Retraction based Direct Search Methods for Derivative Free Riemannian Optimization

Vyacheslav Kungurtsev* Francesco Rinaldi† Damiano Zeffiro‡

February 22, 2022

Abstract

Direct search methods represent a robust and reliable class of algorithms for solving black-box optimization problems. In this paper, we explore the application of those strategies to Riemannian optimization, wherein minimization is to be performed with respect to variables restricted to lie on a manifold. More specifically, we consider classic and line search extrapolated variants of direct search, and, by making use of retractions, we devise tailored strategies for the minimization of both smooth and nonsmooth functions. As such we analyze, for the first time in the literature, a class of retraction based algorithms for minimizing nonsmooth objectives on a Riemannian manifold without having access to (sub)derivatives. Along with convergence guarantees we provide a set of numerical performance illustrations on a standard set of problems.
Keywords: Direct search, derivative free optimization, Riemannian manifold, retraction.
AMS subject classifications: 90C06, 90C30, 90C56.

1 Introduction

Riemannian optimization, or solving minimization problems constrained on a Riemannian manifold embedded in an Euclidean space, is an important and active area of research considering the numerous problems in data science, robotics, and other settings wherein there is an important 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. In cases of dynamics subject to significant epistemic uncertainty and the necessity of performing a simulation to compute a function evaluation, derivatives may be unavailable. This paper presents the introduction of a classic set of DFO algorithms, namely direct search, to the case of Riemannian optimization. For classic references of Riemannian optimization and DFO, see, e.g., [1] and [5, 10, 20], respectively.

Formally, let $\mathcal{M}$ be a smooth manifold embedded in $\mathbb{R}^n$. We are interested here in the problem

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

with f continuous and bounded below. We consider both the case of $f(x)$ being continuously differentiable, as well as the more general nonsmooth case.

Direct search methods (see, e.g., [19] and references therein) belong to the class of algorithms that are mesh based, rather than model based. This distinction presents a binary taxonomy of DFO algorithms: on the one hand we have those based on approximating gradient information using function evaluations and constructing approximate local models, while on the other hand we have those based on sampling a

*Department of Computer Science, Czech Technical University, Czech Republic. Research supported by the OP VVV project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics" (kunguvya@fel.cvut.cz)
†Dipartimento di Matematica "Tullio Levi-Civita", Università di Padova, Italy (rinaldi@math.unipd.it)
‡Dipartimento di Matematica "Tullio Levi-Civita", Università di Padova, Italy (damiano.zeffiro@math.unipd.it)

pre-defined grid of points for the next iteration. Thus direct search is particularly suitable for black box cases wherein it is unknown the degree to which any model would have much veracity.

To the best of our knowledge, thorough studies of DFO on Riemannian manifolds have only been carried out recently in the literature. In [21], the authors focus on a model based method using a two point function approximation for the gradient. The paper [27] presents a specialized Polak-Ribiéere-Polyak procedure for finding a zero of a tangent vector field on a Riemannian manifold. In [12], the author focuses on a specific class of manifolds (reductive homogeneous spaces, including several matrix manifolds) where, thanks to the properties of exponential maps, a straightforward extension of mesh adaptive direct search methods (see, e.g., [4, 5]) and probabilistic direct search strategies [14] is possible. Some DFO methods and nonsmooth problems on Riemannian manifolds without convergence analysis can be found in [16] and references therein.

Thus our paper presents the first analysis of retraction based direct search strategies on Riemannian manifolds, and the first analysis of a DFO algorithm for minimizing nonsmooth objectives in Riemannian optimization. In particular, we first adapt, thanks to the use of retractions, a classic direct search scheme (see, e.g., [10, 19]) and a linesearch based scheme (see, e.g., [11, 22, 23, 24] for further details on this class of methods) to deal with the minimization of a given smooth function over a manifold. Then, inspired by the ideas in [13], we extend the two proposed strategies to the nonsmooth case.

The remainder of this paper is as follows. In Section 2, we present some definitions. In Section 3, we present and prove convergence for a direct search method applicable for continuously differentiable f. In Section 4, we consider the case of f not being continuously differentiable, and only Lipschitz continuous. We present some numerical results in Section 5 and conclude in Section 6.

2 Definitions and notation

We now introduce some notation for the formalism we use in this article. We refer the reader to, e.g., [1, 7] for an overview of the relevant background.

Let $T\mathcal{M}$ be the tangent manifold and for $x \in \mathcal{M}$ let $T_x\mathcal{M}$ be the tangent bundle to $\mathcal{M}$ in x. We assume that $\mathcal{M}$ is a Riemannian manifold, i.e., for x in $\mathcal{M}$, we have a scalar product $\langle \cdot, \cdot \rangle_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$ smoothly dependent from x. Let $\mathrm{dist}(\cdot, \cdot)$ be the distance induced by the scalar product, so that for $x, y \in \mathcal{M}$ we have that $\mathrm{dist}(x, y)$ is the length of the shortest geodesic connecting x and y. Furthermore, let $\nabla_{\mathcal{M}}$ be the Levi-Cita connection for $\mathcal{M}$, and $\Gamma : T\mathcal{M} \times \mathcal{M} \to T\mathcal{M}$ the parallel transport with respect to $\nabla_{\mathcal{M}}$, with $\Gamma_x^y(v) \in T_y\mathcal{M}$ transport of the vector $v \in T_x\mathcal{M}$. We define P_x 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.

We write $\{a_k\}$ as a shorthand for $\{a_k\}_{k \in I}$ when the index set I is clear from the context. We also use the shorthand notations $T_k\mathcal{M}, \mathsf{P}_k, \langle \cdot, \cdot \rangle_k, \| \cdot \|_k$, Γ_i^j for $T_{x_k}\mathcal{M}, \mathsf{P}_{x_k}, \langle \cdot, \cdot \rangle_{x_k}, \| \cdot \|_{x_k}$ and $\Gamma_{x_i}^{x_j}$.

We define the distance dist^* between vectors in different tangent spaces in a standard way using parallel transport (see for instance [6]): for $x, y \in M$, $v \in T_xM$ and $w \in T_yM$,

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

and for a sequence $\{(y_k, v_k)\}$ in $T\mathcal{M}$ we write $v_k \to v$ if $y_k \to y$ in $\mathcal{M}$ and $\mathrm{dist}^*(v_k, v) \to 0$.

As it is common in the Riemannian optimization literature (see, e.g., [2]), to define our tentative descent directions we use a retraction $R : T\mathcal{M} \to \mathcal{M}$. We assume $R \in C^1(T\mathcal{M}, \mathcal{M})$, with

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

(true in any compact subset of $T\mathcal{M}$ given the C^1 regularity of R, without any further assumptions), and that the sufficient decrease property holds: for any L-Lipschitz smooth f,

$$f(R(x, d)) \leq f(x) + \langle \mathrm{grad} f(x), d \rangle + L\|d\|^2. \tag{4}$$

3 Smooth optimization problems

In this section, we consider solving (1) with the objective satisfying $f \in C^1(\mathcal{M})$. Recall that we can define the Riemannian gradient as

$$\mathrm{grad} f(x) = \mathsf{P}_x(\nabla f(x)), \tag{5}$$

for given $x \in \mathcal{M}$.

3.1 Preliminaries

First, we assume that the objective function f has a Lipschitz continuous gradient on the manifold.

Assumption 3.1. *There exists $L_f > 0$ such that for all $x \in \mathcal{M}$*

$$\mathrm{dist}^*(\mathrm{grad} f(x), \mathrm{grad} f(y)) = \|\Gamma_x^y \mathrm{grad} f(x) - \mathrm{grad} f(y)\| \leq L_f \|\mathrm{grad} f(x)\|, \tag{6}$$

Like in the unconstrained case, the Lipschitz gradient property implies the standard descent property.

Proposition 3.1. *Assume that M is compact and R is a C^2 retraction. If condition* (6) *holds, then the sufficient decrease property* (4) *holds for some constant $L > 0$.*

The proof can be found in the appendix. An analogous property, but under the stronger assumption that f has Lipschitz gradient as a function in $\mathbb{R}^n$, is proved in [8].
Another assumption we make in this context is that the gradient norm is globally bounded.

Assumption 3.2. *There exists $M_f > 0$ such that,*

$$\|\mathrm{grad} f(x)\| \leq M_f, \tag{7}$$

for every $x \in \mathcal{M}$.

For each of the algorithms in this section, we further assume that, at each iteration k, we have a positive spanning basis $\{p_k^j\}_{j \in [1:K]}$ of the tangent space $T_{x_k} M$ of the iterate x_k (further details on how to get a positive spanning basis can be found, e.g., in [10]). More specifically, we assume that the basis stays bounded and does not become degenerate during the algorithm, that is,

Assumption 3.3. *There exists $B > 0$ such that*

$$\max_{j \in [1:K]} \|p_k^j\| \leq B, \tag{8}$$

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

$$\max_{i \in [1:K]} \langle r, p_k^j \rangle \geq \tau \|r\|, \tag{9}$$

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

3.2 Direct search algorithm

We present here our Riemannian Direct Search method based on Spanning Bases (RDS-SB) for smooth objectives as Algorithm 1.

This procedure resembles the standard direct search algorithm for unconstrained derivative free optimization (see, e.g., [10, 19]) with two significant modifications. First, at every iteration a positive spanning basis 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 $T_{x_k^j} \mathcal{M}$ to the manifold.

Algorithm 1 RDS-SB

Input: $x_0 \in \mathcal{M}$, $\gamma_1 \in (0,1)$, $\gamma_2 \geq 1$, $\alpha_0 > 0$, $\rho > 0$
for $k = 0, 1, ...$ **do**
  Compute a positive spanning basis $\{p_k^j\}_{j=1:K}$ of $T_k\mathcal{M}$
  for $j = 1, ..., K$ **do**
    Let $x_k^j = R(x_k, \alpha_k p_k^j)$
    if $f(x_k^j) \leq f(x_k) - \rho\alpha_k^2$ **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 $f(x_k^j) > f(x_k) - \rho\alpha_k^2$ 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

3.3 Convergence analysis

Now we show asymptotic global convergence of the method. Using a similar structure of reasoning as in standard convergence derivations for direct search, we prove that the gradient evaluated at iterates associated with unsuccessful steps must converge to zero, and extend the property to the remaining iterates, using the Lipschitz continuity of the gradient.

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

Lemma 3.1. *If* $f(R(x_k, \alpha_k p_k^j)) > f(x_k) - \gamma\alpha_k^2$, *then*

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

Proof. To start with, we have

$$\begin{aligned} f(x_k) - \gamma\alpha_k^2 < f(R(x, \alpha_k p_k^j)) \leq f(x_k) + \alpha_k \langle \mathrm{grad} f(x_k), p_k^j \rangle + L\alpha_k^2 \|p_k^j\|^2 \\ \leq f(x_k) + \alpha_k \langle \mathrm{grad} f(x_k), p_k^j \rangle + L\alpha_k^2 B^2 \,, \end{aligned} \tag{11}$$

where we used (4) in the second inequality, and (8) in the third one. The above inequality can be rewritten as

$$\alpha_k \langle \mathrm{grad} f(x_k), p_k^j \rangle + \alpha_k^2 (LB^2 + \gamma) > 0. \tag{12}$$

Given that $\alpha_k > 0$, the above is true iff

$$\alpha_k > -\frac{\langle \mathrm{grad} f(x_k), p_k^j \rangle}{(LB^2 + \gamma)} \,, \tag{13}$$

which rearranged gives the thesis. □

From this we can infer a bound on the gradient with respect to the stepsize.

Lemma 3.2. *If iteration* k *is unsuccessful, then*

$$\|\mathrm{grad} f(x_k)\| \leq \frac{\alpha_k (2LB^2 + \gamma)}{\tau} \,. \tag{14}$$

Proof. If iteration k is unsuccessful, equation (10) must hold for every $j \in [1:K]$. We obtain the thesis by applying the positive spanning property (9) in the RHS:

$$\alpha_k(LB^2+\gamma) > \max_{j\in[1:K]} -\langle \mathrm{grad} f(x_k), p_k^j\rangle \geq \tau \|\mathrm{grad} f(x_k)\| \,. \tag{15}$$

□

Finally, we are able to show convergence of the gradient norm using the lemmas above and appropriate arguments regarding the step sizes.

Theorem 3.1. *For the sequence $\{x_k\}$ generated by Algorithm 1 we have*

$$\lim_{k\to\infty} \|\mathrm{grad} f(x_k)\| = 0 \,. \tag{16}$$

Proof. To start with, clearly $\alpha_k \to 0$ since the objective is bounded below, $\{f(x_k)\}$ is non increasing 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

$$\|\mathrm{grad} f(x_k)\| \leq \varepsilon \left(\frac{(2LB^2+\gamma)}{\tau} + L_f L_r B \frac{\gamma_2}{\gamma_2 - 1} \right) , \tag{17}$$

which clearly implies the thesis given that ε is arbitrary.
First, (17) is satisfied for $k \geq \bar{k}$ if the step k is unsuccessful by Lemma 3.2:

$$\|\mathrm{grad} f(x_k)\| \leq \frac{\alpha_k(2LB^2+\gamma)}{\tau} \leq \frac{\varepsilon(2LB^2+\gamma)}{\tau} \,, \tag{18}$$

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. 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

$$\sum_{i=0}^{j-1} \alpha_{k+i} \leq \sum_{i=0}^{j-1} \varepsilon \gamma_2^{i-j+1} \leq \varepsilon \sum_{h=0}^{\infty} \gamma_2^{-h} = \varepsilon \frac{\gamma_2}{\gamma_2 - 1} \,. \tag{19}$$

Then

$$\begin{aligned} \mathrm{dist}(x_k, x_{k+j}) &\leq \sum_{i=0}^{j-1} \mathrm{dist}(x_{k+i}, x_{k+i+1}) = \sum_{i=0}^{j-1} \mathrm{dist}(x_{k+i}, R(x_{k+i}, \alpha_{k+i} p_{k+i}^{j(k+i)})) \\ &\leq \sum_{i=0}^{j-1} L_r \alpha_{k+i} B \leq L_r B \varepsilon \frac{\gamma_2}{\gamma_2 - 1} \,. \end{aligned} \tag{20}$$

where we used (3) together with (8) in the second inequality, and (19) in the third one.
In turn,

$$\begin{aligned} \|\mathrm{grad} f(x_k)\| &\leq \mathrm{dist}^*(\mathrm{grad} f(x_k), \mathrm{grad} f(x_{k+j})) + \|\mathrm{grad} f(x_{k+j})\| \\ &\leq L_f \,\mathrm{dist}(x_k, x_{k+j}) + \frac{\varepsilon(2LB^2+\gamma)}{\tau} \leq \varepsilon \left(\frac{2LB^2+\gamma}{\tau} + L_f L_r B \frac{\gamma_2}{\gamma_2 - 1} \right) , \end{aligned} \tag{21}$$

where we used (6) and (18) with $k+j$ instead of k for the first and second summand respectively in the second inequality, and (20) in the last one. □

3.4 Incorporating an extrapolation linesearch

The works [23, 24] introduced the use of an extrapolating line search 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. We found that the same technique can be applied in the Riemannian setting to good effect. We present here our Riemannian Direct Search with Extrapolation method based on Spanning Bases (RDSE-SB) for smooth objectives. The scheme is presented in detail as Algorithm 2. As we can easily see, the method uses a specific stepsize for each direction in the positive spanning basis, so that instead of α_k we have 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 of the objective is obtained.

When analyzing the RDSE-SB method, we assume that the following continuity condition holds.

Assumption 3.4. *For every $l, m \in \mathbb{N}$, $j \in [1:K]$, there exists a constant $L_\Gamma > 0$ such that*

$$\operatorname{dist}^*(p_l^j, p_m^j) \leq L_\Gamma \operatorname{dist}(x_l, x_m)\,. \tag{22}$$

We refer the reader to [24] for a slightly weaker continuity condition in an Euclidean setting.

Algorithm 2 RDSE-SB

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

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

if $f(R(x_k, \alpha d)) > f(x) - \gamma\alpha^2$ **then**
  $(0, \gamma_1\alpha)$
end if
while $f(R(x_k, \alpha d)) < f(x) - \gamma\alpha^2$ **do**
  Set $\alpha = \gamma_2\alpha$
end while
Return $(\alpha/\gamma_2, \alpha/\gamma_2)$

We now proceed to prove the asymptotic convergence of this method.

Lemma 3.3. *We have, at every iteration k, that the following inequality holds:*

$$-\langle \operatorname{grad} f(x_k), p_k^{j(k)} \rangle < \tilde{\alpha}_{k+1}^{j(k)} \frac{\gamma_2}{\gamma_1}(2LB^2 + \gamma). \tag{23}$$

Proof. It is immediate to check that we must always have

$$f(R(x_k, \Delta_k p_k^{j(k)})) > f(x_k) - \gamma\Delta_k^2, \tag{24}$$

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

$$-\langle \operatorname{grad} f(x_k), p_k^{j(k)} \rangle < \Delta_k(2LB^2 + \gamma) \leq \tilde{\alpha}_{k+1}^{j(k)} \frac{\gamma_2}{\gamma 1}(2LB^2 + \gamma)\,, \tag{25}$$

where we used Lemma 3.1 in the first inequality. □

Theorem 3.2. *For $\{x_k\}$ generated by Algorithm 2, we have*

$$\lim_{k\to\infty} \|\mathrm{grad} f(x_k)\| \to 0\,. \tag{26}$$

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 3.3 we have

$$-\langle \mathrm{grad} f(x_k), p_k^{j(k)}\rangle < \bar\alpha_k c_1\,, \tag{27}$$

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

$$\begin{aligned} &-\langle \mathrm{grad} f(x_k), p_k^i\rangle \leq -\langle \mathrm{grad} f(x_{k+h}), p_{k+h}^i\rangle + |\langle \mathrm{grad} f(x_{k+h}), p_{k+h}^i\rangle - \langle \mathrm{grad} f(x_k), p_k^i\rangle| \\ \leq& c_1\bar\alpha_{k+h} + |\langle \mathrm{grad} f(x_{k+h}), p_{k+h}^i\rangle - \langle \mathrm{grad} f(x_k), p_k^i\rangle|\,, \end{aligned} \tag{28}$$

for $h \leq K$ such that $k+h = j(i)$, and where in the second inequality we used (27) with $k+h$ instead of k. For the second summand appearing in the RHS of (28), we can write the following bound

$$\begin{aligned} &|\langle \mathrm{grad} f(x_{k+h}), p_{k+h}^i\rangle - \langle \mathrm{grad} f(x_k), p_k^i\rangle| = |\langle \mathrm{grad} f(x_{k+h}), p_{k+h}^i\rangle - \langle \Gamma_k^{k+h}\mathrm{grad} f(x_k), \Gamma_k^{k+h} p_k^i\rangle| \\ \leq& |\langle \mathrm{grad} f(x_{k+h}) - \Gamma_k^{k+h}\mathrm{grad} f(x_k), p_{k+h}^i\rangle| + |\langle \Gamma_k^{k+h}\mathrm{grad} f(x_k), p_{k+h}^i - \Gamma_k^{k+h} p_k^i\rangle| \\ &+|\langle \mathrm{grad} f(x_{k+h}) - \Gamma_k^{k+h}\mathrm{grad} f(x_k), p_{k+h}^i - \Gamma_k^{k+h} p_k^i\rangle| \\ \leq& L_f \,\mathrm{dist}(x_k, x_{k+h})\|p_{k+h}^i\| + L_\Gamma \|\mathrm{grad} f(x_k)\|\,\mathrm{dist}(x_{k+h}, x_k) + L_f L_\Gamma\,\mathrm{dist}(x_k, x_{k+h})^2 \\ \leq& (L_f B + L_\Gamma M_f + L_f L_\Gamma\,\mathrm{dist}(x_{k+h}, x_k))\,\mathrm{dist}(x_{k+h}, x_k)\,, \end{aligned} \tag{29}$$

where in the second inequality we used the Cauchy-Schwartz inequality together with the Assumptions on the Lipschitz property of the iterates (6) and (22), while in the third inequality we used conditions (8) and (7).
We can now bound $\mathrm{dist}(x_k, x_{k+h})$ as follows

$$\begin{aligned} &\mathrm{dist}(x_{k+h}, x_k) \leq \sum_{l=0}^{h-1} \mathrm{dist}(x_{k+l+1}, x_{k+l}) \\ =& \sum_{l=0}^{h-1} \mathrm{dist}(x_{k+l}, R(x_{k+l}, \bar\alpha_{k+l} p_{k+l}^{j(k+l)})) \leq \sum_{l=0}^{h-1} L_r \bar\alpha_{k+l}\|p_{k+l}^{j(k+l)}\| \\ \leq& BL_r \sum_{l=0}^{h-1} \bar\alpha_{k+l} \leq hBL_r \max_{l\in[0:h-1]} \bar\alpha_{k+l} \\ \leq& KBL_r \max_{l\in[0:K]} \bar\alpha_{k+l}\,, \end{aligned} \tag{30}$$

where we used (3) in the second inequality, (8) in the third one, and $h \leq K$ in the last one.
Now let $\Delta_k = \max_{l\in[0:K]} \bar\alpha_{k+l}$, so that in particular $\Delta_k \to 0$. We apply (30) to the RHS of (29) and obtain

$$|\langle \mathrm{grad} f(x_{k+h}), p_{k+h}^i\rangle - \langle \mathrm{grad} f(x_k), p_k^i\rangle| \leq (L_f B + L_\Gamma M_f + L_f L_\Gamma c_2\Delta_k)c_2\Delta_k \to 0\,, \tag{31}$$

for $k\to\infty$ and $c_2 = KBL_r$. Finally, for every $i \in [1:K]$

$$-\langle \mathrm{grad} f(x_k), p_k^i\rangle \leq c_1\bar\alpha_{k+h} + (L_f B + L_\Gamma M_f + L_f L_\Gamma c_2\Delta_k)c_2\Delta_k \to 0\,, \tag{32}$$

and the thesis follows after observing that, by (9),

$$\|\mathrm{grad} f(x_k)\| \leq \frac{1}{\tau} \max_{i\in[1:K]} -\langle \mathrm{grad} f(x_k), p_k^i\rangle \to 0\,, \tag{33}$$

where the convergence of the gradient norm to zero is a consequence of (32). □

4 Nonsmooth objectives

Now we proceed to present and study direct search methods in the context where f is Lipschitz continuous and bounded from below, but not necessarily continuously differentiable. The algorithms we devise are built around the ideas given in [13], 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, we first need to define the Clarke directional derivative for a point $x \in M$. The standard approach is to write the function in coordinate charts and take the standard Clarke derivative in an Euclidean space (see, e.g., [17] and [18]). Formally, given a chart (φ, U) at $x \in M$ and $v \in T_x M$, we define

$$f^\circ(x, v) = \tilde{f}(\varphi(x), d\varphi(x)v)', \tag{34}$$

for $\tilde{f}(y) = f(\varphi^{-1}(y))$. The following lemma shows the relationship between definition (34) and a directional derivative like object defined with retractions.

Lemma 4.1. *If $(y_k, q_k) \to (x, d)$ and $t_k \to 0$,*

$$f^\circ(x, d) \geq \limsup_{k\to\infty} \frac{f(R(y_k, t_k q_k)) - f(y_k)}{t_k}. \tag{35}$$

The proof is rather technical and we defer it to the appendix.

4.2 Refining subsequences

We now adapt the definition of refining subsequence used in the analysis of direct search methods (see, e.g., [3, 13]) to the Riemannian setting. Let (x_k, d_k) be a sequence in $T\mathcal{M}$.

Definition 4.1. *We say that the subsequence $\{x_{i(k)}\}$ is refining if $x_{i(k)} \to x$, and if for every $d \in T_x\mathcal{M}$ with $\|d\|_x = 1$ there is a further subsequence $\{j(i(k))\}$ such that*

$$\lim_{k\to\infty} d_{j(i(k))} = d. \tag{36}$$

We now give a sufficient condition for a sequence to be refining.

Proposition 4.1. *If $x_{i(k)} \to x^*$, $\bar{d}_{i(k)}$ is dense in the unit sphere, and $d_{i(k)} = \mathsf{P}_k(\bar{d}_{i(k)})/\|\mathsf{P}_k(\bar{d}_{i(k)})\|_k$ for $\mathsf{P}_k(\bar{d}_{i(k)}) \neq 0$ and $d_{i(k)} = 0$ otherwise, then it holds that the subsequence $\{x_{i(k)}\}$ is refining.*

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

$$\lim_{k\to\infty} d_{j(i(k))} = \lim_{k\to\infty} \frac{\mathsf{P}_k(\bar{d}_{j(i(k))})}{\|\mathsf{P}_k(\bar{d}_{j(i(k))})\|_k} = \frac{\mathsf{P}_{x^*}(\bar{d})}{\|\mathsf{P}_{x^*}(\bar{d})\|_{x^*}} = \frac{\bar{d}}{\|\bar{d}\|_{x^*}} = d, \tag{37}$$

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

We present here our Riemannian Direct Search method based on Dense Directions (RDS-DD) for nonsmooth objectives. The scheme is presented in detail as Algorithm 4. The algorithm performs three simple steps at an iteration k. First, a given search direction is suitably projected onto 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

Algorithm 4 RDS-DD

Input: $x_0 \in \mathbb{R}^n$, $\alpha_0 > 0$, $\gamma > 0, \gamma_1 \in (0,1), \gamma_2 \geq 1$, $\{\bar{d}_k\}$ dense in $S(0,1)$
for $k = 0, 1, \ldots$ **do**
  Let $d_k = \mathsf{P}_k(\bar{d}_k)/\|\mathsf{P}_k(\bar{d}_k)\|_k$ if $\mathsf{P}_k(\bar{d}_k) \neq 0$, 0 otherwise
  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

of the objective function is satisfied (and the stepsize is expanded), otherwise the iterate stays the same (and the stepsize is reduced).

Thanks to the theoretical tools previously introduced, we can easily prove that a suitable subsequence of unsuccessful iterations of the RDS-DD method converges to a Clarke stationary point.

Theorem 4.1. *Let $\{x_k\}$ be generated by Algorithm 4. If $\{x_{i(k)}\}$ is refining, with $x_{i(k)} \to x^*$, and $i(k)$ is an unsuccessful iteration for every $k \in \mathbb{N} \cup \{0\}$, then x^* is Clarke stationary.*

Proof. Clearly 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 $i(k)$

$$f(R(x_{i(k)}, \alpha_{i(k)} d_{i(k)})) - f(x_{i(k)}) > -\gamma\alpha_{i(k)}^2 \,. \tag{38}$$

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

$$\limsup_{k\to\infty} \frac{f(R(y_k, t_k q_k)) - f(y_k)}{t_k} \geq \limsup_{k\to\infty} -\gamma\alpha_{i(k)} = 0 \,, \tag{39}$$

thanks to (38), and by applying Lemma 4.1 we get

$$f^\circ(x^*, d) \geq \limsup_{k\to\infty} \frac{f(R(y_k, t_k q_k)) - f(y_k)}{t_k} \geq 0 \,, \tag{40}$$

which implies the thesis since d is arbitrary. □

4.4 Direct search with extrapolation for nonsmooth objectives

We present here our Riemannian Direct Search method with Extrapolation based on Dense Directions (RDSE-DD) for nonsmooth objectives. The detailed scheme is given in Algorithm 5. As we can easily see, 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 a linesearch is performed using Algorithm 3 to hopefully obtain a new point that guarantees a sufficient decrease.

Algorithm 5 RDSE-DD

Input: $x_0 \in \mathbb{R}^n$, $\alpha_0 > 0$, $\gamma > 0, \gamma_1 \in (0,1), \gamma_2 \geq 1$, $\{\bar{d}_k\}$ dense in $S(0,1)$.
for $k = 0, 1, \ldots$ **do**
  Let $d_k = \mathsf{P}_k(\bar{d}_k)/\|\mathsf{P}_k(\bar{d}_k)\|_k$ if $\mathsf{P}_k(\bar{d}_k) \neq 0$, 0 otherwise.
  Compute $\alpha_k, \tilde{\alpha}_{k+1}$ with **Linesearchprocedure**$(\tilde{\alpha}_k, x_k, d_k, \gamma, \gamma_1, \gamma_2)$
  Set $x_{k+1} = R(x_k, \alpha_k d_k)$
end for

Once again, by exploiting the theoretical tools previously introduced, we can straightforwardly prove that a suitable subsequence of the RDSE-DD iterations converges to a Clarke stationary point. It is interesting to notice that, thanks to the use of the linesearch strategy, we are not restricted to considering unsuccessful iterations this time.

Theorem 4.2. *Let $\{x_k\}$ be generated by Algorithm 5. If $\{x_{i(k)}\}$ is refining, with $x_{i(k)} \to x^*$, then x^* is Clarke stationary.*

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

$$f(R(x_k, \beta_k d_k)) - f(x_k) > -\gamma\beta_k^2 \,. \tag{41}$$

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

5 Numerical results

We now report the results of some numerical experiments of the algorithms described in this paper on a set of simple but illustrative example problems. The comparison among the algorithms is carried out by using data and performance profiles [25]. 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

$$f(x_k) \leq f_L + \tau(f(x_0) - f_L) \,, \tag{42}$$

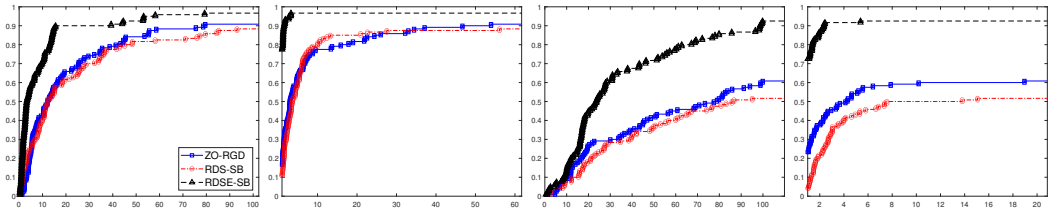
where $0 < \tau < 1$ and f_L is the best objective function value achieved by any solver on problem p. Then, performance and data profiles of solver s are the following functions

$$\begin{aligned} \rho_s(\alpha) &= \frac{1}{|P|}\left|\left\{p \in P : \frac{t_{p,s}}{\min\{t_{p,s'} : s' \in S\}} \leq \alpha\right\}\right|, \\ d_s(\kappa) &= \frac{1}{|P|}\left|\{p \in P : t_{p,s} \leq \kappa(n_p+1)\}\right| , \end{aligned}$$

where n_p is the dimension of problem p.
We used a budget of $100(n_p + 1)$ function evaluations in all cases and two different precisions for the condition (42), that is $\tau \in \{10^{-1}, 10^{-3}\}$. We consider randomly generated instances of well-known optimization problems over manifolds from [1, 7, 16]. A brief description of those problems as well as the details of our implementation can be found in the appendix (see Sections 7.2, 7.3 and 7.4). The size of the ambient space for the instances varies from 2 to 200. We would finally like to highlight that, in Section 7.5, we report further detailed numerical results, splitting the problems 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 Figure 1, we include the results related to 8 smooth instances of problem (1) from [1, 7], each with 15 different problem dimensions (from 2 to 200), for a total number of 60 tested instances. We compared our methods, that is RDS-SB and RDSE-SB, with the zeroth order gradient descent (ZO-RGD, [21, 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. By taking a look at the detailed results in Section 7.5, we can also see how the gap between RDSE-SB and the other two algorithms gets larger as the problem dimension grows.



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

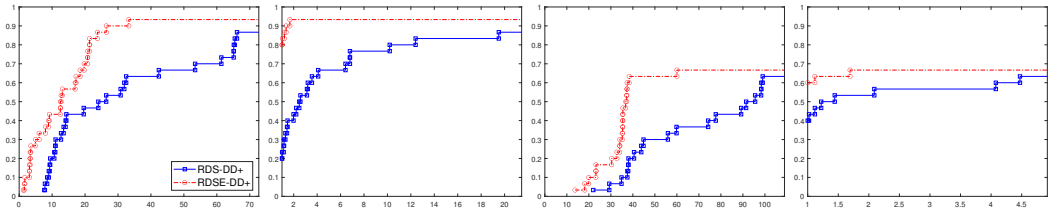
Figure 1: Smooth case: results for all the instances

5.2 Nonsmooth problems

We finally report a preliminary comparison between a direct search strategy and a linesearch strategy on two nonsmooth instances of (1) from [16], each with 15 different problem sizes (from 2 to 200), thus getting a total number of 30 tested instances.

In the direct search strategy (RDS-DD+), we apply the RDS-SB method until $\alpha_{k+1} \leq \alpha_\epsilon$, at which point we switch to the nonsmooth version RDS-DD. Analogously, in the linesearch strategy (RDSE-DD+), we apply the RDSE-SB method until $\max_{j\in[1:K]} \tilde{\alpha}^j_{k+1} \leq \alpha_\epsilon$, at which point we switch to the nonsmooth version RDSE-DD. Both strategies use a threshold parameter $\alpha_\epsilon > 0$ to switch from the smooth to the nonsmooth DFO algorithm. We refer the reader to [13] and references therein for other direct search strategies combining coordinate and dense directions.

We report, in Figure 2, the comparison between the two considered strategies. As in the smooth case, the linesearch based strategy outperforms the simple direct search one. By taking a look at the detailed results in Section 7.5, we can once again see 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., $\tau = 10^{-3}$

Figure 2: Nonsmooth case: results for all the instances

6 Conclusion

In this paper, we presented direct search algorithms with and without an extrapolation linesearch for minimizing functions over a Riemannian manifold. We 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. We found also 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.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*, Princeton University Press, Princeton, 2009.

[2] P.-A. Absil and J. Malick, *Projection-like retractions on matrix manifolds*, SIAM J. Optim., 22 (2012), pp. 135–158.

[3] C. Audet and J. E. Dennis Jr, *Analysis of generalized pattern searches*, SIAM J. Optim., 13 (2002), pp. 889–903.

[4] C. Audet and J. E. Dennis Jr, *Mesh adaptive direct search algorithms for constrained optimization*, SIAM J. Optim., 17 (2006), pp. 188–217.

[5] C. Audet and W. Hare, *Derivative-free and blackbox optimization*, vol. 2 of Operations Research,Financ.Engin., Springer, 2017.

[6] D. Azagra, J. Ferrera, and F. López-Mesas, *Nonsmooth analysis and hamilton–jacobi equations on riemannian manifolds*, J. Funct. Anal., 220 (2005), pp. 304–361.

[7] N. Boumal, *An introduction to optimization on smooth manifolds*, 2022, `http://sma.epfl.ch/~nboumal/book/index.html` (accessed 2022-02-10).

[8] N. Boumal, P.-A. Absil, and C. Cartis, *Global rates of convergence for nonconvex optimization on manifolds*, IMA J. Numer. Anal., 39 (2019), pp. 1–33.

[9] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, *Manopt, a Matlab toolbox for optimization on manifolds*, Journal of Machine Learning Research, 15 (2014), pp. 1455–1459.

[10] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to derivative-free optimization*, MOS SIAM Ser. Optim., SIAM, Philadelphia, 2009.

[11] A. Cristofari and F. Rinaldi, *A derivative-free method for structured optimization problems*, SIAM J. Optim., 31 (2021), pp. 1079–1107.

[12] D. W. Dreisigmeyer, *Direct search methods on reductive homogeneous spaces*, J. Optim. Theory Appl., 176 (2018), pp. 585–604.

[13] G. Fasano, G. Liuzzi, S. Lucidi, and F. Rinaldi, *A linesearch-based derivative-free approach for nonsmooth constrained optimization*, SIAM J. Optim., 24 (2014), pp. 959–992.

[14] S. Gratton, C. W. Royer, L. N. Vicente, and Z. Zhang, *Direct search based on probabilistic descent*, SIAM J. Optim., 25 (2015), pp. 1515–1541.

[15] R. Hosseini and S. Sra, *Matrix manifold optimization for gaussian mixtures*, Advances in Neural Information Processing Systems, 28 (2015), pp. 910–918.

[16] S. Hosseini, B. S. Mordukhovich, and A. Uschmajew, *Nonsmooth optimization and its applications*, International Series of Numerical Mathematics, Springer International Publishing, 2019.

[17] S. Hosseini and M. Pouryayevali, *Nonsmooth optimization techniques on riemannian manifolds*, J. Optim. Theory Appl., 158 (2013), pp. 328–342.

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

[19] T. G. Kolda, R. M. Lewis, and V. Torczon, *Optimization by direct search: New perspectives on some classical and modern methods*, SIAM Rev., 45 (2003), pp. 385–482.

[20] J. Larson, M. Menickelly, and S. M. Wild, *Derivative-free optimization methods*, Acta Numer., 28 (2019), pp. 287–404.

[21] J. Li, K. Balasubramanian, and S. Ma, *Zeroth-order optimization on riemannian manifolds*, (2020), https://arxiv.org/abs/2003.11238.

[22] G. Liuzzi, S. Lucidi, and M. Sciandrone, *Sequential penalty derivative-free methods for nonlinear constrained optimization*, SIAM J. Optim., 20 (2010), pp. 2614–2635.

[23] S. Lucidi and M. Sciandrone, *A derivative-free algorithm for bound constrained optimization*, Comput. Optim. Appl., 21 (2002), pp. 119–142.

[24] S. Lucidi and M. Sciandrone, *On the global convergence of derivative-free methods for unconstrained optimization*, SIAM J. Optim., 13 (2002), pp. 97–116.

[25] J. J. Moré and S. M. Wild, *Benchmarking derivative-free optimization algorithms*, SIAM J. Optim., 20 (2009), pp. 172–191.

[26] B. Vandereycken, *Riemannian and multilevel optimization for rank-constrained matrix problems*. PhD thesis, Department of Computer Science, KU Leuven, 2010, http://www.unige.ch/math/vandereycken/papers/phd_Vandereycken.pdf (accessed 2022-02-10).

[27] T.-T. Yao, Z. Zhao, Z.-J. Bai, and X.-Q. Jin, *A riemannian derivative-free polak–ribiére–polyak method for tangent vector field*, Numerical Algorithms, 86 (2021), pp. 325–355.

7 Appendix

7.1 Proofs

In order to prove Proposition 3.1 we first need the following lemma.

Lemma 7.1. *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*

$$h^\circ(\tilde{y}, \tilde{v}) \geq \limsup_{k\to\infty} \frac{h(\tilde{y}_k + t_k\tilde{v}_k) - h(\tilde{y}_k)}{t_k}\,. \tag{43}$$

Proof. We have

$$|h(\tilde{y}_k + t_k\tilde{v}_k) - h(\tilde{y}_k + t_k\tilde{v})| \leq t_k L_h \|\tilde{v} - \tilde{v}_k\| = o(t_k)\,, \tag{44}$$

with L_h the Lipschitz constant of h. Then

$$\begin{aligned} &\limsup_{k\to\infty} \frac{h(\tilde{y}_k + t_k\tilde{v}_k) - h(\tilde{y}_k)}{t_k} = \limsup_{k\to\infty} \frac{h(\tilde{y}_k + t_k\tilde{v}) + o(t_k) - h(\tilde{y}_k)}{t_k} \\ =&\limsup_{k\to\infty} \frac{h(\tilde{y}_k + t_k\tilde{v}) - h(\tilde{y}_k)}{t_k} \leq h^\circ(\tilde{y}, \tilde{v})\,, \end{aligned} \tag{45}$$

where we used (44) in the first equality, and with the inequality true by definition of the Clarke derivative. □

Proof of Proposition 3.1. Let (φ) be a chart defined in a neighborhood U of $x \in M$. We use the notation $(\tilde{x}, \tilde{d}) = (\varphi(x), d\varphi(x)d)$ for $(x, d) \in T\mathcal{M}$. We pushforward 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 $d, q \in T_xM$ we define $g(\tilde{d}, \tilde{q}) = \langle d, q\rangle_x$, $\|\tilde{d} - \tilde{q}\|_{\tilde{x}} = \|d - q\|_x$, and $\tilde{\Gamma}^{\tilde{y}}_{\tilde{x}}(\tilde{d}) = \Gamma^y_x(d)$. 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 $g(\mathrm{grad}\tilde{f}(\tilde{x}), \tilde{d}) = \langle \nabla\tilde{f}(x), d\rangle$ 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 first prove (4) in x for some constant $L > 0$ and any d with $\|d\| \leq B$ for some $B > 0$. Equivalently, we want to prove

$$\tilde{f}(\tilde{R}(\tilde{x}, \tilde{d})) \leq \tilde{f}(\tilde{x}) + g(\mathrm{grad}\tilde{f}(\tilde{x}), \tilde{d}) + \frac{L}{2}\|\tilde{d}\|_{\tilde{x}}^2 \,. \tag{46}$$

for $\tilde{d}$ s.t. $\|\tilde{d}\| \leq B$.
By compactness we can choose (φ, U) and $B > 0$ in such a way that, for every $\tilde{y} \in \tilde{U}_1 \subset \tilde{U}$ 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 and $B > 0$ independent from $\tilde{x}, \tilde{y}, \tilde{d}$.
First, since $\tilde{R}$ is in particular C^1 regular

$$\tilde{R}(\tilde{x}, \tilde{d}) = \tilde{x} + O(\|\tilde{d}\|_{\tilde{x}}) \,, \tag{47}$$

and by smoothness of the parallel transport

$$\tilde{\Gamma}_{\tilde{x}}^{\tilde{y}} \tilde{q} = \tilde{q} + O(\|\tilde{x} - \tilde{y}\|) \,. \tag{48}$$

Furthermore,

$$\mathrm{grad}\tilde{f}(\tilde{x} + \tilde{q}) = \tilde{\Gamma}_{\tilde{x}}^{\tilde{x}+\tilde{q}} \mathrm{grad}\tilde{f}(\tilde{x}) + O(\mathrm{dist}(\tilde{x}, \tilde{x} + \tilde{q})) \,, \tag{49}$$

by the Lipschitz continuity assumption (6), and consequently

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

where we used (3) in the last equality.
Finally, since, $\frac{d}{dt}\tilde{R}(\tilde{x}, t\tilde{d})$ is C^1 regular, we also have

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

where we used (48) in the third equality, and (3) in the last one. Again by compactness, for $\tilde{y} \in \tilde{U}_1$, $t \leq 1$, $\|\tilde{q}\|, \|\tilde{d}\| \leq B$ 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 (46) reasoning as follows:

$$\begin{aligned} &\tilde{f}(\tilde{R}(\tilde{x}, \tilde{d})) - \tilde{f}(\tilde{R}(\tilde{x}, 0)) = \tilde{f}(\tilde{R}(\tilde{x}, \bar{t}q)) - \tilde{f}(\tilde{R}(\tilde{x}, 0)) \\ =&\int_0^{\bar{t}} \frac{d}{dt}\tilde{f}(\tilde{R}(\tilde{x} + t\tilde{q}))dt = \int_0^{\bar{t}} g(\mathrm{grad}f(\tilde{R}(\tilde{x}, t\tilde{q})), \frac{d}{dt}\tilde{R}(\tilde{x}, t\tilde{d}))dt \\ =&\int_0^{\bar{t}} g(\tilde{\Gamma}_{\tilde{x}}^{\tilde{R}(\tilde{x},t\tilde{q})} \mathrm{grad}\tilde{f}(\tilde{x}) + O(t\|\tilde{q}\|), \tilde{\Gamma}_{\tilde{x}}^{\tilde{R}(\tilde{x},t\tilde{d})} \tilde{d} + O(t\|\tilde{q}\|))dt \\ =&\int_0^{\bar{t}} \left(g(\tilde{\Gamma}_{\tilde{x}}^{\tilde{R}(\tilde{x},t\tilde{q})} \mathrm{grad}\tilde{f}(\tilde{x}), \tilde{\Gamma}_{\tilde{x}}^{\tilde{R}(\tilde{x},t\tilde{d})} \tilde{d}) + O(t\|\tilde{q}\|) \right) dt \\ =&g(\mathrm{grad}f(\tilde{x}), \tilde{d}) + O(\bar{t}^2\|\tilde{q}\|) = g(\mathrm{grad}f(\tilde{x}), \tilde{d}) + O(\|\tilde{d}\|^2) \,, \end{aligned} \tag{52}$$

where we used (50) and (51) in the fourth inequality. To conclude, notice that the above argument does not depend from the choice of $\tilde{x} \in \tilde{U}_1$, so that it can be extended to every $\tilde{y} \in \tilde{U}_1$ and then by compactness to every $y \in M$. □

Proof of Lemma 4.1. With the notation introduced in the proof of Proposition 3.1, without loss of generality we assume that U is bounded and that φ can be extended to a neighborhood containing the closure of U.
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

$$\tilde{R}(\tilde{y}, \tilde{v}) = \tilde{y} + \tilde{v} + O(\|\tilde{v}\|^2) \,, \tag{53}$$

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

$$\|\tilde{\Gamma}_{\tilde{x}}^{\tilde{x}_k}\tilde{q} - \tilde{q}\| \leq O\left(\|\tilde{x} - \tilde{x}_k\|\right), \tag{54}$$

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

$$\begin{aligned}\|\tilde{d}_k - \tilde{d}\| &\leq \|\tilde{d}_k - \tilde{\Gamma}_{\tilde{x}}^{\tilde{x}_k}\tilde{d}\| + \|\tilde{\Gamma}_{\tilde{x}}^{\tilde{x}_k}\tilde{d} - \tilde{d}\| \leq O\left(\|\tilde{d}_k - \tilde{\Gamma}_{\tilde{x}}^{\tilde{x}_k}(\tilde{d})\|_{\tilde{x}}\right) + O\left(\|\tilde{x} - \tilde{x}_k\|\right) \\ &= O\left(\|d_k - \Gamma_x^{x_k}(d)\|_x\right) + O\left(\|\tilde{x} - \tilde{x}_k\|\right) = o(1),\end{aligned} \tag{55}$$

where in the second inequality we used (54), 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

$$\begin{aligned}\|\tilde{v}_k - \tilde{d}\| &= \frac{1}{t_k}\|\tilde{R}(\tilde{x}_k, t_k\tilde{d}_k) - \tilde{x}_k - t_k\tilde{d}\| \leq \frac{1}{t_k}(\|R(\tilde{x}_k, t_k\tilde{d}_k) - \tilde{x}_k - t_k\tilde{d}_k\| + t_k\|d_k - \tilde{d}_k\|) \\ &= \frac{1}{t_k}(O(t_k^2\|\tilde{d}_k\|^2) + t_k o(1)) = o(1),\end{aligned} \tag{56}$$

where we used (53) and (55) for the first and the second summand in the second equality. In other words, $\tilde{v}_k \to \tilde{d}$. To conclude,

$$\begin{aligned}\limsup_{k\to\infty} \frac{f(R(y_k, t_k d_k)) - f(y_k)}{t_k} &= \limsup_{k\to\infty} \frac{\tilde{f}(\tilde{R}(\tilde{y}_k, t_k\tilde{d}_k)) - \tilde{f}(\tilde{y}_k)}{t_k} \\ = \limsup_{k\to\infty} \frac{\tilde{f}(\tilde{y}_k + t_k\tilde{v}_k) - \tilde{f}(\tilde{y}_k)}{t_k} &\geq \tilde{f}^\circ(\tilde{x}, \tilde{d}) = f^\circ(x, d),\end{aligned} \tag{57}$$

where in the inequality we were able to apply (7.1) because $\tilde{v}_k \to \tilde{d}$ by (56). □

7.2 Implementation details

For all the problems, the manifold structure we used was the one available in the MANOPT library [9]. After a basic tuning phase, we set the algorithm parameters as follows: we used $\gamma_1 = 0.61$, $\gamma_2 = 1$ and $\gamma = 0.77$ 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+, we considered the same parameters of the smooth case for RDS-SB and RDSE-SB, setting $\alpha_\epsilon = 10^{-3}$, and for both RDS-DD and RDSE-DD used $\gamma_1 = 0.95$, $\gamma_2 = 2$, and $\gamma = 1$.
The positive spanning basis was obtained both in Algorithm 1 and Algorithm 2 by projecting the positive spanning basis $(e_1, ..., e_n, -e_1, ..., -e_n)$ 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.
We generated the starting point and the parameters related to the instances either with MATLAB rand function or by using the random element generators implemented in the MANOPT library.

7.3 Smooth problems

We describe here the 8 smooth instances of problem (1) from [1, 7].

7.3.1 Largest eigenvalue, singular value, and top singular values problem

In the largest eigenvalue problem [7, Section 2.3], given a symmetric matrix $A \in S(n,n) = \{A \in \mathbb{R}^{n\times n} \mid A = A^\top\}$, we are interested in computing

$$\max_{x\in\mathbb{S}^{n-1}} x^\top Ax\,. \tag{58}$$

The largest singular value problem [7, Section 2.3] can be formulated generalizing (58): given $A \in \mathbb{R}^{m\times h}$, we are interested in

$$\max_{x\in\mathbb{S}^{m-1}, y\in\mathbb{S}^{h-1}} x^\top A y\,. \tag{59}$$

Notice how the domain in (58) and (59) are a sphere and the product of two spheres respectively.
Finally, to compute the sum of the top r singular values, as explained in [7, Section 2.5] it suffices to solve

$$\max_{X\in S(m,r), Y\in S(h,r)} X^\top A Y\,, \tag{60}$$

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

7.3.2 Dictionary learning

The dictionary learning problem [7, Section 2.4] can be formulated as

$$\begin{array}{ll} \min & \|Y - DC\| + \lambda\|C\|_1, \\ \text{s.t.} & D \in \mathbb{R}^{d\times h}, C \in \mathbb{R}^{h\times k},\ \|D_1\| = ... = \|D_h\| = 1\,, \end{array} \tag{61}$$

for a fixed $Y \in \mathbb{R}^{d\times k}$, $\lambda > 0$, $\|\cdot\|_1$ the ℓ_1- norm, and $D_1, ..., D_h$ the columns of D.
In our implementation we smooth the objective by using a smoothed version $\|\cdot\|_{1,\varepsilon}$ of $\|\cdot\|_1$

$$\|C\|_{1,\varepsilon} = \sum_{i,j} \sqrt{C_{i,j}^2 + \varepsilon^2}\,. \tag{62}$$

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

7.3.3 Synchronization of rotations

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

$$\mathrm{SO}(d) = \{R \in \mathbb{R}^{d\times d} \mid R^\top R = I_d \text{ and } \det(R) = 1\}\,. \tag{63}$$

In the synchronization of rotations problem [7, Section 2.6], we need to find rotations $R_1, ..., R_h \in \mathrm{SO}(d)$ 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

$$\min_{\hat{R}_1,...,\hat{R}_h\in\mathrm{SO}(d)} \sum_{(i,j)\in E} \|\hat{R}_i - H_{ij}\hat{R}_j\|^2\,. \tag{64}$$

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

7.3.4 Low-rank matrix completion

The low rank matrix completion problem [7, Section 2.7] can be written, for a fixed matrix $M \in \mathbb{R}^{m\times h}$, as

$$\begin{array}{ll} \min & \sum_{(i,j)\in\Omega}(X_{ij} - M_{ij})^2, \\ s.t. & X \in \mathbb{R}^{m\times h},\ \mathrm{rank}(X) = r\,, \end{array} \tag{65}$$

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., [26]).

7.3.5 Gaussian mixture models

In the Gaussian mixture model problem [7, Section 2.8], we are interested in computing a maximum likelihood estimation for a given set of observations $x_1, ..., x_h$:

$$\max_{\substack{\hat{u}_1,...,\hat{u}_k \in \mathbb{R}^d \\ \hat{\Sigma}_1,...,\hat{\Sigma}_k \in \mathrm{Sym}(d)^+, \\ w \in \Delta_+^{K-1}}} \sum_{i=1}^{h} \log \left(\sum_{k=1}^{K} w_k \frac{1}{\sqrt{2\pi \det(\Sigma_k)}} e^{\frac{(x-\mu_k)^\top \Sigma_k^{-1} (x-\mu_k)}{2}} \right), \tag{66}$$

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

$$\mathrm{Sym}(d)^+ = \{X \in \mathbb{R}^{d \times d} \mid X = X^\top, \, X \succ 0\} \tag{67}$$

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, we considered the case $K = 2$ and the reformulation proposed in [15], which does not use the unconstrained variables $(\hat{u}_1, ..., \hat{u}_k)$.

7.3.6 Procrustes problem

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

$$\min_{x \in \mathcal{M}} \|AX - B\|_F^2 , \tag{68}$$

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

7.4 Nonsmooth problems

We report two nonsmooth problems taken from [16].

7.4.1 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

$$\min_{x \in \mathbb{S}^{n-1}} \|Qx\|_1 . \tag{69}$$

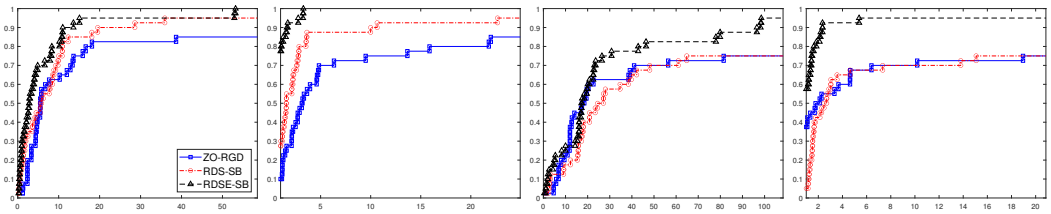
7.4.2 Nonsmooth low-rank matrix completion

In the nonsmooth version of the low rank matrix completion problem (65) the Euclidean norm is replaced with the l_1 norm, so that in the objective we have a sum of absolute values:

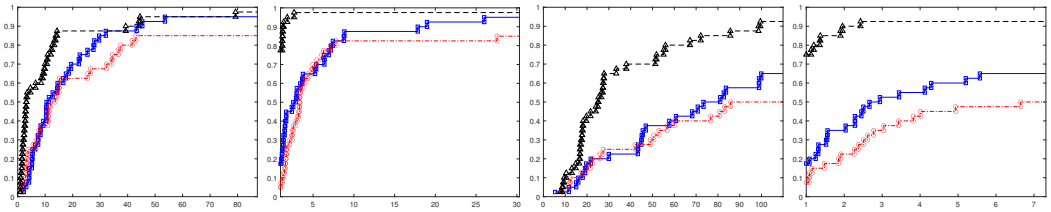
$$\begin{array}{ll} \min & \sum_{(i,j) \in \Omega} |X_{ij} - M_{ij}|, \\ s.t. & X \in \mathbb{R}^{m \times n}, \, \mathrm{rank}(X) = r . \end{array} \tag{70}$$

7.5 Additional numerical results

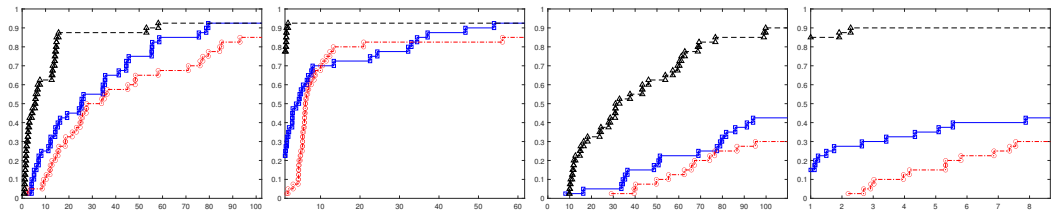
We include here the performance and data profiles split by problem size.



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

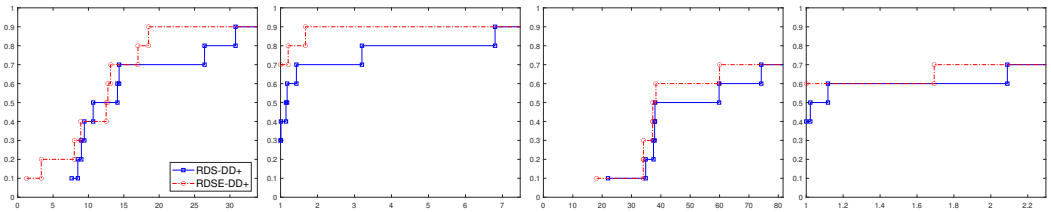


(e) Data p., $\tau = 10^{-1}$ (f) Perf. p., $\tau = 10^{-1}$ (g) Data p., $\tau = 10^{-3}$ (h) Perf. p., $\tau = 10^{-3}$

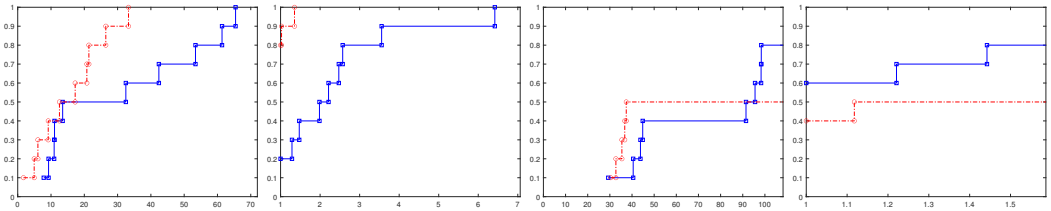


(i) Data p., $\tau = 10^{-1}$ (j) Perf. p., $\tau = 10^{-1}$ (k) Data p., $\tau = 10^{-3}$ (l) Perf. p., $\tau = 10^{-3}$

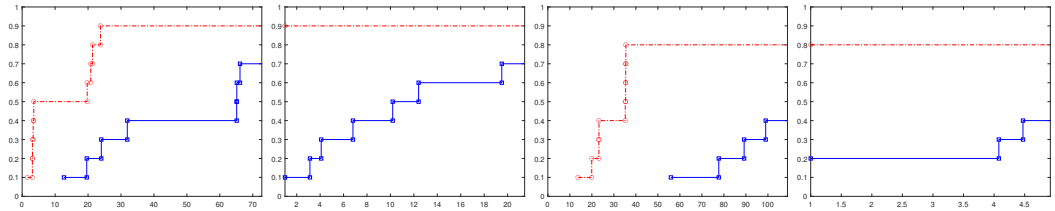
Figure 3: From top to bottom: results for small, medium and large instances in the smooth case.



(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., $\tau = 10^{-3}$

Figure 4: From top to bottom: results for small, medium and large instances in the nonsmooth case.