.1 Preliminaries

A Lipschitz continuous gradient assumption is first presented.

Assumption 1 There exists $L_f > 0$ such that for all $x, y \in \mathcal{M}$

$$\operatorname{dist}^*(\operatorname{grad} f(x), \operatorname{grad} f(y)) = \left\| \Gamma_x^y \operatorname{grad} f(x) - \operatorname{grad} f(y) \right\| \leq L_f \operatorname{dist}(x, y). \tag{4}$$

The next assumption generalizes the standard descent property.

Assumption 2 There exists $L > 0$ so that for every $x \in \mathcal{M} \cap \mathcal{L}_0$, $d \in T_x\mathcal{M}$

$$f(R(x, d)) \leq f(x) + \langle \operatorname{grad} f(x), d \rangle + \frac{L}{2} \|d\|^2. \tag{5}$$

Under suitable assumptions, the Lipschitz gradient property implies the generalized standard descent property.

Proposition 3.1 *Assume that $\mathcal{L}_0$ is compact, f is Lipschitz continuous and that R is a C^2 retraction. Then, Assumption 1 implies Assumption 2.*

The proof can be found in Appendix. It should be noted that Proposition 3.1 is a key tool to extend convergence properties from the unconstrained case to the Riemannian case. To the best of our knowledge, this result is new to the literature. Under the stronger assumption that f has Lipschitz gradient as a function in $\mathbb{R}^n$, the standard descent property (5) was proven for retractions in [9].

For each algorithm in this section, it is further assumed that, at each iteration k, a positive spanning set (as defined, e.g., in [11]) $\{p_k^j\}_{j\in[1:K]}$ is available for the tangent space $T_k M$. This positive spanning set is assumed to stay bounded and not become degenerate during the algorithm, that is,

Assumption 3 There exists $B > 0$ such that

$$\max_{j\in[1:K]} \left\| p_k^j \right\| \leq B, \tag{6}$$

for every $k \in \mathbb{N}$. Furthermore, there is a constant $\tau > 0$ such that

$$\max_{i\in[1:K]} \langle r, p_k^j \rangle \geq \tau \left\| r \right\|, \tag{7}$$

for every $k \in \mathbb{N}$ and $r \in T_{x_k} M$.

3.2 Direct Search Algorithm

Here, the Riemannian Direct Search method based on Spanning Bases (RDS-SB) for smooth objectives is presented as Algorithm 1.

This procedure resembles the standard direct search algorithm for unconstrained derivative free optimization (see, e.g., [11, 21]) with two significant modifications. First, at every iteration a positive spanning set is computed for the current tangent vector space $T_k\mathcal{M}$. As this space is expected to change at every iteration, it is not possible to use the same standard positive spanning sets appearing in the classic algorithms. Second, the candidate point x_k^j is computed by retracting the step $\alpha_k p_k^j$ from the current tangent space $T_k\mathcal{M}$ to the manifold, ensuring satisfaction of the geometric constraint.

3.3 Convergence Analysis

In this section, asymptotic global convergence of the method is shown. First it is proved that the gradient, in unsuccessful iterates, must be bounded by a constant proportional to the stepsize (Lemma 3.2). This is a well-known condition in the unconstrained case (see, e.g., [30, Theorem 1]), extended to the Riemannian case thanks to Proposition 3.1.

Algorithm 1 RDS-SB

Input: $x_0 \in \mathcal{M}$, $\gamma_1 \in (0,1)$, $\gamma_2 \geq 1$, $\alpha_0 > 0$, $\gamma > 0$
for $k = 0, 1, \ldots$ **do**
  Compute a positive spanning set $\{p_k^j\}_{j=1:K}$ of $T_k\mathcal{M}$
  for $j = 1, \ldots, K$ **do**
    Let $x_k^j = R(x_k, \alpha_k p_k^j)$
    if $f(x_k^j) \leq f(x_k) - \gamma\alpha_k^2$ **then**
      $\alpha_{k+1} = \gamma_2\alpha_k$, $x_{k+1} = x_k^j$
      Declare the step k successful
      Break
    end if
  end for
  if $f(x_k^j) > f(x_k) - \gamma\alpha_k^2$ for $j \in [1:K]$ **then**
    $\alpha_{k+1} = \gamma_1\alpha_k$, $x_{k+1} = x_k$
    Declare the step k unsuccessful
  end if
end for

Given that the stepsize converges to zero, the bound implies that the gradient converges to zero for unsuccessful steps. It is then proved, using the Lipschitz continuity of the gradient, that the gradient converges to zero for successful steps as well.

The first lemma states a bound on the scalar product between the gradient and the descent direction for an unsuccessful iteration.

Lemma 3.1 *Let $f \in C^1(\Omega)$, $\{x_k\}$ generated by Algorithm 1, and let Assumptions 2, 3 hold.*
If $f(R(x_k, \alpha_k p_k^j)) > f(x_k) - \gamma\alpha_k^2$, then

$$\alpha_k(LB^2/2 + \gamma) > -\langle grad f(x_k), p_k^j \rangle. \tag{8}$$

Proof To start with, we have

$$\begin{aligned} f(x_k) - \gamma\alpha_k^2 < f(R(x, \alpha_k p_k^j)) &\leq f(x_k) + \alpha_k \langle \mathrm{grad} f(x_k), p_k^j \rangle + \frac{L}{2}\alpha_k^2 \left\| p_k^j \right\|^2 \\ &\leq f(x_k) + \alpha_k \langle \mathrm{grad} f(x_k), p_k^j \rangle + \frac{L}{2}\alpha_k^2 B^2, \end{aligned} \tag{9}$$

where we used (5) in the second inequality, and (6) in the third one. The above inequality can be rewritten as

$$\alpha_k \langle \mathrm{grad} f(x_k), p_k^j \rangle + \alpha_k^2(LB^2/2 + \gamma) > 0. \tag{10}$$

Given that $\alpha_k > 0$, the above is true if and only if

$$\alpha_k > -\frac{\langle \mathrm{grad} f(x_k), p_k^j \rangle}{(LB^2/2 + \gamma)}, \tag{11}$$

which rearranged gives the thesis. □

From this, a bound on the gradient with respect to the stepsize is inferred.

Lemma 3.2 *Let $f \in C^1(\Omega)$, $\{x_k\}$ generated by Algorithm 1, and let Assumptions 2, 3 hold. If iteration k is unsuccessful, then*

$$\|grad f(x_k)\| \leq \frac{\alpha_k(LB^2/2+\gamma)}{\tau}. \tag{12}$$

Proof If iteration k is unsuccessful, Eq. (8) must hold for every $j \in [1:K]$. We obtain the thesis by applying the positive spanning property (7) in the RHS:

$$\alpha_k(LB^2/2+\gamma) > \max_{j\in[1:K]} -\langle \text{grad} f(x_k), p_k^j\rangle \geq \tau \|\text{grad} f(x_k)\|. \tag{13}$$

□

Finally, convergence of the gradient norm to zero is shown using the lemmas above and appropriate arguments regarding the stepsizes.

Theorem 3.1 *Let $f \in C^1(\Omega)$, $\{x_k\}$ generated by Algorithm 1, and let Assumptions 1, 2, 3 hold. For the sequence $\{x_k\}$ generated by Algorithm 1*

$$\lim_{k\to\infty} \|grad f(x_k)\| = 0. \tag{14}$$

Proof To start with, it holds that $\alpha_k \to 0$ since the objective is bounded below, $\{f(x_k)\}$ is nonincreasing with $f(x_{k+1}) \leq f(x_k) - \gamma\alpha_k^2$ if the step k is successful, and so there can be a finite number of successful steps with $\alpha_k \geq \varepsilon$ for any $\varepsilon > 0$.

For a fixed $\varepsilon > 0$, let $\bar{k}$ such that $\alpha_k \leq \varepsilon$ for every $k \geq \bar{k}$. We now show that, for every $\varepsilon > 0$ and $k \geq \bar{k}$ large enough, we have

$$\|\text{grad} f(x_k)\| \leq \varepsilon\left(\frac{(LB^2/2+\gamma)}{\tau} + L_f L_r B\frac{\gamma_2}{\gamma_2-1}\right), \tag{15}$$

which implies the thesis given that ε is arbitrary.

First, Eq. (15) is satisfied for $k \geq \bar{k}$ if the step k is unsuccessful by Lemma 3.2:

$$\|\text{grad} f(x_k)\| \leq \frac{\alpha_k(LB^2/2+\gamma)}{\tau} \leq \frac{\varepsilon(LB^2/2+\gamma)}{\tau}, \tag{16}$$

using $\alpha_k \leq \varepsilon$ in the second inequality.

If the step k is successful, then let j be the minimum positive index such that the step $k+j$ is unsuccessful. Notice that such a j exists because $\alpha_k \to 0$ which implies by the Algorithm's construction an infinite subsequence of unsuccessful steps. We have that $\alpha_{k+i} = \alpha_k\gamma_2^i$ for $i \in [0:j-1]$, and since $\alpha_{k+j-1} \leq \varepsilon$ by induction we get $\alpha_{k+i} \leq \varepsilon\gamma_2^{i-j+1}$. Therefore,

$$\sum_{i=0}^{j-1}\alpha_{k+i} \leq \sum_{i=0}^{j-1}\varepsilon\gamma_2^{i-j+1} \leq \varepsilon\sum_{h=0}^{\infty}\gamma_2^{-h} = \varepsilon\frac{\gamma_2}{\gamma_2-1}. \tag{17}$$

Then,

$$\begin{aligned}\operatorname{dist}(x_k, x_{k+j}) &\leq \sum_{i=0}^{j-1} \operatorname{dist}(x_{k+i}, x_{k+i+1}) = \sum_{i=0}^{j-1} \operatorname{dist}(x_{k+i}, R(x_{k+i}, \alpha_{k+i} p_{k+i}^{j(k+i)})) \\ &\leq \sum_{i=0}^{j-1} L_r \alpha_{k+i} B \leq L_r B \varepsilon \frac{\gamma_2}{\gamma_2 - 1}, \end{aligned} \tag{18}$$

where we used (3) together with (6) in the second inequality, and (17) in the third one.

In turn,

$$\begin{aligned}\|\operatorname{grad} f(x_k)\| &\leq \operatorname{dist}^*(\operatorname{grad} f(x_k), \operatorname{grad} f(x_{k+j})) + \left\|\operatorname{grad} f(x_{k+j})\right\| \\ &\leq L_f \operatorname{dist}(x_k, x_{k+j}) + \frac{\varepsilon(LB^2/2 + \gamma)}{\tau} \\ &\leq \varepsilon \left(\frac{LB^2/2 + \gamma}{\tau} + L_f L_r B \frac{\gamma_2}{\gamma_2 - 1} \right), \end{aligned} \tag{19}$$

where we used (4) and (16) with $k + j$ instead of k for the first and second summand, respectively, in the second inequality, and (18) in the last one. □

3.4 Incorporating an Extrapolation Linesearch

The work [25, 26] introduced the use of an extrapolating linesearch that tests the objective on variable inputs farther away from the current iterate than the tentative point obtained by direct search on a given direction (i.e., an element of the positive spanning set). Such a thorough exploration of the search directions ultimately yields better performances in practice by computing longer successfully objective-decreasing steps. In this work, it is shown that the same technique can be applied in the Riemannian setting to good effect. In particular, in this section our Riemannian Direct Search with Extrapolation method based on Spanning Bases (RDSE-SB) for smooth objectives is presented. The scheme is described in detail as Algorithm 2, which can be viewed as a Riemannian version of [26, Algorithm 2].

The method uses a specific stepsize for each direction in the positive spanning set, so that instead of α_k there is a set of stepsizes $\{\alpha_k^j\}_{j\in[1:K]}$ for every $k \in \mathbb{N}_0$. Furthermore, a retraction-based linesearch procedure (see Algorithm 3) is used to better explore a given direction in case a sufficient decrease in the objective is obtained.

When analyzing the RDSE-SB method, due to the changes in the tangent space, the same positive spanning set cannot be kept for different iterates as is done in the unconstrained case (see [26, Algorithm 2, Step 2 and 3]). Therefore, using the distance dist^* to compare different tangent spaces, a novel condition is introduced here ensuring some continuity in the choice of the positive spanning set.

Assumption 4 For every $k \in \mathbb{N}$, $j \in [1 : K]$, there exists a constant $L_\Gamma > 0$ such that

$$\operatorname{dist}^*(p_k^j, p_{k+1}^j) \leq L_\Gamma \operatorname{dist}(x_k, x_{k+1}). \tag{20}$$

When $\mathcal{M}$ is embedded in $\mathbb{R}^n$ and $\mathcal{L}_0$ is compact, it is easy to see that condition (20) holds if $\{p_k^j\}_{j\in[1:K]}$ is the projection of a positive spanning set of $\mathbb{R}^n$ (independent from k) into $T_k\mathcal{M}$, using that $T_x\mathcal{M}$ varies smoothly with x.

It is now convenient to define, for $k \le l$, $\tilde{\Gamma}_k^l = \Gamma_{l-1}^l \circ \ldots \circ \Gamma_k^{k+1}$, where for $k = l$ the composition on the RHS is empty and we set $\tilde{\Gamma}_k^l$ equal to the identity. Let also

$$d(k,l) = \sum_{i=0}^{l-k-1} \operatorname{dist}(x_{k+i}, x_{k+i+1}). \tag{21}$$

The following lemma, which links the directions of the positive spanning sets in different iterates, holds:

Lemma 3.3 *Let $f \in C^1(\mathcal{M})$, $\{x_k\}$ be generated by Algorithm 2, and Assumptions 1, 3, 4 hold. For $k \in \mathbb{N}$, $j \ge 0$, $i \in [1:K]$:*

$$\begin{aligned}|\langle \operatorname{grad} f(x_k), p_k^i\rangle - \langle \operatorname{grad} f(x_{k+j}), p_{k+j}^i\rangle| &\le L_\Gamma \|\operatorname{grad} f(x_k)\| \, d(k,k+j)\\ &+L_f d(k,k+j).\end{aligned} \tag{22}$$

Proof First,

$$\begin{aligned}\left\|\tilde{\Gamma}_k^{k+h} p_k^i - p_{k+h}^i\right\| &= \left\|\sum_{j=0}^{h-1}(\tilde{\Gamma}_{k+j}^{k+h} p_{k+j}^i - \tilde{\Gamma}_{k+j+1}^{k+h} p_{k+j+1}^i)\right\|\\ &\le \sum_{j=0}^{h-1}\left\|\tilde{\Gamma}_{k+j}^{k+h} p_{k+j}^i - \tilde{\Gamma}_{k+j+1}^{k+h} p_{k+j+1}^i\right\| = \sum_{j=0}^{h-1}\left\|\tilde{\Gamma}_{k+j+1}^{k+h}(\Gamma_{k+j}^{k+j+1} p_{k+j}^i - p_{k+j+1}^i)\right\|\\ &= \sum_{j=0}^{h-1}\left\|\Gamma_{k+j}^{k+j+1} p_{k+j}^i - p_{k+j+1}^i\right\| \le \sum_{j=0}^{h-1} L_\Gamma \operatorname{dist}(x_{k+j}, x_{k+j+1}) = L_\Gamma d(k,k+h),\end{aligned} \tag{23}$$

where we used (20) in the last inequality. Analogously, from (4) it follows

$$|\left\|\operatorname{grad} f(x_{k+h}) - \tilde{\Gamma}_k^{k+h}\operatorname{grad} f(x_k)\right\| \le L_f d(k,k+h). \tag{24}$$

We can then conclude

$$\begin{aligned}|\langle \operatorname{grad} f(x_{k+h}), p_{k+h}^i\rangle - \langle \operatorname{grad} f(x_k), p_k^i\rangle| &= |\langle \operatorname{grad} f(x_{k+h}), p_{k+h}^i\rangle\\ &\quad -\langle \tilde{\Gamma}_k^{k+h}\operatorname{grad} f(x_k), \tilde{\Gamma}_k^{k+h} p_k^i\rangle|\\ &= |\langle \operatorname{grad} f(x_{k+h}) - \tilde{\Gamma}_k^{k+h}\operatorname{grad} f(x_k), p_{k+h}^i\rangle\\ &\quad -\langle \tilde{\Gamma}_k^{k+h}\operatorname{grad} f(x_k), \tilde{\Gamma}_k^{k+h} p_k^i - p_{k+h}^i\rangle|\\ &\le |\langle \operatorname{grad} f(x_{k+h})\\ &\quad -\tilde{\Gamma}_k^{k+h}\operatorname{grad} f(x_k), p_{k+h}^i\rangle| + |\langle \tilde{\Gamma}_k^{k+h}\operatorname{grad} f(x_k), \tilde{\Gamma}_k^{k+h} p_k^i - p_{k+h}^i\rangle|\end{aligned}$$

$$
\begin{aligned}
&\leq \left\| \operatorname{grad} f(x_{k+h}) - \tilde{\Gamma}_k^{k+h} \operatorname{grad} f(x_k) \right\| \left\| p_{k+h}^i \right\| \\
&\quad + \left\| \tilde{\Gamma}_k^{k+h} \operatorname{grad} f(x_k) \right\| \left\| \tilde{\Gamma}_k^{k+h} p_k^i - p_{k+h}^i \right\| \\
&\leq B L_f d(k, k+h) + L_\Gamma d(k, k+h) \left\| \operatorname{grad} f(x_k) \right\|, \qquad (25)
\end{aligned}
$$

where we used (23), (24) and (6) in the last inequality. □

Algorithm 2 RDSE-SB

1: **Input:** $x_0 \in \mathbb{R}^n$, $\{\alpha_0^j\}_{j\in[1:K]}$, $\gamma > 0$, $\gamma_1 \in (0,1)$, $\gamma_2 \geq 1$.
2: **for** $k = 0, 1, \ldots$ **do**
3: Compute a positive spanning set $\{p_k^j\}_{j\in[1:K]}$ of $T_k\mathcal{M}$
4: Set $j(k) = \mod(k, n)$, $\alpha_k^i = \tilde{\alpha}_k^i$ and $\tilde{\alpha}_{k+1}^i = \tilde{\alpha}_k^i$ for $i \in [1:K] \setminus \{j(k)\}$.
5: Compute $\alpha_k^{j(k)}, \tilde{\alpha}_{k+1}^{j(k)}$ with **linesearch procedure**$(\tilde{\alpha}_k^{j(k)}, x_k, p_k^{j(k)}, \gamma, \gamma_1, \gamma_2)$
6: Set $x_{k+1} = R(x_k, \alpha_k^{j(k)} p_k^{j(k)})$
7: **end for**

Algorithm 3 Linesearchprocedure$(x, \alpha, d, \gamma, \gamma_1, \gamma_2)$

1: **if** $f(R(x_k, \alpha d)) > f(x) - \gamma\alpha^2$ **then**
2: **Return** $(0, \gamma_1\alpha)$
3: **end if**
4: **while** $f(R(x_k, \alpha d)) \leq f(x) - \gamma\alpha^2$ **do**
5: Set $\alpha = \gamma_2\alpha$
6: **end while**
7: **Return** $(\alpha/\gamma_2, \alpha/\gamma_2)$

Asymptotic convergence of this method is proved in the remaining part of this section.

Lemma 3.4 *Let $f \in C^1(\mathcal{M})$, $\{x_k\}$ generated by Algorithm 2, and let Assumptions 2, 3 hold. At every iteration k, the following inequality holds:*

$$
-\langle \operatorname{grad} f(x_k), p_k^{j(k)} \rangle < \tilde{\alpha}_{k+1}^{j(k)} \frac{\gamma_2}{\gamma_1} (LB^2/2 + \gamma). \qquad (26)
$$

Proof It is immediate to check that we must always have

$$
f(R(x_k, \Delta_k p_k^{j(k)})) > f(x_k) - \gamma \Delta_k^2, \qquad (27)
$$

for $\Delta_k = \frac{1}{\gamma_1}\tilde{\alpha}_{k+1}^{j(k)}$ if the linesearch procedure terminates at the second line, and $\Delta_k = \gamma_2\tilde{\alpha}_{k+1}^{j(k)}$ if the linesearch procedure terminates in the last line. Then in both cases

$$
-\langle \operatorname{grad} f(x_k), p_k^{j(k)} \rangle < \Delta_k (LB^2/2 + \gamma) \leq \tilde{\alpha}_{k+1}^{j(k)} \frac{\gamma_2}{\gamma 1} (LB^2/2 + \gamma), \qquad (28)
$$

where we used Lemma 3.1 in the first inequality. □

Assumption 4 makes it possible to extend [26, Proposition 5.2] to the Riemannian case.

Theorem 3.2 *Let $f \in C^1(\mathcal{M})$, $\{x_k\}$ be generated by Algorithm 1, and let Assumptions 1, 2, 3 and 4 hold. We have*

$$\lim_{k\to\infty} \|grad f(x_k)\| \to 0. \tag{29}$$

Proof Let $\bar{\alpha}_k = \max_{j\in[1:K]} \tilde{\alpha}_{k+1}^{j(k)}$, so that $\bar{\alpha}_k \to 0$ since $\tilde{\alpha}_k^{j(k)} \to 0$, reasoning as in the proof of Theorem 3.1. As a consequence of Lemma 3.4, we have

$$-\langle \text{grad} f(x_k), p_k^{j(k)} \rangle < \bar{\alpha}_k c_1, \tag{30}$$

for the constant $c_1 = \frac{\gamma_2}{\gamma_1}(LB^2/2 + \gamma)$ independent from $j(k)$.
It remains to bound $\langle \text{grad} f(x_k), p_k^i \rangle$ for $i \neq j(k)$. To start with, we have the following bound:

$$\begin{aligned} -\langle \text{grad} f(x_k), p_k^i \rangle &\leq -\langle \text{grad} f(x_{k+h}), p_{k+h}^i \rangle + |\langle \text{grad} f(x_{k+h}), p_{k+h}^i \rangle \\ &\quad -\langle \text{grad} f(x_k), p_k^i \rangle| \\ &\leq c_1 \bar{\alpha}_{k+h} + |\langle \text{grad} f(x_{k+h}), p_{k+h}^i \rangle - \langle \text{grad} f(x_k), p_k^i \rangle|, \end{aligned} \tag{31}$$

for $h \leq K$ such that $i = j(k+h)$, and where in the second inequality we used (30) with $k+h$ instead of k. For the second summand appearing in the RHS of (31), from Lemma 3.3 it follows

$$\begin{aligned} |\langle \text{grad} f(x_{k+h}), p_{k+h}^i \rangle - \langle \text{grad} f(x_k), p_k^i \rangle| &\leq L_f d(k, k+h) B \\ + L_\Gamma \|\text{grad} f(x_k)\| \, d(k, k+h). \end{aligned} \tag{32}$$

We can now bound $d(k, k+h)$ as follows

$$\begin{aligned} d(k, k+h) &= \sum_{l=0}^{h-1} \text{dist}(x_{k+l+1}, x_{k+l}) \\ &= \sum_{l=0}^{h-1} \text{dist}(x_{k+l}, R(x_{k+l}, \bar{\alpha}_{k+l} p_{k+l}^{j(k+l)})) \leq \sum_{l=0}^{h-1} L_r \bar{\alpha}_{k+l} \left\| p_{k+l}^{j(k+l)} \right\| \\ &\leq B L_r \sum_{l=0}^{h-1} \bar{\alpha}_{k+l} \leq h B L_r \max_{l\in[0:h-1]} \bar{\alpha}_{k+l} \\ &\leq K B L_r \max_{l\in[0:K]} \bar{\alpha}_{k+l}, \end{aligned} \tag{33}$$

where we used (3) in the second inequality, (6) in the third one, and $h \leq K$ in the last one.

Let $\Delta_k = \max_{l\in[0:K]} \bar{\alpha}_{k+l}$, so that in particular $\Delta_k \to 0$.
For every $i \in [1:K]$:

$$
\begin{aligned}
-\langle \operatorname{grad} f(x_k), p_k^i \rangle &\leq c_1 \bar{\alpha}_{k+h} + L_f d(k, k+h) B + L_\Gamma \|\operatorname{grad} f(x_k)\| d(k, k+h) \\
&\leq c_2 \Delta_k + c_3 \Delta_k \|\operatorname{grad} f(x_k)\|, \qquad (34)
\end{aligned}
$$

for $c_2 = c_1 + L_f B^2 K L_r$ and $c_3 = K B L_r L_\Gamma$. Then, applying (7) and (34), we get

$$
\tau \|\operatorname{grad} f(x_k)\| \leq \max_{i\in[1:K]} -\langle \operatorname{grad} f(x_k), p_k^i \rangle \leq c_2 \Delta_k + c_3 \Delta_k \|\operatorname{grad} f(x_k)\| \qquad (35)
$$

and rearranging, for k large enough so that $\tau - c_3 \Delta_k > 0$,

$$
\|\operatorname{grad} f(x_k)\| \leq \frac{c_2 \Delta_k}{\tau - c_3 \Delta_k} \to 0, \qquad (36)
$$

as desired. □

4 Nonsmooth Objectives

In this section, some direct search methods are studied in the context where f is Lipschitz continuous, and bounded from below, but not necessarily continuously differentiable. The algorithms detailed here are built around the ideas given in [15], where the authors consider direct search methods for nonsmooth objectives in Euclidean space.

4.1 Clarke Stationarity for Nonsmooth Functions on Riemannian Manifolds

In order to perform our analysis, a definition of the Clarke directional derivative for a point $x \in \mathcal{M}$ is needed. The standard approach is to write the function in coordinate charts and take the standard Clarke derivative in an Euclidean space (see, e.g., [19, 20]). Formally, given a chart (φ, U) at $x \in \mathcal{M}$ and $v \in T_x\mathcal{M}$,

$$
f^\circ(x; v) = \tilde{f}^\circ(\varphi(x); \mathrm{d}\varphi(x)v), \qquad (37)
$$

for $\tilde{f}(y) = f(\varphi^{-1}(y))$. The following lemma shows the relationship between definition (37) and a directional derivative like object defined with retractions. This nontrivial result is the key tool allowing us to extend the analysis of direct search methods on $\mathbb{R}^n$ to the Riemannian setting.

Lemma 4.1 *Let f be Lipschitz continuous. If $(y_k, q_k) \to (x, d)$ and $t_k \to 0$,*

$$
f^\circ(x; d) \geq \limsup_{k\to\infty} \frac{f(R(y_k, t_k q_k)) - f(y_k)}{t_k}. \qquad (38)
$$

The proof is rather technical and thus deferred to the Appendix.

4.2 Refining Subsequences

The definition of refining subsequence used in the analysis of direct search methods (see, e.g., [3, 15]) is adapted here to the Riemannian setting. Let (x_k, d_k) be a sequence in $T\mathcal{M}$.

Definition 4.1 The subsequence $\{x_{i(k)}\}$ is refining if $x_{i(k)} \to x^*$, and $i(k)$ is unsuccessful for every k. In this case, the limit x^* is called a refined point.

Definition 4.2 Given a refining subsequence $\{x_{i(k)}\}$ with refined point x^*, a direction $d \in T_x\mathcal{M}$ with $\|d\|_x = 1$ is said to be a refined direction if for a further subsequence $\{j(i(k))\}$

$$\lim_{k\to\infty} \operatorname{dist}^*(d_{j(i(k))}, d) = 0. \tag{39}$$

A sufficient condition for the directions in a refined point to be refining is now given, assuming that the manifold is embedded in $\mathbb{R}^n$ and that the directions are obtained projecting from the unit sphere to the tangent spaces.

Proposition 4.1 *If $x_{i(k)}$ is a refining subsequence, $\bar{d}_{i(k)}$ is dense in the unit sphere,*

$$d_{i(k)} = \frac{\mathsf{P}_k(\bar{d}_{i(k)})}{\|\mathsf{P}_k(\bar{d}_{i(k)})\|_k},$$

for $\mathsf{P}_k(\bar{d}_{i(k)}) \neq 0$ and $d_{i(k)} = 0$; otherwise, then every $d \in T_{x^*}\mathcal{M}$ with $\|d\|_{x^*} = 1$ is refining.

Proof Fix $d \in T_{x^*}\mathcal{M}$, with $\|d\|_{x^*} = 1$, and let $\bar{d} = d/\|d\|$. By density, $\bar{d}_{j(i(k))} \to \bar{d}$ for a proper choice of the subsequence $\{j(i(k))\}$. Then,

$$\lim_{k\to\infty} d_{j(i(k))} = \lim_{k\to\infty} \frac{\mathsf{P}_k(\bar{d}_{j(i(k))})}{\left\|\mathsf{P}_k(\bar{d}_{j(i(k))})\right\|_k} = \frac{\mathsf{P}_{x^*}(\bar{d})}{\left\|\mathsf{P}_{x^*}(\bar{d})\right\|_{x^*}} = \frac{\bar{d}}{\left\|\bar{d}\right\|_{x^*}} = d, \tag{40}$$

where in the second equality we used the continuity of P_x and of the norm $\|\cdot\|_x$, and in the third equality we used $\mathsf{P}_{x^*}(\bar{d}) = \bar{d}$ since $\bar{d} \in T_{x^*}\mathcal{M}$ by construction. □

4.3 Direct Search for Nonsmooth Objectives

Our Riemannian Direct Search method based on Dense Directions (RDS-DD) for nonsmooth objectives is presented here. The scheme is presented in detail as Algorithm 4. The algorithm performs three simple steps at an iteration k. First, a search direction is selected randomly in the current tangent space. Then, a tentative point is generated by retracting the step $\alpha_k d_k$ from the tangent space to the manifold. Such a point is then eventually accepted as the new iterate if a sufficient decrease condition of the objective function is satisfied (and the stepsize is expanded); otherwise, the iterate stays the same. (And the stepsize is reduced.)

Algorithm 4 RDS-DD

Input: $x_0 \in \mathbb{R}^n$, $\alpha_0 > 0$, $\gamma > 0$, $\gamma_1 \in (0,1)$, $\gamma_2 \geq 1$
for $k = 0, 1, \ldots$ **do**
  Sample d_k randomly in $\{d \in T_k\mathcal{M} \mid \|d\|_{x_k} = 1\}$
  if $f(R(x_k, \alpha_k d_k)) \leq f(x) - \gamma\alpha_k^2$ **then**
    $x_{k+1} = R(x_k, \alpha_k d_k)$, $\alpha_{k+1} = \gamma_2 \alpha_k$
  else
    $x_{k+1} = x_k$, $\alpha_{k+1} = \gamma_1 \alpha_k$
  end if
end for

Thanks to the theoretical tools previously introduced, and in particular to the relation between retractions and the Clarke directional derivative proved in Lemma 4.1, it showed in a straightforward way that a suitable subsequence of unsuccessful iterations of the RDS-DD method converges to a Clarke stationary point.

Theorem 4.1 *Let f be Lipschitz continuous and $\{x_k\}$ be generated by Algorithm 4. If $\{x_{i(k)}\}$ is refining, with $x_{i(k)} \to x^*$, and every $d \in T_{x^*}\mathcal{M}$ with $\|d\|_{x^*} = 1$ is a refining direction, x^* is Clarke stationary.*

Proof By the same assumptions as in the smooth case $\alpha_k \to 0$ and in particular $\alpha_{i(k)} \to 0$. Since by assumption $i(k)$ is an unsuccessful step, we have, for every $i(k)$,

$$f(R(x_{i(k)}, \alpha_{i(k)} d_{i(k)})) - f(x_{i(k)}) > -\gamma\alpha_{i(k)}^2. \tag{41}$$

Let $d \in T_{x^*}\mathcal{M}$ with $\|d\|_{x^*} = 1$, let $\{j(i(k))\}$ be such that $d_{j(i(k))} \to d$, and let $y_k = x_{j(i(k))}$, $q_k = d_{j(i(k))}$, $t_k = \alpha_{j(i(k))}$. We have

$$\limsup_{k\to\infty} \frac{f(R(y_k, t_k q_k)) - f(y_k)}{t_k} \geq \limsup_{k\to\infty} -\gamma\alpha_{i(k)} = 0, \tag{42}$$

thanks to (41), and by applying Lemma 4.1 we get

$$f^\circ(x^*; d) \geq \limsup_{k\to\infty} \frac{f(R(y_k, t_k q_k)) - f(y_k)}{t_k} \geq 0, \tag{43}$$

which implies the thesis since d is arbitrary. □

4.4 Direct Search with Linesearch Extrapolation for Nonsmooth Objectives

Our Riemannian Direct Search method with linesearch Extrapolation based on Dense Directions (RDSE-DD) for nonsmooth objectives is presented here. It can be seen as an extension to the Riemannian setting of the DFN$_{simple}$ algorithm introduced in [15] for the Euclidean setting with bound constraints. The detailed scheme is given in Algorithm 5. The algorithm performs just two simple steps at an iteration k. First, a given search direction is suitably projected on the current tangent space. Then,

Algorithm 5 RDSE-DD

1: **Input:** $x_0 \in \mathbb{R}^n$, $\alpha_0 > 0$, $\gamma > 0$, $\gamma_1 \in (0, 1)$, $\gamma_2 \geq 1$.
2: **for** $k = 0, 1, \ldots$ **do**
3: Sample d_k randomly in $\{d \in T_k\mathcal{M} \mid \|d\|_{x_k} = 1\}$
4: Compute $\alpha_k, \tilde{\alpha}_{k+1}$ with **linesearch procedure**$(\tilde{\alpha}_k, x_k, d_k, \gamma, \gamma_1, \gamma_2)$
5: Set $x_{k+1} = R(x_k, \alpha_k d_k)$
6: **end for**

a linesearch is performed using Algorithm 3 to hopefully obtain a new point that guarantees a sufficient decrease.

Once again, by exploiting the theoretical tools previously introduced, it is proved in a straightforward way that a suitable subsequence of the RDSE-DD iterations converges to a Clarke stationary point. Thanks to the use of the linesearch strategy, the following result is not restricted to considering unsuccessful iterations. Given the lack of such iterations, for the purposes of Definition 4.1, every converging subsequence generated by Algorithm 5 is considered as refining.

Theorem 4.2 *Let f be Lipschitz continuous and $\{x_k\}$ be generated by Algorithm 5. If $\{x_{i(k)}\}$ is refining, with $x_{i(k)} \to x^*$ and every $d \in T_{x^*}\mathcal{M}$ with $\|d\|_{x^*} = 1$ is a refining direction, then x^* is Clarke stationary.*

Proof Let $\beta_k = \tilde{\alpha}_{k+1}/\gamma_1$ if the linesearch procedure exits before the loop, and $\beta_k = \gamma_2\tilde{\alpha}_k$ otherwise, so that in particular $\beta_k \to 0$. Then by definition of the linesearch procedure, for every k

$$f(R(x_k, \beta_k d_k)) - f(x_k) > -\gamma\beta_k^2. \tag{44}$$

The rest of the proof is analogous to that of Theorem 4.1. □

5 Numerical Results

In this section, results of some numerical experiments of the algorithms described in this paper on a set of simple but illustrative example problems are presented. The comparison among the algorithms is carried out by using data and performance profiles [27]. Specifically, let S be a set of algorithms and P a set of problems. For each $s \in S$ and $p \in P$, let $t_{p,s}$ be the number of function evaluations required by algorithm s on problem p to satisfy the condition

$$f(x_k) \leq f_L + \tau(f(x_0) - f_L), \tag{45}$$

where $0 < \tau < 1$ and f_L is the best objective function value achieved by any solver on problem p. Then, the performance and data profiles of solver s are defined, respectively, by the following functions

$$\rho_s(\alpha) = \frac{1}{|P|}\left|\left\{p \in P : \frac{t_{p,s}}{\min\{t_{p,s'} : s' \in S\}} \leq \alpha\right\}\right|,$$

$$d_s(\kappa) = \frac{1}{|P|} \left| \left\{ p \in P : t_{p,s} \leq \kappa(n_p + 1) \right\} \right|,$$

where n_p is the dimension of problem p.

A budget of $100(n_p + 1)$ function evaluations is used in all cases, and two different precisions for the condition (45), that is, $\tau \in \{10^{-1}, 10^{-3}\}$. Randomly generated instances of well-known optimization problems over manifolds from [1, 8, 18] are considered. A brief description of those problems as well as the details of our implementation can be found in Appendix (see Sects. 7.3, 7.4 and 7.5). The size of the ambient space for the instances varies from 2 to 200. In the results, the problems are split by ambient space dimension: between 2 and 15 for small instances, between 16 and 50 for medium instances, and between 51 and 200 for large instances.

5.1 Smooth Problems

In Fig. 1, the results related to 8 smooth instances of problem (1) from [1, 8] are included, each with 15 different problem dimensions (from 2 to 200), for a total number of 60 tested instances, split as described above. Our methods, that is, RDS-SB and RDSE-SB, are compared with the zeroth-order gradient descent (ZO-RGD, [23, Algorithm 1]).

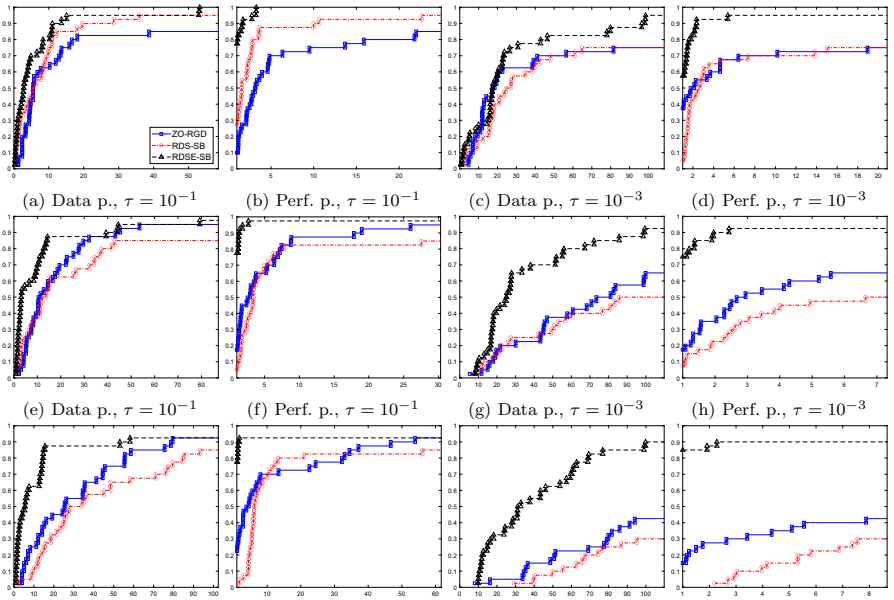
The results clearly show that RDSE-SB performs better than RDS-SB and ZO-RGD both in efficiency and reliability for both levels of precision. It can also be seen how the gap between RDSE-SB and the other two algorithms gets larger as the problem dimension grows.

5.2 Nonsmooth Problems

Here, a preliminary comparison is reported between a direct search strategy, a linesearch strategy, and ZO-RGD on two nonsmooth instances of (1) from [18], each with 15 different problem sizes (from 2 to 200), thus getting a total number of 30 tested instances, split by dimension as for smooth instances. It should be noted that while in the unconstrained setting the performance of zeroth-order (sub)gradient descent methods on nonsmooth objectives has been analyzed (see, e.g., [28]), there are, to the best of our knowledge, no convergence guarantees in the Riemannian setting.

In the direct search strategy (RDS-DD+), the RDS-SB method is applied until $\alpha_{k+1} \leq \alpha_\epsilon$, at which point the nonsmooth version RDS-DD is used. Analogously, in the linesearch strategy (RDSE-DD+), the RDSE-SB method is applied until $\max_{j \in [1:K]} \tilde{\alpha}_{k+1}^j \leq \alpha_\epsilon$, at which point the nonsmooth version RDSE-DD is used. Both strategies use a threshold parameter $\alpha_\epsilon > 0$ to switch from the smooth to the nonsmooth DFO algorithm. The reader is referred to [15] and references therein for other direct search strategies combining coordinate and dense directions.

In Fig. 2, the comparison between the considered strategies is reported. As in the smooth case, the linesearch-based strategy outperforms both the simple direct search and the zeroth-order one. It can once again be seen how the gap between the algorithms gets larger as the problem dimension gets large enough.

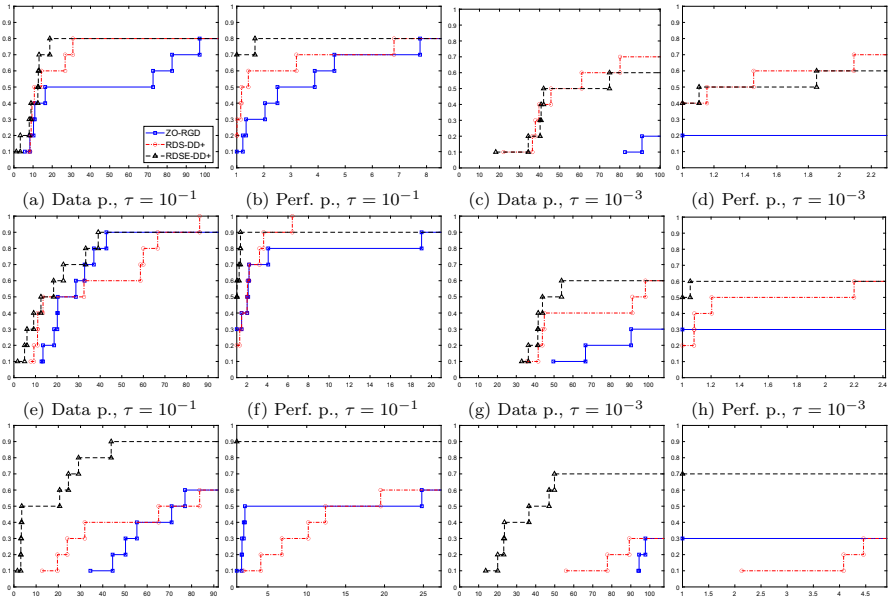


(a) Data p., $\tau = 10^{-1}$ (b) Perf. p., $\tau = 10^{-1}$ (c) Data p., $\tau = 10^{-3}$ (d) Perf. p., $\tau = 10^{-3}$

(e) Data p., $\tau = 10^{-1}$ (f) Perf. p., $\tau = 10^{-1}$ (g) Data p., $\tau = 10^{-3}$ (h) Perf. p., $\tau = 10^{-3}$

(i) Data p., $\tau = 10^{-1}$ (j) Perf. p., $\tau = 10^{-1}$ (k) Data p., $\tau = 10^{-3}$ (l) Perf. p., $\tau = 10^{-3}$

Fig. 1 From top to bottom: results for small, medium, and large instances in the smooth case



(a) Data p., $\tau = 10^{-1}$ (b) Perf. p., $\tau = 10^{-1}$ (c) Data p., $\tau = 10^{-3}$ (d) Perf. p., $\tau = 10^{-3}$

(e) Data p., $\tau = 10^{-1}$ (f) Perf. p., $\tau = 10^{-1}$ (g) Data p., $\tau = 10^{-3}$ (h) Perf. p., $\tau = 10^{-3}$

(i) Data p., $\tau = 10^{-1}$ (j) Perf. p., $\tau = 10^{-1}$ (k) Data p., $\tau = 10^{-3}$ (l) Perf. p., $\tau = 10^{-3}$

Fig. 2 From top to bottom: results for small, medium, and large instances in the nonsmooth case

6 Conclusion

In this paper, direct search algorithms with and without an extrapolation linesearch for minimizing functions over a Riemannian manifold are presented. It was found that, modulo modifications to account for the changing vector space structure with the iterations, direct search strategies provide guarantees of convergence for both smooth and nonsmooth objectives. It was also found that in practice, in our numerical experiments, the extrapolation linesearch speeds up the performance of direct search in both cases, and it appears that it even outperforms a gradient approximation-based zeroth-order Riemannian algorithm in the smooth case. As a natural extension for future work, considering the stochastic case would be a reasonable next step.

Funding Open access funding provided by Università degli Studi di Padova within the CRUI-CARE Agreement.

Data availability The datasets generated during the current study are available from the corresponding author on reasonable request.

7 Appendix

7.1 Proof of Proposition 3.1

Before proving Proposition 3.1, a local version is stated and proved.

Lemma A.1 *Let Assumption 1 hold and $R \in C^2(T\mathcal{M}, \mathcal{M})$. Then, for every $x \in \mathcal{M}$ there is a neighborhood U_x of x, and constants L_x and D_x such that for every $y \in U_x$, $d \in T_y\mathcal{M}$ with $\|d\| \leq D_x$*

$$f(R(y,d)) \leq f(y) + \langle grad f(y), d\rangle + \frac{L_x}{2}\|d\|^2 . \tag{46}$$

Proof Let (φ) be a chart defined in a neighborhood U of $x \in \mathcal{M}$. We can take the neighborhood small enough so that for y, z varying in U the parallel transport Γ_y^z depends smoothly on y, z and is uniquely defined. We use the notation $(\tilde{x}, \tilde{d}) = (\varphi(x), \mathrm{d}\varphi(x)d)$ for $(x, d) \in T\mathcal{M}$. We push forward the manifold and the related structure with the chart φ, i.e., for $\bar{\varphi} = \varphi^{-1}$ we define $\tilde{f} = f \circ \bar{\varphi}$, $\tilde{U} = \varphi(U)$, $\tilde{R}(\tilde{y}, \tilde{d}) = R(y, d)$; for $d, q \in T_x\mathcal{M}$ we define $g(\tilde{d}, \tilde{q}) = \langle d, q\rangle_x$, $\|\tilde{d} - \tilde{q}\|_{\tilde{x}} = \|d - q\|_x$, and $\tilde{\Gamma}_{\tilde{x}}^{\tilde{y}}(\tilde{d}) = \Gamma_x^y(d)$. With slight abuse of notation, we use $\mathrm{dist}(\tilde{x}, \tilde{y})$ to denote $\mathrm{dist}(x, y)$. We also define as $\mathrm{grad}\,\tilde{f}$ the gradient of $\tilde{f}$ with respect to the scalar

product g, so that $g(\operatorname{grad}\tilde{f}(\tilde{x}),\tilde{d}) = \langle \nabla \tilde{f}(x), d\rangle$ for any $\tilde{d} \in \mathbb{R}^m$. Importantly, by the equivalence of norms in $\mathbb{R}^m$ we can use $O(\|\tilde{d}\|_x)$ and $O(\|\tilde{d}\|)$ interchangeably.

We can choose (φ, U) and $B > 0$ in such a way that for some neighborhood $\tilde{U}_x \subset \tilde{U}$ of $\tilde{x}$, for every $\tilde{y} \in \tilde{U}_x$ and $\tilde{d}$ with $\|\tilde{d}\|_{\tilde{y}} \leq B$ we have $\tilde{R}(\tilde{y},\tilde{d}) \in \tilde{U}_2 \subset \tilde{U}$, with $\tilde{U}_2$ compact.

With this notation, we need to prove

$$\tilde{f}(\tilde{R}(\tilde{y},\tilde{d})) \leq \tilde{f}(\tilde{y}) + g(\operatorname{grad}\tilde{f}(\tilde{y}),\tilde{d}) + \frac{L_x}{2}\left\|\tilde{d}\right\|_{\tilde{y}}^2, \tag{47}$$

for $\tilde{d}$ such that $\|\tilde{d}\|_{\tilde{y}} \leq B$, $\tilde{y} \in \tilde{U}_x \subset \tilde{U}$ and some $L_x > 0$.

First, since $\tilde{R}$ is in particular C^1 regular

$$\tilde{R}(\tilde{x},\tilde{d}) = \tilde{x} + O(\left\|\tilde{d}\right\|_{\tilde{x}}), \tag{48}$$

and by the local smoothness of the parallel transport, for $\tilde{y}, \tilde{z} \in \tilde{U}_2$, we have

$$\tilde{\Gamma}_{\tilde{y}}^{\tilde{z}}\tilde{q} = \tilde{q} + O(\|\tilde{y} - \tilde{z}\|). \tag{49}$$

Furthermore,

$$\begin{aligned}\operatorname{grad}\tilde{f}(\tilde{R}(\tilde{y},\tilde{q})) &= \tilde{\Gamma}_{\tilde{y}}^{\tilde{R}(\tilde{y},\tilde{q})}\operatorname{grad}\tilde{f}(\tilde{y}) + O(\operatorname{dist}(\tilde{y},\tilde{R}(\tilde{y},\tilde{q})))\\ &= \tilde{\Gamma}_{\tilde{y}}^{\tilde{R}(\tilde{y},\tilde{q})}\operatorname{grad}\tilde{f}(\tilde{y}) + O(\|\tilde{q}\|),\end{aligned} \tag{50}$$

where we used (4) in the first equality and (3) in the second equality.

Finally, since, $\frac{\mathrm{d}}{\mathrm{d}t}\tilde{R}(\tilde{y}, t\tilde{q})$ is C^1 regular, we also have

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\tilde{R}(\tilde{y}, t\tilde{q})|_{t=h} &= \frac{\mathrm{d}}{\mathrm{d}t}\tilde{R}(\tilde{y}, t\tilde{q})|_{t=0} + O(\|h\tilde{q}\|)\\ &= \tilde{q} + O(h\,\|\tilde{q}\|) = \tilde{\Gamma}_{\tilde{y}}^{R(\tilde{y},h\tilde{q})}\tilde{q} + O(\|R(\tilde{y},h\tilde{q}) - \tilde{y}\|) + O(h\,\|\tilde{q}\|)\\ &= \tilde{\Gamma}_{\tilde{y}}^{R(\tilde{y},h\tilde{q})}\tilde{q} + O(h\,\|\tilde{q}\|),\end{aligned} \tag{51}$$

where we used (49) in the third equality, and (3) in the last one. Again by compactness, for $\tilde{y} \in \tilde{U}_1$, $t \leq 1$, $\|\tilde{q}\| \leq B$ the implicit constants can be taken with no dependence from the variables.

Now, for $\tilde{d}$ s.t. $\tilde{d} \leq B$ define $\tilde{q} = B\tilde{d}/\|\tilde{d}\|$, so that $\tilde{d} = \bar{t}\tilde{q}$ for $\bar{t} = \|\tilde{d}\|/B$. Then, we obtain (47) reasoning as follows:

$$\begin{aligned}&\tilde{f}(\tilde{R}(\tilde{y},\tilde{d})) - \tilde{f}(\tilde{R}(\tilde{y},0)) = \tilde{f}(\tilde{R}(\tilde{y},\bar{t}\tilde{q})) - \tilde{f}(\tilde{R}(\tilde{y},0))\\ &= \int_0^{\bar{t}}\frac{\mathrm{d}}{\mathrm{d}t}\tilde{f}(\tilde{R}(\tilde{y}+t\tilde{q}))\mathrm{d}t = \int_0^{\bar{t}} g(\operatorname{grad}f(\tilde{R}(\tilde{y},t\tilde{q})), \frac{\mathrm{d}}{\mathrm{d}t}\tilde{R}(\tilde{y},t\tilde{d}))\mathrm{d}t\\ &= \int_0^{\bar{t}} g(\tilde{\Gamma}_{\tilde{y}}^{\tilde{R}(\tilde{y},t\tilde{q})}\operatorname{grad}\tilde{f}(\tilde{y}) + O(t\,\|\tilde{q}\|), \tilde{\Gamma}_{\tilde{y}}^{\tilde{R}(\tilde{y},t\tilde{d})}\tilde{d} + O(t\,\|\tilde{q}\|))dt\end{aligned}$$

$$= \int_0^{\tilde{t}} \left(g(\tilde{\Gamma}_{\tilde{y}}^{\tilde{R}(\tilde{y},t\tilde{q})} \operatorname{grad} \tilde{f}(\tilde{y}), \tilde{\Gamma}_{\tilde{y}}^{\tilde{R}(\tilde{y},t\tilde{d})} \tilde{d}) + O(t \|\tilde{q}\|) \right) \mathrm{d}t$$

$$= g(\operatorname{grad} f(\tilde{y}), \tilde{d}) + O(\tilde{t}^2 \|\tilde{q}\|) = g(\operatorname{grad} f(\tilde{y}), \tilde{d}) + O(\left\|\tilde{d}\right\|^2), \tag{52}$$

where we used (50) and (51) in the fourth equality, and the implicit constant for the $O(\|\tilde{d}\|^2)$ term can be taken as some $L_x > 0$ independent from $\tilde{d}$ and $\tilde{y}$. □

Proof (Proposition 3.1) By the compactness of $\mathcal{L}_0$, the local property of Lemma A.1 can be extended to all $\mathcal{L}_0$: For some $\tilde{L}, B > 0$, (5) holds for every $x \in \mathcal{L}_0 \cap \mathcal{M}$, $d \in T_x\mathcal{M}$ with $\|d\| \leq B$. Let

$$M_f = \max_{x \in \mathcal{L}_0} \|\operatorname{grad} f(x)\| . \tag{53}$$

We now claim that when $\|d\| > B$, (5) holds with $L = \frac{2(M_f + L_0 L_r)}{2B}$, for L_0 Lipschitz constant of f. Indeed in this case, we have

$$\begin{aligned} f(R(x,d)) &\leq f(x) + L_0 \operatorname{dist}(x, R(x,d)) \leq f(x) + L_0 L_r \|d\| \\ &= f(x) - M_f \|d\| + (M_f + L_0 L_r) \|d\| \leq f(x) + \langle \operatorname{grad} f(x), d \rangle \\ &\quad + \frac{2(M_f + L_0 L_r)}{2B} B \|d\| \\ &\leq f(x) + \langle \operatorname{grad} f(x), d \rangle + \frac{2(M_f + L_0 L_r)}{2B} \|d\|^2 , \end{aligned} \tag{54}$$

as desired. Combining the results obtained for the case $\|d\| \leq B$ and $\|d\| > B$, we obtain the desired result for $L = \max\left(\frac{2(L_0 L_r + M_f)}{B}, \tilde{L}\right)$. □

7.2 Proof of Lemma 4.1

In order to prove Lemma 4.1, the following lemma is needed, which will be proved with an argument analogous to the one used in the proof of [5, Proposition 3.9].

Lemma A.2 *For a Lipschitz continuous function $h : \mathbb{R}^m \to \mathbb{R}$, $\tilde{y}, \tilde{v} \in \mathbb{R}^m$, if $\tilde{y}_k \to \tilde{y}$, $\tilde{v}_k \to \tilde{v}$, and $t_k \to 0$, then*

$$h^\circ(\tilde{y}; \tilde{v}) \geq \limsup_{k \to \infty} \frac{h(\tilde{y}_k + t_k \tilde{v}_k) - h(\tilde{y}_k)}{t_k}. \tag{55}$$

Proof We have

$$|h(\tilde{y}_k + t_k \tilde{v}_k) - h(\tilde{y}_k + t_k \tilde{v})| \leq t_k L_h \|\tilde{v} - \tilde{v}_k\| = o(t_k), \tag{56}$$

with L_h being the Lipschitz constant of h. Then,

$$\limsup_{k \to \infty} \frac{h(\tilde{y}_k + t_k \tilde{v}_k) - h(\tilde{y}_k)}{t_k} = \limsup_{k \to \infty} \frac{h(\tilde{y}_k + t_k \tilde{v}) + o(t_k) - h(\tilde{y}_k)}{t_k}$$

$$= \limsup_{k\to\infty} \frac{h(\tilde{y}_k + t_k\tilde{v}) - h(\tilde{y}_k)}{t_k} \leq h^\circ(\tilde{y};\tilde{v}), \tag{57}$$

where we used (56) in the first equality, and with the inequality true by definition of the Clarke derivative. □

Proof (Lemma 4.1) We use the notation introduced in the proof of Proposition 3.1, so that in particular $\tilde{x} = \varphi(x)$ and $\tilde{d} = \mathrm{d}\varphi(x)d$. Without loss of generality, we assume that U is bounded, that φ can be extended to a neighborhood containing the closure of U, $\{x_k\} \subset U$, and that the parallel transport $\Gamma_x^y v$ depends smoothly from $x, y \in U$ and $v \in T_x\mathcal{M}$.

First, since pushforward $\tilde{R}$ of a C^2 retraction on $\mathbb{R}$ is a C^2 retraction itself of $T\mathbb{R}^m$ on $\mathbb{R}^m$, we have the Taylor expansion

$$\tilde{R}(\tilde{y},\tilde{v}) = \tilde{y} + \tilde{v} + O(\|\tilde{v}\|^2), \tag{58}$$

with the implicit constant uniform for $\tilde{y}$ varying in $\tilde{U}$ and $\tilde{v}$ chosen in $\mathbb{R}^m$.

Second, for any fixed constant $B > 0$, by continuity we have

$$\left\| \tilde{\Gamma}_{\tilde{x}}^{\tilde{x}_k}\tilde{q} - \tilde{q} \right\| \leq O\left(\|\tilde{x} - \tilde{x}_k\|\right), \tag{59}$$

for $k \to \infty$, $\tilde{q} \in \mathbb{R}^m$ with $\|\tilde{q}\| \leq B$, and with a uniform implicit constant.

Therefore,

$$\begin{aligned}\left\|\tilde{d}_k - \tilde{d}\right\| &\leq \left\|\tilde{d}_k - \tilde{\Gamma}_{\tilde{x}}^{\tilde{x}_k}\tilde{d}\right\| + \left\|\tilde{\Gamma}_{\tilde{x}}^{\tilde{x}_k}\tilde{d} - \tilde{d}\right\| \leq O\left(\left\|\tilde{d}_k - \tilde{\Gamma}_{\tilde{x}}^{\tilde{x}_k}(\tilde{d})\right\|_{\tilde{x}}\right) + O\left(\|\tilde{x} - \tilde{x}_k\|\right) \\ &= O\left(\left\|d_k - \Gamma_x^{x_k}(d)\right\|_x\right) + O\left(\|\tilde{x} - \tilde{x}_k\|\right) = o(1),\end{aligned} \tag{60}$$

where in the second inequality we used (59), and in the last equality we used $d_k \to d$ together with $\tilde{x}_k \to \tilde{x}$.

Let now $\tilde{v}_k = (\tilde{R}(\tilde{x}_k, t_k\tilde{d}_k) - \tilde{x}_k)/t_k$. Then,

$$\begin{aligned}\left\|\tilde{v}_k - \tilde{d}\right\| &= \frac{1}{t_k}\left\|\tilde{R}(\tilde{x}_k, t_k\tilde{d}_k) - \tilde{x}_k - t_k\tilde{d}\right\| \\ &\leq \frac{1}{t_k}\left(\left\|R(\tilde{x}_k, t_k\tilde{d}_k) - \tilde{x}_k - t_k\tilde{d}_k\right\| + t_k\left\|\tilde{d}_k - \tilde{d}\right\|\right) \\ &= \frac{1}{t_k}\left(O(t_k^2\left\|\tilde{d}_k\right\|^2) + t_k o(1)\right) = o(1),\end{aligned} \tag{61}$$

where we used (58) and (60) for the first and the second summand in the second equality. In other words, $\tilde{v}_k \to \tilde{d}$. To conclude,

$$\begin{aligned}&\limsup_{k\to\infty} \frac{f(R(y_k, t_k d_k)) - f(y_k)}{t_k} = \limsup_{k\to\infty} \frac{\tilde{f}(\tilde{R}(\tilde{y}_k, t_k\tilde{d}_k)) - \tilde{f}(\tilde{y}_k)}{t_k} \\ &= \limsup_{k\to\infty} \frac{\tilde{f}(\tilde{y}_k + t_k\tilde{v}_k) - \tilde{f}(\tilde{y}_k)}{t_k} \leq \tilde{f}^\circ(\tilde{x};\tilde{d}) = \tilde{f}^\circ(\varphi(x); \mathrm{d}\varphi(x)d) = f^\circ(x;d),\end{aligned} \tag{62}$$

where in the inequality we were able to apply Lemma A.2 because $\tilde{v}_k \to \tilde{d}$ by (61), and the last equality follows by the definition (37). □

7.3 Implementation Details

For all the problems, the manifold structure used was the one available in the MANOPT library [10].

After a basic tuning phase, the algorithm parameters were set as follows: $\gamma_1 = 0.61$, $\gamma_2 = 1$ and $\gamma = 0.77$ were used for Algorithm 1, $\gamma_1 = 0.81$, $\gamma_2 = 3.12$, and $\gamma = 0.11$ for Algorithm 2, and the stepsize $1.64/n$ (recall that n is the dimension of the ambient space) for the ZO-RGD method.

For the nonsmooth strategies RDS-DD+ and RDSE-DD+, the same parameters of the smooth case for RDS-SB and RDSE-SB were considered, setting $\alpha_\epsilon = 10^{-4}$, and for both RDS-DD and RDSE-DD used $\gamma_1 = 0.95$, $\gamma_2 = 2$, and $\gamma = 1$. When dealing with the nonsmooth case, the stepsize used for ZO-RGD was the same as the one considered in the smooth case.

The positive spanning set was obtained both in Algorithm 1 and Algorithm 2 by projecting the positive spanning set $(e_1, \ldots, e_n, -e_1, \ldots, -e_n)$ of the ambient space $\mathbb{R}^n$ on the tangent space. The initial stepsize was set to 1 for all the direct search methods, with no fine tuning.

The starting point and the parameters related to the instances were generated either with MATLAB rand function or by using the random element generators implemented in the MANOPT library.

7.4 Smooth Problems

Here, the 8 smooth instances of problem (1) from [1, 8] are described.

7.4.1 Largest Eigenvalue, Singular Value, and Top Singular Values Problem

In the largest eigenvalue problem [8, Section 2.3], given a symmetric matrix $A \in \mathbb{S}_{n-1} := \{A \in \mathbb{R}^{n\times n} \mid A = A^\top\}$, the goal is to compute

$$\max_{x \in S(0,1)} x^\top A x. \tag{63}$$

The largest singular value problem [8, Section 2.3] can be formulated generalizing (63): Given $A \in \mathbb{R}^{m\times h}$, the problem to solve is

$$\max_{x \in S(0,1), y \in S(0,1)} x^\top A y. \tag{64}$$

Notice how the domain in (63) and (64) is a sphere and the product of two spheres, respectively.

Finally, to compute the sum of the top r singular values, as explained in [8, Section 2.5] it suffices to solve

$$\max_{X \in St(m,r), Y \in St(h,r)} X^\top A Y, \tag{65}$$

for $St(a, b)$ the Stiefel manifold with dimensions (a, b).

7.4.2 Dictionary Learning

The dictionary learning problem [8, Section 2.4] can be formulated as

$$\begin{aligned} \min_{D \in \mathbb{R}^{d \times h}, C \in \mathbb{R}^{h \times k}} \quad & \|Y - DC\| + \lambda \|C\|_1 \\ \text{s.t.} \quad & \|D_1\| = \ldots = \|D_h\| = 1 \end{aligned} \tag{66}$$

for a fixed $Y \in \mathbb{R}^{d \times k}$, $\lambda > 0$, $\|\cdot\|_1$ the ℓ_1- norm, and $D_1, \ldots, D_h$ the columns of D.

In our implementation, the objective is smoothed by using a smoothed version $\|\cdot\|_{1,\varepsilon}$ of $\|\cdot\|_1$

$$\|C\|_{1,\varepsilon} = \sum_{i,j} \sqrt{C_{i,j}^2 + \varepsilon^2}. \tag{67}$$

In our tests, the solution $\bar{C}$ is generated using MATLAB sprand function, with a density of 0.3, set the regularization parameter λ to 0.01 and ε to 0.001.

7.4.3 Synchronization of Rotations

Let $\mathrm{SO}(d)$ be the special orthogonal group:

$$\mathrm{SO}(d) = \{R \in \mathbb{R}^{d \times d} \mid R^\top R = I_d \text{ and } \det(R) = 1\}. \tag{68}$$

In the synchronization of rotations problem [8, Section 2.6], rotations $R_1, \ldots, R_h \in \mathrm{SO}(d)$ must be retrieved from noisy measurements H_{ij} of $R_i R_j^{-1}$, for every $(i, j) \in E$, a subset of $\binom{h}{2}$ (the set of couples of distinct elements in $[1 : h]$). The objective is then

$$\min_{\hat{R}_1, \ldots, \hat{R}_h \in \mathrm{SO}(d)} \sum_{(i,j) \in E} \left\| \hat{R}_i - H_{ij} \hat{R}_j \right\|^2. \tag{69}$$

In our tests, the case $h = 2$ is considered for simplicity.

7.4.4 Low-rank Matrix Completion

The low-rank matrix completion problem [8, Section 2.7] can be written, for a fixed matrix $M \in \mathbb{R}^{m\times h}$, as

$$\begin{aligned} \min_{X\in\mathbb{R}^{m\times h}} \quad & \sum_{(i,j)\in\Omega} \left(X_{ij} - M_{ij}\right)^2, \\ \text{s.t.} \quad & \operatorname{rank}(X) = r, \end{aligned} \tag{70}$$

given a positive integer $r > 0$ and a subset of indices $\Omega \subset [1:m] \times [1:h]$. It can be proven that the optimization domain, that is, the matrices in $\mathbb{R}^{m\times n}$ with fixed rank r, can be given a Riemannian manifold structure (see, e.g., [29]).

7.4.5 Gaussian Mixture Models

In the Gaussian mixture model problem [8, Section 2.8], the goal is to compute a maximum likelihood estimation for a given set of observations $x_1, \ldots, x_h$:

$$\max_{\substack{\hat{u}_1,\ldots,\hat{u}_k\in\mathbb{R}^d \\ \hat{\Sigma}_1,\ldots,\hat{\Sigma}_k\in\mathrm{Sym}(d)^+, \\ w\in\Delta_+^{K-1}}} \sum_{i=1}^{h} \log\left(\sum_{k=1}^{K} w_k \frac{1}{\sqrt{2\pi\det(\Sigma_k)}} e^{\frac{(x-\mu_k)^\top \Sigma_k^{-1}(x-\mu_k)}{2}}\right), \tag{71}$$

where $\mathrm{Sym}(d)^+$ is the manifold of positive definite matrices

$$\mathrm{Sym}(d)^+ = \{X \in \mathbb{R}^{d\times d} \mid X = X^\top,\ X \succ 0\}, \tag{72}$$

and Δ_+^{K-1} is the subset of strictly positive elements of the simplex Δ^{K-1}, which can be given a manifold structure. In our tests, the case $K = 2$ is considered with the reformulation proposed in [17], which does not use the unconstrained variables $(\hat{u}_1, \ldots, \hat{u}_k)$.

7.4.6 Procrustes Problem

The Procrustes problem [1] is the following linear regression problem, for fixed $A \in \mathbb{R}^{l\times n}$ and $B \in \mathbb{R}^{l\times p}$:

$$\min_{X\in\mathrm{St}(n,p)} \|AX - B\|_F^2, \tag{73}$$

In our tests, the variable $X \in \mathbb{R}^{n\times p}$ is assumed to be in the Stiefel manifold $\mathrm{St}(n, p)$, a choice leading to the so-called unbalanced orthogonal Procrustes problem.

7.5 Nonsmooth Problems

Here, two nonsmooth problems taken from [18] are described.

7.5.1 Sparsest Vector in a Subspace

Given an orthonormal matrix $Q \in \mathbb{R}^{m\times n}$, the problem of finding the sparsest vector in the subspace generated by the columns of Q can be relaxed as

$$\min_{x\in\mathbb{S}^{n-1}} \|Qx\|_1 . \tag{74}$$

7.5.2 Nonsmooth Low-rank Matrix Completion

In the nonsmooth version of the low-rank matrix completion problem (70) the Euclidean norm is replaced with the l_1 norm, so that the objective consists in a sum of absolute values:

$$\begin{aligned} \min_{X\in\mathbb{R}^{m\times n}} \quad & \sum_{(i,j)\in\Omega} \left|X_{ij} - M_{ij}\right|, \\ \text{s.t.} \quad & \operatorname{rank}(X) = r. \end{aligned} \tag{75}$$

References

1. Absil, P.-A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton (2009)
2. Absil, P.-A., Malick, J.: Projection-like retractions on matrix manifolds. SIAM J. Optim. **22**, 135–158 (2012)
3. Audet, C., Dennis, J.E., Jr.: Analysis of generalized pattern searches. SIAM J. Optim. **13**, 889–903 (2002)
4. Audet, C., Le Digabel, S., Peyrega, M.: Linear equalities in blackbox optimization. Comput. Optim. Appl. **61**, 1–23 (2015)
5. Audet, C., Dennis, J.E., Jr.: Mesh adaptive direct search algorithms for constrained optimization. SIAM J. Optim. **17**, 188–217 (2006)
6. Audet, C., Hare, W.: Derivative-Free and Blackbox Optimization. Springer (2017)
7. Azagra, D., Ferrera, J., López-Mesas, F.: Nonsmooth analysis and Hamilton–Jacobi equations on Riemannian manifolds. J. Funct. Anal. **220**, 304–361 (2005)
8. Boumal, N.: An Introduction to optimization on smooth manifolds. http://sma.epfl.ch/nboumal/book/index.html (2022). Accessed 10 Feb 2022
9. Boumal, N., Absil, P.-A., Cartis, C.: Global rates of convergence for nonconvex optimization on manifolds. IMA J. Numer. Anal. **39**, 1–33 (2019)
10. Boumal, N., Mishra, B., Absil, P.-A., Sepulchre, R.: Manopt, a Matlab toolbox for optimization on manifolds. J. Mach. Learn. Res. **15**, 1455–1459 (2014)
11. Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to Derivative-Free Optimization. SIAM, Philadelphia (2009)
12. Cristofari, A., Rinaldi, F.: A derivative-free method for structured optimization problems. SIAM J. Optim. **31**, 1079–1107 (2021)
13. Dreisigmeyer, D.W.: Equality constraints, Riemannian manifolds and direct search methods. https://optimization-online.org/wp-content/uploads/2007/08/1743.pdf (2006). Accessed 21 Mar 2023
14. Dreisigmeyer, D.W.: Direct search methods on reductive homogeneous spaces. J. Optim. Theory Appl. **176**, 585–604 (2018)
15. Fasano, G., Liuzzi, G., Lucidi, S., Rinaldi, F.: A linesearch-based derivative-free approach for nonsmooth constrained optimization. SIAM J. Optim. **24**, 959–992 (2014)
16. Gratton, S., Royer, C.W., Vicente, L.N., Zhang, Z.: Direct search based on probabilistic descent. SIAM J. Optim. **25**, 1515–1541 (2015)
17. Hosseini, R., Sra, S.: Matrix manifold optimization for Gaussian mixtures. NIPS **28**, 910–918 (2015)

18. Hosseini, S., Mordukhovich, B.S., Uschmajew, A.: Nonsmooth Optimization and Its Applications. International Series of Numerical Mathematics, Springer (2019)
19. Hosseini, S., Pouryayevali, M.: Nonsmooth optimization techniques on Riemannian manifolds. J. Optim. Theory Appl. **158**, 328–342 (2013)
20. Hosseini, S., Uschmajew, A.: A Riemannian gradient sampling algorithm for nonsmooth optimization on manifolds. SIAM J. Optim. **27**, 173–189 (2017)
21. Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: new perspectives on some classical and modern methods. SIAM Rev. **45**, 385–482 (2003)
22. Larson, J., Menickelly, M., Wild, S.M.: Derivative-free optimization methods. Acta Numer. **28**, 287–404 (2019)
23. Li, J., Balasubramanian, K., Ma, S.: Zeroth-order optimization on Riemannian manifolds. https://arxiv.org/abs/2003.11238 (2020)
24. Liuzzi, G., Lucidi, S., Sciandrone, M.: Sequential penalty derivative-free methods for nonlinear constrained optimization. SIAM J. Optim. **20**, 2614–2635 (2010)
25. Lucidi, S., Sciandrone, M.: A derivative-free algorithm for bound constrained optimization. Comput. Optim. Appl. **21**, 119–142 (2002)
26. Lucidi, S., Sciandrone, M.: On the global convergence of derivative-free methods for unconstrained optimization. SIAM J. Optim. **13**, 97–116 (2002)
27. Moré, J.J., Wild, S.M.: Benchmarking derivative-free optimization algorithms. SIAM J. Optim. **20**, 172–191 (2009)
28. Nesterov, Y., Spokoiny, V.: Random gradient-free minimization of convex functions. Found. Comput. Math. **17**, 527–566 (2017)
29. Vandereycken, B.: Riemannian and multilevel optimization for rank-constrained matrix problems. PhD Thesis. Department of Computer Science, KU Leuven. http://www.unige.ch/math/vandereycken/papers/phd_Vandereycken.pdf (2010). Accessed 10 Feb 2022
30. Vicente, L.N.: Worst case complexity of direct search. EURO J. Comput. Optim. **1**, 143–153 (2013)
31. Yao, T.-T., Zhao, Z., Bai, Z.-J., Jin, X.-Q.: A Riemannian derivative-free Polak–Ribiére–Polyak method for tangent vector field. Numer. Algorithms **86**, 325–355 (2021)