Noname manuscript No.
(will be inserted by the editor)

A descent method for nonsmooth multiobjective optimization problems on Riemannian manifolds

Chunming Tang · Hao He · Jinbao Jian · Miantao Chao

Received: date / Accepted: date

Abstract In this paper, a descent method for nonsmooth multiobjective optimization problems on complete Riemannian manifolds is proposed. The objective functions are only assumed to be locally Lipschitz continuous instead of convexity used in existing methods. A necessary condition for Pareto optimality in Euclidean space is generalized to the Riemannian setting. At every iteration, an acceptable descent direction is obtained by constructing a convex hull of some Riemannian ε-subgradients. And then a Riemannian Armijo-type line search is executed to produce the next iterate. The convergence result is established in the sense that a point satisfying the necessary condition for Pareto optimality can be generated by the algorithm in a finite number of iterations. Finally, some preliminary numerical results are reported, which show that the proposed method is efficient.

Keywords Multiobjective optimization · Riemannian manifolds · Descent method · Pareto optimality · Convergence analysis

Mathematics Subject Classification (2000) 65K05 · 90C30

1 Introduction

In the field of optimization, minimizing multiple objective functions at the same time is called multiobjective optimization. Usually, these objective functions are conflicting with each other, instead of having some common minimum

Chunming Tang, Hao He, and Miantao Chao
College of Mathematics and Information Science, Center for Applied Mathematics of Guangxi, Guangxi University, Nanning, 530004, P. R. China

Jinbao Jian, Corresponding author
College of Mathematics and Physics; Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Center for Applied Mathematics and Artificial Intelligence, Guangxi Minzu University, Nanning 530006, P. R. China
jianjb@gxu.edu.cn

points. For example, producers want to create higher value and make the cost as low as possible in production and manufacturing. Similar problems arise in many applications such as engineering design [17], management science [4, 25], environmental analysis [32, 19], etc. Due to its wide practical applications, multiobjective optimization has always been a hot topic, and a rich literature was produced; see the monographs [1, 14, 38] and the references therein.

In most cases, the solution of multiobjective optimization problem is not a single point but a set of all optimal compromises, namely, the Pareto set. In traditional multiobjective optimization, one of the most popular methods is the scalarization approach [23], whose idea is to convert a multiobjective problem to a single or a family of single objective optimization problems. However, in this method, the users need to select some necessary parameters because they are not known in advance, which may bring an additional cost. To overcome this shortcoming, there have other methods to solve such optimization problems, such as descent methods [20, 16, 24], Newton-type methods [21, 36], proximal point methods [8, 11] and proximal bundle methods [35], etc. These methods are almost all developed from a single objective optimization. In this paper, we are particularly interested in the case where the objective functions are not necessarily differentiable or convex. Recently, Gebken and Peitz [24] proposed a descent method for locally Lipschitz multiobjective optimization, in which an acceptable descent direction for all objectives is selected as the element which has the smallest norm in the negative convex hull of certain subgradients of the objective functions.

In recent years, many traditional single objective optimization theories and methods have been extended from Euclidean space to Riemannian manifolds; see, e.g., [2, 5, 27, 28, 30, 31, 37, 40]. Comparatively, for Riemannian multiobjective optimization, the relevant literature is very scarce, especially for nonsmooth cases. In [9] and [10], a steepest descent method and an inexact version with Armijo rule for multiobjective optimization in the Riemannian context are presented, respectivley. Both methods require the objective functions to be continuously differentiable for partial convergence, and further assume that the objective vector function is quasi-convex and the manifold has nonnegative curvature for full convergence. In [12], a proximal point method for nonsmooth multiobjective optimization on Hadamard manifold is developed. In [6], a subgradient-type method for Riemannian nonsmooth multiobjective optimization is presented, which requires the objective vector function to be convex. In [18], a trust region method for Riemannian smooth multiobjective optimization problems is proposed. As far as we know, numerical results are not reported in the existing literature for Riemannian nonsmooth multiobjective optimization.

Based on the above observations, the aim of this paper is to develop a practical implementable method for nonconvex nonsmooth multiobjective optimization problems on general Riemannian manifolds. More precisely, we propose a descent method for locally Lipschitz multiobjective optimization problems on complete Riemannian manifolds, which can be regarded as an extension of the work [24] in Euclidean space. To the best of our knowledge, our work is the first to consider the setting discussed here. The classical necessary condition of the Pareto optimal points for nonsmooth multiobjective optimization (see [34]) is generalized to the Riemannian setting. The Riemannian ε-subdifferential is introduced by using the isometric vector transports which satisfy a locking condition [29]. And then show that there exists a common descent direction for each objective, which is just the element with the smallest norm in the set consisting of the negative convex hull of the Riemannian ε-subdifferentials of all objective functions. Of course, it is generally not easy to compute the ε-subdifferentials of a nonsmooth function especially when its domain is a manifold. In order to save computational effort, inspired by the strategy adopted in the traditional methods [33, 24], we use the convex hull of a special set to approximate the convex hull of Riemannian ε-subdifferentials of all objective functions. For this set, at the beginning, it consists of a single ε-subgradient of each objective function, then some new ε-subgradients are systematically computed and added to enrich the set, until the element with the smallest norm in its convex hull is an acceptable direction for each objective function. Furthermore, a Riemannian Armijo-type line search is executed to produce the next iterate. The convergence result is established in the sense that an (ε, δ)-critical point which is an approximation of the Pareto optimal point can be generated in a finite number of iterations. Finally, some preliminary numerical results are reported, which show that the proposed method is efficient.

This paper is organized as follows. In section 2, we recall some basic notations and definitions regarding Riemannian manifolds and locally Lipschitz function. In section 3, the necessary condition for Pareto optimality regarding locally Lipschitz multiobjective optimization problems is generalized to Riemannian manifolds, and the details of our method is presented. In section 4, we establish the convergence result of our method. In section 5, some preliminary numerical experiments are given.

2 Preliminaries

Throughout of the paper, we denote by $\mathrm{cl}S$ and $\mathrm{conv}S$ the closure and the convex hull of a set S, respectively. Letting $\mathcal{M}$ be a complete d-dimensional ($d \geq 1$) smooth manifold endowed with a Riemannian metric $\langle \cdot, \cdot \rangle_x$ on the tangent space $T_x\mathcal{M}$, we denote by $\|\cdot\|_x$ the norm which induced by Riemannian metric. We will often omit subscripts when they do not cause confusion and simply write $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ to $\langle \cdot, \cdot \rangle_x$ and $\| \cdot \|_x$, respectively. The Riemannian distance from x to y is denoted by $\mathrm{dist}(x,y)$, where the points $x, y \in \mathcal{M}$. Denote $\mathbb{B}(x, \sigma) = \{y \in \mathcal{M} |\ \mathrm{dist}(x, y) < \sigma\}$, and the tangent bundle by $T\mathcal{M}$.

Firstly, we introduce the definition of locally Lipschitz functions on Riemannian manifolds; see, e.g., [27].

Definition 2.1 Let $x \in \mathcal{M}$, $L > 0$, and $\mathcal{N} \subset \mathcal{M}$ be a neighborhood of x. If $f : \mathcal{M} \to \mathbb{R}$ satisfies

$$|f(x) - f(y)| \leq L\mathrm{dist}(x, y), \text{ for all } y \in \mathcal{N},$$

we say that f is Lipschitz continuous near x with the constant L. Furthermore, if for all $x \in \mathcal{M}$, f is Lipschitz continuous near x, then we say that f is a locally Lipschitz (continuous) function on $\mathcal{M}$.

Now we consider the Riemannian nonsmooth multiobjective optimization problem:

$$\min_{x \in \mathcal{M}} F(x) := \begin{pmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{pmatrix}, \tag{1}$$

where $F : \mathcal{M} \to \mathbb{R}^m$ is called objective vector function, and the components $f_i : \mathcal{M} \to \mathbb{R}$ for $i \in \{1, \cdots, m\}$ are called objective functions, which are assumed to be locally Lipschitz continuous on $\mathcal{M}$. Clearly, the concept of optimality for real-valued function no longer applies, since the objective function of problem (1) is vector valued. So we introduce the following so-called Pareto optimality (see [15, Ch. 2]), and our aim is to find (approximate) Pareto optimal points on manifolds.

Definition 2.2 Let $x \in \mathcal{M}$. If there is no $y \in \mathcal{M}$ such that

$$f_i(y) \leq f_i(x) \ \ \forall i \in \{1, \cdots, m\} \text{ and } f_j(y) < f_j(x) \ \text{ for some } j \in \{1, \cdots, m\},$$

then we say that x is a Pareto optimal point for the problem (1). Pareto set is the set consisting of all Pareto optimal points.

For optimization problems posed on nonlinear manifolds, the concept of retraction can help us to develop a theory which is similar to line search methods in $\mathbb{R}^n$; see [2, Def. 4.1.1].

Definition 2.3 A smooth mapping $R : T\mathcal{M} \to \mathcal{M}$ is called a retraction on a manifold $\mathcal{M}$ if it has the following properties:
(i) $R_x(0_x) = x$, where 0_x denotes the zero element of $T_x\mathcal{M}$;
(ii) with the canonical identification $T_{0_x}T_x\mathcal{M} \simeq T_x\mathcal{M}$, R_x satisfies

$$\mathrm{D}R_x(0_x) = \mathrm{id}_{T_x\mathcal{M}},$$

where R_x is the restriction of R to $T_x\mathcal{M}$, and $\mathrm{id}_{T_x\mathcal{M}}$ denotes the identity mapping on $T_x\mathcal{M}$.

It is further assumed that there is a constant κ such that

$$\mathrm{dist}(R_x(\xi_x), x) \leq \kappa \|\xi_x\| \tag{2}$$

for all $x \in \mathcal{M}$ and $\xi_x \in T_x\mathcal{M}$. Intuitively, the inequality (2) implies that the distance between x and $R_x(\xi_x)$ is bounded when the vector ξ_x is bounded.

Furthermore, it can ensure that the point $R_x(\xi_x)$ is in the neighborhood of x when $\|\xi_x\|$ is small enough. This does not constitute any restriction in most cases of interest; see [27]. Denote $B_R(x,r) = \{R_x(\eta_x)|\ \|\eta_x\| < r\}$, which is an open ball centered at x with radius r if the retraction R is the exponential mapping; see [29].

Since we will work in different tangent spaces, it is necessary to introduce the concept of vector transport (see [2, Def. 8.1.1]). As its name implies, it serves to move vectors in different tangent spaces to the same tangent space. Particularly, parallel translation along geodesics is a vector transport.

Definition 2.4 A smooth mapping $\mathcal{T} : T\mathcal{M} \oplus T\mathcal{M} \to T\mathcal{M} : (\eta_x, \xi_x) \mapsto \mathcal{T}_{\eta_x}(\xi_x)$ is said to be a vector transport associated to a retraction R if for all (η_x, ξ_x), the following conditions hold:
(i) $\mathcal{T}_{\eta_x} : T_x\mathcal{M} \to T_{R_x(\eta_x)}\mathcal{M}$ is a linear map;
(ii) $\mathcal{T}_{0_x}(\xi_x) = \xi_x$ for all $\xi_x \in T_x\mathcal{M}$.

Briefly, if $\xi_x \in T_x\mathcal{M}$ and $R(\eta_x) = y$, then $\mathcal{T}_{\eta_x}$ transports vector ξ_x from the tangent space of $\mathcal{M}$ at x to the tangent space at y. In order to obtain convergence results, the following conditions are also required.

- The vector transport $\mathcal{T}$ preserves inner products, i.e.,

$$\langle \mathcal{T}_{\eta_x}(\xi_x), \mathcal{T}_{\eta_x}(\zeta_x)\rangle = \langle \xi_x, \zeta_x\rangle. \tag{3}$$

- The following locking condition is satisfied for $\mathcal{T}$, i.e.,

$$\mathcal{T}_{\xi_x}(\xi_x) = \beta_{\xi_x}\mathcal{T}_{R_{\xi_x}}(\xi_x), \quad \beta_{\xi_x} = \frac{\|\xi_x\|}{\|\mathcal{T}_{R_{\xi_x}}(\xi_x)\|}, \tag{4}$$

  where

$$\mathcal{T}_{R_{\eta_x}}(\xi_x) = \mathrm{D}R_x(\eta_x)[\xi_x] = \frac{\mathrm{d}}{\mathrm{d}t}R_x(\eta_x + t\xi_x)|_{t=0}.$$

The above conditions are satisfied with $\beta_{\xi_x} = 1$ if the retraction and vector transport are selected as the exponential map and parallel transport, respectively; see [28] for more details. For simplicity, the following intuitive notations are used:

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

In addition, it is necessary to introduce the notion of injectivity radius for R_x, since we need to transport subgradients from tangent spaces at some points lying in the neighborhood of $x \in \mathcal{M}$ to the tangent space at x; see [29].

Definition 2.5 Let

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

where $B(0_x, \varepsilon) = \{\xi_x|\ \|\xi_x\| < \varepsilon\} \subset T_x\mathcal{M}$. Furthermore, for this retraction R, the injectivity radius of $\mathcal{M}$ is defined as

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

Remark 2.1 (i) As in usual we assume that $\iota(\mathcal{M}) > 0$, and that an explicit positive lower bound of $\iota(\mathcal{M})$ is available, which will be used as an input of the algorithm. In fact, when $\mathcal{M}$ is compact, we at least know that $\iota(\mathcal{M}) > 0$. (ii) Clearly, $\mathcal{T}_{x\leftarrow y}(\xi_x)$ is well defined for all $y \in B_R(x, \iota(x))$, especially, for all $y \in B_R(x, \iota(\mathcal{M}))$. In what follows, it will always be ensured that $\mathcal{T}_{x\leftarrow y}(\xi_x)$ is well defined when we use it.

We close this section by recalling the notion of Riemannian subdifferential, which is an extension of the classical Clarke subdifferential; see [29]. If X is a Hilbert space, and ϕ is a locally Lipschitz function defined from X to $\mathbb{R}$, the Clarke generalized directional derivative of ϕ at x in direction v is defined as

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

and the Clarke subdifferential of ϕ at x is then given by

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

Definition 2.6 Let $f : \mathcal{M} \to \mathbb{R}$ be a locally Lipschitz function and for $x \in \mathcal{M}$, denote $\hat{f}_x = f \circ R_x$. The Riemannian directional derivative of f at x in direction $p \in T_x\mathcal{M}$ is defined as

$$f^\circ(x; p) = \hat{f}_x^\circ(0_x, p),$$

where $\hat{f}_x^\circ(0_x, p)$ is the Clarke generalized directional derivative of $\hat{f}_x : T_x\mathcal{M} \to \mathbb{R}$ at 0_x in direction $p \in T_x\mathcal{M}$. Then the Riemannian subdifferential of f at x is defined as

$$\partial f(x) = \partial \hat{f}_x(0_x).$$

The following lemma shows some important properties of the Riemannian subdifferential, which are similar to those of the Clarke subdifferential in Hilbert space; see [29, Thm. 2.2].

Lemma 2.1 *Let $f : \mathcal{M} \to \mathbb{R}$ be a locally Lipschitz function, then the set $\partial f(x)$ is a nonempty, compact and convex subset of $T_x\mathcal{M}$, and $\|\xi\| \le L$ for all $\xi \in \partial f(x)$, where L is the Lipschitz constant near x.*

The Riemannian ε-subdifferential and ε-subgradient of f can be also defined; see [29].

Definition 2.7 Let $\varepsilon \in \left(0, \frac{1}{2}\iota(x)\right)$, [1] then the Riemannian ε-subdifferential of a locally Lipschitz function f on a Riemannian manifold $\mathcal{M}$ at x is defined as

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

where $\beta_\eta = \frac{\|\eta\|}{\|\mathcal{T}_{R_\eta}(\eta)\|}$. Every element of $\partial_\varepsilon f(x)$ is called a (Riemannian) ε-subgradient.

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

Lemma 2.2 *The set $\partial_\varepsilon f(x)$ is a nonempty, compact and convex subset of $T_x\mathcal{M}$.*

Proof From [29, Thm. 2.15] we know that the set $\partial_\varepsilon f(x)$ is bounded. This together with Lemma 2.1 and Definition 2.7 shows the claim. □

3 A descent method for Riemannian nonsmooth multiobjective optimization

In this section, we present the details of our algorithm, which contains two procedures that help us to find a descent direction. We first introduce the definition of global weak Pareto optimal and local weak Pareto optimal for problem (1) as follows; see [15, Ch. 2].

Definition 3.1 Let $x \in \mathcal{M}$. If there is no $y \in \mathcal{M}$ such that $f_i(y) < f_i(x)$ for all $i = 1, \cdots, m$, then we say that x is a weak Pareto optimal of problem (1). If there exists some $\sigma > 0$ such that x is a (weak) Pareto optimal on $\mathbb{B}(x,\sigma)\cap\mathcal{M}$, then we say that x is a local (weak) Pareto optimal of problem (1).

It is clear that if x is a Pareto optimal of problem (1), then it is a global weak Pareto optimal of problem (1), so it also must be a local weak Pareto optimal of problem (1). Next, we generalize the necessary condition for Pareto optimality in Euclidean space (see [34]) to the Riemannian setting.

Theorem 3.1 *Let $x \in \mathcal{M}$ be a local weak Pareto optimal of problem (1), then we have*

$$0_x \in \mathrm{conv}G(x), \tag{5}$$

where $G(x) = \bigcup_{i=1}^m \partial f_i(x)$.

Proof We first show that $\overline{G}(x) = \emptyset$, where

$$\overline{G}(x) = \{d \in T_x\mathcal{M} |\ \langle d, \xi\rangle < 0 \text{ for all } \xi \in G(x)\}.$$

By Definition 3.1, there exists a $\sigma > 0$ such that for every $y \in \mathcal{M} \cap \mathbb{B}(x,\sigma)$ there is an index $i \in \{1, \cdots, m\}$ such that inequality $f_i(y) \geq f_i(x)$ holds. Let $d \in T_x\mathcal{M}$ be arbitrary, then there exist sequences $\{d_k\} \subseteq T_x\mathcal{M}$ and $\{t_k\}$ such that $d_k \to d$ and $t_k \downarrow 0$. Set $\tilde{\varepsilon} = \min\{\frac{\sigma}{\kappa}, \iota(x)\}$ and $\varepsilon \in (0, \tilde{\varepsilon})$. It is clear that there exists a constant $N > 0$ such that $t_k d_k \in B(0_x, \varepsilon)$ for all $k > N$. By (2), we have $R_x(t_k d_k) \in \mathcal{M} \cap \mathbb{B}(x,\sigma)$ for all $k > N$. Then for every $k > N$ there exists an index $i_k \in \{1, \cdots, m\}$ such that $f_{i_k}(R_x(t_k d_k)) \geq f_{i_k}(x)$. Since m is finite, there must be an index $i_0 \in \{1, \cdots, m\}$ and subsequences $\{d_{k_j}\} \subset \{d_k\}$ and $\{t_{k_j}\} \subset \{t_k\}$ such that

$$f_{i_0}\left(R_x(t_{k_j} d_{k_j})\right) \geq f_{i_0}(x) = f_{i_0}(R_x(0_x)),$$

for all $k_j > N$. Denote $K = \{k_j |\ k_j > N\}$. Since $\hat{f}_{i_0 x} = f_{i_0} \circ R_x$ is a locally Lipschiz function on $B(0_x, \varepsilon)$, by the mean value theorem ([13, Thm. 2.3.7]), it follows that for all $\bar{k} \in K$, there exists a $\tilde{t}_{\bar{k}} \in (0, t_{\bar{k}})$ such that

$$\hat{f}_{i_0 x}(t_{\bar{k}} d_{\bar{k}}) - \hat{f}_{i_0 x}(0_x) \in \langle \partial \hat{f}_{i_0 x}(\tilde{t}_{\bar{k}} d_{\bar{k}}), t_{\bar{k}} d_{\bar{k}}\rangle.$$

Then from [13, Prop. 2.1.2 (b)], we obtain

$$\hat{f}_{i_0x}^{\circ}(\tilde{t}_{\bar{k}}d_{\bar{k}};d_{\bar{k}}) = \max_{\xi\in\partial\hat{f}_{i_0x}(\tilde{t}_{\bar{k}}d_{\bar{k}})}\langle\xi,d_{\bar{k}}\rangle \geq \frac{1}{t_{\bar{k}}}(\hat{f}_{i_0x}(t_{\bar{k}}d_{\bar{k}}) - \hat{f}_{i_0x}(0_x)) \geq 0.$$

Thus, for all $\bar{k}\in K$ we have $\hat{f}_{i_0x}^{\circ}(\tilde{t}_{\bar{k}}d_{\bar{k}};d_{\bar{k}})\geq 0$. Since $d_{\bar{k}}\to d$ and $\tilde{t}_{\bar{k}}d_{\bar{k}}\to 0_x$, from the upper semicontinuous of function $\hat{f}_{i_0x}^{\circ}$ (see [13, Prop. 2.1.1 (b)]) and Definition 2.6, we obtain

$$f_{i_0}^{\circ}(x,d) = \hat{f}_{i_0x}^{\circ}(0_x;d) = \limsup_{\bar{k}\to\infty}\hat{f}_{i_0x}^{\circ}(\tilde{t}_{\bar{k}}d_{\bar{k}};d_{\bar{k}}) \geq 0.$$

By [29, Thm. 2.2 (b)], we have $f_{i_0}^{\circ}(x,d) = \max_{\xi\in\partial f_{i_0}(x)}\langle\xi,d\rangle$. Therefore, there exists a $\bar{\xi}\in\partial f_{i_0}(x)\subseteq G(x)$ such that $\langle\bar{\xi},d\rangle\geq 0$, which implies $d\notin\overline{G}(x)$, and thus $\overline{G}(x)=\emptyset$.

Now, we show that $0_x\in\mathrm{conv}G(x)$. Note that $\overline{G}(x)=\emptyset$, then for any $d\in T_x\mathcal{M}$, there exists some $\xi_0\in G(x)\subseteq\mathrm{conv}G(x)$ such that

$$\langle d,\xi_0\rangle \geq 0. \tag{6}$$

Suppose $0_x\notin\mathrm{conv}G(x)$. Since the sets $\mathrm{conv}G(x)$ and $\{0_x\}$ are closed convex sets, there exist $d\in T_x\mathcal{M}$ and $a\in\mathbb{R}$ such that

$$0 = \langle d,0_x\rangle \geq a \text{ and } \langle d,\xi\rangle < a \text{ for all } \xi\in\mathrm{conv}G(x)$$

according to the separation theorem. The above relations imply that $\langle d,\xi\rangle<0$ for all $\xi\in\mathrm{conv}G(x)$, which is a contradiction with inequality (6). Hence $0_x\in\mathrm{conv}G(x)$. □

From Theorem 3.1 and the previous results, we know that $0_x\in\mathrm{conv}G(x)$ if x is a Pareto optimum of problem (1). Conversely, when the objective functions are strictly convex, the point satisfying (5) is Pareto optimum of problem (1); see the lemma below.

Lemma 3.1 *Suppose that the objective functions of problem (1) are all strictly convex on $\mathcal{M}$.*[2] *Then every point satisfying (5) is Pareto optimal of problem (1).*

Proof By [3, Lem. 1.3], we immediately obtain this result. □

The method proposed in this paper is a descent method based on the line search strategy. In particular, for each iteration k, we hope to find a descent direction $g_k\in T_{x_k}\mathcal{M}$ and a stepsize $t_k>0$ such that $f_i(x_{k+1})<f_i(x_k)$ for all $i\in\{1,\cdots,m\}$, where $x_{k+1}=R_{x_k}(t_kg_k)$. Next, we will explain how to find such g_k.

[2] A function $f:\mathcal{M}\to\mathbb{R}$ is said to be convex if the composition $f\circ\gamma:[-\nu,\nu]\to\mathbb{R}$ is convex for any geodesic segment $\gamma:[-\nu,\nu]\to\mathcal{M}$ with $\nu>0$; see [7]. The function f is said to be strictly convex if the composition $f\circ\gamma:[-\nu,\nu]\to\mathbb{R}$ is strictly convex.

Definition 3.2 For $\varepsilon \in \left(0, \frac{1}{2}\iota(x)\right)$, the ε-subdifferential of the objective vector $F(x)$ of problem (1) is defined as

$$G_\varepsilon(x) := \operatorname{conv}\left(\bigcup_{i=1}^{m} \partial_\varepsilon f_i(x)\right) \subset T_x\mathcal{M}.$$

It is clear that $G_\varepsilon(x)$ is nonempty, convex and compact by Lemma 2.2.

Lemma 3.2 *Let* $\varepsilon \in \left(0, \frac{1}{2}\iota(x)\right)$.
(i) If x is Pareto optimal of problem (1) , then $0_x \in G_\varepsilon(x)$.
(ii) Let $x \in \mathcal{M}$ and

$$\bar{g} := \underset{\xi \in -G_\varepsilon(x)}{\operatorname{argmin}} \|\xi\|. \tag{7}$$

Then, either $\bar{g} = 0_x$ or $\bar{g} \neq 0_x$ and

$$\langle \bar{g}, \xi \rangle \leq -\|\bar{g}\|^2 < 0, \quad \forall \xi \in G_\varepsilon(x). \tag{8}$$

Proof (i) It is obvious that $\operatorname{conv}G(x) \subseteq G_\varepsilon(x)$, then combining with Theorem 3.1 we immediately obtain $0_x \in G_\varepsilon(x)$. (ii) Since the set $G_\varepsilon(x)$ is nonempty and compact, then there exists some $\bar{g}$ such that $\bar{g} = \operatorname{argmin}_{\xi \in -G_\varepsilon(x)} \|\xi\|$. If $\bar{g} \neq 0_x$, we have $-\bar{g} = \operatorname{argmin}_{\xi \in G_\varepsilon(x)} \|\xi\|$ by (7). Note that the set $G_\varepsilon(x)$ is convex, so we have the inequality $\langle \xi - (-\bar{g}), -(-\bar{g}) \rangle \leq 0$, which implies (8). ☐

By Lemma 3.2, we still have the necessary optimality condition $0_x \in G_\varepsilon(x)$ when working with the ε-subdifferential instead of $\operatorname{conv}G(x)$. The following lemma shows that for each objective function f_i, there exists a common lower bound for a stepsize to guarante descent when using the direction $\bar{g}$ defined by (7) as a search direction. We extend the result of [24, Lem. 3.2] to the Riemannian setting as follows.

Lemma 3.3 *Let* $\varepsilon \in \left(0, \frac{1}{2}\iota(x)\right)$ *and $\bar{g}$ be the solution of (7). Then*

$$f_i(R_x(t\bar{g})) \leq f_i(x) - t\|\bar{g}\|^2, \quad \forall\ 0 \leq t \leq \frac{\varepsilon}{\|\bar{g}\|},\ i \in \{1, \cdots, m\}.$$

Proof For all $t \in \left[0, \frac{\varepsilon}{\|\bar{g}\|}\right]$, by Lebourg's mean value theorem [26, Thm. 3.3], there exist $\theta \in (0, 1)$ and $\xi \in \partial f_i(R_x(\theta t\bar{g}))$ such that

$$f_i(R_x(t\bar{g})) - f_i(x) = \langle \xi, \mathrm{D}R_x(\theta t\bar{g})[t\bar{g}] \rangle,\ i \in \{1, \cdots, m\}.$$

It is clear that $\|\theta t\bar{g}\| < \varepsilon$. Combining (3) and the locking condition (4) of the vector transport, we have that

$$f_i(R_x(t\bar{g})) - f_i(x) = \frac{t}{\beta_{\theta t\bar{g}}} \langle \mathcal{T}_{x \leftarrow R_x(\theta t\bar{g})}(\xi), \bar{g} \rangle,\ i \in \{1, \cdots, m\}.$$

Since $\|\theta t\bar{g}\| < \varepsilon$, it follows that $\frac{1}{\beta_{\theta t\bar{g}}} \mathcal{T}_{x \leftarrow R_x(\theta t\bar{g})}(\xi) \in \partial_\varepsilon f_i(x) \subset G_\varepsilon(x)$, then from (8) we obtain

$$f_i(R_x(t\bar{g})) - f_i(x) \leq -t\|\bar{g}\|^2,\ i \in \{1, \cdots, m\}.$$

This completes the proof. ☐

Lemma 3.3 states that $\bar{g}$ is a descent direction of f_i for every $i \in \{1 \cdots, m\}$. However, it is not easy to compute $\bar{g}$ in practice, since the set $G_\varepsilon(x)$ is usually unknown. A natural idea is to approximate $G_\varepsilon(x)$ by the convex hull of a certain set W, which is expected to have at least two properties: (i) it is much easier to compute than $G_\varepsilon(x)$; (ii) $\tilde{g} = \mathrm{argmin}_{\xi \in -\mathrm{conv}W} \|\xi\|$ can be used instead of $\bar{g}$ as an approximate descent direction of f_i for every $i \in \{1 \cdots, m\}$.

Now, we present the details of our algorithm (Algorithm 1) as follows.

Algorithm 1: A descent method for Riemannian nonsmooth multi-objective optimization

---

0 Select an initial point $x_0 \in \mathcal{M}$, $\varepsilon \in \left(0, \frac{1}{2}\iota(\mathcal{M})\right)$, tolerance $\delta > 0$, and Armijo parameters $c \in (0, 1)$, $\alpha > 1$, $t_0 > 0$. Let $k = 0$.

1 Compute an acceptable descent direction: $g_k = \mathcal{P}_{\mathrm{dd}}(x_k, \varepsilon, \delta, c)$, where $\mathcal{P}_{\mathrm{dd}}$ is a procedure given below.

2 If $\|g_k\| \le \delta$, then STOP.

3 Find the smallest integer $\ell \in \left\{0, 1, \cdots, \left[\frac{\ln(t_0\|g_k\|) - \ln\varepsilon}{\ln\alpha}\right]\right\}$ satisfying

$$f_i(R_{x_k}(\alpha^{-\ell} t_0 g_k)) \le f_i(x_k) - \alpha^{-\ell} t_0 c \|g_k\|^2, \quad i \in \{1, \cdots, m\}.$$

If such an ℓ exists, set $t_k = \alpha^{-\ell} t_0$. Otherwise, set $t_k = \frac{\varepsilon}{\|g_k\|}$.

4 Set $x_{k+1} = R_{x_k}(t_k g_k)$, $k = k + 1$ and go to Step 1.

Remark 3.1 In step 1 of Algorithm 1, the aim of the inner procedure $\mathcal{P}_{\mathrm{dd}}$ is to find an acceptable descent direction of f_i for every $i \in \{1 \cdots, m\}$, which uses the substitute $\mathrm{conv}W$ instead of $G_\varepsilon(x)$. In step 3, the symbol $[\,\cdot\,]$ denotes the largest integer that does not exceed $\cdot$. In what follows, we will show that $\frac{\varepsilon}{\|g_k\|}$ is a common descent stepsize for all objective functions when using g_k as the search direction. The line search strategy of step 3 means that if there is a longer stepsize $\alpha^{-\ell} t_0$ than $\frac{\varepsilon}{\|g_k\|}$, then we use $\alpha^{-\ell} t_0$ as the stepsize. Otherwise we use the latter.

Next, we describe how we can obtain a good approximation of $G_\varepsilon(x)$ without requiring full knowledge of the ε-subdifferential. Let $W = \{\xi_1, \cdots, \xi_r\} \subseteq G_\varepsilon(x)$ and

$$\tilde{g} := \underset{g \in -\mathrm{conv}W}{\mathrm{argmin}} \|g\|. \tag{9}$$

If $\tilde{g} \neq 0_x$, then set $c \in (0, 1)$ and check the following inequality:

$$f_i\left(R_x\left(\frac{\varepsilon}{\|\tilde{g}\|}\tilde{g}\right)\right) \le f_i(x) - c\varepsilon\|\tilde{g}\|, \quad \forall\, i \in \{1, \cdots, m\}. \tag{10}$$

If (10) holds, then we can say $\mathrm{conv}W$ is an acceptable approximation for $G_\varepsilon(x)$, and $\tilde{g}$ is an acceptable descent direction. Otherwise, the set $I \subseteq \{1, \cdots, m\}$

consists of the indices for which (10) is not satisfied is nonempty, then we hope to find a new ε-subgradient $\xi' \in G_\varepsilon(x)$ such that $W \cup \xi'$ yields a better approximation to $G_\varepsilon(x)$. The following lemma can help us to find such an ε-subgradient.

Lemma 3.4 *Let $c \in (0,1)$, $W = \{\xi_1, \cdots, \xi_r\} \subseteq G_\varepsilon(x)$ and $\tilde{g} \neq 0_x$ be the solution of (9). If*

$$f_j\left(R_x\left(\frac{\varepsilon}{\|\tilde{g}\|}\tilde{g}\right)\right) > f_j(x) - c\varepsilon\|\tilde{g}\| \tag{11}$$

for some $j \in \{1, \cdots, m\}$, then there exist some $t' \in \left[0, \frac{\varepsilon}{\|\tilde{g}\|}\right]$ and $\xi' \in \partial f_j(R_x(t'\tilde{g}))$ such that

$$\langle \beta_{t'\tilde{g}}^{-1} \mathcal{T}_{x \leftarrow R_x(t'\tilde{g})} \xi', \tilde{g} \rangle > -c\|\tilde{g}\|^2, \tag{12}$$

and

$$\xi = \beta_{t'\tilde{g}}^{-1} \mathcal{T}_{x \leftarrow R_x(t'\tilde{g})} \xi' \in G_\varepsilon(x) \setminus \mathrm{conv}W. \tag{13}$$

Proof Suppose for all $t' \in \left[0, \frac{\varepsilon}{\|\tilde{g}\|}\right]$ and $\xi' \in \partial f_j(R_x(t'\tilde{g}))$ we have

$$\langle \beta_{t'\tilde{g}}^{-1} \mathcal{T}_{x \leftarrow R_x(t'\tilde{g})} \xi', \tilde{g} \rangle \leq -c\|\tilde{g}\|^2. \tag{14}$$

Next we show that it is impossible. In fact, by Lebourg's mean value theorem, there exist $\theta \in (0,1)$ and $\tilde{\xi} \in \partial f_j\left(R_x\left(\theta\frac{\varepsilon}{\|\tilde{g}\|}\tilde{g}\right)\right)$ such that

$$f_j\left(R_x\left(\frac{\varepsilon}{\|\tilde{g}\|}\tilde{g}\right)\right) - f_j(x) = \left\langle \tilde{\xi}, \mathrm{D}R_x\left(\theta\frac{\varepsilon}{\|\tilde{g}\|}\tilde{g}\right)\left[\frac{\varepsilon}{\|\tilde{g}\|}\tilde{g}\right]\right\rangle.$$

Note that $\|\theta\frac{\varepsilon}{\|\tilde{g}\|}\tilde{g}\| < \varepsilon$, then using (3) and the locking condition (4) of the vector transport, we obtain

$$f_j\left(R_x\left(\frac{\varepsilon}{\|\tilde{g}\|}\tilde{g}\right)\right) - f_j(x) = \frac{\varepsilon}{\|\tilde{g}\|\beta_{\theta\frac{\varepsilon}{\|\tilde{g}\|}\tilde{g}}}\left\langle \mathcal{T}_{x \leftarrow R_x\left(\theta\frac{\varepsilon}{\|\tilde{g}\|}\tilde{g}\right)}(\tilde{\xi}), \tilde{g}\right\rangle.$$

This together with $\theta \in (0,1)$ and (14) shows that

$$f_j\left(R_x\left(\frac{\varepsilon}{\|\tilde{g}\|}\tilde{g}\right)\right) - f_j(x) \leq \frac{\varepsilon}{\|\tilde{g}\|}(-c\|\tilde{g}\|^2) = -c\varepsilon\|\tilde{g}\|,$$

which is a contradiction with (11), and therefore (12) holds.

Finally, we prove (13). Note that $t' \in \left[0, \frac{\varepsilon}{\|\tilde{g}\|}\right]$, thus $\|t'\tilde{g}\| \leq \varepsilon$, which implies that $\xi = \beta_{t'\tilde{g}}^{-1} \mathcal{T}_{x \leftarrow R_x(t'\tilde{g})} \xi' \in G_\varepsilon(x)$. If $\xi \in \mathrm{conv}W$, we have $\langle \xi, \tilde{g} \rangle \leq -\|\tilde{g}\|^2 < -c\|\tilde{g}\|^2$, which is a contradiction with (12). Thus (13) holds. $\square$

Lemma 3.4 above only implies that there exist t' and ξ' satisfying (12) without showing a way how to obtain them. Now, we present a procedure ($\mathcal{P}_{\rm ns}$) which can help us to compute such t' and ξ' in practice. For simplicity, denote

$$h_j(t) := f_j(R_x(t\tilde{g})) - f_j(x) + ct\|\tilde{g}\|^2$$

for $j \in I$, i.e., f_j is not satisfied with (10).

Procedure: Find a new ε-subgradient: $(t, \tilde{\xi}_s^j) = \mathcal{P}_{\rm ns}(j, x_k, \tilde{g}_s, \varepsilon, c)$

0 Set $a = 0, b = \frac{\varepsilon}{\|\tilde{g}_s\|}$ and $t = \frac{a+b}{2}$.
1 Compute $\tilde{\xi}_s^j \in \partial f_j(R_{x_k}(t\tilde{g}_s))$.
2 If $\langle \beta_{t\tilde{g}_s}^{-1} \mathcal{T}_{x_k \leftarrow R_{x_k}(t\tilde{g}_s)} \tilde{\xi}_s^j, \tilde{g}_s \rangle > -c\|\tilde{g}_s\|^2$ then STOP.
3 If $h_j(b) > h_j(t)$ set $a = t$. Else set $b = t$.
4 Set $t = \frac{a+b}{2}$ and go to step 1.

The following theorem shows some important properties of the procedure $\mathcal{P}_{\rm ns}$, which can be viewed as an extension of [24, Lem. 3.4].

Theorem 3.2 *For the current point x_k, let $j \in I$ and $\{t_i\}$ be the sequence generated by the procedure $\mathcal{P}_{\rm ns}$.*

(i) If $\{t_i\}$ is finite, then some ξ' was found to satisfy (12).

(ii) If $\{t_i\}$ is infinite, then it converges to some $\bar{t} \in \left[0, \frac{\varepsilon}{\|\tilde{g}_s\|}\right]$ such that either there is some $\xi' \in \partial f_j(R_{x_k}(\bar{t}\tilde{g}_s))$ satisfying (12) or $0 \in \partial h_j(\bar{t})$.

Proof (i) If $\{t_i\}$ is finite, by construction, the procedure $\mathcal{P}_{\rm ns}$ must have stopped in step 2, then some $\xi' = \tilde{\xi}_s^j$ was found to satisfy (12).

(ii) If $\{t_i\}$ is infinite, it is clear that $\{t_i\}$ must be convergent to some $\bar{t} \in \left[0, \frac{\varepsilon}{\|\tilde{g}_s\|}\right]$. Additionally, we have $h_j(0) = 0$ and $h_j\left(\frac{\varepsilon}{\|\tilde{g}_s\|}\right) > 0$ since (10) is violated for the index j. Let $\{a_i\}$ and $\{b_i\}$ be the sequences corresponding to a and b in procedure $\mathcal{P}_{\rm ns}$.

We first show that $\bar{t} \neq 0$. Suppose by contradiction that $\bar{t} = 0$. By the construction of $\mathcal{P}_{\rm ns}$, we have $h_j(t_i) \geq h_j(b_i)$ for all $i \in \mathbb{N}$. Then

$$h_j(t_i) \geq h_j(b_i) = h_j(t_{i-1}) \geq h_j(b_{i-1}) = \cdots = h_j(t_1) \geq h_j(b_1) = h_j\left(\frac{\varepsilon}{\|\tilde{g}_s\|}\right) > 0.$$

Due to the continuity of h_j, we obtain $h_j(0) = \lim_{i\to\infty} h_j(t_i) \geq h_j\left(\frac{\varepsilon}{\|\tilde{g}_s\|}\right) > 0$, which is a contradiction.

So we must have $\bar{t} > 0$. Furthermore, it is clear that $h_j(b_i) > h_j(a_i)$ for all $i \in \mathbb{N}$ by the construction of $\mathcal{P}_{\rm ns}$. Since the function h_j is locally Lipschiz continuous on $\left[0, \frac{\varepsilon}{\|\tilde{g}_s\|}\right]$, by the mean value theorem there is some $r_i \in [a_i, b_i]$ such that

$$0 < h_j(b_i) - h_j(a_i) \in (b_i - a_i)\partial h_j(r_i).$$

It is obvious that $\lim_{i\to\infty} r_i = \bar{t}$ and $a_i < b_i$, thus we have $\partial h_j(r_i) \cap \mathbb{R}_+ \neq \emptyset$ for all $i \in \mathbb{N}$. Due to the upper semicontinuity of ∂h_j, there must be some $v \in \partial h_j(\bar{t})$ with $v \geq 0$. By [26, Prop. 3.1], we obtain

$$0 \leq v \in \partial h_j(\bar{t}) \subseteq \langle \partial f_j(R_{x_k}(\bar{t}\tilde{g}_s)), \mathrm{D}R_{x_k}(\bar{t}\tilde{g}_s)[\tilde{g}_s]\rangle + c\|\tilde{g}_s\|^2. \tag{15}$$

Thus, if there is some $\xi_0 \in \partial f_j(R_{x_k}(\bar{t}\tilde{g}_s))$ with $0 < \langle \xi_0, \mathrm{D}R_{x_k}(\bar{t}\tilde{g}_s)[\tilde{g}_s]\rangle + c\|\tilde{g}_s\|^2$, i.e. $\langle \xi_0, \mathrm{D}R_{x_k}(\bar{t}\tilde{g}_s)[\tilde{g}_s]\rangle > -c\|\tilde{g}_s\|^2$, then using (3) and the locking condition (4) of vector transport, we have that

$$\langle \beta_{\bar{t}\tilde{g}_s}^{-1} \mathcal{T}_{x_k \leftarrow R_{x_k}(\bar{t}\tilde{g}_s)} \xi_0, \tilde{g}_s\rangle > -c\|\tilde{g}_s\|^2,$$

which shows that $\xi' = \xi_0 \in \partial f_j(R_{x_k}(\bar{t}\tilde{g}_s))$ satisfies (12). Otherwise, we obtain

$$\langle \xi, \mathrm{D}R_{x_k}(\bar{t}\tilde{g}_s)[\tilde{g}_s]\rangle + c\|\tilde{g}_s\|^2 \leq 0, \quad \forall \xi \in \partial f_j(R_{x_k}(\bar{t}\tilde{g}_s)).$$

This along (15) implies $0 = v \in \partial h_i(\bar{t})$. □

We note that the procedure $\mathcal{P}_{\mathrm{ns}}$ will stop after finitely many iterations in practice; see [24, Remark 3.1]. Based on this procedure, it is possible to construct another procedure that can compute an acceptable descent direction of f_i for $i \in \{1 \cdots, m\}$, namely procedure $\mathcal{P}_{\mathrm{dd}}$ used in step 1 of Algorithm 1.

Procedure: Compute a descent direction: $g_k = \mathcal{P}_{\mathrm{dd}}(x_k, \varepsilon, \delta, c)$

0 Compute $\xi^i \in \partial_\varepsilon f_i(x_k)$ for all $i \in \{1, \cdots, m\}$. Set $W_0 = \{\xi^1, \cdots, \xi^m\}$ and $s = 0$.

1 Compute $\tilde{g}_s = \underset{g \in -\mathrm{conv} W_s}{\mathrm{argmin}} \|g\|$.

2 If $\|\tilde{g}_s\| \leq \delta$, set $g_k = \tilde{g}_s$ and STOP.

3 Find all indices for which there is no sufficient decrease:

$$I_s = \left\{ j \in \{1, \cdots, m\} : \ f_j\left(R_{x_k}\left(\frac{\varepsilon}{\|\tilde{g}_s\|}\tilde{g}_s\right)\right) > f_j(x_k) - c\varepsilon\|\tilde{g}_s\| \right\}.$$

If $I_s = \emptyset$, set $g_k = \tilde{g}_s$, then STOP.

4 For each $j \in I_s$, compute $(t, \tilde{\xi}_s^j) = \mathcal{P}_{\mathrm{ns}}(j, x_k, \tilde{g}_s, \varepsilon, c)$, and set $\xi_s^j = \beta_{t\tilde{g}_s}^{-1} \mathcal{T}_{x_k \leftarrow R_{x_k}(t\tilde{g}_s)} \tilde{\xi}_s^j$.

5 Set $W_{s+1} = W_s \cup \{\xi_s^j : \ j \in I_s\}$, $s = s + 1$ and go to step 1.

We show that the procedure $\mathcal{P}_{\mathrm{dd}}$ is well defined, that is, it terminates in a finite number of iterations, and then an acceptable descent direction is produced.

Theorem 3.3 *The procedure $\mathcal{P}_{\mathrm{dd}}$ terminates in a finite number of iterations. In addition, let $\hat{g}$ be the last element of $\{\tilde{g}_s\}$, then either $\|\hat{g}\| \leq \delta$ or $\hat{g}$ is an acceptable descent direction, that is*

$$f_i\left(R_{x_k}\left(\frac{\varepsilon}{\|\hat{g}\|}\hat{g}\right)\right) \leq f_i(x_k) - c\varepsilon\|\hat{g}\|, \quad \forall i \in \{1, \cdots, m\}.$$

Proof Suppose by contradiction that $\{\tilde{g}_s\}$ is an infinite sequence. Let $s \geq 1$ and $j \in I_{s-1}$. By the construction of $\mathcal{P}_{\rm dd}$, it follows that $\xi^j_{s-1} \in W_s$ and $-\tilde{g}_{s-1} \in {\rm conv}W_{s-1} \subseteq {\rm conv}W_s$. Since $\tilde{g}_s = {\rm argmin}_{g\in -{\rm conv}W_s}\|g\|$, for all $\lambda \in (0,1)$, we have

$$\begin{aligned}\|\tilde{g}_s\|^2 &< \|-\tilde{g}_{s-1} + \lambda(\xi^j_{s-1} + \tilde{g}_{s-1})\| \\ &= \|\tilde{g}_{s-1}\|^2 - 2\lambda\langle\tilde{g}_{s-1}, \xi^j_{s-1}\rangle - 2\lambda\|\tilde{g}_{s-1}\|^2 + \lambda^2\|\xi^j_{s-1} + \tilde{g}_{s-1}\|^2. \quad (16)\end{aligned}$$

Note that $j \in I_{s-1}$, then by step 4 of $\mathcal{P}_{\rm dd}$ and $\mathcal{P}_{\rm ns}$, we obtain

$$\langle\tilde{g}_{s-1}, \xi^j_{s-1}\rangle > -c\|\tilde{g}_{s-1}\|^2. \quad (17)$$

Additionally, since $G_\varepsilon(x_k)$ is a compact subset of $T_{x_k}\mathcal{M}$, there is a constant $C > 0$ such that $\|\xi\| \leq C$ for all $\xi \in G_\varepsilon(x)$. Thus

$$\|\xi^j_{s-1} + \tilde{g}_{s-1}\| \leq 2C. \quad (18)$$

Combining (16) with (17) and (18), we have

$$\begin{aligned}\|\tilde{g}_s\|^2 &< \|\tilde{g}_{s-1}\|^2 + 2\lambda c\|\tilde{g}_{s-1}\|^2 - 2\lambda\|\tilde{g}_{s-1}\|^2 + 4\lambda^2C^2 \\ &= \|\tilde{g}_{s-1}\|^2 - 2\lambda(1-c)\|\tilde{g}_{s-1}\|^2 + 4\lambda^2C^2.\end{aligned}$$

Let $\lambda = \frac{1-c}{4C^2}\|\tilde{g}_{s-1}\|^2$, then it follows from $c \in (0,1)$ and $\|g_{k-1}\| \leq C$ that $\lambda \in (0,1)$. Therefore

$$\begin{aligned}\|\tilde{g}_s\|^2 &< \|\tilde{g}_{s-1}\|^2 - 2\frac{(1-c)^2}{4C^2}\|\tilde{g}_{s-1}\|^4 + \frac{(1-c)^2}{4C^2}\|\tilde{g}_{s-1}\|^4 \\ &= \left(1 - \frac{(1-c)^2}{4C^2}\|\tilde{g}_{s-1}\|^2\right)\|\tilde{g}_{s-1}\|^2.\end{aligned}$$

Since the $\mathcal{P}_{\rm dd}$ does not terminate, it holds $C \geq \|\tilde{g}_{s-1}\| > \delta$. Thus

$$\|\tilde{g}_s\|^2 < \left(1 - \left(\frac{1-c}{2C}\delta\right)^2\right)\|\tilde{g}_{s-1}\|^2.$$

Set $\tau = 1 - (\frac{1-c}{2C}\delta)^2 \in (0,1)$. By recursion, we obtain

$$\|\tilde{g}_s\|^2 < \tau\|\tilde{g}_{s-1}\|^2 < \tau^2\|\tilde{g}_{s-2}\|^2 < \cdots < \tau^{s-1}\|\tilde{g}_1\|^2 \leq \tau^{s-1}C^2.$$

This shows that $\|\tilde{g}_s\| \leq \delta$ for sufficiently large s, which is a contradiction. □

4 Convergence analysis

In this section, we establish the convergence of Algorithm 1. The conception of (ε, δ)-critical is extended from $\mathbb{R}^n$ (see [24, Def. 3.2]) to Riemannian manifolds. Under the assumption of at least one objective function of problem (1) is bounded below, we show that the sequence $\{x_k\}$ generated by Algorithm 1 is finite with the last element being (ε, δ)-critical.

Definition 4.1 Let $x \in \mathcal{M}$, $\varepsilon \in \left(0, \frac{1}{2}\iota(\mathcal{M})\right)$ and $\delta > 0$. Then, x is called (ε, δ)-critical, if

$$\min_{g \in -G_\varepsilon(x)} \|g\| \leq \delta.$$

Clearly, if a point $x \in \mathcal{M}$ satisfies (5), then it is an (ε, δ)-critical point, but the converse is not necessarily true.

Theorem 4.1 *Assume that at least one objective function of problem (1) is bounded below. Let $\{x_k\}$ be the sequence generated by Algorithm 1. Then $\{x_k\}$ is finite with the last element being (ε, δ)-critical.*

Proof Suppose by contradiction that $\{x_k\}$ is infinite. Then, we have $\|g_k\| > \delta$ for all $k \in \mathbb{N}$. If $t_k = \alpha^{-\ell} t_0$ in step 3 of Algorithm 1, then we have $\alpha^{-\ell} t_0 \geq \frac{\varepsilon}{\|g_k\|}$. This together with (3) shows that, for all $i \in \{1, \cdots, m\}$,

$$\begin{aligned} f_i(R_{x_k}(t_k g_k)) - f_i(x_k) &= f_i(R_{x_k}(\alpha^{-\ell} t_0 g_k)) - f_i(x_k) \\ &\leq -\alpha^{-\ell} t_0 c \|g_k\|^2 \\ &\leq -c\varepsilon \|g_k\| \\ &< -c\varepsilon\delta. \end{aligned} \tag{19}$$

Conversely, if $t_k = \frac{\varepsilon}{\|g_k\|}$, we have $\frac{\varepsilon}{\|g_k\|} \geq \alpha^{-\ell} t_0$, then from Theorem 3.3, the last inequality in (19) can be also obtained. In summary, we can conclude that $\{f_i(x_k)\}$ is unbounded below for each $i \in \{1, \cdots, m\}$, which is a contradiction.Thus the sequence $\{x_k\}$ is finite.

Let x_* and g_* be the last elements of $\{x_k\}$ and $\{g_k\}$, respectively. Since the algorithm stopped, we must have $\|g_*\| \leq \delta$ by step 2 of Algorithm 1. On the other hand, by the construction of the procedures $\mathcal{P}_{\text{dd}}$ and $\mathcal{P}_{\text{ns}}$, there is a set $W_* \subseteq G_\varepsilon(x_*)$ such that $g_* = \operatorname{argmin}_{g \in -\operatorname{conv} W_*} \|g\|$. Thus

$$\min_{g \in -G_\varepsilon(x_*)} \|g\| \leq \min_{g \in -\operatorname{conv}(W_*)} \|g\| = \|g_*\| \leq \delta,$$

which completes the proof. □

5 Numerical results

In this section, we will present numerical results of several examples for our method. Most of the objective functions of these examples are of the classic optimization problems on Riemannian manifolds.

Example 5.1 We first consider a simple problem. Let $m = 2$ in problem (1), and set $\mathcal{M} = S^2$ which is the Euclidean unit sphere in $\mathbb{R}^3$, $f_1(x) = \max(0.5x_1 + x_3, 0.3x_2 + 1.5x_3)$ and $f_2(x) = |x_1 - 0.5| + x_2 + x_3$.

Example 5.2 Recently, many researchers are interested in the geometric median on a Riemannian manifold $\mathcal{M}$ (see[31, 22]). Let $y_1, \cdots, y_q \in \mathcal{M}$ be some given points, $w = (w_1, \cdots, w_q)^T \in \mathbb{R}^q_+$ and $\sum_{j=1}^q w_j = 1$ be the corresponding weights. This problem is to minimize $\sum_{j=1}^q w_j \mathrm{dist}(x, y_j)$ on $\mathcal{M}$. Now, we consider the multiobjective setting and set $\mathcal{M} = S^{p-1}$ which is the Euclidean unit sphere in $\mathbb{R}^p$ and $f_i(x) = \sum_{j=1}^{q^i} w^i_j \mathrm{dist}(x, y^i_j)$, where $y^i_1, \cdots, y^i_{q^i} \in \mathcal{M}$, $w^i \in \mathbb{R}^{q^i}_+$ with $\sum_{j=1}^{q^i} w^i_j = 1$ for all $i \in \{1, \cdots, m\}$.

Example 5.3 Eigenvalue problems are ubiquitous in scientific research and practical applications, such as physical science and engineering design, etc. Let A be a real symmetric matrix, the eigenvalue problem can be transformed into a Rayleigh quotient problem whose objective function is $\frac{x^T Ax}{x^T x}$. This problem can be further viewed as an optimization problem on a sphere to minimize $x^T Ax$; see [2]. Also, we consider the multiobjective setting, set $\mathcal{M} = S^{p-1}$ and $f_i(x) = x^T A_i x$, where A_i is a real symmetric matrix for each $i \in \{1, \cdots, m\}$.

Example 5.4 The l_1-regularized least squares problem (named as Lasso) was proposed in [39], which has been used heavily in machine learning and basis pursuit denoising, etc. The cost function of this problem is $\frac{1}{2}\|Ax - b\|^2 + \lambda\|x\|_1$, where $A \in \mathbb{R}^{n\times p}$, $b \in \mathbb{R}^p$ and $\lambda > 0$. Here, we restrict x to the unit sphere and consider the objective functions $f_i(x) = \frac{1}{2}\|A_i x - b_i\|^2 + \lambda_i\|x\|_1$, where $A_i \in \mathbb{R}^{n\times p}$, $b_i \in \mathbb{R}^p$ and $\lambda_i > 0$ for all $i \in \{1, \cdots, m\}$.

On sphere S^{p-1}, the Riemannian metric is inherited from the ambient space $\mathbb{R}^p$, and the Riemannian distance $\mathrm{dist}(x, y) = \arccos\langle x, y\rangle$. Moreover, for all instances, the exponential map and the parallel transport are employed as a retraction and vector transport, respectively. More precisely, the retraction is as follows

$$R_x(\xi) := \exp_x(\xi) = \cos(\|\xi\|)x + \sin(\|\xi\|)\frac{\xi}{\|\xi\|},$$

where $\xi \in T_x S^{p-1}$. The vector transport associated with R is given by

$$\mathcal{T}_{x\to\gamma(t)} := \left(I_p + (\cos(\|\dot\gamma(0)t\|) - 1)uu^T - \sin(\|\dot\gamma(0)t\| x u^T\right)\xi,$$

where γ is a geodesic on S^{p-1} with $\gamma(0) = x$ and $u = \frac{\dot\gamma(0)}{\|\dot\gamma(0)\|}$. Note that $\mathcal{T}_{x\leftarrow y}(\xi_y) = \mathcal{T}_{y\to\sigma(1)}$, where $\sigma(t) = \exp_y(tv)$ denotes the geodesic connecting y to x, and v can be computed by $v = \mathrm{dist}(x, y)\frac{(I - xx^T)(y - x)}{\|(I - xx^T)(y - x)\|}$. Therefore

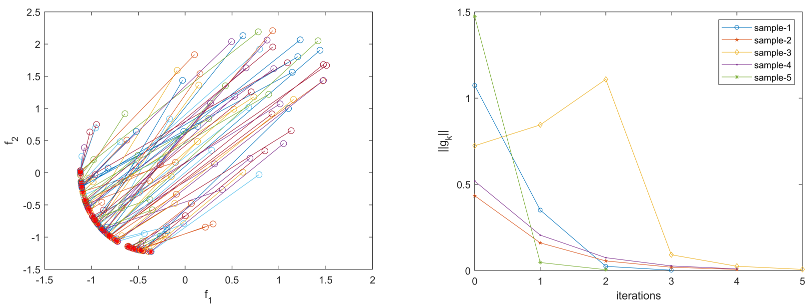
$$\mathcal{T}_{x\leftarrow y}(\xi_y) = \left(I_p + (\cos\|v\| - 1)uu^T - \sin\|v\| yu^T\right)\xi_y \;\text{ with }\; u = v/\|v\|,$$

which is well defined for all $y \neq \pm x$; see [27].

All tests are implemented in MATLAB R2018b using IEEE double precision arithmetic and run on a laptop equipped with Intel Core i7, CPU 2.60

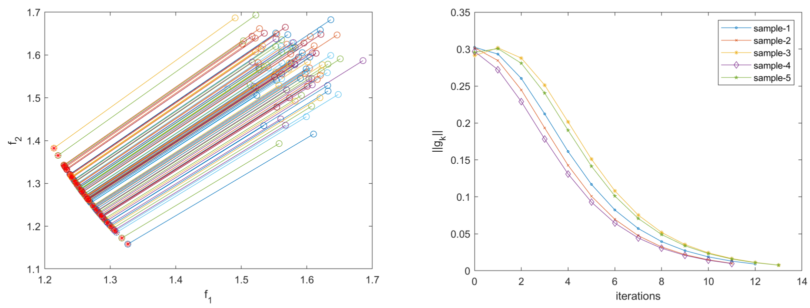
GHz and 16 GB of RAM. The quadratic programming solver **quadprog.m** in the MATLAB optimization toolbox is used to solve the convex quadratic problem in step 1 of the procedure $\mathcal{P}_{\mathrm{dd}}$. For all examples, we set the algorithm parameters as follows: $\varepsilon = 10^{-4}$, $\delta = 10^{-3}$, $c = 0.25$, $\alpha = 2$, $t_0 = 1$.

The numerical results are shown in Figs. 1–7. In particular, the left of each picture shows the value space generated by our algorithm for 100 random starting points, and the right is the variation of the norm of g_k with the number of iterations for five different random starting points. In Fig. 2, we set $p = 100$, $m = 2$, $q^1 = 6$, $w^1 = (0.1, 0.1, 0.1, 0.2, 0.2, 0.3)^T$, $q^2 = 4$, $w^2 = (0.1, 0.2, 0.3, 0.4)^T$ and y^i_j is randomly generated for $i = 1, 2$, $j = 1, \cdots, q^i$. In Fig. 3, we set $p = 100$, $m = 3$, $q^1 = 6$, $w^1 = (0.1, 0.1, 0.1, 0.2, 0.2, 0.3)^T$, $q^2 = 4$, $w^2 = (0.1, 0.2, 0.3, 0.4)^T$, $q^3 = 5$, $w^3 = (0.1, 0.1, 0.2, 0.3, 0.3)$, and y^i_j is randomly generated for $i = 1, 2, 3$, $j = 1, \cdots, q^i$.



(a) (b)

Fig. 1 Numerical results for Example 5.1.



(a) (b)

Fig. 2 Numerical results for Example 5.2 with $m = 2$.

In Figs. 1(a)–7(a), the hollow points indicate the objective vector values of the initial points, and these marked with red stars are the objective vector values of the final points (namely, the (ε, δ)-critical points). From these figures, we see that nearly all of the final points generated by our method are (approximate) Pareto optimal points for the corresponding examples, except that several points marked with black circles in Fig. 4(a) are not Pareto optimal,

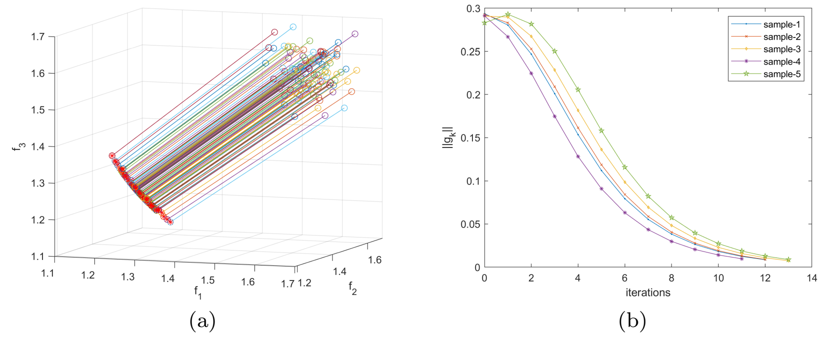


Fig. 3 Numerical results for Example 5.2 with $m = 3$.

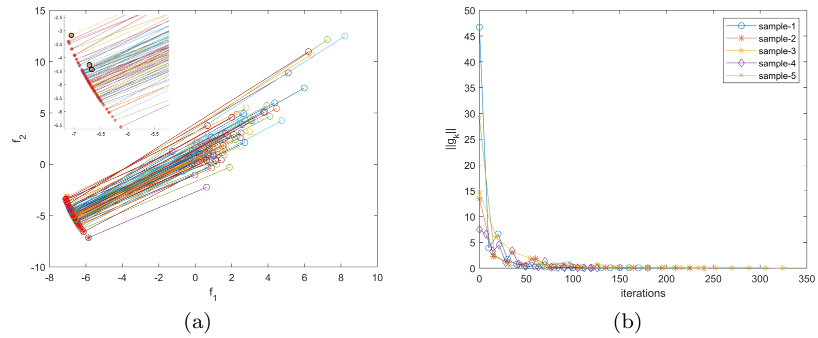


Fig. 4 Numerical results for Example 5.3 with $p = 100$, $m = 2$, and A_i being randomly generated for $i = 1, 2$.

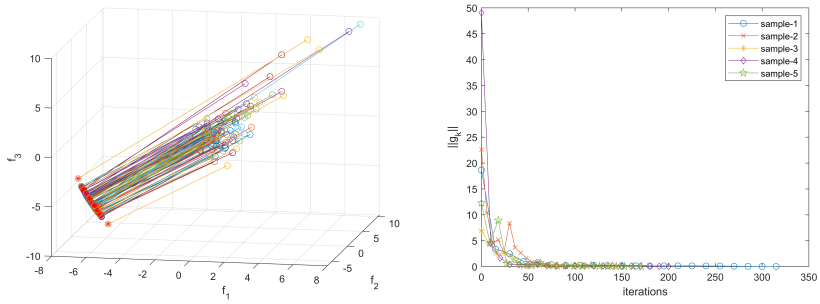
which might be local (weak) Pareto optimal points. Thus, we can obtain the approximations of Pareto sets for the above examples by Algorithm 1 when some reasonable number of starting points are given. Furthermore, in Table 1, we list the average number of iterations (ANI) for 100 random starting points for all examples. In summary, the preliminary numerical results show that our method is effective and promising.

Table 1 Average number of iterations

Example	5.1	5.2 ($m = 2$)	5.2 ($m = 3$)	5.3 ($m = 2$)	5.3 ($m = 3$)	5.4 ($m = 2$)	5.4 ($m = 3$)
ANI	4.4	13.0	13.0	266.3	192.4	98.8	102.1

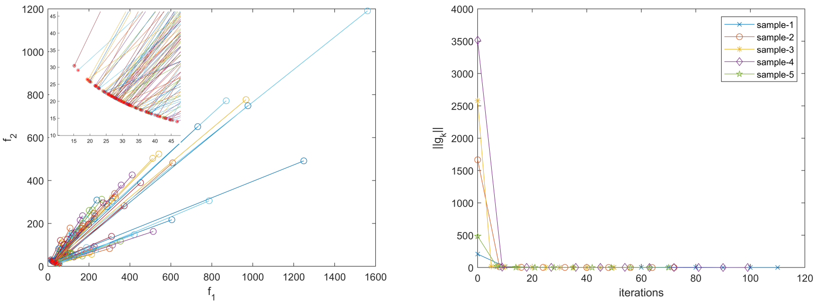
6 Conclusions

In this paper, we have presented a descent method for multiobjective optimization problems with locally Lipschitz components on complete Riemannian manifolds. Our setting is much more general than certain convexities assumed



(a) (b)

Fig. 5 Numerical results for Example 5.3 with $p = 100$, $m = 3$, and A_i being randomly generated for $i = 1, 2, 3$.

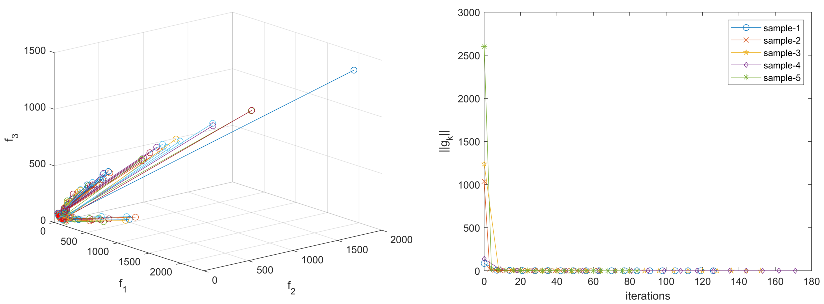


(a) (b)

Fig. 6 Numerical results for Example 5.4 with $n = 100$, $p = 60$, $m = 2$, $\lambda_1 = 0.01$, $\lambda_2 = 0.02$, A_i and b_i being randomly generated for $i = 1, 2$.

in the existing works. To avoid computing the Riemannian ε-subdifferential of the objective vector function, a convex hull of some Riemannian ε-subgradients is constructed to obtain an acceptable descent direction, which greatly reduces the computational complexity. Furthermore, we extend a necessary condition of the Pareto optimal points to the Riemannian setting. Finite convergence of the proposed algorithm is obtained under the assumption that at least one objective function is bounded below and the employed retraction and vector transport satisfy certain conditions. Finally, some preliminary numerical results illustrate the effectiveness of our method. As a future work, similar to the idea in [29], we may further extend the norm in (9) by the P-norm and choose the direction as $P\tilde{g}$, where P is a positive definite matrix.

Funding This work was supported by the National Natural Science Foundation of China (12271113, 12171106, 12061013) and Guangxi Natural Science Foundation (2020GXNSFDA238017).



(a) (b)

Fig. 7 Numerical results for Example 5.4 with $n = 100$, $p = 60$, $m = 3$, $\lambda_1 = 0.01$, $\lambda_2 = 0.02$, $\lambda_3 = 0.02$, A_i and b_i being randomly generated for $i = 1, 2, 3$.

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

References

1. Abraham, A., Jain, L.: Evolutionary Multiobjective Optimization. Springer (2005)
2. Absil, P.A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press (2008)
3. Attouch, H., Garrigos, G., Goudou, X.: A dynamic gradient approach to Pareto optimization with nonsmooth convex objective functions. J. Math. Anal. Appl. **422**(1), 741-771 (2015)
4. Bagchi, U.: Simultaneous minimization of mean and variation of flow time and waiting time in single machine systems. Oper. Res. **37**(1), 118-125 (1989)
5. Boumal, N.: An Introduction to Optimization on Smooth Manifolds. Cambridge University Press (2023)
6. Bento, G.C., Cruz Neto, J.X.: A subgradient method for multiobjective optimization on Riemannian manifolds. J. Optim. Theory Appl. **159**(1), 125-137 (2013)
7. Bento, G.C., Melo, J.G.: Subgradient method for convex feasibility on Riemannian manifolds. J. Optim. Theory Appl. **152**(3), 773-785 (2012)
8. Bonnel, H., Iusem, A.N., Svaiter, B.F.: Proximal methods in vector optimization. SIAM J. Optim. **15**(4), 953-970 (2005)
9. Bento, G.C., Ferreira, O.P., Oliveira, P.R.: Unconstrained steepest descent method for multicriteria optimization on Riemannian manifolds. J. Optim. Theory Appl. **154**, 88-107 (2012)
10. Bento, G.C., Cruz Neto, J.X., Santos, P.S.M: An inexact steepest descent method for multicriteria optimization on Riemannian manifolds. J. Optim. Theory Appl. **159**, 108-124 (2013)
11. Bento, G.C., Cruz Neto, J.X., López, G., Soubeyran, A., Souza, J.C.O.: The proximal point method for locally Lipschitz functions in multiobjective optimization with application to the compromise problem. SIAM J. Optim. **28**(2), 1104-1120 (2018)
12. Bento, G.C., Cruz Neto, J.X., Meireles, L.V.: Proximal point method for locally Lipschitz functions in multiobjective optimization of Hadamard manifolds. J. Optim. Theory Appl. **179**, 37-52 (2018)
13. Clarke, F.: Optimization and Nonsmooth Analysis. Society for Industrial and Applied Mathematics (1983)
14. Deb, K.: Multi-objective optimisation using evolutionary algorithms: an introduction. Springer (2011)

15. Ehrgott, M.: Multicriteria Optimization. Springer, New York (2005)
16. El Moudden, M., El Mouatasim, A.: Accelerated diagonal steepest descent method for unconstrained multiobjective optimization. J. Optim. Theory Appl. **188**(1), 220-242 (2021)
17. Engau, A., Wiecek, M.M.: 2D decision-making for multicriteria design optimization. Structural and Multidisciplinary Optimization. **34**, 301-315 (2007)
18. Eslami, N., Najafi, B., Vaezpour, S.M.: A trust region method for solving multicriteria optimization problems on Riemannian manifolds. J. Optim. Theory Appl. **196**(1), 212-239 (2023)
19. Fliege, J.: OLAF-a general modeling system to evaluate and optimize the location of an air polluting facility. OR Spektrum. **23**(1), 117–136 (2001)
20. Fliege, J., Svaiter, B.F.: Steepest descent methods for multicriteria optimization. Math. Methods Oper. Res. **51**(3), 479-494 (2000)
21. Fliege, J., Drummond, L.G., Svaiter, B.F.: Newton's method for multiobjective optimization. SIAM J. Optim. **20**(2), 602-626 (2009)
22. Fletcher, P. T., Venkatasubramanian, S., Joshi, S.: The geometric median on Riemannian manifolds with application to robust atlas estimation. NeuroImage. **45**(1), S143-S152 (2009)
23. Geoffrion, A.M.: Proper efficiency and the theory of vector maximization. J. Math. Anal. Appl. **22**(3), 618-630 (1968)
24. Gebken, B., Peitz, S.: An efficient descent method for locally Lipschitz multiobjective optimization problems. J. Optim. Theory Appl. **188**(3), 696-723 (2021)
25. Gravel, M., Martel, J.M., Madeau, R., Price, W., Tremblay, R.: A multicriterion view of optimal ressource allocation in job-shop production. Eur. J. Oper. Res. **61**(1-2), 230-244 (1992)
26. Hosseini, S., Pouryayevali, M.R.: Generalized gradients and characterization of epi-Lipschitz sets in Riemannian manifolds. Nonlinear Anal. **74**(12), 3884-3895 (2011)
27. Hosseini, S., Uschmajew, A.: A Riemannian gradient sampling algorithm for nonsmooth optimization on manifolds. SIAM J. Optim. **27**(1), 173-189 (2017)
28. Huang, W., Gallivan, K.A., Absil, P.A.: A Broyden class of quasi-Newton methods for Riemannian optimization. SIAM J. Optim. **25**(3), 1660-1685 (2015)
29. Hosseini, S., Huang, W., Yousefpour, R.: Line search algorithms for locally Lipschitz functions on Riemannian manifolds. SIAM J. Optim. **28**(1), 596-619 (2018)
30. Hu, J., Liu, X., Wen, Z.W., Yuan, Y.X.: A brief introduction to manifold optimization. J. Oper. Res. Soc. China **8**(2), 199-248 (2020)
31. Hoseini Monjezi, N., Nobakhtian, S., Pouryayevali, M.R.: A proximal bundle algorithm for nonsmooth optimization on Riemannian manifolds. IMA J. Numer. Anal. **43**(1), 293-325 (2023)
32. Leschine, T.M., Wallenius, H., Verdini, W.A.: Interactive multiobjective analysis and assimilative capacity-based ocean disposal decisions. Eur. J. Oper. Res. **56**(2), 278-289 (1992)
33. Mahdavi-Amiri, N., Yousefpour, R.: An effective nonsmooth optimization algorithm for locally Lipschitz functions. J. Optim. Theory Appl. **155**(1), 180-195 (2012)
34. Mäkelä, M.M., Eronen, V.P., Karmitsa, N.: On Nonsmooth Multiobjective Optimality Conditions with Generalized Convexities, pp. 333–357. Springer, New York (2014)
35. Mäkelä, M.M., Karmitsa, N., Wilppu, O.: Proximal bundle method for nonsmooth and nonconvex multiobjective optimization. Mathematical modeling and optimization of complex structures, pp. 191-204. Springer (2016)
36. Povalej, Ž.: Quasi-Newton's method for multiobjective optimization. J. Comput. Appl. Math. **255**, 765-777 (2014)
37. Sato, H.: Riemannian Optimization and Its Applications. Springer Nature, Switzerland (2021)
38. Sawaragi, Y., Nakayama, H., Tanino, T.: Theory of multiobjective optimization. Elsevier (1985).
39. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B. **58**(1), 267-288 (1996)
40. Zhu, X.: A Riemannian conjugate gradient method for optimization on the Stiefel manifold. Comput. Optim. Appl. **67**, 73-110 (2017)