

Hybrid Riemannian Conjugate Gradient Methods with Global Convergence Properties *

Hiroyuki Sakai Hideaki Iiduka

May 26, 2020

Abstract

This paper presents Riemannian conjugate gradient methods and global convergence analyses under the strong Wolfe conditions. The main idea of the proposed methods is to combine the good global convergence properties of the Dai-Yuan method with the efficient numerical performance of the Hestenes-Stiefel method. One of the proposed algorithms is a generalization to Riemannian manifolds of the hybrid conjugate gradient method of the Dai and Yuan in Euclidean space. The proposed methods are compared well numerically with the existing methods for solving several Riemannian optimization problems.

1 Introduction

This paper focuses on the conjugate gradient method. Nonlinear conjugate gradient methods in Euclidean space are a class of important methods for solving unconstrained optimization problems. In [10], Hestenes and Stiefel developed a conjugate gradient method for solving linear systems with a symmetric positive-definite matrix of coefficients. In [7], Fletcher and Reeves extended the conjugate gradient method to unconstrained nonlinear optimization problems. Theirs is the first nonlinear conjugate gradient method in Euclidean space. Al-Baali [3] indicated that the Fletcher-Reeves method converges globally and generates the descent direction with an inexact line search when the step size satisfies the strong Wolfe conditions [22, 23]. Polak and Ribière [13] introduced a conjugate gradient method with good numerical performance. Dai and Yuan [4] introduced a conjugate gradient method with a better global convergence property than that of the Fletcher-Reeves method. The Hestenes-Stiefel and Polak-Ribière-Polyak methods do not always converge under the strong Wolfe conditions, and for this reason, hybrid conjugate gradient methods have been presented in [5, 11, 19]. Touati-Ahmed and Storey [19], and Hu and Storey [11] proposed methods combining the Fletcher-Reeves and Polak-Ribière-Polyak methods. Moreover, Dai and Yuan [5] proposed the hybrid conjugate gradient

*This work was supported by JSPS KAKENHI Grant Number JP18K11184.

method, which combines the Dai-Yuan method and the Hestenes-Stiefel method. These nonlinear conjugate gradient methods in Euclidean space are summarized by Hager and Zhang in [8].

The conjugate gradient method in Euclidean space is applicable to a Riemannian manifold. In [18], Smith introduced the notion of Riemannian optimization using the exponential map and parallel translation. However, using the exponential map or parallel translation on a Riemannian manifold is generally not computationally efficient. Absil, Mahony, and Sepulchre [2] proposed to use a mapping called a retraction that approximates the exponential map. Moreover, they introduced the notion of vector transport, which approximates parallel transport. In addition, Ring and Wirth [14] introduced generalized line search methods (e.g., the Wolfe conditions [22, 23]) on Riemannian manifolds.

Using the retraction and vector transport, Ring and Wirth [14] presented a Fletcher-Reeves type of nonlinear conjugate gradient method on Riemannian manifolds. They indicated that the Fletcher-Reeves methods have a global convergence property under the strong Wolfe conditions. However, their convergence analysis assumed that the vector transport does not increase the norm of the search direction vector, which is not the standard assumption (see [16, Section 5]). To remove this unnatural assumption, Sato and Iwai [16] introduced the notion of scaled vector transport [16, Definition 2.2]. They proved that by using scaled vector transport, the Fletcher-Reeves method on a Riemannian manifold generates a descent direction at every iteration and converges globally without impractical assumptions. Similarly, in [15], Sato used scaled vector transport in a convergence analysis. He indicated that the Dai-Yuan-type Riemannian conjugate gradient method generates a descent direction at every iteration and converges globally under the Wolfe conditions. This means that the Dai-Yuan method has a better global convergence property than that of the Fletcher-Reeves method on Riemannian manifolds, since the latter has to resort to the *strong* Wolfe conditions, whereas the former only requires the Wolfe conditions.

In this paper, we propose hybrid Riemannian conjugate gradient methods exploiting the idea used in the paper [5]. One of the methods we propose has already been used in numerical experiments (e.g., [9, (43)], [17, Table 1]), but no convergence analysis has yet been presented for it. Our methods combine the good numerical performance of the Hestenes-Stiefel method with the efficient global convergence property of the Dai-Yuan method. Moreover, we present convergence analyses of our methods. The proofs are along the lines of [5, Theorem 2.3], except that the step-size assumption is stronger than that of the Euclidean case. This is due to the use of scaled vector transport. Our hybrid methods converge globally if the size of the parameter, which is used to determine the search direction, with respect to that of the Dai-Yuan method is in a certain range (Theorem 3.2). We provide two examples which satisfy such a condition. In numerical experiments, we show that our hybrid methods outperform the Dai-Yuan and Polak-Ribière-Polyak methods.

This paper is organized as follows. Section 2 reviews the fundamentals of Riemannian geometry and Riemannian optimization. Section 3 proposes the hy-

brid Riemannian conjugate gradient methods and presents global convergence analyses for them. Section 4 compares our methods with the existing Riemannian conjugate gradient methods through numerical experiments. Section 5 concludes the paper with mention of future work.

2 Riemannian Conjugate Gradient Methods

Let us start by reviewing the nonlinear conjugate gradient methods in Euclidean space. The search direction η_k of the nonlinear conjugate gradient method is determined by $\eta_0 = -\nabla f(x_0)$ and

$$\eta_{k+1} = -\nabla f(x_{k+1}) + \beta_{k+1}\eta_k, \quad (1)$$

where $x_0 \in \mathbb{R}^n$, $\beta_0 = 0$, and β_k is a parameter to be suitably defined. Well-known formulas for β_k are the Fletcher-Reeves (FR) [7], Dai-Yuan (DY) [4], Polak-Ribière-Polyak (PRP) [13], and Hestenes-Stiefel (HS) [10] formulas, given by

$$\beta_k^{\text{FR}} = \frac{\|\nabla f(x_k)\|^2}{\|\nabla f(x_{k-1})\|^2}, \quad (2)$$

$$\beta_k^{\text{DY}} = \frac{\|\nabla f(x_k)\|^2}{\eta_{k-1}^\top y_{k-1}}, \quad (3)$$

$$\beta_k^{\text{PRP}} = \frac{\nabla f(x_k)^\top y_{k-1}}{\|\nabla f(x_{k-1})\|^2}, \quad (4)$$

$$\beta_k^{\text{HS}} = \frac{\nabla f(x_k)^\top y_{k-1}}{\eta_{k-1}^\top y_{k-1}}, \quad (5)$$

respectively, where $y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$.

In the Euclidean space setting, a line search optimization algorithm updates the current iterate x_k to the next iterate x_{k+1} with the updating formula,

$$x_{k+1} = x_k + \alpha_k \eta_k, \quad (6)$$

where $\alpha_k > 0$ is a positive step size. One often chooses a step size $\alpha_k > 0$ to satisfy the Wolfe conditions [22, 23], namely,

$$f(x_k + \alpha_k \eta_k) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^\top \eta_k, \quad (7)$$

$$\nabla f(x_k + \alpha_k \eta_k)^\top \eta_k \geq c_2 \nabla f(x_k)^\top \eta_k, \quad (8)$$

where $0 < c_1 < c_2 < 1$. When the step size satisfies the following condition, which is a substitute of (8):

$$|\nabla f(x_k + \alpha_k \eta_k)^\top \eta_k| \leq c_2 |\nabla f(x_k)^\top \eta_k|, \quad (9)$$

we call (7) and (9) the strong Wolfe conditions.

In [5], Dai and Yuan proved that the method defined by (1) and (6) produces a descent search direction at every iteration and converges globally if the step size $\alpha_k > 0$ satisfies (7) and (8), and β_k satisfies

$$-\sigma \leq \frac{\beta_k}{\beta_k^{\text{DY}}} \leq 1,$$

where $\sigma := (1 - c_2)/(1 + c_2)$ and c_2 is a constant in the second condition (8). In this paper, we extend these choices of the parameter β_k to Riemannian manifolds.

Now we will briefly outline Riemannian optimization, especially the Riemannian conjugate gradient method, by summarizing [2]. Moreover, we will introduce relevant notation of Riemannian geometry.

Let (M, g) be a Riemannian manifold with a Riemannian metric g , and let $T_x M$ be the tangent vector space of M at a point of $x \in M$. In addition, let TM be the tangent bundle of M , which is defined by $TM = \bigcup_{x \in M} T_x M$. Let $f : M \rightarrow \mathbb{R}$ be a smooth objective function. Throughout this paper, to simplify the notation, we will write the Riemannian metric $g(\cdot, \cdot)$ as $\langle \cdot, \cdot \rangle$. Given a smooth function $f : M \rightarrow \mathbb{R}$, the gradient of f at a point $x \in M$, denoted by $\text{grad } f(x)$, is defined as the unique element of $T_x M$ that satisfies

$$df_x(\xi) = \langle \text{grad } f(x), \xi \rangle_x \quad (\xi \in T_x M).$$

An unconstrained optimization problem on a Riemannian manifold M is expressed as follows:

Problem 2.1. *Let $f : M \rightarrow \mathbb{R}$ be smooth. Then, we would like to*

$$\begin{aligned} & \text{minimize} \quad f(x), \\ & \text{subject to} \quad x \in M. \end{aligned}$$

In order to generalize line search optimization algorithms to Riemannian manifolds, we will use the notions of a retraction and vector transport (see [2]), which are defined as follows:

Definition 2.1 (Retraction). *Let M be a manifold and TM be a tangent bundle of a manifold M . Any smooth map $R : TM \rightarrow M$ is called a retraction on M , if it has the following properties.*

- $R_x(0_x) = x$, where 0_x denotes the zero element of $T_x M$;
- With the canonical identification $T_{0_x} T_x M \simeq T_x M$, R_x satisfies $DR_x(0_x)[\xi] = \xi$ for all $\xi \in T_x M$,

where R_x denotes the restriction of R to $T_x M$ and DR is the differential of R (see [2, Section 3]).

Definition 2.2 (Vector transport). *Let M be a manifold and TM be a tangent bundle of M . Any smooth map $\mathcal{T} : TM \oplus TM \rightarrow TM$, where \oplus denotes the Whitney sum, is called vector transport on M , if it has the following properties.*

- There exists a retraction R , called the retraction associated with \mathcal{T} , such that $\mathcal{T}_\eta(\xi) \in T_{R_x(\eta)}M$ for all $x \in M$, and for all $\eta, \xi \in T_xM$;
- $\mathcal{T}_{0_x}(\xi) = \xi$ for all $\xi \in T_xM$;
- $\mathcal{T}_\eta(a\xi + b\zeta) = a\mathcal{T}_\eta(\xi) + b\mathcal{T}_\eta(\zeta)$ for all $a, b \in \mathbb{R}$, and for all $\eta, \xi, \zeta \in T_xM$.

where $\mathcal{T}_\eta(\xi)$ denotes $\mathcal{T}(\eta, \xi)$.

In this paper, we will focus on the differentiated retraction \mathcal{T}^R as a vector transport, defined by

$$\mathcal{T}_\eta^R(\xi) := DR_x(\eta)[\xi] \quad (\xi \in T_xM), \quad (10)$$

where $x \in M$ and $\eta \in T_xM$. It is easy to prove that \mathcal{T}^R satisfies the properties of Definition 2.1 (see [2, Chapter 8]).

In Riemannian optimization, by using a retraction R and vector transport \mathcal{T} on M , we can generalize the updating formula (6) and the search direction of the conjugate gradient method (1) to, respectively,

$$x_{k+1} = R_{x_k}(\alpha_k \eta_k), \quad (11)$$

$$\eta_{k+1} = -\text{grad}f(x_{k+1}) + \beta_{k+1} \mathcal{T}_{\alpha_k \eta_k}(\eta_k), \quad (12)$$

where $\alpha_k > 0$ is a positive step size (see [2]). We call the search direction η_k a descent direction if η_k satisfies

$$\langle \text{grad}f(x_k), \eta_k \rangle_{x_k} < 0.$$

Moreover, the line search conditions (7) and (8) can be generalized to Riemannian manifolds as follows:

$$f(R_{x_k}(\alpha_k \eta_k)) \leq f(x_k) + c_1 \alpha_k \langle \text{grad}f(x_k), \eta_k \rangle_{x_k}, \quad (13)$$

$$\langle \text{grad}f(R_{x_k}(\alpha_k \eta_k)), DR_{x_k}(\alpha_k \eta_k)[\eta_k] \rangle_{R_{x_k}(\alpha_k \eta_k)} \geq c_2 \langle \text{grad}f(x_k), \eta_k \rangle_{x_k}, \quad (14)$$

where $0 < c_1 < c_2 < 1$ (see [15, 16]). We call (13) the Armijo condition. Moreover, the second of the strong Wolfe conditions (9) can be rewritten as

$$\left| \langle \text{grad}f(R_{x_k}(\alpha_k \eta_k)), DR_{x_k}(\alpha_k \eta_k)[\eta_k] \rangle_{R_{x_k}(\alpha_k \eta_k)} \right| \leq c_2 \left| \langle \text{grad}f(x_k), \eta_k \rangle_{x_k} \right|. \quad (15)$$

Sato and Iwai [16] introduced the notion of scaled vector transport. A scaled vector transport of the k -th iterate $\mathcal{T}^{(k)}$ associated with \mathcal{T}^R is defined by

$$\mathcal{T}_{\alpha_k \eta_k}^{(k)}(\eta_k) := \begin{cases} \mathcal{T}_{\alpha_k \eta_k}^R(\eta_k), & \text{if } \|\mathcal{T}_{\alpha_k \eta_k}^R(\eta_k)\|_{x_{k+1}} \leq \|\eta_k\|_{x_k}, \\ \frac{\|\eta_k\|_{x_k}}{\|\mathcal{T}_{\alpha_k \eta_k}^R(\eta_k)\|_{R_{\alpha_k \eta_k}(\eta_k)}} \mathcal{T}_{\alpha_k \eta_k}^R(\eta_k), & \text{otherwise.} \end{cases} \quad (16)$$

Note that scaled vector transport does not satisfy the properties of Definition 2.2. Thus, we cannot call this vector transport with mathematical exactitude; however, by using scaled vector transport, we often obtain good convergence properties for the Riemannian conjugate gradient methods.

Scaled vector transport $\mathcal{T}^{(k)}$ satisfies the following inequalities:

$$\left| \left\langle \text{grad}f(x_{k+1}), \mathcal{T}_{\alpha_k \eta_k}^{(k)}(\eta_k) \right\rangle_{x_{k+1}} \right| \leq \left| \left\langle \text{grad}f(x_{k+1}), \mathcal{T}_{\alpha_k \eta_k}^R(\eta_k) \right\rangle_{x_{k+1}} \right| \quad (17)$$

and

$$\left\| \mathcal{T}_{\alpha_k \eta_k}^{(k)}(\eta_k) \right\|_{x_{k+1}} \leq \|\eta_k\|_{x_k}. \quad (18)$$

Now, we would like to verify that inequality (17) holds. From the definition of scaled vector transport (16), we obtain

$$\left| \left\langle \text{grad}f(x_{k+1}), \mathcal{T}_{\alpha_k \eta_k}^{(k)}(\eta_k) \right\rangle_{x_{k+1}} \right| = \left| \left\langle \text{grad}f(x_{k+1}), s^{(k)} \mathcal{T}_{\alpha_k \eta_k}^R(\eta_k) \right\rangle_{x_{k+1}} \right|,$$

where $s^{(k)}$ denotes

$$s^{(k)} := \min \left\{ 1, \frac{\|\eta_k\|_{x_k}}{\|\mathcal{T}_{\alpha_k \eta_k}^R(\eta_k)\|_{x_{k+1}}} \right\} \leq 1.$$

Therefore, it follows that

$$\begin{aligned} \left| \left\langle \text{grad}f(x_{k+1}), \mathcal{T}_{\alpha_k \eta_k}^{(k)}(\eta_k) \right\rangle_{x_{k+1}} \right| &= s^{(k)} \left| \left\langle \text{grad}f(x_{k+1}), \mathcal{T}_{\alpha_k \eta_k}^R(\eta_k) \right\rangle_{x_{k+1}} \right| \\ &\leq \left| \left\langle \text{grad}f(x_{k+1}), \mathcal{T}_{\alpha_k \eta_k}^R(\eta_k) \right\rangle_{x_{k+1}} \right|, \end{aligned}$$

which leads to (17). Obviously, (16) implies (18).

Throughout this paper, we will replace vector transport \mathcal{T} by scaled vector transport $\mathcal{T}^{(k)}$ in (12). Therefore, the $(k+1)$ -th search direction of the Riemannian conjugate gradient method is determined by

$$\eta_{k+1} = -\text{grad}f(x_{k+1}) + \beta_{k+1} \mathcal{T}_{\alpha_k \eta_k}^{(k)}(\eta_k). \quad (19)$$

In (19), β_{k+1} is also given by generalizations of the formulas (2), (3), (4), and

(5), i.e.,

$$\beta_k^{\text{FR}} = \frac{\|\text{grad}f(x_k)\|_{x_k}^2}{\|\text{grad}f(x_{k-1})\|_{x_{k-1}}^2}, \quad (20)$$

$$\beta_k^{\text{DY}} = \frac{\|\text{grad}f(x_k)\|_{x_k}^2}{\left\langle \text{grad}f(x_k), \mathcal{T}_{\alpha_{k-1}\eta_{k-1}}^{(k-1)}(\eta_{k-1}) \right\rangle_{x_k} - \left\langle \text{grad}f(x_{k-1}), \eta_{k-1} \right\rangle_{x_{k-1}}}, \quad (21)$$

$$\beta_k^{\text{PRP}} = \frac{\left\langle \text{grad}f(x_k), \text{grad}f(x_k) - \mathcal{T}_{\alpha_{k-1}\eta_{k-1}}^{(k-1)}(\text{grad}f(x_{k-1})) \right\rangle_{x_k}}{\|\text{grad}f(x_{k-1})\|_{x_{k-1}}^2}, \quad (22)$$

$$\beta_k^{\text{HS}} = \frac{\left\langle \text{grad}f(x_k), \text{grad}f(x_k) - \mathcal{T}_{\alpha_{k-1}\eta_{k-1}}^{(k-1)}(\text{grad}f(x_{k-1})) \right\rangle_{x_k}}{\left\langle \text{grad}f(x_k), \mathcal{T}_{\alpha_{k-1}\eta_{k-1}}^{(k-1)}(\eta_{k-1}) \right\rangle_{x_k} - \left\langle \text{grad}f(x_{k-1}), \eta_{k-1} \right\rangle_{x_{k-1}}}. \quad (23)$$

We call these formulas the Fletcher-Reeves, Dai-Yuan, Polak-Ribière-Polyak and Hestenes-Stiefel formulas, respectively. In the next section, we propose a new choice of β_k .

In [16], Sato and Iwai proved that by using the scaled vector transport $\mathcal{T}^{(k)}$ substitute of \mathcal{T} in (12) and a step size which satisfies the strong Wolfe conditions (13) and (15), the Fletcher-Reeves type conjugate gradient method defined by (11), (19), and (20) generates sequences that converge globally. Similarly, in [15], Sato indicated that if we use scaled vector transport, with a step size satisfying the Wolfe conditions, (13) and (14), the Dai-Yuan type conjugate gradient method defined by (11), (19), and (21) generates globally convergent sequences.

3 Riemannian Hybrid Conjugate Gradient Method and Its Global Convergence Analysis

3.1 Proposed hybrid Riemannian conjugate gradient method

This section describes the Riemannian conjugate gradient descent method using a hybrid β_k , which exploits the idea described in [5].

Let r_k be the size of β_k with respect to β_k^{DY} defined by (21), namely,

$$r_k := \frac{\beta_k}{\beta_k^{\text{DY}}}. \quad (24)$$

We will prove that, for the method defined by (11) and (19), the search direction η_k is a descent direction at every iteration and the method converges globally if the step size $\alpha_k > 0$ satisfies the strong Wolfe conditions (13) and (15), and the scalar β_k is such that

$$-\sigma \leq r_k \leq 1, \quad (25)$$

where $\sigma := (1 - c_2)/(1 + c_2) > 0$ and c_2 denotes the constant in the second of the strong Wolfe conditions (15). Furthermore, since the following two choices of β_k :

$$\beta_k = \max\{0, \min\{\beta_k^{\text{DY}}, \beta_k^{\text{HS}}\}\} \quad (26)$$

and

$$\beta_k = \max\{-\sigma\beta_k^{\text{DY}}, \min\{\beta_k^{\text{DY}}, \beta_k^{\text{HS}}\}\} \quad (27)$$

satisfy the condition (25), we can use either of these hybrid formulas β_k defined by (26) and (27) as the scalar in (19). The above two choices of β_k are examples of the hybrid methods in Euclidean space [5]. This implies that our hybrid method is a generalization of the method in [5]. The parameter (26) is used in the numerical experiments of [9, (43)] and [17, Table 1]. The hybrid methods using (26) and (27) combine the good global convergence properties of the Dai-Yuan method (21) with the efficient numerical performance of the Hestenes-Stiefel method (23). Now, we note that, in Euclidean space, the hybrid methods using (26) and (27) converge globally under the Wolfe conditions (7) and (8), whereas, on a Riemannian manifold, the hybrid methods need the strong Wolfe conditions (13) and (15) to converge globally. In Section 4, we provide a numerical evaluation showing that the Riemannian conjugate gradient methods with the hybrid β_k defined by (26) and (27) perform better than the Polak-Ribière-Polyak method.

3.2 Global convergence analysis

Zoutendijk's theorem is described on Riemannian manifolds as follows:

Theorem 3.1 (Zoutendijk [14]). *Let (M, g) be a Riemannian manifold and R be a retraction on M . Let $f : M \rightarrow \mathbb{R}$ be a smooth, bounded below function with the following property: there exists $L > 0$ such that*

$$|D(f \circ R_x)(t\eta)[\eta] - D(f \circ R_x)(0_x)[\eta]| \leq Lt \quad (\eta \in T_x M, \|\eta\|_x = 1, x \in M, t \geq 0).$$

Suppose that in the line search optimization algorithm (11), each step size $\alpha_k > 0$ satisfies the strong Wolfe conditions (13) and (15), and each search direction η_k is a descent direction. Then the following series converges:

$$\sum_{k=0}^{\infty} \frac{\langle \text{grad} f(x_k), \eta_k \rangle_{x_k}^2}{\|\eta_k\|_{x_k}^2} < \infty. \quad (28)$$

The proof of this theorem is along the lines of Zoutendijk's theorem in Euclidean space (see [14, Theorem 3.3]). Next, we will prove the main convergence theorem.

Theorem 3.2. *Let $f : M \rightarrow \mathbb{R}$ be a function satisfying the assumptions of Zoutendijk's theorem. If each $\alpha_k > 0$ satisfies the strong Wolfe conditions (13)*

and (15), and if β_k is such that¹ $-\sigma \leq r_k \leq 1$, then any sequence $\{x_k\}$ generated by the Riemannian conjugate gradient method defined by (11) and (19) satisfies

$$\liminf_{k \rightarrow \infty} \|\text{grad}f(x_k)\|_{x_k} = 0. \quad (29)$$

Let us start with a brief outline of the proof strategy of Theorem 3.2, with an emphasis on the main difficulty that has to be overcome in order to generalize the proof in [5, Theorem 2.3] to manifolds. The flow of our proof is the same as in [5]. First, we show that the search direction in each iteration of the hybrid methods is the descent direction. Therefore, the assumption, "each search direction η_k is a descent direction", of Zoutendijk's theorem is satisfied. Then, assuming that equation (29) does not hold, the proof is completed by deriving a contradiction.

In general Riemannian manifolds, the inner product of tangent vectors at different points cannot be defined, so the inner product is taken using scaled vector transport. However, the use of scaled vector transport causes a problem that does not occur in Euclidean space. Specifically, the absolute value is required for the inequality in (36) when generalizing to the Riemannian manifold.

Theorem 3.2. If $\text{grad}f(x_{k_0}) = 0$ for some k_0 , then (29) follows. Thus, it is sufficient to prove (29) only when $\text{grad}f(x_k) \neq 0$ for all $k \geq 0$.

First, we prove that each search direction η_k is a descent direction by induction. For $\eta_0 = -\text{grad}f(x_0)$, it is obvious that η_0 is a descent direction.

Assume that η_{k-1} is a descent direction. Then, we find that

$$\begin{aligned} & \langle \text{grad}f(x_k), \eta_k \rangle_{x_k} \\ &= \left\langle \text{grad}f(x_k), -\text{grad}f(x_k) + \beta_k \mathcal{T}_{\alpha_{k-1}\eta_{k-1}(\eta_{k-1})}^{(k-1)} \right\rangle_{x_k} \\ &= -\|\text{grad}f(x_k)\|_{x_k}^2 + r_k \frac{\|\text{grad}f(x_k)\|_{x_k}^2 \left\langle \text{grad}f(x_k), \mathcal{T}_{\alpha_{k-1}\eta_{k-1}(\eta_{k-1})}^{(k-1)} \right\rangle_{x_k}}{\left\langle \text{grad}f(x_k), \mathcal{T}_{\alpha_{k-1}\eta_{k-1}(\eta_{k-1})}^{(k-1)} \right\rangle_{x_k} - \langle \text{grad}f(x_{k-1}), \eta_{k-1} \rangle_{x_{k-1}}} \\ &= \frac{\langle \text{grad}f(x_{k-1}), \eta_{k-1} \rangle_{x_{k-1}}}{\left\langle \text{grad}f(x_k), \mathcal{T}_{\alpha_{k-1}\eta_{k-1}(\eta_{k-1})}^{(k-1)} \right\rangle_{x_k} - \langle \text{grad}f(x_{k-1}), \eta_{k-1} \rangle_{x_{k-1}}} \|\text{grad}f(x_k)\|_{x_k}^2 \\ &\quad + \frac{(r_k - 1) \left\langle \text{grad}f(x_k), \mathcal{T}_{\alpha_{k-1}\eta_{k-1}(\eta_{k-1})}^{(k-1)} \right\rangle_{x_k}}{\left\langle \text{grad}f(x_k), \mathcal{T}_{\alpha_{k-1}\eta_{k-1}(\eta_{k-1})}^{(k-1)} \right\rangle_{x_k} - \langle \text{grad}f(x_{k-1}), \eta_{k-1} \rangle_{x_{k-1}}} \|\text{grad}f(x_k)\|_{x_k}^2 \end{aligned} \quad (30)$$

where the first equation comes from (19) and the second equation comes from $\beta_k = r_k \beta_k^{\text{DY}}$ and (21). Accordingly, (21) ensures that

$$\begin{aligned} & \langle \text{grad}f(x_k), \eta_k \rangle_{x_k} \\ &= \left\{ \langle \text{grad}f(x_{k-1}), \eta_{k-1} \rangle_{x_{k-1}} + (r_k - 1) \left\langle \text{grad}f(x_k), \mathcal{T}_{\alpha_{k-1}\eta_{k-1}(\eta_{k-1})}^{(k-1)} \right\rangle_{x_k} \right\} \beta_k^{\text{DY}}, \end{aligned}$$

¹The formulas defined by (26) and (27) satisfy $-\sigma \leq r_k \leq 1$.

which, together with (24), implies that

$$\begin{aligned}\beta_k &= r_k \beta_k^{\text{DY}} \\ &= \frac{r_k \langle \text{grad} f(x_k), \eta_k \rangle_{x_k}}{\langle \text{grad} f(x_{k-1}), \eta_{k-1} \rangle_{x_{k-1}} + (r_k - 1) \langle \text{grad} f(x_k), \mathcal{T}_{\alpha_{k-1} \eta_{k-1}}^{(k-1)}(\eta_{k-1}) \rangle_{x_k}}.\end{aligned}$$

Let l_k and ξ_k be

$$l_k := \frac{\langle \text{grad} f(x_k), \mathcal{T}_{\alpha_{k-1} \eta_{k-1}}^{(k-1)}(\eta_{k-1}) \rangle_{x_k}}{\langle \text{grad} f(x_{k-1}), \eta_{k-1} \rangle_{x_{k-1}}}, \quad (31)$$

$$\xi_k := \frac{r_k}{1 + (r_k - 1)l_k}. \quad (32)$$

Using (31) and (32), we obtain

$$\begin{aligned}\beta_k &= r_k \beta_k^{\text{DY}} \\ &= \xi_k \frac{\langle \text{grad} f(x_k), \eta_k \rangle_{x_k}}{\langle \text{grad} f(x_{k-1}), \eta_{k-1} \rangle_{x_{k-1}}}.\end{aligned} \quad (33)$$

Furthermore, let ζ_k be

$$\zeta_k := \frac{1 + (r_k - 1)l_k}{l_k - 1}. \quad (34)$$

Then, (30) guarantees that

$$\langle \text{grad} f(x_k), \eta_k \rangle_{x_k} = \zeta_k \|\text{grad} f(x_k)\|_{x_k}^2. \quad (35)$$

On the other hand, since α_k satisfies the strong Wolfe conditions, (15) implies that

$$\left| \langle \text{grad} f(x_k), \text{DR}_{x_{k-1}}(\alpha_{k-1} \eta_{k-1})[\eta_{k-1}] \rangle_{x_k} \right| \leq c_2 \left| \langle \text{grad} f(x_{k-1}), \eta_{k-1} \rangle_{x_{k-1}} \right|,$$

which, together with (10), (17) and (31) implies that

$$\begin{aligned}|l_k| &= \frac{\left| \langle \text{grad} f(x_k), \mathcal{T}_{\alpha_{k-1} \eta_{k-1}}^{(k-1)}(\eta_{k-1}) \rangle_{x_k} \right|}{\left| \langle \text{grad} f(x_{k-1}), \eta_{k-1} \rangle_{x_{k-1}} \right|} \\ &\leq \frac{\left| \langle \text{grad} f(x_k), \mathcal{T}_{\alpha_{k-1} \eta_{k-1}}^R(\eta_{k-1}) \rangle_{x_k} \right|}{\left| \langle \text{grad} f(x_{k-1}), \eta_{k-1} \rangle_{x_{k-1}} \right|} \\ &= \frac{\left| \langle \text{grad} f(x_k), \text{DR}_{x_{k-1}}(\alpha_{k-1} \eta_{k-1})[\eta_{k-1}] \rangle_{x_k} \right|}{\left| \langle \text{grad} f(x_{k-1}), \eta_{k-1} \rangle_{x_{k-1}} \right|} \leq c_2.\end{aligned} \quad (36)$$

This means $|l_k| \leq c_2 < 1$, which implies $l_k - 1 < 0$. Similar to equation (2.18) in [5], we obtain $1 + (r_k - 1)l_k > 0$. Hence,

$$\zeta_k = \frac{1 + (r_k - 1)l_k}{l_k - 1} < 0,$$

which, together with (35), implies that η_k is a descent direction. Thus, induction shows that each η_k is a descent direction.

Finally, we prove (29) by contradiction. Assume that

$$\liminf_{k \rightarrow \infty} \|\text{grad}f(x_k)\|_{x_k} > 0.$$

Then, noting $\|\text{grad}f(x_k)\|_{x_k} \neq 0$ for all k , there exists $\gamma > 0$ such that

$$\|\text{grad}f(x_k)\|_{x_k} \geq \gamma > 0.$$

for all k . Since (19) means that

$$\eta_k + \text{grad}f(x_k) = \beta_k \mathcal{T}_{\alpha_{k-1}\eta_{k-1}}^{(k-1)}(\eta_{k-1}),$$

taking the norms of the above equation and its square, it follows that

$$\|\eta_k\|_{x_k}^2 = \beta_k^2 \|\mathcal{T}_{\alpha_{k-1}\eta_{k-1}}^{(k-1)}(\eta_{k-1})\|_{x_k}^2 - 2 \langle \text{grad}f(x_k), \eta_k \rangle_{x_k} - \|\text{grad}f(x_k)\|_{x_k}^2.$$

Similar to equation (2.21) in [5], by dividing both sides of the above equation by $\langle \text{grad}f(x_k), \eta_k \rangle_{x_k}^2 \neq 0$, (33) and (35) give,

$$\begin{aligned} \frac{\|\eta_k\|_{x_k}^2}{\langle \text{grad}f(x_k), \eta_k \rangle_{x_k}^2} &= \xi_k^2 \frac{\|\mathcal{T}_{\alpha_{k-1}\eta_{k-1}}^{(k-1)}(\eta_{k-1})\|_{x_k}^2}{\langle \text{grad}f(x_{k-1}), \eta_{k-1} \rangle_{x_{k-1}}^2} \\ &\quad + \frac{1}{\|\text{grad}f(x_k)\|_{x_k}^2} \left\{ 1 - \left(1 + \frac{1}{\zeta_k} \right)^2 \right\}. \end{aligned} \quad (37)$$

Similar to equation (2.24) in [5], we obtain

$$|1 + (r_k - 1)l_k| \geq |r_k|,$$

which, together with (32), implies

$$|\xi_k| \leq 1.$$

From the above inequality with (37) and (18), we obtain

$$\begin{aligned} \frac{\|\eta_k\|_{x_k}^2}{\langle \text{grad}f(x_k), \eta_k \rangle_{x_k}^2} &\leq \frac{\|\mathcal{T}_{\alpha_{k-1}\eta_{k-1}}^{(k-1)}(\eta_{k-1})\|_{x_k}^2}{\langle \text{grad}f(x_{k-1}), \eta_{k-1} \rangle_{x_{k-1}}^2} + \frac{1}{\|\text{grad}f(x_k)\|_{x_k}^2} \\ &\leq \frac{\|\eta_{k-1}\|_{x_{k-1}}^2}{\langle \text{grad}f(x_{k-1}), \eta_{k-1} \rangle_{x_{k-1}}^2} + \frac{1}{\|\text{grad}f(x_k)\|_{x_k}^2}. \end{aligned}$$

Using the above inequality recursively and noting the hypothesis, $\|\text{grad}f(x_k)\|_{x_k} \geq \gamma > 0$, and $\|\eta_0\|_{x_0}^2 = \|\text{grad}f(x_0)\|_{x_0}^2$, it follows that

$$\frac{\|\eta_k\|_{x_k}^2}{\langle \text{grad}f(x_k), \eta_k \rangle_{x_k}^2} \leq \sum_{i=0}^k \frac{1}{\|\text{grad}f(x_i)\|_{x_i}^2} \leq \sum_{i=0}^k \frac{1}{\gamma^2} = \frac{k+1}{\gamma^2}.$$

This means

$$\frac{\langle \text{grad}f(x_k), \eta_k \rangle_{x_k}^2}{\|\eta_k\|_{x_k}^2} \geq \frac{\gamma^2}{k+1},$$

which indicates

$$\sum_{k=0}^{\infty} \frac{\langle \text{grad}f(x_k), \eta_k \rangle_{x_k}^2}{\|\eta_k\|_{x_k}^2} \geq \sum_{k=0}^{\infty} \frac{\gamma^2}{k+1} = \infty.$$

This contradicts (28) in Zoutendijk's theorem and completes the proof. \square \square

4 Numerical Experiments

This section compares the performances of the existing Riemannian conjugate gradient methods with those of the proposed methods. We solved 7 types of Riemann optimization problems (Problem 4.1–4.7) on several manifolds and objective functions. We solved these problems 10 times with each algorithm, that is, 70 times in total. Then, we calculated a performance profile [6] for each algorithm to show the advantages of our algorithms. Our experiments used the source code of `pymanopt` (<https://github.com/pymanopt>, see [20]). In particular, the Riemannian conjugate gradient method was implemented in `pymanopt`, so we changed only the parameter β_k for the experiments.

4.1 The Rayleigh-quotient minimization problem on the unit sphere

Problem 4.1 is the Rayleigh-quotient minimization problem on the unit sphere (see [2, Chapter 4.6]). The optimal solutions of Problem 4.1 are the unit eigenvectors of A associated with the smallest eigenvalue (see [2, Chapter 2]).

Problem 4.1. For $A \in \mathcal{S}_{++}^n$,

$$\begin{aligned} & \text{minimize} && f(x) = x^\top A x, \\ & \text{subject to} && x \in \mathbb{S}^{n-1} := \{x \in \mathbb{R}^n : \|x\| = 1\}, \end{aligned}$$

where \mathcal{S}_{++}^n denotes the set of all symmetric positive-definite matrices.

In the experiments, we set $n = 100$ and generated a matrix $A \in \mathcal{S}_{++}^n$ with randomly chosen elements by using `sklearn.datasets.make_spd_matrix`.

4.2 Computation of Stability Number

For an undirected graph G , a stable set in G is a set of vertices, which are mutually nonadjacent. We define $S(G)$ as the size of a maximum stable set in G . In [12], Motzkin and Straus showed that the computation of the stability number of graphs problem is equivalent to Problem 4.2. Specifically, the value of the objective function in the global optimal solution of Problem 4.2 is equal to $S(G)^{-1}$. In addition, Yuan, Gu, Lai, and Wen [24, Section 5.3] considered the problem as a Riemannian optimization problem.

Problem 4.2. *Let $G = (V, E)$ be an undirected graph.*

$$\begin{aligned} \text{minimize} \quad & f(x) = \sum_{i=1}^n x_i^4 + 2 \sum_{(i,j) \in E} x_i^2 x_j^2, \\ \text{subject to} \quad & x \in \mathbb{S}^{n-1} := \{x \in \mathbb{R}^n : \|x\| = 1\}, \end{aligned}$$

where $n = |V|$ and $\|\cdot\|$ denotes the Euclidean norm.

In the experiments, we set $n = 20$ and generated a graph $G = (V, E)$ randomly by using `networkx.fast_gnp_random_graph`. Here, we set the probability for edge creation to $1/4$.

4.3 The brockett-cost-function minimization problem on a Stiefel manifold

Problem 4.3 is the Brockett-cost-function minimization problem on a Stiefel manifold (see [2, Chapter 4.8]).

Problem 4.3. *For $A \in \mathcal{S}_{++}^n$ and $N = \text{diag}(\mu_0, \dots, \mu_p)$ ($0 \leq \mu_0 \leq \dots \leq \mu_p$),*

$$\begin{aligned} \text{minimize} \quad & f(X) = \text{tr}(X^\top A X N) \\ \text{subject to} \quad & X \in \text{St}(p, n) := \{X \in \mathbb{R}^{n \times p} : X^\top X = I_p\}. \end{aligned}$$

In the experiments, we set $p = 5$, $n = 20$ and $N := \text{diag}(1, \dots, p)$ and generated a matrix $A \in \mathcal{S}_{++}^n$ with randomly chosen elements by using `sklearn.datasets.make_spd_matrix`.

4.4 The closest unit norm column approximation problem

Problem 4.4 is the closest unit norm column approximation problem, whose implementation is given in `pymanopt/examples/closest_unit_norm_column_approximation.py`.

Problem 4.4. *For $A \in \mathbb{R}^{m \times n}$,*

$$\begin{aligned} \text{minimize} \quad & f(X) = \|X - A\|_F^2 \\ \text{subject to} \quad & X \in \mathcal{OB}(m, n) := \{X \in \mathbb{R}^{m \times n} : \text{ddiag}(X^\top X) = I_m\}, \end{aligned}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\text{ddiag}(X)$ denotes a diagonal matrix whose diagonal elements are those of X .

In the experiments, we set $m = 10$ and $n = 1000$ and generated a matrix $A \in \mathbb{R}^{m \times n}$ with randomly chosen elements by using `numpy.random.randn`.

4.5 Off-diagonal cost function minimization

In [1, Section 3], Absil and Gallivan introduced a cost function on oblique manifolds, which is an off-diagonal cost function written as

$$f(X) := \sum_{i=1}^N \|X^\top C_i X - \text{ddiag}(X^\top C_i X)\|_F^2,$$

where C_i ($i = 1, 2, \dots, N$) are symmetric matrices. Problem 4.5 is one of minimizing the off-diagonal cost function on an oblique manifold.

Problem 4.5. For $C_i \in \mathcal{S}^n$ ($i = 1, \dots, N$),

$$\begin{aligned} \text{minimize} \quad & f(X) = \sum_{i=1}^N \|X^\top C_i X - \text{ddiag}(X^\top C_i X)\|_F^2 \\ \text{subject to} \quad & X \in \mathcal{OB}(n, p) := \{X \in \mathbb{R}^{n \times p} : \text{ddiag}(X^\top X) = I_p\}, \end{aligned}$$

where \mathcal{S}^n denotes the set of all symmetric matrices.

In the experiments, we set $N = 5$, $n = 10$ and $p = 5$ and generated 5 matrices $B_i \in \mathbb{R}^{n \times n}$ ($i = 1, 2, \dots, 5$) with randomly chosen elements by using `numpy.random.randn`. We set symmetric matrices $C_i \in \mathcal{S}^n$ as $C_i := (B_i + B_i^\top)/2$ ($i = 1, 2, \dots, 5$).

4.6 The low-rank matrix approximation problem

Problem 4.6 is the low-rank matrix approximation problem whose implementation is given in `pymanopt/examples/low_rank_matrix_approximation.py`.

Problem 4.6. For $A \in \mathbb{R}^{m \times n}$,

$$\begin{aligned} \text{minimize} \quad & f(X) = \|X - A\|_F^2, \\ \text{subject to} \quad & X \in M_k := \{X \in \mathbb{R}^{m \times n} : \text{rank}(X) = k\}. \end{aligned}$$

In the experiments, we set $m = 100$, $n = 80$ and $k = 4$ and generated a matrix $A \in \mathbb{R}^{m \times n}$ with randomly chosen elements by using `numpy.random.randn`.

4.7 The robust matrix completion problem

Problem 4.7 is the robust matrix completion problem, discussed by Vandereycken [21, Section 1.1 (1.5)].

Problem 4.7. For $A \in \mathbb{R}^{m \times n}$, and a subset Ω of the complete set of entries $\{1, \dots, m\} \times \{1, \dots, n\}$,

$$\begin{aligned} \text{minimize} \quad & f(X) = \|P_\Omega(X - A)\|_F^2, \\ \text{subject to} \quad & X \in M_k := \{X \in \mathbb{R}^{m \times n} : \text{rank}(X) = k\}, \end{aligned}$$

where

$$P_{\Omega} : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}, X_{ij} \mapsto \begin{cases} X_{ij} & (i, j) \in \Omega \\ 0 & (i, j) \notin \Omega \end{cases}.$$

In the experiments, we set $m = 10$, and $m = 8$ and $k = 4$, and Ω contained each pair $(i, j) \in \{1, \dots, m\} \times \{1, \dots, n\}$ with probability $1/2$. Moreover, we used a matrix $A \in \mathbb{R}^{m \times n}$ that was generated with randomly chosen elements by using `numpy.random.randn`.

We used line search algorithms for the strong Wolfe conditions (13) and (15) with $c_1 = 0.0001$ and $c_2 = 0.9$. We determined that a sequence had converged to an optimal solution if the stopping condition,

$$\|\text{grad}f(x_k)\|_{x_k} < 10^{-6}$$

was satisfied.

The experiments used a MacBook Air (2017) with a 1.8 GHz Intel Core i5, 8 GB 1600 MHz DDR3 memory, and macOS Mojave version 10.14.5 operating system. The algorithms were written in Python 3.7.6 with the NumPy 1.17.3 package and the Matplotlib 3.1.1 package. We modified the strong Wolfe line search provided as `scipy.optimize.line_search` in the SciPy package, to compute the step size in (11).

For comparison, we chose two Riemannian conjugate gradient methods, i.e., the Dai-Yuan method (21) and the Polak-Ribière-Polyak method (22). Below, we call the hybrid methods using (26) and (27), Hybrid1 and Hybrid2, respectively.

Table 1 and 2 summarize the results such as the average and median values of the above 70 experiments. In particular, Table 1 shows summary statistics for the number of iterations and Table 2 shows those for the elapsed time. From Table 1 and 2, we can see that the hybrid methods converge to optimal solutions in fewer iterations and in less time than the DY and PRP methods.

Table 1: Summary statistics on the iteration of 70 experiments of the Riemannian optimization problems.

	DY	PRP	Hybrid1	Hybrid2
mean	1438.7	399.9	212.2	235.0
std	1765.9	513.8	217.0	212.0
min	46	21	20	20
median	570.5	161	129	135
max	7061	2037	952	803

Then, we calculate the performance profiles [6]. The performance profile $P_s : \mathbb{R} \rightarrow [0, 1]$ is defined as follows: let \mathcal{P} and \mathcal{S} be the set of problems and solvers, respectively. For each $p \in \mathcal{P}$ and $s \in \mathcal{S}$, we define $t :=$

Table 2: Summary statistics on the elapsed time of 70 experiments of the Riemannian optimization problems.

	DY	PRP	Hybrid1	Hybrid2
mean	18.70	4.39	2.56	2.91
std	26.22	4.53	2.13	2.46
min	0.43	0.14	0.15	0.14
median	10.34	2.74	2.31	2.44
max	147.46	20.95	10.74	11.42

(computing time required to solve problem p by solver s). We define the performance ratio $r_{p,s}$ as

$$r_{p,s} := \frac{t_{p,s}}{\min_{s' \in \mathcal{S}} t_{p,s'}}.$$

Next, we define the performance profile, for all $\tau \in \mathbb{R}$, as

$$P_s(\tau) := \frac{\text{size}\{p \in \mathcal{P} : r_{p,s} \leq \tau\}}{\text{size}\mathcal{P}},$$

where $\text{size}A$ denotes the number of elements of a set A .

Figure 1 plots the performance profile of each algorithm versus the number of iterations. It shows that the hybrid methods have much higher performance than the DY method. Moreover, the hybrid methods outperform the PRP method. Also, it can be seen that Hybrid1 is superior to Hybrid2. Figure 2 plots the performance profiles of each algorithm versus the elapsed time. We can see that the hybrid methods are superior to both DY and PRP. In particular, they perform much better than the DY method. In addition, Hybrid1 is again superior to Hybrid2.

5 Conclusion and Future Work

This paper presented hybrid Riemannian conjugate gradient methods and showed their global convergence properties. It compared them numerically with the existing Riemannian conjugate gradient methods on several Riemannian optimization problems. The results of the numerical experiments demonstrated the efficiency of the hybrid methods.

Various hybrid conjugate methods have been proposed for Euclidean space, such as

$$\beta_k = \max\{0, \min\{\beta_k^{\text{PRP}}, \beta_k^{\text{FR}}\}\}.$$

The hybrid conjugate methods in Euclidean space are summarized in [8]. We will present more hybrid methods and convergence analyses in a future paper.

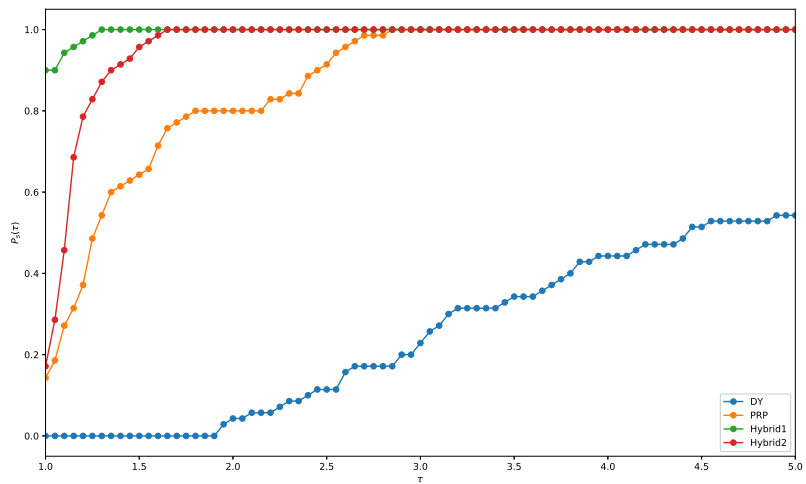


Figure 1: Performance profile versus number of iterations.

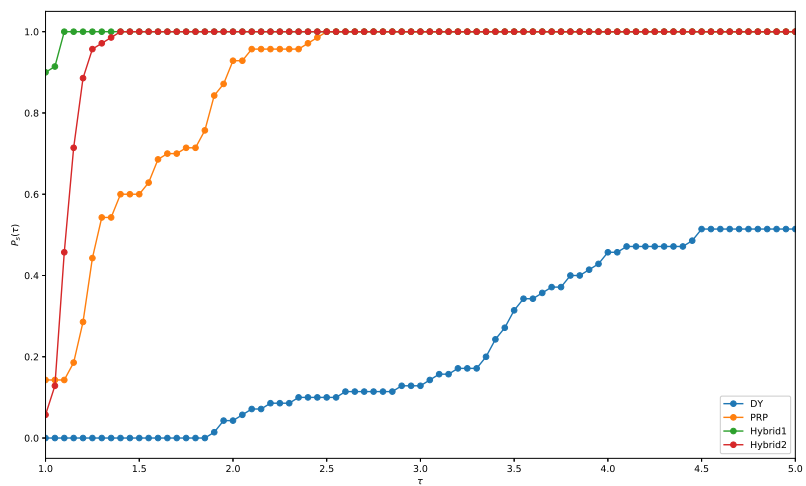


Figure 2: Performance profile versus elapsed time.

6 Acknowledgment

We are sincerely grateful to the editor and the anonymous reviewer for helping us improve the original manuscript. This work was supported by JSPS KAKENHI Grant Number JP18K11184.

References

- [1] P.-A. Absil and K. A. Gallivan. Joint diagonalization on the oblique manifold for independent component analysis. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 5, pages V–V, 2006.
- [2] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.
- [3] M. Al-Baali. Descent property and global convergence of the Fletcher-Reeves method with inexact line search. *IMA Journal of Numerical Analysis*, 5(1):121–124, 1985.
- [4] Y.-H. Dai and Y. Yuan. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM Journal on optimization*, 10(1):177–182, 1999.
- [5] Y.-H. Dai and Y. Yuan. An efficient hybrid conjugate gradient method for unconstrained optimization. *Annals of Operations Research*, 103(1-4):33–47, 2001.
- [6] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213, 2002.
- [7] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *The computer journal*, 7(2):149–154, 1964.
- [8] W. W. Hager and H. Zhang. A survey of nonlinear conjugate gradient methods. *Pacific journal of Optimization*, 2(1):35–58, 2006.
- [9] S. Hawe, M. Kleinsteuber, and K. Diepold. Analysis operator learning and its application to image reconstruction. *IEEE Transactions on Image Processing*, 22(6):2138–2150, 2013.
- [10] M. R. Hestenes and E. Stiefel. *Methods of conjugate gradients for solving linear systems*. NBS Washington, DC, 1952.
- [11] Y. Hu and C. Storey. Global convergence result for conjugate gradient methods. *Journal of Optimization Theory and Applications*, 71(2):399–405, 1991.
- [12] T. S. Motzkin and E. G. Straus. Maxima for graphs and a new proof of a theorem of turn. *Canadian Journal of Mathematics*, 17:533–540, 1965.

- [13] E. Polak and G. Ribière. Note sur la convergence de méthodes de directions conjuguées. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 3(R1):35–43, 1969.
- [14] W. Ring and B. Wirth. Optimization methods on Riemannian manifolds and their application to shape space. *SIAM Journal on Optimization*, 22(2):596–627, 2012.
- [15] H. Sato. A Dai-Yuan-type Riemannian conjugate gradient method with the weak Wolfe conditions. *Computational Optimization and Applications*, 64(1):101–118, 2016.
- [16] H. Sato and T. Iwai. A new, globally convergent Riemannian conjugate gradient method. *Optimization*, 64(4):1011–1031, 2015.
- [17] S. E. Selvan, U. Amato, K. A. Gallivan, C. Qi, M. F. Carfora, M. Larobina, and B. Alfano. Descent algorithms on oblique manifold for source-adaptive ica contrast. *IEEE Transactions on Neural Networks and Learning Systems*, 23(12):1930–1947, 2012.
- [18] S. T. Smith. Optimization techniques on Riemannian manifolds. *Fields Institute Communications*, 3(3):113–135, 1994.
- [19] D. Touati-Ahmed and C. Storey. Efficient hybrid conjugate gradient techniques. *Journal of Optimization Theory and Applications*, 64(2):379–397, 1990.
- [20] J. Townsend, N. Koep, and S. Weichwald. Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *The Journal of Machine Learning Research*, 17(1):4755–4759, 2016.
- [21] B. Vandereycken. Low-rank matrix completion by Riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013.
- [22] P. Wolfe. Convergence conditions for ascent methods. *SIAM Review*, 11(2):226–235, 1969.
- [23] P. Wolfe. Convergence conditions for ascent methods. ii: Some corrections. *SIAM Review*, 13(2):185–188, 1971.
- [24] H. Yuan, X. Gu, R. Lai, and Z. Wen. Global optimization with orthogonality constraints via stochastic diffusion on manifold. *Journal of Scientific Computing*, 80(2):1139–1170, 2019.