

**FIRST-ORDER METHODS FOR
MULTI-OBJECTIVE AND RIEMANNIAN OPTIMIZATION
PROBLEMS**

KANGMING CHEN

**FIRST-ORDER METHODS FOR
MULTI-OBJECTIVE AND RIEMANNIAN OPTIMIZATION
PROBLEMS**

by
KANGMING CHEN

Submitted in partial fulfillment of
the requirement for the degree of
DOCTOR OF INFORMATICS
(Applied Mathematics and Physics)



**KYOTO UNIVERSITY
KYOTO 606–8501, JAPAN
FEBRUARY 2026**

Preface

Multi-objective optimization constitutes a fundamental decision-making framework designed to address problems involving multiple, often conflicting, criteria. Unlike single-objective optimization, which seeks a unique solution, multi-objective optimization aims to find a set of optimal trade-offs known as the Pareto front. While various methodologies such as scalarization and evolutionary algorithms have been developed to solve these problems, contemporary research has increasingly pivoted toward first-order descent methods to address the high dimensionality and rigorous computational demands of modern applications. Rather than relying on scalarization techniques that convert multiple objectives into a single function, these descent-based methods seek a common direction to simultaneously improve all objective functions. By iteratively guiding the solution toward a Pareto stationary point, these methods provide a direct and scalable approach essential for large-scale optimization settings.

In parallel, a distinct but equally critical class of optimization problems involves constraints that confine feasible solutions to nonlinear spaces, specifically differentiable manifolds. Traditional optimization methods designed for Euclidean spaces often fail to respect the intrinsic geometry of these spaces, potentially leading to suboptimal or infeasible results. Riemannian optimization has emerged as a robust framework to address these challenges by redefining classical optimization concepts directly on the manifold. By leveraging intrinsic geometric structures, Riemannian methods offer superior performance and theoretical guarantees for problems governed by constraints such as orthogonality, fixed rank, or positive definiteness. Current advancements in this domain focus on extending the rich algorithmic landscape of Euclidean optimization—particularly first-order methods—to the Riemannian setting.

This thesis lies at the intersection of two active research domains: multi-objective optimization and Riemannian optimization. Its central aim is to develop and analyze first-order algorithms for solving challenging problems that involve multiple objectives, manifold constraints, or a combination of both.

The contributions of this work are threefold. First, we propose a proximal gradient method based on Bregman distances for multi-objective optimization. Classical proximal gradient schemes typically rely on Euclidean distances, which may fail to reflect the intrinsic geometry of the problem. By incorporating Bregman distances, the proposed method adapts to the intrinsic structure of the objective functions and provides a more flexible descent

mechanism. We establish a rigorous convergence analysis, showing that the generated sequence converges to a Pareto stationary point.

Second, we extend nonlinear conjugate gradient methods to vector optimization on Riemannian manifolds, a general framework that encompasses multi-objective optimization as a special case. This work combines vector optimization with Riemannian optimization by extending the classical conjugate gradient method to the manifold setting. We utilize retractions and vector transports to handle the manifold geometry, and establish the global convergence of the proposed method under standard assumptions.

Finally, to address composite problems where projections or proximal mappings are computationally expensive, we introduce the generalized Riemannian conditional gradient methods (generalized Frank–Wolfe algorithms). We describe the proposed algorithm, analyze its asymptotic convergence, and demonstrate its performance through applications such as sparse principal component analysis.

The research presented in this thesis builds upon the works originally published in [42, 40, 39] and provides further elaboration, theoretical depth, and discussion on related topics.

Kangming Chen
February 2026

Acknowledgments

First and foremost, I really thank my supervisor, Associate Professor Ellen Hidemi Fukuda, for her mentorship and support. Her guidance and encouragement have been fundamental to my academic growth. Her rigorous and meticulous attitude toward research has also shaped me, encouraging me to do my best to grow into a qualified researcher.

I am also sincerely grateful to Professor Nobuo Yamashita and Associate Professor Hiroyuki Sato, both highly respected experts in the field, for their helpful comments and constructive suggestions throughout the research process.

I would like to sincerely thank Professor Kenji Kashima, who is my advisor during doctoral period and provided critical comments during my thesis defense.

Also, I would like to express my sincere appreciation to Ms. Yuiko Igura, the Assistant Administrative Staff of system optimization laboratory, for her kind assistance and dedicated support in navigating various administrative matters.

I would like to thank all members of system optimization laboratory for their company.

I am also deeply grateful to my family for their constant encouragement.

Thank you to everyone who made this work possible.

Finally, I gratefully acknowledge the financial support of the Japan Science and Technology Agency (JST) through the Kyoto University Innovation Creation Fellowship and the SPRING Program. This work would not have been possible without these fellowships and the academic environment at Kyoto University.

This experience has become part of my memory and part of myself. Memory has no return, and research has no end. I am here now, and I hope to keep moving forward.

Contents

Preface	v
Acknowledgments	vii
List of Figures	xi
List of Tables	xiii
List of Notations	xv
1 Introduction	1
1.1 Multi-objective optimization	1
1.1.1 Scalarization approaches	2
1.1.2 Metaheuristics	2
1.1.3 Descent methods	2
1.2 Riemannian optimization	3
1.3 Classical first-order methods	5
1.3.1 Proximal gradient method	5
1.3.2 Conjugate gradient method	6
1.3.3 Conditional gradient method	8
1.4 Motivations and contribution	10
1.5 Outline of the thesis	11
2 Preliminaries	13
2.1 Notations and basic definitions	13
2.2 Bregman distances and relative smoothness	14
2.3 Vector and multi-objective optimization	16
2.4 Riemannian manifolds	17
2.4.1 Fundamentals of Riemannian manifolds	17
2.4.2 Operations on Riemannian manifolds	18
2.4.3 Convexity on manifolds	19

3	A proximal gradient method with Bregman distance in multi-objective optimization	21
3.1	Introduction	21
3.2	Preliminaries	23
3.3	The multi-objective proximal gradient method with Bregman distance	24
3.3.1	Constant stepsize	26
3.3.2	Backtracking procedure	26
3.4	Convergence analysis	27
3.4.1	Convergence to Pareto stationary points	27
3.4.2	Convergence rate analysis	30
3.5	Conclusion	34
4	Nonlinear conjugate gradient method for vector optimization on Riemannian manifolds with retraction and vector transport	35
4.1	Introduction	35
4.2	Nonlinear vector Riemannian conjugate gradient method	36
4.3	Global convergence analysis	39
4.3.1	NVRCG method with FR parameter	43
4.3.2	NVRCG method with CD parameter	46
4.3.3	NVRCG method with DY parameter	47
4.3.4	NVRCG method with PRP, HS, and LS parameter	48
4.4	Numerical experiments	50
4.4.1	Examples on Sphere manifold	50
4.4.2	Examples on Stiefel manifold	51
4.5	Conclusion	54
5	Riemannian conditional gradient methods for composite optimization problems	57
5.1	Introduction	57
5.2	Retraction-based convexity and smoothness	58
5.3	Riemannian generalized conditional gradient method	59
5.4	Convergence analysis	61
5.4.1	Analysis with the adaptive and the diminishing step sizes	63
5.4.2	Analysis with the Armijo step size	64
5.5	Solving the subproblem	69
5.6	Accelerated conditional gradient methods	71
5.7	Numerical experiments	72
5.7.1	Sparse PCA on the sphere manifold	72
5.7.2	Sparse PCA on the Stiefel manifold	73
5.7.3	Numerical results	74
5.8	Conclusion	82

6 Conclusions and future works	83
--------------------------------	----

List of Figures

4.1	Example (4.4.1) with A_1 and c .	52
4.2	Example (4.4.1) with A_2 and c .	52
4.3	Example (4.4.1) with A_3 and c .	52
4.4	Example (4.4.2) with random A_1 and $A_2 = A_1 + E$, where E is All-ones matrix.	53
4.5	Example (4.4.2) with random A_1 and $A_2 = A_1 + E$, where E is All-ones matrix, and $0.9\beta^{\text{FR}}$	53
4.6	Example (4.4.3) with random A_1 , B_2 , and $A_2 = A_1 + e$, $B_2 = B_1 + e$, where e is identity matrix	55
4.7	Example (4.4.3) with random A_1 , B_2 , $A_2 = A_1 + e$, $B_2 = B_1 + e$, where e is identity matrix, and 0.9β .	55
4.8	Example (4.4.3) with random A and B	55
4.9	Example (4.4.3) with random A, B and 0.9β .	55
4.10	Example (4.4.4) with random N_1 and N_2 .	56
4.11	Example (4.4.4) with random N_1 , N_2 and 0.9β .	56
4.12	Example (4.4.4) with random N_1 and $N_2 = N_1 + e$, where e is the identity matrix.	56
4.13	Example (4.4.4) with random N_1 , $N_2 = N_1 + e$ and 0.9β , where e is the identity matrix.	56
5.1	Example (5.7.1) with fixed A , different n and $\lambda = 0.1$	76
5.2	Example (5.7.1) with fixed A , $\lambda = 0.1$ and different n for RGCondG and accelerated method	78
5.3	Example (5.7.4) with fixed A , $\lambda = 0.1$, different n and p	81

List of Tables

4.1	Example (4.4.1) with $c_1 = 0.1$ and $c_2 = 0.6$	51
5.1	Average results of 10 random runs for problem (5.7.1) with $\lambda = 0.1$ and fixed A	77
5.2	Average results of 10 random runs for problem (5.7.1) with varying λ and fixed A , $n = 500$	78
5.3	Comparison of Accelerated and RGCondG Methods: Average Results of 10 Random Runs for problem (5.7.1) with $\lambda = 0.1$	79
5.4	Average results of 10 random runs for problem (5.7.4) with $\lambda = 0.1$ and fixed A	80
5.5	Average results of 10 random runs for problem (5.7.4) with varying λ and fixed A , $n = 200, p = 10$	80

List of Notations

Spaces and Sets

$\mathbb{R}^n$	The n -dimensional Euclidean space
$\mathbb{R}_+^m$	The non-negative orthant of $\mathbb{R}^m$
$\mathcal{M}$	A differentiable Riemannian manifold
$T_x\mathcal{M}$	The tangent space to $\mathcal{M}$ at a point x
$T\mathcal{M}$	The tangent bundle of $\mathcal{M}$
$T\mathcal{M} \oplus T\mathcal{M}$	The Whitney sum of the tangent bundle
Δ_n	The standard unit simplex in $\mathbb{R}^n$
$\text{dom } f$	The effective domain of a function f

Scalars, Vectors, and Matrices

x, y, z	Points in Euclidean space or on a manifold
η, ξ, ζ	Tangent vectors in $T_x\mathcal{M}$
u, v	Vectors in the objective space (e.g., $\mathbb{R}^m$)
A, B, X	Matrices
I_n	The identity matrix of size $n \times n$
x_i	The i -th component of a vector x

Norms, Inner Products, and Distances

$\ \cdot\ $	The standard Euclidean norm (i.e., $\ \cdot\ _2$)
$\langle \cdot, \cdot \rangle$	The standard Euclidean inner product
$\ \cdot\ _x$	The norm induced by the Riemannian metric on $T_x\mathcal{M}$
$\langle \cdot, \cdot \rangle_x$	The Riemannian inner product (metric) at x
$\text{dist}(x, y)$	The Riemannian (geodesic) distance between x and y
ω	The kernel (reference) function for Bregman distances
$B_\omega(x, y)$	The Bregman distance between x and y generated by ω

Calculus and Convex Analysis

$\nabla f(x)$	The Euclidean gradient of f at x
$\text{grad } f(x)$	The Riemannian gradient of f at $x \in \mathcal{M}$
$JF(x)$	The Jacobian matrix of a vector-valued function F at x
$f'(x; d)$	The directional derivative of f at x along direction d

$\partial f(x)$ The subdifferential of a convex function f at x

Riemannian Operators

$R_x(\eta)$ A retraction mapping a tangent vector $\eta \in T_x\mathcal{M}$ to $\mathcal{M}$
 $R_x^{-1}(y)$ The inverse retraction mapping a point $y \in \mathcal{M}$ to $T_x\mathcal{M}$
 $\text{Exp}_x(\eta)$ The Riemannian exponential map at x applied to η
 $\text{Exp}_x^{-1}(y)$ The Riemannian logarithm map (inverse of Exp)
 $\mathcal{T}_\eta(\xi)$ Vector transport of tangent vector ξ along direction η
 $P_\gamma^{t_1 \leftarrow t_0}$ Parallel transport along curve γ from t_0 to t_1

Vector Optimization

K A closed, convex, pointed cone inducing a partial order
 K^* The positive polar (dual) cone of K
 $\preceq_K, \prec_K$ Partial orders induced by cone K and its interior
 $\preceq, \prec$ Component-wise partial orders (induced by $\mathbb{R}_+^m$)

Chapter 1

Introduction

Mathematical optimization serves as a fundamental pillar of modern scientific inquiry, engineering design, and data-driven decision-making. It provides a rigorous framework for formulating and solving problems wherein one seeks the optimal solution within a feasible set. While classical optimization provides a solid foundation, the increasing complexity of real-world problems necessitates advanced frameworks that integrate Riemannian geometry or multi-objective criteria. This thesis focuses on two such areas: multi-objective optimization and Riemannian optimization. We investigate the theoretical and algorithmic intersection of these two fields.

In this chapter, we provide an overview of multi-objective optimization and delineate the structure of the thesis.

1.1 Multi-objective optimization

Vector optimization is concerned with the minimization of a vector-valued objective function with respect to a partial ordering induced by a cone. A prominent and widely studied instance of vector optimization is *Multi-Objective Optimization (MOO)*, where the partial ordering is defined by the non-negative orthant $\mathbb{R}_+^m$. Before presenting the problem formulation, we summarize the basic notations used throughout this thesis. Let $\mathbb{R}^n$ denote the n -dimensional Euclidean space. For any vector $x \in \mathbb{R}^n$ or matrix A , the notation $x^\top$ (or $A^\top$) denotes its transpose. Unless specified otherwise, all vectors are considered column vectors. Formally, a multi-objective optimization problem is stated as:

$$\min_{x \in \mathcal{X}} F(x) := (f_1(x), f_2(x), \dots, f_m(x))^\top, \quad (1.1.1)$$

where $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is the vector-valued objective function composed of m individual objective functions $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, and $\mathcal{X} \subseteq \mathbb{R}^n$ denotes the feasible set.

In such problems, the objective functions are inherently conflicting, meaning that the improvement of one objective generally entails the deterioration of another. Consequently, a single point that simultaneously minimizes all objectives typically does not exist. Unlike single-objective optimization, the goal here is to identify a set of optimal trade-off solutions,

known as *Pareto optimal* solutions. The set of objective vectors corresponding to these solutions comprises the *Pareto front*, which characterizes the optimal trade-offs among the conflicting criteria.

Existing methodologies for solving MOOs generally can be divided into three main categories: scalarization approaches, metaheuristics, and descent methods.

1.1.1 Scalarization approaches

A foundational and historically predominant strategy for solving MOPs is the *scalarization approach* [70], wherein the original vector-valued problem is transformed into a Single-Objective Optimization Problem (SOP). This transformation is achieved by aggregating the multiple objectives, a process typically mediated by scalar parameters. The most prominent technique within this class is the weighted sum method. This method assigns a positive weight, $w_i > 0$, to each objective function $f_i(x)$ and seeks to minimize their convex combination:

$$\min_{x \in \mathcal{X}} \sum_{i=1}^m w_i f_i(x).$$

Although this method is simple to implement and works well with standard single-objective solvers, it has significant limitations. First, the optimal weights are rarely known a priori, often requiring a heuristic and iterative tuning process. More fundamentally, for problems characterized by a non-convex Pareto front, the weighted sum method is theoretically incapable of identifying solutions situated within the non-convex regions of the front. Although alternative scalarization techniques, such as the ϵ -constraint method [104], have been proposed to address these issues, they are often computationally expensive and require careful parameter tuning.

1.1.2 Metaheuristics

A second major class of methodologies, which has garnered considerable attention for complex, non-differentiable, or “black-box” problems, is *metaheuristics*. These methods, frequently bio-inspired, operate on a population-based paradigm where a cohort of candidate solutions evolves in parallel. Multi-Objective Evolutionary Algorithms (MOEAs) serve as the primary exemplar, with seminal algorithms such as the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [51, 135] and the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) [140, 116] achieving widespread adoption.

The main advantage of these methods is their ability to produce an approximation of the entire Pareto front in a single run. However, a major limitation is the lack of theoretical convergence guarantees. Moreover, their performance is often highly sensitive to hyperparameter configurations, such as population size and mutation rates. And they usually require a large number of function evaluations. This can be too expensive for problems where the objective function is costly to compute.

1.1.3 Descent methods

In response to the theoretical limitations of metaheuristics and the parameter-dependency challenges of scalarization techniques, a significant body of research over the past two decades has focused on *multi-objective descent methods*. These methods represent direct generalizations of classical single-objective optimization algorithms. Their fundamental aim is to generate a sequence of iterates that achieve simultaneous improvement across all objective functions. The core principle involves identifying a common descent direction $d \in \mathbb{R}^n$ at a given iterate x , such that $\nabla f_i(x)^\top d < 0$ for all $i = 1, \dots, m$. This direction is typically computed by solving a tractable subproblem, often formulated as a strongly convex quadratic program.

The multi-objective steepest descent method was originally proposed by Fliege et al. [60]. This foundational work stimulated the extension of numerous classical algorithms to the multi-objective setting [64]. Following this trajectory, multi-objective variants have been developed for a variety of fundamental methods, including Newton’s method [59], conjugate gradient methods [99], quasi-Newton methods [107], projected gradient methods [62, 63, 77], and subgradient methods [48].

Proximal-based methods have also been widely studied, including the proximal point method [20, 27, 45, 76] and the proximal gradient method [16, 32, 110]. Recently, the conditional gradient (Frank-Wolfe) method was also extended to multi-objective optimization [8, 7]. Other techniques, such as trust-region [35] and augmented Lagrangian methods [47], have also been developed. A primary advantage of these methods is that they retain the theoretical convergence properties of single-objective algorithms. This makes them suitable for applications where convergence guarantees are important.

1.2 Riemannian optimization

While mathematical optimization traditionally relies on the linear structure of Euclidean spaces, many contemporary problems present a different challenge. In fields such as machine learning, computer vision, and signal processing, it is common to encounter problems where the feasible set is defined by nonlinear, curved constraints. Notable examples include orthogonality constraints, fixed-rank constraints, or the set of symmetric positive definite (SPD) matrices. When standard Euclidean optimization methods are applied, they often ignore the intrinsic geometry of these sets. This can result in overly complex formulations and numerical instability, such as iterates failing to adhere to the constraints.

Manifold optimization, also known as *Riemannian optimization*, provides an alternative approach for solving optimization problems with geometric constraints. Instead of treating the feasible set as a subset of a Euclidean space, this method considers the set itself as the domain, specifically a Riemannian manifold $\mathcal{M}$. This allows constrained problems in $\mathbb{R}^n$ to be reformulated as unconstrained problems on the manifold. The objective is then typically

expressed as:

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

where $f : \mathcal{M} \rightarrow \mathbb{R}$ is a real-valued objective function defined on the manifold. By leveraging the tools of differential geometry, algorithms can be designed to navigate the manifold's intrinsic curvature, thereby automatically satisfying the constraints throughout the optimization process. This often leads to more natural, computationally efficient, and numerically robust methods compared to traditional constrained techniques. While the conceptual foundations of optimizing on manifolds trace back several decades [102, 103], the modern, practicable framework heavily relies on computationally efficient tools, such as retractions, which were systematically unified and popularized by Absil et al. [3].

Building upon this framework, a growing body of research has adapted numerous classical algorithms, including gradient-based methods and trust-region techniques, specifically for optimization on manifolds [10, 1, 11]. Driven by the demands of large-scale applications, there has been a significant focus on developing robust and scalable first-order methods. Consequently, Riemannian versions of conjugate gradient [121, 119], stochastic gradient [124, 26], and proximal gradient methods [43, 86] have been extensively studied. This ongoing development paves the way for tackling increasingly complex problems, including the multi-objective optimization scenarios on manifolds that are central to this thesis [21, 55].

The extension of Euclidean optimization algorithms to the Riemannian setting requires replacing standard vector space operations with their appropriate geometric counterparts. A generic algorithm on a manifold $\mathcal{M}$, which generates a sequence of iterates $x_k \in \mathcal{M}$, exemplifies this adaptation. Such an algorithm typically takes the following form:

$$x_{k+1} = R_{x_k}(\alpha_k \eta_k), \tag{1.2.1}$$

where η_k is a search direction residing in the tangent space $T_{x_k}\mathcal{M}$, $\alpha_k > 0$ is a step size determined by a line-search procedure, and R_{x_k} is a retraction. The retraction is a crucial mapping that takes the step $\alpha_k \eta_k$ from the tangent space back onto the manifold $\mathcal{M}$, generalizing the simple vector addition used in Euclidean updates while ensuring the iteration remains feasible.

Central to constructing the search direction η_k is the Riemannian gradient $\text{grad}f(x)$, typically defined as the projection of the standard Euclidean gradient $\nabla f(x)$ onto the tangent space $T_x\mathcal{M}$. This vector represents the intrinsic direction of steepest ascent of the objective function f on the manifold. Furthermore, for iterative methods that rely on information from previous steps, such as conjugate gradient or quasi-Newton algorithms, a mechanism known as vector transport $\mathcal{T}$ is required. Vector transport generalizes the concept of parallel translation, providing a computationally efficient way to move a tangent vector from one tangent space to another (e.g., from $T_{x_k}\mathcal{M}$ to $T_{x_{k+1}}\mathcal{M}$) for comparison or aggregation. These four components, namely tangent spaces, Riemannian gradients, retractions, and vector transports, constitute the fundamental framework for extending Euclidean optimization algorithms to the manifold setting.

The practical utility and far-reaching impact of the Riemannian optimization framework are underscored by its successful and continually expanding application across diverse scientific and engineering domains. In numerical linear algebra and matrix analysis, it has provided the natural language for fundamental problems such as eigenvalue and singular value decompositions [122, 34], low-rank matrix and tensor completion [30, 68, 134], and various joint diagonalization problems [133, 2]. The framework has become particularly instrumental in modern machine learning and data science. Applications therein range from training novel neural network architectures [87, 88, 67] to dictionary learning [46]. In computer vision and signal processing, Riemannian methods are now essential for tasks such as shape analysis [128], phase retrieval [85, 52]. Further significant applications arise in fields as diverse as control theory [125, 126], quantum technologies [101]. This widespread adoption shows that Riemannian optimization is not just a theoretical concept but a practical tool for solving complex geometric problems. The success of this framework motivates the development of efficient algorithms, particularly the first-order methods studied in this thesis.

1.3 Classical first-order methods

1.3.1 Proximal gradient method

The Proximal Gradient Method has become a pivotal tool in modern optimization, particularly for solving large-scale problems that incorporate non-smooth terms. It is tailored for composite optimization problems of the form:

$$\min_{x \in \mathbb{R}^n} F(x) := f(x) + g(x), \quad (1.3.1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a smooth (typically convex) function with a Lipschitz continuous gradient, and $g : \mathbb{R}^n \rightarrow (-\infty, +\infty]$ is a convex, lower semi-continuous, but potentially non-smooth function. This composite structure is ubiquitous in domains such as machine learning, signal processing, and statistics. Canonical examples include the LASSO formulation in sparse regression, total variation denoising, and matrix completion with nuclear norm regularization.

Origins and Evolution

The proximal gradient method has a long theoretical history, originating well before its common use in data science. The central concept, the proximal operator, was introduced by Moreau in the 1960s [105, 106] as a tool in convex analysis, generalizing the projection onto a convex set. Structurally, the method is a forward-backward splitting algorithm. These methods were analyzed in the late 1970s by Lions and Mercier [96] and Passty [111], mainly within applied mathematics and monotone operator theory. The method was later popularized in the optimization community by Fukushima and Mine [66], who formulated a

generalized proximal point algorithm for convex programming, now known as the proximal gradient method.

A major improvement was the integration of Nesterov's acceleration techniques [109]. While the standard proximal gradient method exhibits a sublinear convergence rate of $\mathcal{O}(1/k)$, researchers demonstrated that an optimal rate of $\mathcal{O}(1/k^2)$ is achievable for convex problems. This realization led to the development of accelerated variants, most notably the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) by Beck and Teboulle [15]. This dramatic improvement in convergence speed solidified the proximal gradient method and its accelerated variants as indispensable tools for large-scale optimization.

Algorithmic Formulation

At the core of the method lies the proximal operator [105]. For a function g and a parameter $t > 0$, it is defined as:

$$\text{prox}_{tg}(y) = \arg \min_{x \in \mathbb{R}^n} \left\{ g(x) + \frac{1}{2t} \|x - y\|^2 \right\}.$$

Intuitively, the proximal operator maps a point y to a unique point x that balances the minimization of g with the proximity to y . Based on this, the iterative scheme of the proximal gradient method is given by:

$$x_{k+1} = \text{prox}_{t_k g}(x_k - t_k \nabla f(x_k)),$$

where $t_k > 0$ is the step size at iteration k . This update rule offers a clear geometric perspective as it is equivalent to minimizing a quadratic approximation of the smooth function f at x_k plus the non-smooth term g . Here, the quadratic term serves as an upper bound justified by the Lipschitz continuity of the gradient. Formally, the next iterate is obtained by:

$$x_{k+1} = \arg \min_{x \in \mathbb{R}^n} \left\{ \langle \nabla f(x_k), x - x_k \rangle + g(x) + \frac{1}{2t_k} \|x - x_k\|^2 \right\}.$$

Here is a summary of the Proximal Gradient Method with a constant step size:

Algorithm 1: Proximal Gradient Method with Constant Stepsize

Step 1 (Initialization): Choose $x^0 \in \mathbb{R}^n$, a step size $t \in (0, 1/L]$, and a tolerance $\varepsilon > 0$. Set $k := 0$.

Step 2 (Proximal Step): Compute the next iterate:

$$x^{k+1} = \arg \min_{x \in \mathbb{R}^n} \left\{ \langle \nabla f(x_k), x - x_k \rangle + g(x) + \frac{1}{2t} \|x - x_k\|^2 \right\}.$$

Step 3 (Stopping Criterion): If $\|x_{k+1} - x_k\| < \varepsilon$, then stop.

Step 4 (Update): Set $k := k + 1$, and go to Step 2.

The convergence of this method is guaranteed under standard assumptions, specifically when the step size t_k is fixed within the interval $(0, 1/L]$, where L is the Lipschitz constant of ∇f . In practice, however, L is often unknown or conservative; thus, a backtracking line search is frequently employed to adaptively determine the step size. Beyond acceleration, contemporary research has expanded the method's scope to stochastic variants, coordinate-descent schemes, and generalizations to non-Euclidean settings, including the incorporation of Bregman distances and extensions to Riemannian manifolds, all of which constitute the core focus of this thesis.

1.3.2 Conjugate gradient method

The conjugate gradient method stands as one of the most celebrated and influential iterative methods in numerical optimization. Its appeal stems from its remarkable efficiency, low memory requirements, and strong theoretical foundations, making it a premier choice for solving large-scale problems. The conjugate gradient method was originally introduced by Fletcher and Reeves in [57], while the Riemannian conjugate gradient method has been discussed in several studies, including [121, 118, 142, 143, 53]. On the other hand, the nonlinear conjugate gradient method for vector optimization was first proposed in [100], and some other choices of parameters were considered later in [74, 73].

Unlike the proximal gradient method, which is tailored for composite non-smooth problems, the CG method is primarily designed for unconstrained optimization of smooth functions,

$$\min_{x \in \mathbb{R}^n} f(x), \quad (1.3.2)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuously differentiable.

Origins and Evolution

The Conjugate Gradient method was proposed by Hestenes and Stiefel in 1952 [82]. Initially, it was not proposed as a general optimization algorithm, but rather as an elegant and highly effective iterative method for solving large, sparse systems of linear equations, $Ax = b$, where A is a SPD matrix. This problem is equivalent to minimizing the quadratic function $f(x) = \frac{1}{2}x^\top Ax - b^\top x$.

The ingenuity of the linear CG method lies in its construction of a sequence of search directions that are mutually conjugate with respect to the matrix A (i.e., $\eta_i^\top A \eta_j = 0$ for $i \neq j$). This property, known as A-orthogonality, ensures that each new direction is linearly independent of the previous ones. This finite termination property represented a significant theoretical advantage over methods like steepest descent, which often exhibits slow, zigzagging convergence.

The pivotal extension of the CG method from solving quadratic problems to minimizing general nonlinear functions occurred in the 1960s. The work of Fletcher and Reeves in 1964 [57] marked the beginning of nonlinear conjugate gradient methods. The challenge

in this transition is that the Hessian matrix $\nabla^2 f(x)$ is no longer a constant SPD matrix A . Thus, the concept of A -orthogonality cannot be directly applied. Instead, the focus shifted to designing the update for the search direction in a way that retains some of the desirable properties of the linear CG method. This led to various formulas for the scalar parameter β , which determines how the previous search direction is combined with the current negative gradient. The most classical of these include the Fletcher–Reeves (FR) formula, the Polak–Ribière (PR) formula [113] (and its Polak–Ribière–Polyak (PRP) modification [113, 114]), and the Hestenes–Stiefel (HS) formula [82], which directly mimics the linear CG update. Subsequent research has introduced numerous other variants, such as the Dai–Yuan (DY) formula [49], each with distinct theoretical properties and practical performance characteristics.

The global convergence of these methods for general functions is a non-trivial subject and typically relies on the enforcement of sufficient decrease conditions via the line search, such as the Wolfe conditions.

Algorithmic Formulation

The nonlinear conjugate gradient method is a line-search-based iterative algorithm designed for unconstrained optimization. Starting from an initial point x_0 , the first search direction is set to the steepest descent direction, $d_0 = -\nabla f(x_0)$. For subsequent iterations $k \geq 0$, the algorithm first computes a step size $\alpha_k > 0$ by performing a line search to approximately minimize f along the current direction d_k . The iterate is then updated as $x_{k+1} = x_k + \alpha_k d_k$.

A distinctive feature of the CG method is the construction of the new search direction d_{k+1} , which is defined as a linear combination of the current negative gradient and the previous search direction:

$$d_{k+1} = -\nabla f(x_{k+1}) + \beta_{k+1} d_k.$$

The scalar parameter β_{k+1} is crucial, as it determines the specific variant of the CG method. To facilitate the presentation of common β_{k+1} formulas, let us denote $g_k := \nabla f(x_k)$ and $y_k := g_{k+1} - g_k$. Standard choices for β_{k+1} include the FR, PRP, HS, Conjugate Descent (CD), DY, and Liu–Storey (LS) formulas [57, 58, 49, 113, 81, 97], defined respectively as:

$$\begin{aligned} \beta_{k+1}^{\text{FR}} &= \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, & \beta_{k+1}^{\text{PRP}} &= \frac{g_{k+1}^\top y_k}{\|g_k\|^2}, \\ \beta_{k+1}^{\text{HS}} &= \frac{g_{k+1}^\top y_k}{y_k^\top d_k}, & \beta_{k+1}^{\text{DY}} &= \frac{\|g_{k+1}\|^2}{y_k^\top d_k}, \\ \beta_{k+1}^{\text{CD}} &= \frac{\|g_{k+1}\|^2}{-g_k^\top d_k}, & \beta_{k+1}^{\text{LS}} &= \frac{g_{k+1}^\top y_k}{-g_k^\top d_k}. \end{aligned}$$

Here is a summary of the classical Nonlinear Conjugate Gradient Method in Euclidean space:

Algorithm 2: Nonlinear Conjugate Gradient Method

Step 1 (Initialization): Choose $x_0 \in \mathbb{R}^n$ and tolerance $\varepsilon > 0$. Compute $g_0 = \nabla f(x_0)$, set $d_0 = -g_0$, and $k := 0$.

Step 2 (Stopping Criterion): If $\|g_k\| < \varepsilon$, then stop.

Step 3 (Line Search & Update): Find step size $\alpha_k > 0$ satisfying standard conditions (e.g., Wolfe). Update iterate:

$$x_{k+1} = x_k + \alpha_k d_k. \quad (1.3.3)$$

Step 4 (Direction Update): Compute $g_{k+1} = \nabla f(x_{k+1})$ and calculate β_{k+1} (e.g., using FR or PRP formula). Update search direction:

$$d_{k+1} = -g_{k+1} + \beta_{k+1} d_k. \quad (1.3.4)$$

Step 5 (Increment): Set $k := k + 1$ and go to Step 2.

The global convergence of this method typically relies on the step size α_k satisfying sufficient decrease conditions, such as the Wolfe conditions. A primary advantage of the CG method is its matrix-free nature; it requires only gradient evaluations and vector operations, avoiding the storage and inversion of the Hessian matrix. This memory efficiency makes it exceptionally well-suited for large-scale optimization problems.

1.3.3 Conditional gradient method

The Conditional Gradient (CondG) method, also known as the Frank-Wolfe (FW) algorithm [61], is a first-order method for constrained smooth convex optimization. It is designed to solve problems of the form

$$\min_{x \in \mathcal{C}} f(x), \quad (1.3.5)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable function, and $\mathcal{C} \subset \mathbb{R}^n$ is a compact convex set. The defining characteristic of the method, which sets it apart from projected gradient and proximal gradient methods, is that it completely avoids projection onto the set $\mathcal{C}$. Instead, at each iteration, it requires solving a linear minimization problem over the constraint set. This “projection-free” nature makes it exceptionally well-suited for problems where projection is computationally expensive, but linear minimization is cheap. The versatility of this method is well demonstrated by its wide range of applications reported in the literature (see, for instance, [94, 50]). Furthermore, the CondG method has recently been extended to multiobjective optimization [7, 8, 69, 44, 72].

Origins and Evolution

The algorithm was proposed by Frank and Wolfe in 1956 [61] for quadratic programming. For a long time, it received less attention because other algorithms had faster theoretical

convergence rates. The method converges at a rate of $O(1/k)$, which is slower than the $O(1/k^2)$ rate of accelerated projected or proximal methods.

Interest in the method was renewed in the early 2010s due to large-scale machine learning [89]. In this context, the slow convergence rate is compensated by the low cost per iteration. For high-dimensional constraints, the projection operation used by other methods is often too expensive. A common example is the spectrahedron, where projection requires a full eigendecomposition with $O(n^3)$ complexity. In contrast, the Frank-Wolfe method only requires computing the leading eigenvector, which is much faster.

A further advantage is structural: each FW iterate is a convex combination of the initial point and a small number of extreme points of $\mathcal{C}$, naturally producing sparse or low-rank solutions. This property is desirable in tasks such as SVM training, portfolio optimization, and sparse signal processing. Numerous variants, including away-step, pairwise, stochastic, and nonconvex extensions, have been developed to improve convergence and broaden the applicability of the method [65, 93, 80, 92, 91].

Algorithmic Formulation

The Frank-Wolfe algorithm operates by iteratively approximating the objective function with a linear Taylor expansion and moving towards the minimizer of this linear model within the feasible set.

The central component of the method is the Linear Minimization Oracle (LMO). At each iteration k , given the current iterate x_k and the gradient $\nabla f(x_k)$, the LMO computes a direction-finding atom $s_k \in \mathcal{C}$ (typically a vertex of the constraint set) by solving:

$$s_k \in \arg \min_{s \in \mathcal{C}} \langle \nabla f(x_k), s \rangle.$$

Once s_k is obtained, the algorithm determines a step size $\gamma_k \in [0, 1]$ and updates the next iterate as a convex combination of the current point and the LMO solution:

$$x_{k+1} = (1 - \gamma_k)x_k + \gamma_k s_k.$$

By construction, since $x_k, s_k \in \mathcal{C}$ and $\mathcal{C}$ is convex, the new iterate x_{k+1} remains feasible without requiring any projection operation. The step size γ_k can be chosen via various strategies, such as a diminishing rule (e.g., $\gamma_k = 2/(k+2)$) or an adaptive line search.

Here is a summary of the Frank-Wolfe Method utilizing a line search strategy:

Algorithm 3: Frank-Wolfe (Conditional Gradient) Method with Line Search

Step 1 (Initialization): Choose $x_0 \in \mathcal{C}$ and a tolerance $\varepsilon > 0$. Set $k := 0$.**Step 2 (Stopping Criterion):** Compute $g_k = \nabla f(x_k)$. Solve

$$s_k = \arg \min_{s \in \mathcal{C}} \langle g_k, s \rangle.$$

Compute the Frank-Wolfe gap $G_k = \langle g_k, x_k - s_k \rangle$. If $G_k < \varepsilon$, then stop.**Step 3 (Line Search):** Find the step size $\gamma_k \in [0, 1]$ by (approximately) solving:

$$\gamma_k = \arg \min_{\gamma \in [0, 1]} f((1 - \gamma)x_k + \gamma s_k). \quad (1.3.6)$$

Step 4 (Update Iterate): Update the next iterate:

$$x_{k+1} = (1 - \gamma_k)x_k + \gamma_k s_k. \quad (1.3.7)$$

Step 5 (Increment): Set $k := k + 1$ and go to Step 2.

The key idea of the Conditional Gradient method is to replace the expensive projection step with simpler linear minimization steps. Convergence rates of $\mathcal{O}(1/k)$ have been established for standard convex problems [9, 89], while recent advancements focus on accelerating the method and extending it to non-convex settings [92, 95].

1.4 Motivations and contribution

Modern optimization problems are becoming increasingly complex and often cannot be solved by classical methods. Key challenges include non-linear geometric constraints (Riemannian manifolds), the need to balance conflicting objectives (vector optimization), and non-smooth objective functions. Specifically, many real-world applications, such as medical imaging on the manifold of positive definite matrices or sparse learning with non-Euclidean geometries, involve objective functions that do not satisfy the standard global smoothness requirements. Classical algorithms often fail to maintain feasibility or suffer from prohibitive computational costs when addressing these structured problems. And while existing manifold methods typically focus on single objectives, real-world engineering also often requires balancing structured constraints with multiple performance metrics. Therefore, there is a need for efficient and theoretically rigorous algorithms to address these issues. This thesis aims to develop algorithms that directly tackle these challenges.

The primary contributions of this work are threefold:

First, we extend the multi-objective proximal gradient method to accommodate a broader class of functions. By incorporating Bregman distances, we replace the restrictive global Lipschitz continuity assumption with the more flexible condition of relative smoothness. This theoretical shift enables the simultaneous optimization of conflicting goals in problems where standard smoothness assumptions fail, such as multi-task learning involving Kullback-Leibler

divergence or Poisson inverse problems, where Bregman-based methods naturally exploit the geometry induced by the objective. We establish the global convergence of the proposed method to Pareto stationary points and provide a rigorous analysis of its convergence rate under both constant and backtracking step-size strategies.

Second, we bridge the gap between vector optimization and manifold optimization by introducing nonlinear conjugate gradient methods for vector-valued functions on Riemannian manifolds. A key theoretical contribution is the extension of the Wolfe and Zoutendijk conditions to the Riemannian vector setting; specifically, we prove the existence of step sizes satisfying these generalized conditions. The convergence analysis encompasses vector extensions of the canonical Fletcher–Reeves, conjugate descent, and Dai–Yuan parameters. Under suitable assumptions, we prove that the sequence generated by the algorithm converges to a Pareto stationary point. Furthermore, we explore alternative parameter choices and present numerical experiments that demonstrate the empirical effectiveness of the methods.

Third, to address the high computational cost of retractions or projections in high-dimensional settings, we propose a Riemannian conditional gradient method for composite optimization. We analyze the convergence of the algorithm and establish a convergence rate of $O(1/k)$ for both adaptive and diminishing step sizes. Furthermore, we prove that an Armijo-based strategy achieves a convergence rate of $O(1/\sqrt{k})$, which corresponds to an iteration complexity of $O(1/\epsilon^2)$. In our numerical experiments, we specifically address the Sparse Principal Component Analysis (Sparse PCA) problem on different manifolds. While other algorithms rely on singular value decompositions or other expensive projections that become computationally expensive as data dimensions grow, our methods could circumvent these costs by solving simpler subproblems.

1.5 Outline of the thesis

The remainder of this thesis is organized as follows.

Chapter 2 presents the necessary preliminaries, establishing the key notations, fundamental definitions, and essential properties of vector optimization and Riemannian geometry that serve as the foundation for the subsequent chapters.

Chapter 3 is dedicated to the multi-objective proximal gradient method. Here, we develop the algorithm using Bregman distances to handle relatively smooth functions and analyze its convergence behavior under different step-size rules.

Chapter 4 explores matrix-free methods for vector optimization on manifolds. We introduce the Riemannian vector nonlinear conjugate gradient methods, detailing the adaptation of the Wolfe conditions and analyzing the convergence properties of various conjugate gradient parameters.

Chapter 5 presents projection-free optimization methods. We propose a Riemannian conditional gradient method for composite problems. We analyze the convergence of the algorithm under three step-size strategies and demonstrate its performance through

numerical experiments.

Finally, Chapter 6 summarizes the main findings of this work and outlines potential directions for future research.

Chapter 2

Preliminaries

In this section, we present the fundamental notations, definitions, and concepts from vector optimization, multi-objective optimization, and Riemannian geometry that are essential for this thesis. We begin with notations and basic definitions used in Euclidean spaces, followed by concepts of Pareto optimality and Bregman distances. We then discuss the main concepts of multi-objective optimization. Finally, we introduce the foundational elements of Riemannian manifolds, including retractions, vector transports, and relevant notions of convexity and smoothness.

2.1 Notations and basic definitions

We denote the Euclidean inner product as $\langle \cdot, \cdot \rangle$, and the Euclidean norm as $\| \cdot \|$. When the norm and the inner product are written without a subscript, they refer to the Euclidean ones. The transpose of a matrix A is denoted by $A^\top$, and the identity matrix with dimension ℓ is written as I_ℓ . The interior, the boundary and the closure of a set S are written as $\text{int}(S)$, $\text{bd}(S)$ and $\text{cl}(S)$, respectively. We also define the relation $\preceq$ ($\prec$) in $\mathbf{R}^m$ as $u \preceq v$ ($u \prec v$) if and only if $u_i \leq v_i$ ($u_i < v_i$) for all $i = 1, \dots, m$.

For a given scalar-valued function $q: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ and a vector-valued function $r: \mathbb{R}^n \rightarrow (\mathbb{R} \cup \{+\infty\})^m$, we use $\nabla q(x) \in \mathbb{R}^n$ and $Jr(y) \in \mathbb{R}^{m \times n}$ to denote, respectively, the gradient of q at $x \in \text{dom } q$, and the Jacobian matrix of r at $y \in \text{dom } r$. Here, $\text{dom } q := \{x \in \mathbb{R}^n: q(x) < +\infty\}$ and $\text{dom } r := \{y \in \mathbb{R}^n: r(y) \prec +\infty\}$ denote the effective domain of q and r , respectively. The directional derivative of q at x in the direction $d \in \mathbb{R}^n$, if it exists, is given by

$$q'(x; d) := \lim_{t \searrow 0} \frac{q(x + td) - q(x)}{t}. \quad (2.1.1)$$

We denote the set of subdifferentiable points of q as $\text{dom } \partial q := \{x \in \mathbb{R}^n: \partial q(x) \neq \emptyset\}$, where $\partial q(x)$ means the subdifferential of q at x in the sense of convex analysis, i.e.,

$$\partial q(x) := \{s \in \mathbb{R}^n: q(y) \geq q(x) + \langle s, y - x \rangle \text{ for all } y \in \mathbb{R}^n\}. \quad (2.1.2)$$

Note that for a proper function q , we have $\partial q(x) = \emptyset$ when $x \notin \text{dom } q$.

2.2 Bregman distances and relative smoothness

In this section, we review the fundamental properties of Bregman distances, followed by the concepts of relative smoothness and relative strong convexity. These definitions rely on a reference function $\omega: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$, which is assumed to be proper, closed, convex, and differentiable on its domain $\text{dom } \omega$. We begin with the definition of the Bregman distance [33].

Definition 2.2.1. *The Bregman distance associated with ω is the function $B_\omega: \text{dom } \omega \times \text{dom } \partial\omega \rightarrow \mathbb{R}$ defined by*

$$B_\omega(x, y) := \omega(x) - \omega(y) - \langle \nabla \omega(y), x - y \rangle, \quad \text{for all } x \in \text{dom } \omega, y \in \text{dom } \partial\omega.$$

Remark 2.2.1. *We highlight several key properties of the Bregman distance:*

- (i) **Geometric Interpretation:** *Geometrically, $B_\omega(x, y)$ represents the vertical distance at x between the function ω and its tangent hyperplane approximation constructed at y .*
- (ii) **Non-negativity:** *Since ω is convex, the first-order convexity condition implies that $B_\omega(x, y) \geq 0$ for all x, y . Furthermore, if ω is strictly convex, then $B_\omega(x, y) = 0$ if and only if $x = y$.*
- (iii) **Coercivity from Strong Convexity:** *If ω is μ -strongly convex with respect to $\|\cdot\|$, it provides a lower bound for the Bregman distance:*

$$B_\omega(x, y) \geq \frac{\mu}{2} \|x - y\|^2. \quad (2.2.1)$$

This property is fundamental for convergence analysis, as it establishes a connection between the Bregman distance and the metric topology.

- (iv) **Asymmetry:** *Unlike a standard metric, the Bregman distance is generally not symmetric, i.e., $B_\omega(x, y) \neq B_\omega(y, x)$, and it does not necessarily satisfy the triangle inequality.*

We illustrate the definition with two canonical examples.

Example 2.2.1 (Euclidean distance). *Consider the kernel function $\omega(x) = \frac{1}{2}\|x\|^2$. Its gradient is $\nabla \omega(y) = y$, and it is 1-strongly convex with respect to the Euclidean norm. The associated Bregman distance is simply half the squared Euclidean distance:*

$$B_\omega(x, y) = \frac{1}{2} \|x - y\|^2. \quad (2.2.2)$$

In this specific case, symmetry holds, i.e., $B_\omega(x, y) = B_\omega(y, x)$.

Example 2.2.2 (Kullback–Leibler divergence). *Consider the negative entropy function defined as $\omega(x) = \sum_{i=1}^n x_i \ln x_i$ for $x \succeq 0$ (with $0 \ln 0 = 0$) and $+\infty$ otherwise. The resulting Bregman distance corresponds to the generalized Kullback–Leibler divergence:*

$$B_\omega(x, y) = \sum_{i=1}^n \left(x_i \ln \frac{x_i}{y_i} - x_i + y_i \right). \quad (2.2.3)$$

We now present several lemmas that will be used in our analysis. In particular, Lemma 2.2.2, which is a consequence of Lemma 2.2.1, plays a pivotal role in analyzing the convergence of proximal gradient methods with Bregman distances.

Lemma 2.2.1 (Three-points identity). *[36, Lemma 3.1] For any $a, b \in \text{dom } \partial\omega$ and $c \in \text{dom } \omega$, the following equality holds:*

$$\langle \nabla\omega(b) - \nabla\omega(a), c - a \rangle = B_\omega(c, a) + B_\omega(a, b) - B_\omega(c, b).$$

Lemma 2.2.2. *[36, Lemma 3.2] Let $\theta : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ be a proper closed convex function. For any $z \in \text{dom } \partial\omega$, if the minimizer $z_+ = \underset{x \in \text{dom } \omega}{\operatorname{argmin}} (\theta(x) + B_\omega(x, z))$ exists and ω is differentiable at z_+ , then:*

$$\theta(x) + B_\omega(x, z) \geq \theta(z_+) + B_\omega(z_+, z) + B_\omega(x, z_+) \quad \text{for all } x \in \text{dom } \omega.$$

Many classical optimization methods, such as the one proposed in [129], rely on the assumption that the objective gradients are globally Lipschitz continuous. For a scalar-valued function q , this requires the existence of a constant $\tilde{L}_q$ such that

$$\|\nabla q(x) - \nabla q(y)\| \leq \tilde{L}_q \|x - y\| \quad \text{for all } x, y \in \text{int}(\text{dom } q). \quad (2.2.4)$$

This condition, however, can be overly restrictive for many functions of interest [98]. The notion of *relative smoothness*, introduced below, provides a less demanding alternative by measuring smoothness relative to the reference function ω rather than the Euclidean norm.

Definition 2.2.2 (Relative smoothness). *[98, Definition 1.1] Let ω be convex and differentiable on $\text{dom } \omega$. A function q is called L_q -smooth relative to ω on $\text{dom } \omega$ if for any $x, y \in \text{int}(\text{dom } \omega)$, there exists a scalar $L_q \geq 0$ such that*

$$q(x) \leq q(y) + \langle \nabla q(y), x - y \rangle + L_q B_\omega(x, y). \quad (2.2.5)$$

This definition provides a quadratic-like upper approximation of q analogous to the classical descent lemma in Euclidean spaces. Similarly, we can define relative strong convexity.

Definition 2.2.3 (Relative strong convexity). *[98, Definition 1.2] Let ω be convex and differentiable on $\text{dom } \omega$. A function q is called μ_q -strongly convex relative to ω on $\text{dom } \omega$ if for any $x, y \in \text{int}(\text{dom } \omega)$, there exists a scalar $\mu_q \geq 0$ such that*

$$q(x) \geq q(y) + \langle \nabla q(y), x - y \rangle + \mu_q B_\omega(x, y). \quad (2.2.6)$$

As noted in [98, Section 2], many classes of optimization problems that fail to satisfy standard Lipschitz continuity do satisfy relative smoothness for an appropriately chosen reference function ω , making this framework highly applicable to a broader range of problems.

2.3 Vector and multi-objective optimization

We first introduce concepts from vector optimization, following [78, 79, 90]. Let $K \subset \mathbb{R}^m$ be a closed, convex, and pointed (i.e., $K \cap (-K) = \{0\}$) cone with a nonempty interior. The partial order in $\mathbb{R}^m$ induced by K is denoted by $\preceq_K$, and defined by

$$u \preceq_K v \Leftrightarrow v - u \in K,$$

and the relation $\prec_K$ induced by $\text{int}(K)$ is defined by

$$u \prec_K v \Leftrightarrow v - u \in \text{int}(K).$$

Let $K^* := \{\omega \in \mathbb{R}^m \mid \omega^\top y \geq 0 \text{ for all } y \in K\}$ be the positive polar (or dual) cone of K . Let $C \subset K^* \setminus \{0\}$ be compact and such that

$$K^* = \text{cone}(\text{conv}(C)),$$

that is, the cone generated by the convex hull of C is equal to K^* . In the general case, a canonical choice is to set $C = \{\omega \in K^* \mid \|\omega\| = 1\}$.

For a given scalar-valued function $q: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ and a vector-valued function $r: \mathbb{R}^n \rightarrow (\mathbb{R} \cup \{+\infty\})^m$, we use $\nabla q(x) \in \mathbb{R}^n$ and $Jr(y) \in \mathbb{R}^{m \times n}$ to denote, respectively, the gradient of q at $x \in \text{dom } q$, and the Jacobian matrix of r at $y \in \text{dom } r$. Here, $\text{dom } q := \{x \in \mathbb{R}^n \mid q(x) < +\infty\}$ and $\text{dom } r := \{y \in \mathbb{R}^n \mid r(y) \prec +\infty\}$ denote the effective domain of q and r , respectively. The directional derivative of q at x in the direction $d \in \mathbb{R}^n$, if it exists, is given by

$$q'(x; d) := \lim_{t \searrow 0} \frac{q(x + td) - q(x)}{t}. \quad (2.3.1)$$

We denote the set of subdifferentiable points of q as $\text{dom } \partial q := \{x \in \mathbb{R}^n \mid \partial q(x) \neq \emptyset\}$, where $\partial q(x)$ means the subdifferential of q at x in the sense of convex analysis, i.e.,

$$\partial q(x) := \{s \in \mathbb{R}^n \mid q(y) \geq q(x) + \langle s, y - x \rangle \text{ for all } y \in \mathbb{R}^n\}. \quad (2.3.2)$$

Note that for a proper function q , we have $\partial q(x) = \emptyset$ when $x \notin \text{dom } q$.

Now, consider the following vector optimization problem:

$$\min_K F(x) \quad \text{s.t. } x \in \mathbb{R}^n,$$

where $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a continuously differentiable function, and the minimization is with respect to the cone K . A point $x^* \in \mathbb{R}^n$ is called *K-Pareto optimal* if there exists no other $x \in \mathbb{R}^n$ with $F(x) \preceq_K F(x^*)$ and $F(x) \neq F(x^*)$. A point $x^* \in \mathbb{R}^n$ is called *weakly K-Pareto optimal* if there exists no other $x \in \mathbb{R}^n$ with $F(x) \prec_K F(x^*)$. It is clear that *K-Pareto optimal* points are always *weakly K-Pareto optimal*, but the converse does not hold universally.

A necessary condition for the K -optimality of x^* is

$$-\text{int}(K) \cap \text{Image}(JF(x^*)) = \emptyset,$$

where $JF(x)$ denotes the Jacobian of F at x . A point $x^* \in \mathbb{R}^n$ that satisfies this condition is called K -critical (or K -stationary) for F . If x is not K -critical, there exists $d \in \mathbb{R}^n$ such that $JF(x)d \in -\text{int}(K)$. Every such vector d is a K -descent direction for F at x .

A particularly important special case is multi-objective optimization, where the cone is the non-negative orthant, $K = \mathbb{R}_+^m$. In this context, the problem is written as in (1.1.1):

$$\min_{x \in \mathbb{R}^n} F(x) := (f_1(x), \dots, f_m(x))^\top.$$

In this case, $K^* = \mathbb{R}_+^m$, and C can be taken as the canonical basis. The partial orders $\preceq_K$ and $\prec_K$ simplify to the component-wise relations $\preceq$ and $\prec$ defined earlier. The optimality concepts are then referred to as Pareto optimality.

- $x^* \in \mathbb{R}^n$ is a *Pareto optimal* point for F , if there is no $x \in \mathbb{R}^n$ such that $F(x) \preceq F(x^*)$ and $F(x) \neq F(x^*)$.
- $x^* \in \mathbb{R}^n$ is a *weakly Pareto optimal* point for F , if there is no $x \in \mathbb{R}^n$ such that $F(x) \prec F(x^*)$.

The set of all (weakly) Pareto optimal values is called the (weakly) Pareto frontier.

For this case ($K = \mathbb{R}_+^m$), a point $x \in \mathbb{R}^n$ is K -critical if and only if $JF(x)d \not\prec 0$ for all $d \in \mathbb{R}^n$, namely,

$$\max_{i \in \{1, \dots, m\}} \nabla f_i(x)^\top d \geq 0, \quad \text{for all } d \in \mathbb{R}^n.$$

This leads to the following definition of stationarity, which generalizes the one given in [60] for differentiable problems to composite problems. We say that $\bar{x}$ is *Pareto stationary* (or *critical*), if and only if,

$$\max_{i=1, \dots, m} f'_i(\bar{x}; d) \geq 0 \quad \text{for all } d \in \mathbb{R}^n.$$

The relationship between this stationarity and optimality is summarized below.

Lemma 2.3.1. [129, Lemma 2.2] *The following assertions hold:*

- If x is weakly Pareto optimal for F , then x is Pareto stationary.*
- Assume that every component f_i of F is convex. If x is Pareto stationary for F , then x is weakly Pareto optimal.*
- Assume that every component f_i of F is strictly convex. If x is Pareto stationary for F , then x is Pareto optimal.*

2.4 Riemannian manifolds

2.4.1 Fundamentals of Riemannian manifolds

In this section, we review the fundamental concepts of Riemannian geometry essential for the subsequent developments. For a more comprehensive treatment, the reader is referred to [3, 28, 120].

A Riemannian manifold $\mathcal{M}$ is a smooth manifold endowed with a Riemannian metric, which is defined as a smoothly varying inner product $\langle \cdot, \cdot \rangle_x$ on the tangent space $T_x\mathcal{M}$ at each point $x \in \mathcal{M}$. This metric provides the necessary geometric structure to measure angles and lengths by inducing a norm on each tangent space, defined as $\|\eta\|_x := \sqrt{\langle \eta, \eta \rangle_x}$ for any $\eta \in T_x\mathcal{M}$. The set of all tangent vectors at a specific point x forms the tangent space $T_x\mathcal{M}$. Furthermore, the disjoint union of these tangent spaces across all points in $\mathcal{M}$ constitutes the tangent bundle, denoted by $T\mathcal{M} := \{(x, \eta) \mid x \in \mathcal{M}, \eta \in T_x\mathcal{M}\}$.

We next recall the notion of derivatives for maps between manifolds. Given a smooth map $F : \mathcal{M} \rightarrow \mathcal{N}$, its derivative (or differential) at $x \in \mathcal{M}$ is a linear map $DF(x) : T_x\mathcal{M} \rightarrow T_{F(x)}\mathcal{N}$. A case of particular importance in optimization involves scalar-valued functions of the form $f : \mathcal{M} \rightarrow \mathbb{R}$. Its derivative acts as a linear functional $Df(x) : T_x\mathcal{M} \rightarrow \mathbb{R}$. The *Riemannian gradient*, denoted by $\text{grad } f(x)$, is defined as the unique tangent vector in $T_x\mathcal{M}$ that represents this differential via the Riemannian metric, satisfying:

$$\langle \text{grad } f(x), \eta \rangle_x = Df(x)[\eta], \quad \text{for all } \eta \in T_x\mathcal{M}. \quad (2.4.1)$$

Finally, to facilitate definitions involving pairs of tangent vectors at the same point (such as vector transport), we define the Whitney sum of the tangent bundle as $T\mathcal{M} \oplus T\mathcal{M} := \{(\xi, d) \mid \xi, d \in T_x\mathcal{M}, x \in \mathcal{M}\}$.

2.4.2 Operations on Riemannian manifolds

Optimization algorithms on manifolds require specific mechanisms to navigate the search space and to compare tangent vectors belonging to different tangent spaces. This section formally defines these essential geometric tools: retractions and vector transports.

In Euclidean space $\mathbb{R}^n$, iterative algorithms typically update a point x_k via vector addition, i.e., $x_{k+1} = x_k + t_k d_k$. However, on a nonlinear Riemannian manifold $\mathcal{M}$, the current point $x_k \in \mathcal{M}$ and the search direction $d_k \in T_{x_k}\mathcal{M}$ reside in distinct spaces. Direct addition is undefined as it would likely result in a point leaving the manifold. To address this, we utilize a retraction, which provides a smooth mapping from the tangent space back onto the manifold.

Definition 2.4.1 (Retraction). [120] A smooth map $R : T\mathcal{M} \rightarrow \mathcal{M}$ is called a retraction on a smooth manifold $\mathcal{M}$ if its restriction to the tangent space $T_x\mathcal{M}$ at any point $x \in \mathcal{M}$, denoted by R_x , satisfies the following two conditions:

- (a) $R_x(0_x) = x$, where 0_x is the zero vector of $T_x\mathcal{M}$;

(b) $DR_x(0_x) = \text{id}_{T_x\mathcal{M}}$, where $\text{id}_{T_x\mathcal{M}}$ is the identity map on $T_x\mathcal{M}$.

The retraction serves as a first-order approximation of the exponential map, allowing us to generalize the standard gradient descent update scheme to

$$x_{k+1} = R_{x_k}(t_k d_k), \quad \text{for } k = 0, 1, 2, \dots,$$

where $d_k \in T_{x_k}\mathcal{M}$ is a descent direction (e.g., $d_k = -\text{grad } f(x_k)$) and $t_k > 0$ is a step size.

A canonical and geometrically natural choice for a retraction is the exponential map, denoted by $\text{Exp}_x(d)$. This mapping is defined by tracing a geodesic, which represents the generalization of a straight line to curved spaces, starting at x with an initial velocity d for a unit time step.

Generally, the Riemannian distance between two points $x, y \in \mathcal{M}$ is defined as the infimum of the lengths of all piecewise smooth curves $\gamma : [0, 1] \rightarrow \mathcal{M}$ such that $\gamma(0) = x$ and $\gamma(1) = y$. This distance is expressed as

$$\text{dist}(x, y) = \inf_{\gamma} \int_0^1 \sqrt{\langle \dot{\gamma}(t), \dot{\gamma}(t) \rangle_{\gamma(t)}} dt.$$

When a unique minimizing geodesic exists between x and y , the inverse of the exponential map is well-defined and is referred to as the logarithm map, denoted by $\text{Log}_x(y)$ or $\text{Exp}_x^{-1}(y)$. Then, the logarithm map retrieves the unique tangent vector at x corresponding to the geodesic segment connecting the two points. Consequently, the Riemannian distance can be conveniently computed using the norm of the logarithm map

$$\text{dist}(x, y) := \|\text{Exp}_x^{-1}(y)\|_x.$$

A related challenge in manifold optimization arises when operations involve vectors from different tangent spaces. For instance, in the nonlinear conjugate gradient method, computing the new direction involves combining the previous search direction $d_{k-1} \in T_{x_{k-1}}\mathcal{M}$ with the current gradient $\text{grad } f(x_k) \in T_{x_k}\mathcal{M}$. Since these vectors lie in disjoint vector spaces, they cannot be directly added. To resolve this, we introduce the concept of vector transport.

Definition 2.4.2 (Vector Transport). [120] A map $\mathcal{T} : T\mathcal{M} \oplus T\mathcal{M} \rightarrow T\mathcal{M}$ is called a vector transport on $\mathcal{M}$ if it satisfies the following conditions:

- (a) There exists a retraction R on $\mathcal{M}$ such that $\mathcal{T}_\eta(\xi) \in T_{R_x(\eta)}\mathcal{M}$ for all $x \in \mathcal{M}$ and $\eta, \xi \in T_x\mathcal{M}$.
- (b) $\mathcal{T}_{0_x}(\xi) = \xi$ for all $x \in \mathcal{M}$ and $\xi \in T_x\mathcal{M}$.
- (c) $\mathcal{T}_\eta(a\xi + b\zeta) = a\mathcal{T}_\eta(\xi) + b\mathcal{T}_\eta(\zeta)$ for all scalars $a, b \in \mathbb{R}$ and tangent vectors $\eta, \xi, \zeta \in T_x\mathcal{M}$.

A theoretically significant instance of vector transport is parallel transport. It moves a tangent vector along a curve, typically a geodesic, such that the vector remains parallel to itself with respect to the Riemannian connection. When the retraction is chosen as the

exponential map, parallel transport along the geodesic $\gamma(t) = \text{Exp}_x(t\eta)$ connecting x and $y = \text{Exp}_x(\eta)$ is denoted by $P_\gamma^{1 \leftarrow 0}$. The map defined by $\mathcal{T}_\eta(\xi) := P_\gamma^{1 \leftarrow 0}(\xi)$ constitutes a valid vector transport. Parallel transport is particularly desirable due to its isometric property, meaning that it preserves the inner product and consequently the norm of transported vectors, which is expressed as $\langle \mathcal{T}_\eta(\xi), \mathcal{T}_\eta(\zeta) \rangle_{R_x(\eta)} = \langle \xi, \zeta \rangle_x$.

2.4.3 Convexity on manifolds

To establish the convergence properties of optimization algorithms on manifolds, we generalize classical Euclidean notions of convexity to the Riemannian setting. We begin with the definition of geodesically convex sets and functions, as detailed in [29].

Definition 2.4.3 (Geodesic Convexity and Strong Convexity). *A subset $\mathcal{X} \subseteq \mathcal{M}$ is called geodesically convex if for any two points $x, y \in \mathcal{X}$, there exists a unique geodesic $\gamma: [0, 1] \rightarrow \mathcal{M}$ such that $\gamma(0) = x$, $\gamma(1) = y$, and the entire path $\gamma(t)$ remains in $\mathcal{X}$ for all $t \in [0, 1]$.*

A function $f: \mathcal{X} \rightarrow \mathbb{R}$ is called geodesically convex if for any $x, y \in \mathcal{X}$ and the connecting geodesic γ , the following inequality holds:

$$f(\gamma(t)) \leq (1-t)f(x) + tf(y), \quad \forall t \in [0, 1]. \quad (2.4.2)$$

Furthermore, f is called geodesically μ -strongly convex for some $\mu > 0$ if:

$$f(\gamma(t)) \leq (1-t)f(x) + tf(y) - \frac{\mu}{2}t(1-t)L(\gamma)^2, \quad \forall t \in [0, 1], \quad (2.4.3)$$

where $L(\gamma)$ denotes the length of the geodesic segment connecting x and y .

As in the Euclidean case, differentiable convex functions on manifolds are characterized by a first-order condition using the Riemannian gradient.

Proposition 2.4.1 (First-order Condition). [137] *Let $h: \mathcal{M} \rightarrow \mathbb{R}$ be a smooth and geodesically convex function. Then, for any $x, y \in \mathcal{M}$ where the inverse exponential map is well-defined, we have:*

$$h(y) \geq h(x) + \langle \text{grad } h(x), \text{Exp}_x^{-1}(y) \rangle_x. \quad (2.4.4)$$

Importantly, geodesic convexity guarantees that local optimality implies global optimality, a fundamental property for optimization.

Theorem 2.4.1 (Global Optimality). [29] *Let $\mathcal{X} \subseteq \mathcal{M}$ be a geodesically convex set and $f: \mathcal{X} \rightarrow \mathbb{R}$ be a geodesically convex function. Then, any local minimizer of f in $\mathcal{X}$ is a global minimizer.*

Chapter 3

A proximal gradient method with Bregman distance in multi-objective optimization

3.1 Introduction

This chapter addresses the unconstrained multi-objective optimization problem previously introduced in (1.1.1), reformulated as follows:

$$\begin{aligned} \min \quad & F(x) \\ \text{s.t.} \quad & x \in \mathbb{R}^n, \end{aligned} \tag{3.1.1}$$

where $F: \mathbb{R}^n \rightarrow (\mathbb{R} \cup \{+\infty\})^m$ is a vector-valued function denoted by $F := (F_1, \dots, F_m)^\top$. We assume that each component function $F_i: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ possesses a composite structure defined by:

$$F_i(x) := f_i(x) + g_i(x), \quad i = 1, \dots, m, \tag{3.1.2}$$

where $f_i: \mathbb{R}^n \rightarrow \mathbb{R}$ is a continuously differentiable function, and $g_i: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is a proper, closed, and convex function. The formulation in (3.1.2) encompasses a wide range of applications, including constrained multi-objective problems (where each g_i represents the indicator function of a common feasible set), machine learning tasks [24, 75], and robust optimization [129].

For composite problems of this nature, Tanabe, Fukuda, and Yamashita [129] proposed a multi-objective proximal gradient method. In their approach, the search direction is determined by solving a subproblem that incorporates a first-order approximation of the differentiable components f_i , the exact non-smooth terms g_i , and a regularization term based on the squared Euclidean distance. Under the assumption that the gradients of f_i are globally Lipschitz continuous, they established that any accumulation point of the generated sequence is a Pareto stationary point. Furthermore, a convergence rate analysis for this method was subsequently provided in [130].

Building upon the preliminary concepts established in Chapter 2.2, this chapter generalizes Euclidean-based methods by adopting the Bregman distance as the proximity measure. This measure satisfies a generalized Pythagorean theorem, which is crucial for extending projection-based techniques to non-Euclidean geometries. Consequently, we utilize the Bregman distance here to capture the specific geometry of the constraints and to mitigate the ill-conditioning of subproblems, a strategy successfully applied in scalar constrained optimization [36, 136].

While the application of Bregman distances has been extensively studied in scalar optimization, including proximal point methods [36] and proximal gradient methods [25, 132], their integration into multi-objective optimization remains limited, with exceptions primarily in proximal point algorithms [45, 127]. Motivated by this gap, we propose a multi-objective proximal gradient method with Bregman distances, modifying the search direction framework established in [129]. The flexibility inherent in the choice of the kernel function allows the resulting subproblems to capture the geometry of the objective function more accurately, potentially improving solution precision and convergence speed. For instance, the Kullback–Leibler divergence is particularly advantageous for optimization over the unit simplex [14], and specific Bregman distances are essential for solving Poisson linear inverse problems where the standard proximal gradient method is inapplicable [12]. Further applications are discussed in [136, 83].

To establish the convergence of our proposed method, we utilize the concept of relative smoothness. As detailed in Chapter 2.2, this notion serves as a generalized descent lemma that relaxes the global Lipschitz gradient continuity requirement. We build upon the framework of Lu et al. [98], which demonstrates that relative smoothness and relative strong convexity are weaker conditions than the standard Lipschitz continuity assumption. Importantly, unlike the setting in [12], the framework in [98] does not require the reference function to be strictly convex. Adopting this less restrictive framework, we extend the notion of relative smoothness to the multi-objective setting for the differentiable parts of the objective function. We explicitly analyze the proposed algorithm under two distinct step-size strategies, namely constant and backtracking, and establish both convergence to Pareto stationary points and asymptotic convergence rates for each case.

It is worth noting that a related unpublished manuscript by Chen et al. [38] addresses vector optimization but does not account for the non-differentiable component g . Furthermore, our analysis offers a more refined global convergence theory as we do not mandate the strict convexity or Lipschitz gradient continuity of the reference function. Similarly, an independent unpublished work by Ansary and Dutta [6] considers composite objective functions; however, their analysis relies on the strong convexity of the reference function and does not address convergence rates.

The remainder of this chapter is organized as follows. Section 3.2 recalls necessary preliminaries, including the standard multi-objective proximal gradient method. In Section 3.3, we present the proposed multi-objective proximal gradient method with Bregman distances, detailing both the constant step-size and backtracking procedures.

Section 3.4 provides a rigorous convergence analysis, establishing global convergence to Pareto stationary points and deriving convergence rates for convex and strongly convex problems. Finally, Section 3.5 summarizes the chapter.

3.2 Preliminaries

In this chapter, we denote the iteration index by a superscript, i.e., x^k denotes the vector at iteration k , and x_j^k denotes its j -th component. Although the definitions of Bregman distances and relative smoothness were detailed in Chapter 2, we explicitly recall a few key properties of the Bregman distance B_ω that are indispensable for the analysis in this chapter.

First, the convexity of the kernel function ω implies the non-negativity of the distance:

$$B_\omega(x, y) \geq 0 \quad \text{for all } x \in \text{dom } \omega, y \in \text{dom } \partial\omega. \quad (3.2.1)$$

Moreover, it follows directly from the definition that

$$x = y \quad \implies \quad B_\omega(x, y) = 0. \quad (3.2.2)$$

It should also be noted that if ω is strictly convex, the converse implication holds as well, i.e., $B_\omega(x, y) = 0$ implies $x = y$.

We briefly review the standard multi-objective proximal gradient method proposed by Tanabe, Fukuda, and Yamashita [129], which serves as the foundation for our proposed algorithm. Consistent with many descent-based algorithms for vector optimization, this method generates a sequence of iterates $\{x^k\}$ via the update rule:

$$x^{k+1} := x^k + t_k d^k,$$

where d^k represents the search direction and $t_k > 0$ is the step size at iteration k . The search direction d^k is determined by solving a specific subproblem that creates a local approximation of the objective function vector. Specifically, d^k is the unique solution to the following unconstrained single-objective optimization problem:

$$d^k := \operatorname{argmin}_{d \in \mathbb{R}^n} \left(\tilde{\psi}_{x^k}(d) + \frac{\ell}{2} \|d\|^2 \right), \quad (3.2.3)$$

where $\ell > 0$ is a regularization parameter. The function $\tilde{\psi}_{x^k}: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ represents the maximum of the linearized approximations of the individual objective components, defined as:

$$\tilde{\psi}_{x^k}(d) := \max_{i=1, \dots, m} \left(\nabla f_i(x^k)^\top d + g_i(x^k + d) - g_i(x^k) \right),$$

assuming $x^k \in \bigcap_{i=1}^m \text{dom } g_i$. Structurally, this subproblem balances the improvement of the linearized model (using the gradient for the differentiable part f_i and the exact nonsmooth term g_i) against a regularization term. In [129], the standard squared Euclidean norm is

employed for regularization. We observe that the subproblem (3.2.3) is well-defined because its objective function is strongly convex due to the quadratic term.

Regarding the step size strategy, Tanabe et al. [129] investigated both a constant step size scheme (fixing $t_k = 1$) and a backtracking line search procedure based on the Armijo condition. In both scenarios, they established that any accumulation point of the generated sequence, provided one exists, is a Pareto stationary point [129, Theorems 4.2 and 4.3]. Furthermore, the convergence rate of this method was subsequently analyzed in [130].

The primary limitation of the formulation in (3.2.3) is its reliance on the Euclidean norm, which implicitly assumes the objective function satisfies standard smoothness conditions, namely the Lipschitz continuity of the gradients. In the subsequent section, we will generalize this framework by replacing the Euclidean regularization with a Bregman distance, thereby extending the method's applicability to relatively smooth functions.

3.3 The multi-objective proximal gradient method with Bregman distance

In this section, we explain in detail the proposed proximal gradient method with Bregman distance. From now on, we suppose that the following assumption holds.

Assumption 3.3.1. *Let $\omega: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ be a reference function that is proper, closed, convex, and differentiable over $\text{dom } \partial\omega$. We assume that f_i is L^i -smooth relative to ω and that $\text{dom } g \subseteq \text{dom } \omega$. Also, the level set $\{x \in \text{dom } g \mid \omega(x) \leq \alpha\}$ is compact for all $\alpha \in \mathbb{R}$.*

We recall that in many works, ω is assumed to be a Legendre function, i.e., it is proper, closed, essentially smooth and strictly convex. Also, essentially smooth means $\text{dom } \partial\omega \neq \emptyset$ or, equivalently, $\text{int}(\text{dom } \omega) \neq \emptyset$ with ω differentiable on $\text{int}(\text{dom } \omega)$ and $\|\nabla\omega(x^k)\| \rightarrow +\infty$ when $\{x^k\} \subset \text{int}(\text{dom } \omega)$, $x^k \rightarrow x \in \text{bd}(\text{dom } \omega)$ (see [117, Theorem 26.1]). In this work, we do not require strict convexity of ω unless it is explicitly specified. Observe also that compactness of the defined level set is guaranteed, for instance, when $\text{dom } g$ is compact, or ω is level bounded.

Using the relatively smooth constants, we also define

$$L := \max_{i=1, \dots, m} L^i. \quad (3.3.1)$$

Now, let us notice that when the stepsize t_k is equal to 1, then $d^k = x^{k+1} - x^k$. Thus, in this case, the subproblem (3.2.3) can be written as

$$x^{k+1} = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \left(\psi_{x^k}(x) + \frac{\ell}{2} \|x - x^k\|^2 \right), \quad (3.3.2)$$

where $\ell > 0$ and $\psi_{x^k}: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ is defined by

$$\psi_{x^k}(x) := \max_{i=1, \dots, m} \left(\nabla f_i(x^k)^\top (x - x^k) + g_i(x) - g_i(x^k) \right). \quad (3.3.3)$$

Based on this, in each iteration k of our proposed method, we consider the following subproblem:

$$x^{k+1} = p_{L_k}(x^k) \in \operatorname{argmin}_{x \in \mathbb{R}^n} \left(\psi_{x^k}(x) + L_k B_\omega(x, x^k) \right), \quad (3.3.4)$$

where ψ_{x^k} is defined in (3.3.3), $L_k > 0$, and B_ω is the Bregman distance associated with ω , supposing Assumption 4.3.2. Clearly, this subproblem is equivalent to (3.3.2) when $\omega(x) = \|x\|^2/2$ (if $L_k = \ell$ for all k). We further define its optimal value as follows:

$$\theta(x^k) := \min_{x \in \mathbb{R}^n} \left(\psi_{x^k}(x) + L_k B_\omega(x, x^k) \right) = \psi_{x^k}(p_{L_k}(x^k)) + L_k B_\omega(p_{L_k}(x^k), x^k). \quad (3.3.5)$$

Now, we observe that the definition of ψ_{x^k} gives $\psi_{x^k}(x^k) = 0$, and $B_\omega(x^k, x^k) = 0$ holds. This means that

$$\theta(x^k) \leq 0 \quad \text{for all } k. \quad (3.3.6)$$

We now prove that $p_{L_k}(x^k)$ is in fact well-defined.

Proposition 3.3.1. *The term $p_{L_k}(x^k)$ given in (3.3.4) is well-defined for all k and $p_{L_k}(x^k) \in \operatorname{dom} \partial \omega$. If in addition ω is strongly convex, the minimizer $p_{L_k}(x^k)$ is unique.*

Proof. The optimization problem given in (3.3.5) can be written as

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} \max_{i=1, \dots, m} \left(\nabla f_i(x^k)^\top (x - x^k) + g_i(x) - g_i(x^k) + L_k B_\omega(x, x^k) \right) \\ &= \min_{x \in \mathbb{R}^n} \max_{i=1, \dots, m} \left(\left\langle \nabla f_i(x^k) - L_k \nabla \omega(x^k), x - x^k \right\rangle + g_i(x) - g_i(x^k) \right. \\ & \quad \left. + L_k \omega(x) - L_k \omega(x^k) \right), \end{aligned}$$

where the equality follows from the definition of $B_\omega(x, x^k)$. Dividing the objective function of the above problem by L_k , we obtain the equivalent problem:

$$\min_{x \in \mathbb{R}^n} \max_{i=1, \dots, m} \left(\left\langle \frac{1}{L_k} \nabla f_i(x^k) - \nabla \omega(x^k), x \right\rangle + \frac{1}{L_k} g_i(x) - \frac{1}{L_k} g_i(x^k) + \omega(x) \right). \quad (3.3.7)$$

For some k , defining

$$\varphi(x) := \max_{i=1, \dots, m} \left(\left\langle \frac{1}{L_k} \nabla f_i(x^k) - \nabla \omega(x^k), x \right\rangle + \frac{1}{L_k} g_i(x) - \frac{1}{L_k} g_i(x^k) \right), \quad (3.3.8)$$

the problem can be written as

$$\min_{x \in \mathbb{R}^n} \Psi(x) := (\varphi(x) + \omega(x)). \quad (3.3.9)$$

The function Ψ is closed since g , ω , and the maximum of closed functions are closed. It is also proper because $\operatorname{dom} g \cap \operatorname{dom} \omega \neq \emptyset$. Moreover, Ψ is convex because it takes the maximum of convex functions (and g_i is convex). From Assumption 4.3.2, $\{x \in \operatorname{dom} g \mid \omega(x) \leq \alpha\}$ is compact for all $\alpha \in \mathbb{R}$, so this problem is solvable. This means that $p_{L_k}(x^k)$ is well defined. Since $\operatorname{dom} \Psi$ is nonempty and convex, it follows that there

exists x_0 in the relative interior of $\text{dom } \Psi$ [13, Theorem 3.17], and consequently, by [13, Theorem 3.18], $\partial \Psi(x_0) \neq \emptyset$. Furthermore, if ω is strongly convex, we conclude that φ is a proper closed and strongly convex function, and hence, from [13, Lemma 9.7], the subproblem has a unique optimal solution in $\text{dom } g \cap \text{dom } \partial \omega$. ■

From Assumption 4.3.2 (in particular, the relative smoothness of f_i) and the definition of L in (3.3.1), if $L_k \geq L$, for all i we have

$$F_i(x^{k+1}) - F_i(x^k) \leq \nabla f_i(x^k)^\top (x^{k+1} - x^k) + g_i(x^{k+1}) - g_i(x^k) + L_k B_\omega(x^{k+1}, x^k). \quad (3.3.10)$$

Since x^{k+1} is the optimal solution of (3.3.4), the maximum in i of the right-hand side of (3.3.10) is less than or equal to zero. Thus, for all i

$$F_i(x^{k+1}) \leq F_i(x^k) \quad \text{for all } k, \quad (3.3.11)$$

that is, the objective functions decrease monotonically. In the following subsections, we consider two stepsize rules for our method.

3.3.1 Constant stepsize

Now consider the constant stepsize, which we set as $L_k = \bar{L}$ for all k , with $\bar{L} > L$ and L is given in (3.3.1). Then the proximal gradient method with Bregman distance is given below.

Algorithm 4: Multi-objective proximal gradient method with Bregman distance and constant stepsize

- Step 1 Choose $L_k := \bar{L}$ with $\bar{L} > L$, $\varepsilon > 0$, $x^0 \in \text{dom } g \cap \text{dom } \partial \omega$, and set $k := 0$.
 - Step 2 Compute $p_{L_k}(x^k)$ by solving subproblem (3.3.4).
 - Step 3 If $\|p_{L_k}(x^k) - x^k\| < \varepsilon$, then stop.
 - Step 4 Set $x^{k+1} := p_{L_k}(x^k)$, $k := k + 1$, and go to Step 2.
-

3.3.2 Backtracking procedure

Now we consider the backtracking procedure. In the beginning, let $L_{-1} = s$ with $s > 0$. At iteration $k \geq 0$, let $L_k = L_{k-1}$. Then, while existing i such that

$$f_i(p_{L_k}(x^k)) > f_i(x^k) + \langle \nabla f_i(x^k), p_{L_k}(x^k) - x^k \rangle + L_k B_\omega(p_{L_k}(x^k), x^k),$$

we set $L_k := \eta L_k$ where $\eta > 1$. In other words, $L_k = L_{k-1} \eta^{j_k}$, where j_k is the smallest nonnegative integer given as follows:

$$j_k := \underset{j \in \{0, 1, 2, \dots\}}{\operatorname{argmin}} \left(F(p_{L_{k-1} \eta^{j_k}}(x^k)) \preceq F(x^k) + JF(x^k)^\top (p_{L_{k-1} \eta^{j_k}}(x^k) - x^k) \right. \quad (3.3.12)$$

$$\left. + L_{k-1} \eta^{j_k} B_\omega(p_{L_{k-1} \eta^{j_k}}(x^k), x^k) \right).$$

The rule above ensures that (3.3.10) is still satisfied at each iteration. In addition, the L_k that the backtracking procedure produces satisfies the following bounds for all k :

$$s \leq L_k \leq \max \{ \eta L, s \}.$$

The inequality $s \leq L_k$ is trivial. To prove the inequality $L_k \leq \max \{ \eta L, s \}$, we note that either $L_k = s$ or $L_k > s$. In the latter case there exists an index $0 \leq k' \leq k$ and some i for which the inequality (3.3.10) is not satisfied with $k = k'$ and replacing L_k with L_k/η . From the relative smoothness of f_i , this implies in particular that $L_k/\eta < L^i \leq L$. Thus, we have shown that $L_k \leq \max \{ \eta L, s \}$. Namely, $L_k \leq \alpha L$, where $\alpha = \max \{ \eta, s/L \}$. We also note that the bounds on L_k can be rewritten as

$$\beta L \leq L_k \leq \alpha L,$$

where

$$\alpha = \begin{cases} \frac{\bar{L}}{L}, & \text{constant,} \\ \max \{ \eta, \frac{s}{L} \}, & \text{backtracking,} \end{cases} \quad \beta = \begin{cases} \frac{\bar{L}}{L}, & \text{constant} \\ \frac{s}{L}, & \text{backtracking.} \end{cases}$$

So the algorithm with backtracking stepsize is given below.

Algorithm 5: Multi-objective proximal gradient method with Bregman distance and backtracking procedure

- Step 1 Choose $s > 0$, $\eta > 1$, $\varepsilon > 0$, $x^0 \in \text{dom } g \cap \text{dom } \partial\omega$, and set $L_{-1} = s$, $k := 0$.
 - Step 2 Compute L_k by solving (3.3.12).
 - Step 3 Compute $p_{L_k}(x^k)$ by solving subproblem (3.3.4).
 - Step 4 If $\|p_{L_k}(x^k) - x^k\| < \varepsilon$, then stop.
 - Step 5 Set $x^{k+1} := p_{L_k}(x^k)$, $k := k + 1$, and go to Step 2.
-

3.4 Convergence analysis

In this section, we prove that the sequences generated by Algorithms 4 and 5 converge to Pareto stationary points and discuss their rate of convergence. From now on, let us assume that an infinite sequence is generated.

3.4.1 Convergence to Pareto stationary points

Now we analyze the convergence of the proximal gradient method with Bregman distance.

Lemma 3.4.1. *Let $\{x^k\}$ be generated by Algorithms 4 or 5 and suppose that $\{F_i(x^k)\}$ is bounded from below for all $i = 1, \dots, m$. Then we have*

$$\lim_{k \rightarrow \infty} B_\omega(x^k, x^{k+1}) = 0. \quad (3.4.1)$$

Proof. At the k th iteration,

$$f_i(x^{k+1}) + g_i(x^{k+1})$$

$$\begin{aligned}
 &= f_i(x^k) + g_i(x^k) + f_i(x^{k+1}) - f_i(x^k) + g_i(x^{k+1}) - g_i(x^k) \\
 &\leq f_i(x^k) + g_i(x^k) + \nabla f_i(x^k)(x^{k+1} - x^k) + L_k B_\omega(x^{k+1}, x^k) + g_i(x^{k+1}) - g_i(x^k) \\
 &\leq f_i(x^k) + g_i(x^k) + \psi_{x^k}(x^{k+1}) + L_k B_\omega(x^{k+1}, x^k) \\
 &\leq f_i(x^k) + g_i(x^k) + \psi_{x^k}(x) + L_k (B_\omega(x, x^k) - B_\omega(x, x^{k+1}))
 \end{aligned}$$

for all x . Here, the first inequality follows from (3.3.11). The second inequality follows from the definition of $\psi_{x^k}(x^{k+1})$. And the third inequality follows from Lemma 2.2.2 with $\theta = \psi_{x^k}/L_k$. Letting $x = x^k$, and recalling (3.2.2) and (3.3.3), we obtain

$$\begin{aligned}
 f_i(x^{k+1}) + g_i(x^{k+1}) &\leq f_i(x^k) + g_i(x^k) - L_k B_\omega(x^k, x^{k+1}) \\
 &\leq f_i(x^k) + g_i(x^k) - \beta L B_\omega(x^k, x^{k+1}).
 \end{aligned}$$

Since $\{F_i(x^k)\}$ is bounded from below from the assumption, there exists $\tilde{F}_i \leq F_i(x^k) = f_i(x^k) + g_i(x^k)$ for all i and k . Adding up the above inequality from $k = 0$ to $k = \hat{k}$, we obtain

$$f_i(x^{\hat{k}+1}) + g_i(x^{\hat{k}+1}) \leq f_i(x^0) + g_i(x^0) - \beta L \sum_{k=0}^{\hat{k}} B_\omega(x^k, x^{k+1}).$$

Because $\beta L > 0$, we have

$$\sum_{k=0}^{\hat{k}} B_\omega(x^k, x^{k+1}) \leq (\beta L)^{-1} (f_i(x^0) + g_i(x^0) - f_i(x^{\hat{k}+1}) - g_i(x^{\hat{k}+1})),$$

and thus

$$\sum_{k=0}^{\hat{k}} B_\omega(x^k, x^{k+1}) < \infty.$$

It then follows from the nonnegativity of $B_\omega(x^k, x^{k+1})$ for all k that

$$\lim_{k \rightarrow \infty} B_\omega(x^k, x^{k+1}) = 0,$$

which completes the proof. ■

In order to give the convergence analysis, we consider the following assumption. In Remark 3.4.1 and Section 3.4.1 we will see that this assumption is actually not so strict.

Assumption 3.4.1. *Let $\{x^k\}$ be a sequence generated by Algorithms 4 or 5. If*

$$\lim_{k \rightarrow \infty} B_\omega(x^k, x^{k+1}) = 0,$$

then

$$z^* := \lim_{k \rightarrow \infty} \{\nabla \omega(x^{k+1}) - \nabla \omega(x^k)\} \in N_{\text{cl}(\text{dom } \partial \omega)}(x^*),$$

where x^ is an accumulation point of $\{x^k\}$.*

Theorem 3.4.1. *If Assumption 3.4.1 holds, then every accumulation point of the sequence $\{x^k\}$ generated by Algorithms 4 or 5, if it exists, is a Pareto stationary point.*

Proof.

From the optimality condition of the subproblem (3.3.4), we have

$$\sum_{i=1}^m \lambda_i^k (\nabla f_i(x^k) + \eta_i(x^{k+1})) + L_k (\nabla \omega(x^{k+1}) - \nabla \omega(x^k)) = 0,$$

where $\eta_i(x^{k+1}) \in \partial g_i(x^{k+1})$, $\sum_{i=1}^m \lambda_i^k = 1$, $\lambda_i^k \geq 0$ for all $i = 1, \dots, m$, and $\lambda_i^k = 0$ when $i \notin \mathcal{I}_{x^k}(x^{k+1})$ and

$$\begin{aligned} \mathcal{I}_{x^k}(x^{k+1}) \\ := \{i \in \{1, \dots, m\} \mid \psi_{x^k}(x^{k+1}) = \nabla f_i(x^k)^\top (x^{k+1} - x^k) + g_i(x^{k+1}) - g_i(x^k)\}. \end{aligned}$$

Because $\{\lambda_i^k\}$ and $\{L_k\}$ are bounded, we assume without loss of generality that there exist λ_i^* with $\sum_{i=1}^m \lambda_i^* = 1$, $\lambda_i^* \geq 0$ for all $i = 1, \dots, m$, and $L^* > 0$ such that

$$\begin{aligned} & \lim_{k \rightarrow \infty} \sum_{i=1}^m \lambda_i^k (\nabla f_i(x^k) + \eta_i(x^{k+1})) + L_k (\nabla \omega(x^{k+1}) - \nabla \omega(x^k)) \\ &= \sum_{i=1}^m \lambda_i^* (\nabla f_i(x^*) + \eta_i(x^*)) + L^* z^* \\ &= 0, \end{aligned}$$

where $\eta_i(x^*) \in \partial g_i(x^*)$ and z^* is given in Assumption 3.4.1. Then, from this assumption and Lemma 3.4.1, we have

$$-\sum_{i=1}^m \lambda_i^* (\nabla f_i(x^*) + \eta_i(x^*)) = L^* z^* \in N_{\text{cl}(\text{dom } \partial \omega)}(x^*),$$

where we use the fact that $L^* > 0$ and $N_{\text{cl}(\text{dom } \partial \omega)}(x^*)$ is a cone. Therefore,

$$\sum_{i=1}^m \langle \lambda_i^* (\nabla f_i(x^*) + \eta_i(x^*)), y - x^* \rangle \geq 0 \quad \text{for all } y \in \text{cl}(\text{dom } \partial \omega).$$

This implies that for all $y \in \text{cl}(\text{dom } \partial \omega)$, there exists at least one i with $\lambda_i^* > 0$ such that

$$\langle \nabla f_i(x^*) + \eta_i(x^*), y - x^* \rangle \geq 0. \quad (3.4.2)$$

Moreover, since g_i is convex, we have $g'_i(x^*; y - x^*) = \sup_{v \in \partial g_i(x^*)} \langle v, y - x^* \rangle$ and thus

$$g'_i(x^*; y - x^*) \geq \langle \eta_i(x^*), y - x^* \rangle.$$

Therefore, from (3.4.2), for all $y \in \text{cl}(\text{dom } \partial \omega)$ we obtain

$$F'_i(x^*; y - x^*) = \langle \nabla f_i(x^*), y - x^* \rangle + g'_i(x^*; y - x^*) \geq 0$$

for some i , which implies that $\max F'_i(x^*; y - x^*) \geq 0$, and thus x^* is Pareto stationary. ■

Remark 3.4.1. We observe that if $x^* \in \text{int}(\text{dom } \partial\omega)$ or if $\nabla\omega$ is Hölder continuous, then $z^* = 0$. This ensures the fulfillment of Assumption 3.4.1, implying that the accumulation point x^* is Pareto stationary. This aligns with the result presented in [38, Theorem 2(iii)], where $\nabla\omega$ is assumed to be Lipschitz continuous.

Special case

We now show that Assumption 3.4.1 is not so strict, by showing an example with Kullback-Leibler divergence. For all $i = 1, \dots, m$, define $g_i(x)$ as the indicator function of $\Delta_n := \{x \in \mathbb{R}^n \mid \sum_{i=1}^n x_i = 1, x \geq 0\}$. Let $\omega(x) = \sum_{i=1}^n x_i \ln x_i$ with constraint condition $x \succeq 0$ for the problem (3.1.1). Note that $\text{cl}(\text{dom } \partial\omega) = \{x \mid x \succeq 0\}$. Also, we recall that in this case, $\nabla\omega$ is not Hölder or Lipschitz continuous.

Suppose that the algorithm generates a sequence $\{x^k\}$. It is well-known that ω is 1-strongly convex over Δ_n , and in this case the following inequality holds:

$$B_\omega(x^{k+1}, x^k) \geq \frac{1}{2} \|x^{k+1} - x^k\|_1,$$

where $\|\cdot\|_1$ denotes the 1-norm. Then, combined with Lemma 3.4.1, we can conclude that $\lim_{k \rightarrow \infty} \|x^{k+1} - x^k\|_1 = 0$, which means, for all i , that

$$\lim_{k \rightarrow \infty} |x_i^{k+1} - x_i^k| = 0. \quad (3.4.3)$$

Now, recall that the optimality condition of the subproblem (3.3.4) is given by

$$\sum_{i=1}^m \lambda_i^k (\nabla f_i(x^k) + \eta_i(x^{k+1})) + L_k (\nabla\omega(x^{k+1}) - \nabla\omega(x^k)) = 0,$$

where $\sum_{i=1}^m \lambda_i^k = 1$, $\lambda_i^k \geq 0$ for all $i = 1, \dots, m$, and $\eta_i(x^{k+1}) \in \partial g_i(x^{k+1})$. Note also that for any index j ,

$$(\nabla\omega(x^{k+1}) - \nabla\omega(x^k))_j = \ln \left(\frac{x_j^{k+1}}{x_j^k} \right).$$

Let us now consider an accumulation point x^* of $\{x^k\}$. For any index j such that $x_j^* = 0$, if $\sum_{i=1}^m \lambda_i^k (\nabla f_i(x^k) + \eta_i(x^{k+1}))_j < 0$, it follows that $\ln(x_j^{k+1}/x_j^k) > 0$ because $L_k > 0$. This shows that $x_j^{k+1} > x_j^k$, which contradicts the fact that $x_j^* = 0$. Consequently, it must be the case that $\lim_{k \rightarrow \infty} \sum_{i=1}^m \lambda_i^k (\nabla f_i(x^k) + \eta_i(x^{k+1}))_j \geq 0$, resulting in $\lim_{k \rightarrow \infty} \ln(x_j^{k+1}/x_j^k) \leq 0$. On the other hand, for an index j such that $x_j^* > 0$, the condition (3.4.3) implies $\lim_{k \rightarrow \infty} \ln(x_j^{k+1}/x_j^k) = 0$.

Furthermore, assuming that $\{\nabla f_i(x^k)\}$ and $\{\eta_i(x^{k+1})\}$ are bounded for all i , then $\{\nabla\omega(x^{k+1}) - \nabla\omega(x^k)\}$ is also bounded. If z^* is its accumulation point, then $z_j^* \leq 0$ for all $x_j^* = 0$ and $z_j^* = 0$ for all $x_j^* > 0$, which means $z^* \in N_{\{x \mid x \geq 0\}}(x^*)$, and thus the Assumption 3.4.1 is satisfied in this case.

Remark 3.4.2. We observe that the merit function $v_\ell(x)$ as defined in [38] is not well defined on $\text{bd}(\text{dom } \partial\omega)$, especially when $\omega(x) = \sum_{i=1}^n x_i \ln x_i$. Similarly, due to this reason, the discourse on the nonconvex case in [41] is questionable.

3.4.2 Convergence rate analysis

Let us now recall a merit function for the multi-objective optimization problem, and use it to estimate the convergence rate. Here, we will discuss only the convex and the strongly convex cases. The merit function is the simple function $u_0: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ defined as follows [131]:

$$u_0(x) := \sup_{y \in \mathbb{R}^n} \min_{i \in \{1, \dots, m\}} \{F_i(x) - F_i(y)\}. \quad (3.4.4)$$

The convex case

Here we use the function $u_0(\cdot)$ to analyze the convergence rate. First, we give the following lemma. Note that we state it with f_i and g_i having general convexity parameters, which turn out to be zero in this subsection. It will not make a difference here, but it will be important in the discussion of the next subsection.

Lemma 3.4.2. *Assume that f_i is μ_i -strongly convex relative to ω and g_i is ν_i -strongly convex relative to ω , and write $\mu := \min_{i \in \{1, \dots, m\}} \mu_i$ and $\nu := \min_{i \in \{1, \dots, m\}} \nu_i$. Then, for all $x \in \text{dom } g$ it follows that*

$$\begin{aligned} \sum_{i=1}^m \beta_i^k (F_i(x^{k+1}) - F_i(x)) &\leq L_k (B_\omega(x, x^k) - B_\omega(x, x^{k+1})) \\ &\quad - \mu B_\omega(x, x^k) - \nu B_\omega(x, x^{k+1}), \end{aligned}$$

where β_i^k satisfies the following conditions:

(i) There exists $\eta_i(x^{k+1}) \in \partial g_i(x^{k+1})$ such that

$$\sum_{i=1}^m \beta_i^k (\nabla f_i(x^k) + \eta_i(x^{k+1})) + L_k (\nabla \omega(x^{k+1}) - \nabla \omega(x^k)) = 0,$$

(ii) $\sum_{i=1}^m \beta_i^k = 1$, $\beta_i^k \geq 0$ ($i \in \mathcal{I}_{x^k}(x^{k+1})$) and $\beta_i^k = 0$ ($i \notin \mathcal{I}_{x^k}(x^{k+1})$), where $\mathcal{I}_{x^k}(x^{k+1}) := \{i \mid \psi_{x^k}(x^{k+1}) = \nabla f_i(x^k)^\top (x^{k+1} - x^k) + g_i(x^{k+1}) - g_i(x^k)\}$.

Proof. For all i , (3.3.10) holds for both Algorithms 4 and 5, that is

$$F_i(x^{k+1}) - F_i(x^k) \leq \nabla f_i(x^k)^\top (x^{k+1} - x^k) + g_i(x^{k+1}) - g_i(x^k) + L_k B_\omega(x^{k+1}, x^k). \quad (3.4.5)$$

The above inequality and relative strong convexity of f_i with modulus μ_i give

$$\begin{aligned} F_i(x^{k+1}) - F_i(x) &= (F_i(x^k) - F_i(x)) + (F_i(x^{k+1}) - F_i(x^k)) \\ &\leq (\nabla f_i(x^k)^\top (x^k - x) - \mu_i B_\omega(x, x^k) + g_i(x^k) - g_i(x)) \\ &\quad + (\nabla f_i(x^k)^\top (x^{k+1} - x^k) + g_i(x^{k+1}) - g_i(x^k) + L_k B_\omega(x^{k+1}, x^k)) \\ &\leq \nabla f_i(x^k)^\top (x^{k+1} - x) + g_i(x^{k+1}) - g_i(x) - \mu B_\omega(x, x^k) + L_k B_\omega(x^{k+1}, x^k) \\ &\leq (\nabla f_i(x^k) + \eta_i(x^{k+1}))^\top (x^{k+1} - x) - \mu B_\omega(x, x^k) - \nu B_\omega(x, x^{k+1}) + L_k B_\omega(x^{k+1}, x^k) \end{aligned}$$

for all $x \in \text{dom } g$, where the second inequality follows from the definition of μ and the last one comes from the relative strong convexity of g_i . Multiplying the above inequality by β_i^k and summing for all $i \in \{1, \dots, m\}$, the conditions (i) and (ii) give

$$\begin{aligned}
 & \sum_{i=1}^m \beta_i^k (F_i(x^{k+1}) - F_i(x)) \\
 & \leq -L_k (\nabla \omega(x^{k+1}) - \nabla \omega(x^k))^\top (x^{k+1} - x) + L_k B_\omega(x^{k+1}, x^k) \\
 & \quad - \mu B_\omega(x, x^k) - \nu B_\omega(x, x^{k+1}) \\
 & = L_k \left(\nabla \omega(x^k)^\top x^{k+1} - \nabla \omega(x^k)^\top x + \nabla \omega(x^{k+1})^\top (x - x^{k+1}) + \omega(x^{k+1}) - \omega(x^k) \right. \\
 & \quad \left. - \nabla \omega(x^k)^\top (x^{k+1} - x^k) \right) - \mu B_\omega(x, x^k) - \nu B_\omega(x, x^{k+1}) \\
 & = L_k \left(\omega(x) - \omega(x^k) - \nabla \omega(x^k)^\top (x - x^k) - \omega(x) + \omega(x^{k+1}) + \nabla \omega(x^{k+1})^\top (x - x^{k+1}) \right) \\
 & \quad - \mu B_\omega(x, x^k) - \nu B_\omega(x, x^{k+1}) \\
 & = L_k (B_\omega(x - x^k) - B_\omega(x - x^{k+1})) - \mu B_\omega(x, x^k) - \nu B_\omega(x, x^{k+1}),
 \end{aligned}$$

where the second and last equalities follow from the definition of Bregman distance. $\blacksquare$

We now prove the convergence rate assuming convexity of the objective functions. The following assumption is considered, which is equivalent to the existence of at least one optimal point in the single-objective case. As suggested in [130, Remark 5.2], it is not particularly strong even in the multi-objective case.

Assumption 3.4.2. *Let X^* be the set of weakly Pareto optimal points for the multi-objective problem, and define the level set of F for $\alpha \in \mathbb{R}^m$ by $\Omega_F(\alpha) := \{x \in \mathbb{R}^n \mid F(x) \preceq \alpha\}$. Then, for all $x \in \Omega_F(F(x^0))$ there exists $x^* \in X^*$ such that $F(x^*) \preceq F(x)$ and*

$$R := \sup_{F^* \in F(X^* \cap \Omega_F(F(x^0)))} \inf_{x \in F^{-1}(\{F^*\})} B_\omega(x, x^0) < \infty.$$

Theorem 3.4.2. *Assume that F_i is convex for all $i \in \{1, \dots, m\}$. Under Assumption 3.4.2, Algorithm 4 (or Algorithm 5) generates a sequence $\{x^k\}$ such that*

$$u_0(x^k) \leq \frac{\alpha LR}{k} \quad \text{for all } k \geq 1,$$

where $\alpha = \frac{\bar{\ell}}{L}$ in the constant stepsize setting and $\alpha = \max\{\eta, \frac{s}{L}\}$ if the backtracking rule is employed.

Proof. From Lemma 3.4.2 and the convexity of f_i and g_i , for all $x \in \mathbb{R}^n$ we have

$$\sum_{i=1}^m \beta_i^k (F_i(x^{k+1}) - F_i(x)) \leq L_k (B_\omega(x - x^k) - B_\omega(x - x^{k+1})).$$

Adding up the above inequality from $k = 0$ to $k = \hat{k}$, we obtain

$$\begin{aligned} \sum_{k=0}^{\hat{k}} \sum_{i=1}^m \beta_i^k (F_i(x^{k+1}) - F_i(x)) &\leq L_k (B_\omega(x, x^0) - B_\omega(x, x^{\hat{k}+1})) \\ &\leq L_k B_\omega(x, x^0). \end{aligned}$$

The rest of the proof follows similarly to the proof of [130, Theorem 5.2]. $\blacksquare$

The strongly convex case

Here, we show that $\{x^k\}$ generated by Algorithms 4 and 5 converge linearly to a Pareto optimal point if F_i is strongly convex relative to ω .

Theorem 3.4.3. *Let $\{x^k\}$ be generated by Algorithm 4 or 5 and suppose that Assumption 3.4.1 holds. Let f_i and g_i have convexity parameters $\mu_i \in \mathbb{R}$ and $\nu_i \in \mathbb{R}$, respectively, and write $\mu := \min_{i \in \{1, \dots, m\}} \mu_i$ and $\nu := \min_{i \in \{1, \dots, m\}} \nu_i$. Then there exists a Pareto optimal point $x^* \in \mathbb{R}^n$ such that for each iteration k ,*

$$(L_k + \nu) B_\omega(x^*, x^{k+1}) \leq (L_k - \mu) B_\omega(x^*, x^k). \quad (3.4.6)$$

Furthermore, assume that ω is σ -strongly convex with $\sigma > 0$, and $\nabla \omega$ is q -Hölder continuous with parameter c and $0 < q \leq 1$. Then, there exists a Pareto optimal point $x^* \in \mathbb{R}^n$ such that for each iteration k ,

$$\|x^{k+1} - x^*\| \leq \sqrt{\frac{c(\alpha L - \mu)}{\sigma(\beta L + \nu)}} \|x^k - x^*\|^{\frac{q+1}{2}},$$

where

$$\alpha = \begin{cases} \frac{\bar{L}}{L}, & \text{constant,} \\ \max\{\eta, \frac{s}{L}\}, & \text{backtracking,} \end{cases} \quad \beta = \begin{cases} \frac{\bar{L}}{L}, & \text{constant} \\ \frac{s}{L}, & \text{backtracking.} \end{cases}$$

Thus, we have

$$\|x^k - x^*\| \leq \left(\sqrt{\frac{c(\alpha L - \mu)}{\sigma(\beta L + \nu)}} \right)^{\sum_{i=0}^{k-1} (\frac{q+1}{2})^i} \|x^0 - x^*\|^{(\frac{q+1}{2})^k},$$

and if $0 < c \leq \frac{\sigma(\beta L + \nu)}{\alpha L - \mu}$ then $0 < \sqrt{\frac{c(\alpha L - \mu)}{\sigma(\beta L + \nu)}} \leq 1$. In particular, if $\nabla \omega$ is Lipschitz continuous, we obtain

$$\|x^k - x^*\| \leq \left(\sqrt{\frac{c(\alpha L - \mu)}{\sigma(\beta L + \nu)}} \right)^k \|x^0 - x^*\|.$$

Proof. Since each F_i is strongly convex relative to ω , the level set of every F_i is bounded. Thus, $\{x^k\}$ has an accumulation point $x^* \in \mathbb{R}^n$. Note that x^* is Pareto stationary from Theorem 3.4.1, and thus Pareto optimal because of the strong convexity assumption. From Lemma 3.4.2, we have

$$\sum_{i=1}^m \beta_i^k (F_i(x^{k+1}) - F_i(x^*)) \leq L_k (B_\omega(x^*, x^k) - B_\omega(x^*, x^{k+1})) - \mu B_\omega(x^*, x^k) - \nu B_\omega(x^*, x^{k+1}),$$

where β_i^k satisfies the conditions (i) and (ii) of Lemma 3.4.2. Since the left-hand side of the above inequality is nonnegative because of (3.3.11), we obtain

$$0 \leq L_k (B_\omega(x^*, x^k) - B_\omega(x^*, x^{k+1})) - \mu B_\omega(x^*, x^k) - \nu B_\omega(x^*, x^{k+1}).$$

Namely,

$$(L_k + \nu) B_\omega(x^*, x^{k+1}) \leq (L_k - \mu) B_\omega(x^*, x^k). \quad (3.4.7)$$

Similiar to the so-called descent lemma [23, Proposition A.24], using the Hölder continuity of $\nabla\omega$, i.e., $\|\nabla\omega(y) - \nabla\omega(x)\| \leq c \|y - x\|^q$, we obtain for all x, y ,

$$\omega(x) \leq \omega(y) + \nabla\omega(y)^\top (x - y) + \frac{c}{2} \|y - x\|^{q+1}.$$

Combined with the σ -strong convexity of ω as well as the definition of Bregman distance, we have

$$\frac{\sigma}{2} \|x - y\|^2 \leq B_\omega(x, y) \leq \frac{c}{2} \|x - y\|^{q+1}.$$

The above inequality and (3.4.7) give

$$\frac{\sigma(L_k + \nu)}{2} \|x^* - x^{k+1}\|^2 \leq \frac{c(L_k - \mu)}{2} \|x^* - x^k\|^{q+1}, \quad (3.4.8)$$

which is equivalent to

$$\|x^{k+1} - x^*\| \leq \sqrt{\frac{c(L_k - \mu)}{\sigma(L_k + \nu)}} \|x^k - x^*\|^{\frac{q+1}{2}}.$$

Using the bounds for L_k , we obtain

$$\|x^{k+1} - x^*\| \leq \sqrt{\frac{c(\alpha L - \mu)}{\sigma(\beta L + \nu)}} \|x^k - x^*\|^{\frac{q+1}{2}},$$

where $\alpha = \frac{\bar{L}}{L}$ in the constant stepsize setting and $\alpha = \max\{\eta, \frac{s}{L}\}$ if the backtracking rule is employed. The result then follows by applying this inequality k times. ■

Note that under strong convexity assumption, from (3.4.6) we have

$$B_\omega(x^*, x^k) \leq \left(\frac{\alpha L - \mu}{\beta L + \nu} \right)^k B_\omega(x^*, x^0). \quad (3.4.9)$$

Moreover, we also have linear convergence (as usual, defined with Euclidean norm) when Lipschitz continuity of $\nabla\omega$ is satisfied. One future work will be to see if this condition can be removed to establish the same rate of convergence in the strongly convex case.

3.5 Conclusion

We proposed a proximal gradient method with Bregman distance for multi-objective optimization problems. We also used two-step size strategies: the constant stepsize and the backtracking strategy. We prove that the sequence generated by the algorithms can converge to a Pareto stationary point and further analyze its convergence rate through a merit function. Finally, we proved the convergence rates for convex ($O(1/k)$), and strongly convex ($O(r^k)$ for some $r \in (0, 1)$) problems. We point out that our proof is based on Assumption [3.4.1](#), which is less strict than assuming Hölder continuity of the gradient of the reference function.

Chapter 4

Nonlinear conjugate gradient method for vector optimization on Riemannian manifolds with retraction and vector transport

4.1 Introduction

Building upon the frameworks of vector optimization and Riemannian geometry presented in Chapter 2, this chapter explores the intersection of these two fields. For single-objective optimization, nonlinear conjugate gradient methods have been successfully extended to the Riemannian setting [120, 142, 123]. In parallel, for Euclidean vector optimization, the nonlinear conjugate gradient method was proposed in [100], and various parameter choices have been investigated in [74, 73, 37, 138, 112].

Although other first-order strategies, including steepest descent [19, 18], subgradient [17], and proximal point methods [22], have been successfully adapted to vector optimization on manifolds, the extension of conjugate gradient methods to this domain remains relatively unexplored.

In this chapter, we propose a general framework for the nonlinear conjugate gradient method for vector optimization on Riemannian manifolds. It is important to distinguish our contribution from related recent works. We acknowledge that Ferreira, Lucambio-Pérez, and Prudente [56] have also investigated similar methods. However, their approach relies on specific geometric operators, namely, the exponential map and parallel transport. While theoretically sound, these operators are often computationally expensive or intractable for many manifolds. In contrast, our framework utilizes general retractions and vector transports, making it applicable to a broader range of problems where computationally efficient approximations are required. A parallel study by Najafi et al. [108] also employs general retractions and vector transports but focuses specifically on multi-objective optimization rather than general vector optimization. Furthermore, their convergence

analysis for the Fletcher–Reeves (FR) and Dai–Yuan (DY) parameters relies on imposing a sufficient descent condition and an additional inequality. In this work, we dispense with these restrictive assumptions. We demonstrate that our algorithm naturally generates descent directions and establish its global convergence not only for the FR and DY parameters but also for the Conjugate Descent (CD) parameter. Additionally, we discuss the properties of several other classical parameters within this Riemannian vector framework.

The remainder of this chapter is organized as follows. In Section 4.2, we formally propose the Nonlinear Vector Riemannian Conjugate Gradient (NVRCG) method, formulated using general retractions and vector transports. Section 4.3 is dedicated to the convergence analysis; we extend the Zoutendijk-type condition to the Riemannian context and prove the convergence of the NVRCG method under the Riemannian extensions of FR, CD, and DY parameters. We also provide preliminary results for the Polak–Ribière–Polyak (PRP), Hestenes–Stiefel (HS), and Liu–Storey (LS) parameters. Numerical experiments illustrating the performance of the proposed methods are presented in Section 4.4. Finally, Section 4.5 summarizes the chapter and outlines potential directions for future research.

4.2 Nonlinear vector Riemannian conjugate gradient method

We consider the following unconstrained vector optimization problem:

$$\begin{aligned} \min_K F(x) \\ \text{s. t. } x \in \mathcal{M}, \end{aligned} \tag{4.2.1}$$

where $F : \mathcal{M} \rightarrow \mathbb{R}^m$ is continuously differentiable and $\mathcal{M}$ is an n -dimensional smooth manifold with a Riemannian metric $\langle \cdot, \cdot \rangle$. Define the function $\varphi : \mathbb{R}^m \rightarrow \mathbb{R}$ by

$$\varphi(y) := \sup\{y^\top w \mid w \in C\}.$$

It is worth noting that $\varphi(y)$ is actually the support function of the set C . Therefore, φ is convex, positively homogeneous, and subadditive. In view of the compactness of C , which is given as $C = \{\omega \in K^* \mid \|\omega\| = 1\}$, φ is well defined. It can easily be seen that when $m = 1$, $\varphi(y) = y$ holds. Then, we have

$$\varphi(DF(x)d) = \sup\{[DF(x)d]^\top w \mid w \in C\}.$$

Note that it can be viewed as a nonlinear programming problem when $K = \mathbb{R}_+^m$. The function φ has some useful properties, which we recall below.

Lemma 4.2.1. [79, Lemma 3.1] *Let $y, y' \in \mathbb{R}^m$. Then, the following statements hold:*

- (1) $\varphi(y + y') \leq \varphi(y) + \varphi(y')$ and $\varphi(y) - \varphi(y') \leq \varphi(y - y')$;
- (2) If $y \preceq_K y'$, then $\varphi(y) \leq \varphi(y')$; if $y \prec_K y'$, then $\varphi(y) < \varphi(y')$;

(3) φ is Lipschitz continuous with constant 1 .

From the definition of function φ , we note that:

- d is a K -descent direction for K at x if and only if $\varphi(DF(x)d) < 0$,
- x is K -critical if and only if $\varphi(DF(x)d) \geq 0$ for all d .

Now, consider the extension of the notion of the steepest descent direction to the vector-valued case. We denote it as $v: \mathbb{R}^n \rightarrow \mathbb{R}^n$ and define it as

$$v(x) := \arg \min_d \left\{ \varphi(DF(x)d) + \frac{\|d\|_x^2}{2} \mid d \in T_x \mathcal{M} \right\}. \quad (4.2.2)$$

Also, let $\theta: \mathbb{R}^n \rightarrow \mathbb{R}$ be defined as $\theta(x) := \varphi(DF(x)v(x)) + \|v(x)\|^2/2$. We note that when $m = 1$ and $C = \{1\}$, we have $v(x) = -\nabla f(x)$, that is, we retrieve the classical steepest descent direction in nonlinear optimization.

Remark 4.2.1. For multiobjective optimization, where $K = \mathbb{R}_+^m$, with C given by the canonical basis of $\mathbb{R}^m$, the direction $v(x)$ can be computed by solving

$$\begin{aligned} \min \quad & \alpha + \frac{1}{2} \|d\|_x^2 \\ \text{s. t.} \quad & [JF(x)d]_i \leq \alpha, \quad i = 1, \dots, m, \\ & \langle x, d \rangle_x = 0, \end{aligned}$$

which is a convex quadratic problem with linear inequality and equality constraints.

With the above definition, we can easily obtain the following lemma.

Lemma 4.2.2. [79, lemma 3.3]

- (1) If x is K -critical, then $v(x) = 0$ and $\theta(x) = 0$.
- (2) If x is not K -critical, then $v(x) \neq 0$, $\theta(x) < 0$, $\varphi(DF(x)v(x)) < -\frac{\|v(x)\|^2}{2} < 0$, and $v(x)$ is a K -descent direction for F at x .
- (3) The mappings v and θ are continuous.

In order to get a proper decrease, line search procedures are crucial. In particular, for nonlinear conjugate gradient methods, the so-called Wolfe conditions are widely used, and thus we extend them to vector Riemannian manifolds. Let $e \in K$ be given such that

$$0 < \langle \omega, e \rangle \leq 1 \quad \text{for all } \omega \in C.$$

In particular, for vector optimization, where $K = \mathbb{R}_+^m$ and C is given by the canonical basis of $\mathbb{R}^m$, we can take $e = [1, \dots, 1]^\top \in \mathbb{R}^m$. Letting $0 < c_1 < c_2 < 1$, we propose the (weak) Wolfe conditions as follows:

$$F(R_{x_k}(t_k d_k)) \preceq_K F(x_k) + c_1 t_k \varphi(DF(x_k)[d_k]) e, \quad (4.2.3)$$

$$\varphi(DF(R_{x_k}(t_k d_k))[DR_{x_k}(t_k d_k)[d_k]]) \geq c_2 \varphi(DF(x_k)[d_k]), \quad (4.2.4)$$

and the strong Wolfe conditions are given by

$$\begin{aligned} F(R_{x_k}(t_k d_k)) &\preceq_K F(x_k) + c_1 t_k \varphi(DF(x_k)[d_k]) e, \\ |\varphi(DF(R_{x_k}(t_k d_k))[DR_{x_k}(t_k d_k)[d_k]])| &\leq c_2 |\varphi(DF(x_k)[d_k])|. \end{aligned} \quad (4.2.5)$$

Note that the first condition is the same for both (weak) Wolfe and strong Wolfe conditions, and it is usually called Armijo condition.

In the Euclidean case, the search direction of the nonlinear vector conjugate gradient method is given by

$$d_{k+1} = v(x_{k+1}) + \beta_{k+1} d_k,$$

for $k \geq 0$, where $\beta_{k+1} \in \mathbb{R}$ is a scalar algorithmic parameter. To extend it to the Riemannian case, in this chapter, we use a vector transport called the differentiated retraction [120]. Using $\mathcal{T}^k : T_{x_k} \mathcal{M} \rightarrow T_{x_{k+1}} \mathcal{M}$ with $\mathcal{T}^k(d_k) := \mathcal{T}_{t_k d_k}^R(d_k) = DR_{x_k}(t_k d_k)[d_k]$, we have

$$d_{k+1} = v(x_{k+1}) + \beta_{k+1} \mathcal{T}^k(d_k).$$

For simplicity, we also define

$$\psi_{x,d}(td) := \varphi(DF(R_x(td))[DR_x(td)[d]]),$$

and then $\psi_{x_k, d_k}(0) = \varphi(DF(x_k)[d_k])$.

As reviewed in Chapter 1, several types of β have been proposed for single-objective optimization. We explicitly recall the following six types of β_{k+1} which are considered standard:

$$\begin{aligned} \beta_{k+1}^{\text{FR}} &= \frac{\|\nabla f(x_{k+1})\|_2^2}{\|\nabla f(x_k)\|_2^2}, \\ \beta_{k+1}^{\text{DY}} &= \frac{\|\nabla f(x_{k+1})\|_2^2}{(\nabla f(x_{k+1}) - \nabla f(x_k))^\top d_k}, \\ \beta_{k+1}^{\text{CD}} &= \frac{\|\nabla f(x_{k+1})\|_2^2}{-\nabla f(x_k)^\top d_k}, \\ \beta_{k+1}^{\text{PRP}} &= \frac{(\nabla f(x_{k+1}))^\top (\nabla f(x_{k+1}) - \nabla f(x_k))}{\|\nabla f(x_k)\|_2^2}, \\ \beta_{k+1}^{\text{HS}} &= \frac{(\nabla f(x_{k+1}))^\top (\nabla f(x_{k+1}) - \nabla f(x_k))}{(\nabla f(x_{k+1}) - \nabla f(x_k))^\top d_k}, \\ \beta_{k+1}^{\text{LS}} &= \frac{(\nabla f(x_{k+1}))^\top (\nabla f(x_{k+1}) - \nabla f(x_k))}{-\nabla f(x_k)^\top d_k} \end{aligned}$$

Also, the Riemannian versions of these types of β are provided in [121]. Here, we extend

them to vector optimization on Riemannian manifolds as follows:

$$\begin{aligned}\beta_k^{\text{FR}} &= \frac{\psi_{x_k, v_k}(0)}{\psi_{x_{k-1}, v_{k-1}}(0)}, \\ \beta_k^{\text{CD}} &= \frac{\psi_{x_k, v_k}(0)}{\psi_{x_{k-1}, d_{k-1}}(0)}, \\ \beta_k^{\text{DY}} &= \frac{-\psi_{x_k, v_k}(0)}{\psi_{x_{k-1}, d_{k-1}}(t_{k-1}d_{k-1}) - \psi_{x_{k-1}, d_{k-1}}(0)}, \\ \beta_k^{\text{PRP}} &= \frac{-\psi_{x_k, v_k}(0) + \psi_{x_k, \mathcal{T}^{k-1}(v_k)}(0)}{-\psi_{x_{k-1}, v_{k-1}}(0)}, \\ \beta_k^{\text{LS}} &= \frac{-\psi_{x_k, v_k}(0) + \psi_{x_{k-1}, \mathcal{T}^{k-1}(v_k)}(0)}{-\psi_{x_{k-1}, d_{k-1}}(0)}, \\ \beta_k^{\text{HS}} &= \frac{-\psi_{x_k, v_k}(0) + \psi_{x_{k-1}, \mathcal{T}^{k-1}(v_k)}(0)}{\psi_{x_k, \mathcal{T}^{k-1}(d_{k-1})}(0) - \psi_{x_{k-1}, d_{k-1}}(0)}.\end{aligned}$$

Noting that β_k^{FR} , β_k^{DY} , β_k^{HS} , and β_k^{PRP} were also proposed in [108].

Now we state our nonlinear conjugate gradient method for vector optimization problem (4.2.1) on Riemannian manifolds in Algorithm 6. In each iteration, the algorithm first computes the steepest descent direction by solving the subproblem (4.2.2) and then calculates the actual descent direction according to the updated formula, which also depends on the choice of β . Steps 3 and 4 give the proper step size and update the iteration. We will give a convergence analysis of the algorithm in the next section.

Algorithm 6: Nonlinear vector Riemannian conjugate gradient method (NVRCG)

Step 0. Let $x_0 \in \mathcal{M}$ and initialize $k \leftarrow 0$.

Step 1. Compute $v(x_k)$ as in (4.2.2). If $v(x_k) = 0$, then stop.

Step 2. Compute

$$d_k = \begin{cases} v(x_k), & \text{if } k = 0, \\ v(x_k) + \beta_k \mathcal{T}^{k-1}(d_{k-1}), & \text{if } k \geq 1, \end{cases}$$

where β_k is an algorithmic parameter.

Step 3. Compute a step size $t_k > 0$ by a line search procedure and set

$$x_{k+1} = R_{x_k}(t_k d_k).$$

Step 4. Set $k \leftarrow k + 1$, and go to Step 1.

4.3 Global convergence analysis

In this section, we prove the convergence of the algorithm with each type of β . First, we propose some assumptions for convergence analysis.

Assumption 4.3.1. *There exist constant $L > 0$, for all $x \in \mathcal{M}, d \in T_x \mathcal{M}$ with $\|d\|_x = 1, t \geq 0$, such that*

$$\|D(F \circ R_x)(td)[d] - D(F \circ R_x)(0)[d]\|_x \leq Lt.$$

Remark: When $F = [f_1, \dots, f_m]^\top$, we can simply give such an assumption for each f_i , and take $L = \max\{L_i, i = 1, \dots, m\}$.

Assumption 4.3.2. *Set $\mathcal{L} = \{x \mid F(x) \preceq_K F(x_0)\}$. All monotonically nonincreasing sequences in $F(\mathcal{L})$ are bounded below, i.e., if $\{G_k\}_{k \in \mathbb{N}} \subset F(\mathcal{L})$ and $G_{k+1} \preceq_K G_k$ for all k , then there exists $\mathcal{G} \in \mathbb{R}^m$ such that $\mathcal{G} \preceq_K G_k$ for all k .*

With Assumptions 4.3.1 and 4.3.2, we can prove that the method we propose satisfies Zoutendijk's type condition when d_k is a K -descent direction. We will prove d_k is a K -descent direction in each specific type of β later.

Proposition 4.3.1. *Suppose that Assumptions 4.3.1 and 4.3.2 hold. Consider the iteration, where d_k is a K -descent direction for F at x_k , and t_k satisfies the Wolfe conditions. Then, we have*

$$\sum_{k \geq 0} \frac{\varphi^2(DF(x_k)[d_k])}{\|d_k\|_{x_k}^2} < \infty.$$

Proof. Under Assumption 4.3.1, from the Wolfe conditions and Lemma 4.2.1, we have

$$\begin{aligned} & (c_2 - 1) \varphi(DF(x_k)[d_k]) \\ & \leq \varphi(DF(R_{x_k}(t_k d_k)) [DR_{x_k}(t_k d_k)[d_k]]) - \varphi(DF(x_k)[d_k]) \\ & \leq \|DF(R_{x_k}(t_k d_k)) DR_{x_k}(t_k d_k)[d_k] - DF(x_k) d_k\|_{x_k} \\ & = \|D(F \circ R_{x_k})(t_k d_k)[d_k] - D(F \circ R_{x_k})(0)[d_k]\|_{x_k} \\ & \leq \|d_k\|_{x_k} \left\| D(F \circ R_{x_k}) \left(t_k \|d_k\|_{x_k} \frac{d_k}{\|d_k\|_{x_k}} \right) \left[\frac{d_k}{\|d_k\|_{x_k}} \right] - D(F \circ R_{x_k})(0) \left[\frac{d_k}{\|d_k\|_{x_k}} \right] \right\|_{x_k} \\ & \leq Lt_k \|d_k\|_{x_k}^2. \end{aligned} \tag{4.3.1}$$

The first inequality is derived from the Wolfe condition (4.2.4), while the second inequality is obtained through the utilization of Lemma 4.2.1(3). The equality arises from the utilization of the chain rule for composite functions. Additionally, the third inequality is derived based on the fundamental definition of the norm, and the final inequality is a direct consequence of Assumption 4.3.1.

Since $\varphi(DF(x_k)[d_k]) < 0$ and $\|d_k\| \neq 0$, we obtain

$$\frac{\varphi^2(DF(x_k)[d_k])}{\|d_k\|_{x_k}^2} \leq Lt_k \frac{\varphi(DF(x_k)[d_k])}{c_2 - 1}.$$

By the Wolfe conditions, we have that $\{F(x_k)\}_{k \geq 0}$ is monotonically decreasing and that

$$F(x_{k+1}) - F(x_0) \preceq_K c_1 \sum_{j=0}^k t_j \varphi(DF(x_j) d_j) e$$

for all $k \geq 0$. Then, by the assumption, since $\{F(x_k)\}_{k \geq 0} \subset \mathcal{L}$, there exists $\mathcal{F} \in \mathbb{R}^m$ such that

$$\mathcal{F} - F(x_0) \preceq_K c_1 \sum_{j=0}^k t_j \varphi(DF(x_j)d_j) e$$

for all $k \geq 0$. Therefore, for all $k \geq 0$ and $w \in C$, we obtain

$$\langle \mathcal{F} - F(x_0), w \rangle \leq c_1 \langle e, w \rangle \sum_{j=0}^k t_j \varphi(DF(x_j)d_j).$$

This together with $c_2 \in (0, 1)$ yields

$$\frac{1}{c_2 - 1} \min \{ \langle \mathcal{F} - F(x_0), \bar{w} \rangle \mid \bar{w} \in C \} \geq c_1 \langle e, w \rangle \sum_{j=0}^k t_j \frac{\varphi(DF(x_j)d_j)}{c_2 - 1} > 0.$$

Therefore, we conclude that

$$\sum_{k \geq 0} t_k \frac{\varphi(DF(x_k)[d_k])}{c_2 - 1} < \infty,$$

which applies

$$\sum_{k \geq 0} \frac{\varphi^2(DF(x_k)[d_k])}{\|d_k\|_{x_k}^2} < \infty.$$

■

In the following, we prove that there exist intervals of step sizes satisfying both the weak and the strong Wolfe conditions.

Proposition 4.3.2. *Let F be of class C^1 defined on a smooth Riemannian manifold. Assume that d is a K -descent direction for F at x and F is lower bounded. Then, there exists $t > 0$ that satisfies the Wolfe and strong Wolfe conditions.*

Proof. For given $w \in C$, define $\phi_w, l_w : \mathbb{R} \rightarrow \mathbb{R}$ by

$$\phi_w(t) = F(R_x(td))^\top w$$

and

$$l_w(t) = F(x)^\top w + c_1 t \varphi(DF(x)[d]).$$

Observe that $\phi_w(0) = l_w(0)$. Because d is a K -descent direction, we have $\phi_w(0) \leq \varphi(DF(x)[d]) < c_1 \varphi(DF(x)[d]) < 0$. So there exists $t > 0$ such that $F(R_x(td))^\top w = F(x)^\top w + c_1 t \varphi(DF(x)[d])$. Let $\hat{t}$ be the smallest number among such t . Because $e^\top w \leq 1$, we have for any $t \in (0, \hat{t})$,

$$F(R_x(td))^\top w \leq F(x)^\top w + c_1 t \varphi(DF(x)[d])$$

$$\leq F(x)^\top w + c_1 t \varphi(DF(x)[d]) e^\top w.$$

Therefore condition (4.2.3) holds.

Since $\phi_w(0) = l_w(0)$ and $\phi_w(\hat{t}) = l_w(\hat{t})$, there must exist $\tilde{t} \in (0, \hat{t})$ such that

$$D\phi_w(\tilde{t}) = c_1 \varphi(DF(x)[d]).$$

Then, we have

$$\varphi(DF(R_x(\tilde{t}d))[DR_x(\tilde{t}d)[d]]) \geq c_1 \varphi(DF(x)[d]).$$

From (4.3.1), we know that $\varphi(DF(R_x(td))[DR_x(td)[d]])$ can be seen as a continuous function of t , and by the intermediate value theorem, there exists $\bar{t} \in (0, \tilde{t})$, and because $0 < c_1 < c_2 < 1$, we obtain

$$0 > \varphi(DF(R_x(\bar{t}d))[DR_x(\bar{t}d)[d]]) = c_1 \varphi(DF(x_k)[d_k]) \geq c_2 \varphi(DF(x)[d]).$$

Therefore, conditions (4.2.4) and (4.2.5) hold. ■

Remark 4.3.1. *This proposition just showed that there exists t such that $\varphi(DF(R_x(td))[DR_x(td)[d]]) < 0$, but this does not mean it is true for all t . The exact relationship remains to be discussed.*

Now just set $v_k = v(x_k)$, and we notice that in similar methods it usually imports a sufficient descent condition [100, 108], that is, for all k , there exists a positive α such that

$$\varphi(DF(x_k)[d_k]) \leq \alpha \varphi(DF(x_k)[v_k]). \quad (4.3.2)$$

Remark 4.3.2. (4.3.2) is a normal assumption in many descent methods. This constraint is also used in the proof of general β in vector optimization [100, Thm 4.2(i)], but if we assume this, it is equivalent to assuming that d_k is a K -descent direction, which we want to avoid. So we do not use it in the following part.

From the NVRCG algorithm step 2, we have

$$\begin{aligned} \varphi(DF(x_k)[d_k]) &\leq \varphi(DF(x_k)v_k) + \beta_k \varphi(DF(x_k)\mathcal{T}^{k-1}(d_{k-1})) \\ &= \varphi(DF(x_k)v_k) + \beta_k \varphi(DF(R(x_{k-1}))\mathcal{T}^{k-1}(d_{k-1})), \end{aligned}$$

namely,

$$\psi_{x_k, d_k}(0) \leq \psi_{x_k, v_k}(0) + \beta_k \psi_{x_{k-1}, d_{k-1}}(t_{k-1}d_{k-1}). \quad (4.3.3)$$

From this inequality, we can easily get the following proposition.

Proposition 4.3.3. *Assume that β_k is defined as follows. Then d_k is a K -descent direction.*

$$\beta_k \in \begin{cases} [0, \infty) & \text{if } \psi_{x_{k-1}, d_{k-1}}(t_{k-1}d_{k-1}) \leq 0, \\ [0, -\psi_{x_k, v_k}(0)/\psi_{x_{k-1}, d_{k-1}}(t_{k-1}d_{k-1}) & \text{if } \psi_{x_{k-1}, d_{k-1}}(t_{k-1}d_{k-1}) > 0. \end{cases}$$

Theorem 4.3.1. [19, Theorem 5.1] *Each accumulation point of the sequence $\{x_k\}$ is a Pareto stationary point.*

Proof. The $v(x)$ we defined is a steepest descent direction, and the proof of this theorem is similar to that of Theorem 5.1 in [19]. ■

4.3.1 NVRCG method with FR parameter

Now we consider the vector extension of the FR parameter on Riemannian manifolds:

$$\beta_k^{\text{FR}} = \frac{\psi_{x_k, v_k}(0)}{\psi_{x_{k-1}, v_{k-1}}(0)}.$$

Under some assumptions, the global convergence can be proved if the parameter β is bounded in magnitude by β_k^{FR} .

Proposition 4.3.4. *Assuming $\beta_k^{\text{FR}} \geq \beta_k \geq 0$ and t_k satisfies the Strong Wolfe conditions with $0 < c_1 < c_2 \leq 1/(1 + m_k)$, there exists $m_k > 0$ at each iteration. If $v(x_k) \neq 0$ for each $k \geq 0$, d_k is a K -descent direction and the following inequality holds:*

$$1 - \frac{c_2}{1 - c_2 m_k} \leq \frac{\psi_{x_k, d_k}(0)}{\psi_{x_k, v_k}(0)} \leq \frac{1}{1 - c_2 m_k}. \quad (4.3.4)$$

Proof. According to the definition of $\psi_{x_k, v_k}(0)$, we have $\psi_{x_k, v_k}(0) < 0$ for all k , and $\beta_k^{\text{FR}} > 0$.

Now we prove the statement by induction.

When $k = 0$, we have $\psi_{x_0, d_0}(0) = \psi_{x_0, v_0}(0) < 0$, then the inequality holds.

Assume the inequality still holds for arbitrary $k - 1$, namely,

$$1 - \frac{c_2}{1 - c_2 m_{k-1}} \leq \frac{\psi_{x_{k-1}, d_{k-1}}(0)}{\psi_{x_{k-1}, v_{k-1}}(0)} \leq \frac{1}{1 - c_2 m_{k-1}}.$$

Recalling (4.3.3) and considering the definition of ψ and ψ is bounded, there must exists integer $m > 0$ satisfying the inequality:

$$\psi_{x_k, d_k}(0) \geq \psi_{x_k, v_k}(0) - m\beta_k |\psi_{x_{k-1}, d_{k-1}}(t_{k-1} d_{k-1})|. \quad (4.3.5)$$

Let m be the smallest one among such integers and $m_k = \max\{m, m_{k-1}\}$. Dividing by $\psi_{x_k, v_k}(0)$, which is less than 0, on both sides and combining with the definition of β^{FR} and strong Wolfe condition (4.2.5), we have

$$\begin{aligned} \frac{\psi_{x_k, d_k}(0)}{\psi_{x_k, v_k}(0)} &\leq \frac{\psi_{x_k, v_k}(0) - m\beta_k |\psi_{x_{k-1}, d_{k-1}}(t_{k-1} d_{k-1})|}{\psi_{x_k, v_k}(0)} \\ &= 1 - m \frac{|\beta_k \psi_{x_{k-1}, d_{k-1}}(t_{k-1} d_{k-1})|}{\psi_{x_k, v_k}(0)} \end{aligned}$$

$$\begin{aligned}
 &\leq 1 + mc_2\beta_k \frac{\psi_{x_{k-1},d_{k-1}}(0)}{\psi_{x_k,v_k}(0)} \\
 &\leq 1 + mc_2 \frac{\psi_{x_{k-1},d_{k-1}}(0)}{\psi_{x_{k-1},v_{k-1}}(0)} \\
 &\leq 1 + mc_2 \frac{1}{1 - c_2m_{k-1}} \\
 &\leq \frac{1}{1 - c_2m_k}.
 \end{aligned}$$

Recalling (4.3.3) again, dividing $\psi_{x_k,v_k}(0)$ on both sides, combining with the definition of β^{FR} and strong Wolfe conditions, we have

$$\begin{aligned}
 \frac{\psi_{x_k,d_k}(0)}{\psi_{x_k,v_k}(0)} &\geq \frac{\psi_{x_k,v_k}(0) + \beta_k \psi_{x_{k-1},d_{k-1}}(t_{k-1}d_{k-1})}{\psi_{x_k,v_k}(0)} \\
 &= 1 + \frac{\beta_k \psi_{x_{k-1},d_{k-1}}(t_{k-1}d_{k-1})}{\psi_{x_k,v_k}(0)} \\
 &\geq 1 - c_2\beta_k \frac{\psi_{x_{k-1},d_{k-1}}(0)}{\psi_{x_k,v_k}(0)} \\
 &\geq 1 - c_2 \frac{\psi_{x_{k-1},d_{k-1}}(0)}{\psi_{x_{k-1},v_{k-1}}(0)} \\
 &\geq 1 - c_2 \frac{1}{1 - c_2m_{k-1}} \\
 &\geq 1 - \frac{c_2}{1 - c_2m_k}.
 \end{aligned}$$

This completes the proof. ■

Theorem 4.3.2. *Consider the NVRCG algorithm. Suppose that Assumptions 4.3.1 and 4.3.2 hold and that*

$$\sum_{k \geq 0} \frac{1}{\|d_k\|_{x_k}^2} = \infty. \quad (4.3.6)$$

If $0 < \beta_k \leq \beta_k^{\text{FR}}$, the assumptions of Proposition 4.3.4 hold and t_k satisfies the strong Wolfe conditions, then $\liminf_{k \rightarrow \infty} \|v(x_k)\| = 0$.

Proof. To complete the proof by contradiction that there exists a positive constant ϵ such that

$$\|v_k\|_{x_k} \geq \epsilon > 0 \quad \text{for all } k \geq 0. \quad (4.3.7)$$

By the definition of d_k and the positiveness of β_k , and recall (4.3.3), we obtain that

$$0 \leq -\psi_{x_k,v_k}(0) \leq -\psi_{x_k,d_k}(0) + \beta_k \psi_{x_{k-1},d_{k-1}}(t_{k-1}d_{k-1}). \quad (4.3.8)$$

Thus, considering that t_k satisfies the strong Wolfe conditions, we have

$$0 \leq -\psi_{x_k,v_k}(0) \leq -\psi_{x_k,d_k}(0) - c_2\beta_k \psi_{x_{k-1},d_{k-1}}(0). \quad (4.3.9)$$

Hence, taking the square on both sides of the last inequality, we conclude that

$$\psi_{x_k, v_k}^2(0) \leq \psi_{x_k, d_k}^2(0) + c_2^2 \beta_k^2 \psi_{x_{k-1}, d_{k-1}}^2(0) + 2c_2 \psi_{x_k, d_k}(0) \beta_k \psi_{x_{k-1}, d_{k-1}}(0). \quad (4.3.10)$$

After some algebra in the right hand side (using $2ab \leq 2a^2 + \frac{1}{2}b^2$ with $a = c_2 \psi_{x_k, d_k}(0)$ and $b = \beta_k \psi_{x_{k-1}, d_{k-1}}(0)$), the last inequality becomes

$$\sigma \psi_{x_k, v_k}^2(0) \leq \psi_{x_k, d_k}^2(0) + \frac{\beta_k^2}{2} \psi_{x_{k-1}, d_{k-1}}^2(0), \quad (4.3.11)$$

where $\sigma := (1 + 2c_2^2)^{-1}$. Then we adjust the terms to obtain

$$\begin{aligned} \psi_{x_k, d_k}^2(0) &\geq \sigma \psi_{x_k, v_k}^2(0) - \frac{1}{2} \beta_k^2 \psi_{x_{k-1}, d_{k-1}}^2(0) \\ &\geq \sigma \psi_{x_k, v_k}^2(0) - \frac{1}{2} (\beta_k^{\text{FR}})^2 \psi_{x_{k-1}, d_{k-1}}^2(0) \\ &\geq \sigma \psi_{x_k, v_k}^2(0) - \frac{1}{2} (\beta_k^{\text{FR}})^2 \left(\sigma \psi_{x_{k-1}, v_{k-1}}^2(0) - \frac{1}{2} (\beta_{k-1}^{\text{FR}})^2 \psi_{x_{k-2}, d_{k-2}}^2(0) \right) \\ &\geq \sigma \left(\psi_{x_k, v_k}^2(0) - \frac{1}{2} (\beta_k^{\text{FR}})^2 \psi_{x_{k-1}, v_{k-1}}^2(0) + \left(-\frac{1}{2}\right)^2 (\beta_k^{\text{FR}})^2 (\beta_{k-1}^{\text{FR}})^2 \psi_{x_{k-2}, v_{k-2}}^2(0) + \dots \right. \\ &\quad \left. + \left(-\frac{1}{2}\right)^{k-1} (\beta_k^{\text{FR}})^2 \dots (\beta_2^{\text{FR}})^2 \psi_{x_1, v_1}^2(0) \right) + \left(-\frac{1}{2}\right)^k (\beta_k^{\text{FR}})^2 \dots (\beta_1^{\text{FR}})^2 \psi_{x_0, v_0}^2(0) \\ &\geq \sigma \left(\psi_{x_k, v_k}^2(0) - \frac{1}{2} \psi_{x_k, v_k}^2(0) + \dots + \left(-\frac{1}{2}\right)^{k-1} \psi_{x_k, v_k}^2(0) \right) + \left(-\frac{1}{2}\right)^k \psi_{x_k, v_k}^2(0) \\ &\geq \sigma \psi_{x_k, v_k}^2(0) \sum_{j=1}^k \left(-\frac{1}{2}\right)^{j-1} + \left(-\frac{1}{2}\right)^k \psi_{x_k, v_k}^2(0) \\ &\geq \frac{1}{4} \sigma \|v_k\|_{x_k}^4 \sum_{j=1}^k \left(-\frac{1}{2}\right)^{j-1} + \left(-\frac{1}{2}\right)^k \frac{1}{4} \|v_k\|_{x_k}^4. \end{aligned}$$

The second inequality comes from $\beta_k \leq \beta_k^{\text{FR}}$. The final inequality is derived as follows: because v_k is a K -descent direction and x_k is not a K -critical point, we have $\varphi(DF(x_k)v_k) + \|v_k\|_{x_k}^2/2 < 0$, then we have $\frac{1}{4} \|v_k\|_{x_k}^4 < \psi_{x_k, v_k}^2(0)$. For sufficiently large values of k , there exists a threshold value k_0 such that for $k > k_0$, $\left| \left(-\frac{1}{2}\right)^k \frac{1}{4} \|v_k\|_{x_k}^4 \right| \leq \frac{1}{8} \sigma \|v_k\|_{x_k}^4 \sum_{j=1}^k \left(-\frac{1}{2}\right)^{j-1}$. Because $1/3 < \sigma < 1$ and $\sum_{j=1}^k \left(-\frac{1}{2}\right)^{j-1} \geq 1/2$, we have

$$\begin{aligned} \psi_{x_k, d_k}^2(0) &\geq \frac{1}{4} \sigma \|v_k\|_{x_k}^4 \sum_{j=1}^k \left(-\frac{1}{2}\right)^{j-1} - \frac{1}{8} \sigma \|v_k\|_{x_k}^4 \sum_{j=1}^k \left(-\frac{1}{2}\right)^{j-1} \\ &= \frac{1}{8} \sigma \|v_k\|_{x_k}^4 \sum_{j=1}^k \left(-\frac{1}{2}\right)^{j-1} \\ &\geq \frac{1}{16} \sigma \epsilon^4. \end{aligned}$$

Thus, we conclude

$$\frac{\psi_{x_k, d_k}^2(0)}{\|d_k\|_{x_k}^2} \geq \frac{\sigma \epsilon^4}{16 \|d_k\|_{x_k}^2} > 0,$$

which is in contradiction with Zoutendijk's condition. ■

4.3.2 NVRCG method with CD parameter

Now we consider the vector extension of the CD parameter in Riemannian optimization:

$$\beta_k^{\text{CD}} = \frac{\psi_{x_k, v_k}(0)}{\psi_{x_{k-1}, d_{k-1}}(0)}.$$

Proposition 4.3.5. *Assume that $0 \leq \beta_k \leq \beta_k^{\text{CD}}$ and that t_k satisfies the strong Wolfe conditions. If, for each $k \geq 0$, $v(x_k) \neq 0$, then the following inequality holds:*

$$\psi_{x_k, d_k}(0) \leq (1 - c_2)\psi_{x_k, v_k}(0) < 0. \quad (4.3.12)$$

Proof. Recalling the inequality (4.3.3) and dividing by $\psi_{x_k, v_k}(0)$, which is less than 0, on both sides, we have

$$\begin{aligned} \frac{\psi_{x_k, d_k}(0)}{\psi_{x_k, v_k}(0)} &\geq \frac{\psi_{x_k, v_k}(0) + \beta_k \psi_{x_{k-1}, d_{k-1}}(t_{k-1} d_{k-1})}{\psi_{x_k, v_k}(0)} \\ &= 1 + \frac{\beta_k \psi_{x_{k-1}, d_{k-1}}(t_{k-1} d_{k-1})}{\psi_{x_k, v_k}(0)} \\ &\geq 1 - c_2 \beta_k \frac{\psi_{x_{k-1}, d_{k-1}}(0)}{\psi_{x_k, v_k}(0)} \\ &= 1 - c_2 \frac{\beta_k}{\beta_k^{\text{CD}}} \\ &\geq 1 - c_2. \end{aligned}$$

The second inequality comes from the strong Wolfe condition (4.2.5) and the third inequality is using the definition of β_k^{CD} . We also have the inequality that $\psi_{x_k, v_k}(0) < 0$ that comes from the definition, here comes $\psi_{x_k, d_k}(0) \leq (1 - c_2)\psi_{x_k, v_k}(0) < 0$. ■

Theorem 4.3.3. *Consider the NVRCG algorithm. Suppose that Assumptions 4.3.1 and 4.3.2 hold and that*

$$\sum_{k \geq 0} \frac{1}{\|d_k\|_{x_k}^2} = \infty. \quad (4.3.13)$$

If $0 \leq \beta_k \leq \beta_k^{\text{CD}}$ and t_k satisfies the strong Wolfe conditions, then $\liminf_{k \rightarrow \infty} \|v(x_k)\|_{x_k} = 0$ holds.

Proof. Assume by contradiction as in Theorem 4.3.2 that there exists a positive constant ϵ such that

$$\|v_k\|_{x_k} \geq \epsilon > 0 \quad \text{for all } k \geq 0.$$

Because v_k is a K -descent direction and x^k is not a K -critical point, we have $\psi_{x_k, v_k}(0) + \|v_k\|_{x_k}^2 / 2 < 0$, and thus $\frac{1}{4} \|v_k\|_{x_k}^4 < \psi_{x_k, v_k}^2(0)$. Combining this with (4.3.12) we have

$$\begin{aligned} \psi_{x_k, d_k}^2(0) &\geq (1 - c_2)^2 \psi_{x_k, v_k}^2(0) \\ &\geq \frac{1}{4} (1 - c_2)^2 \|v_k\|_{x_k}^4 \\ &\geq \frac{1}{4} (1 - c_2)^2 \epsilon^4. \end{aligned}$$

Then, we have

$$\frac{\psi_{x_k, d_k}^2(0)}{\|d_k\|_{x_k}^2} \geq \frac{1}{4} (1 - c_2)^2 \epsilon^4 \frac{1}{\|d_k\|_{x_k}^2} > 0,$$

leading to a contradiction with Zoutendjik's condition. ■

4.3.3 NVRCG method with DY parameter

Consider the β^{DY} vector parameter on Riemannian manifolds:

$$\beta_k^{\text{DY}} = \frac{-\psi_{x_k, v_k}(0)}{\psi_{x_{k-1}, d_{k-1}}(t_{k-1} d_{k-1}) - \psi_{x_{k-1}, d_{k-1}}(0)}.$$

Using some simple derivations, we have

$$\begin{aligned} \beta_k^{\text{DY}} \psi_{x_{k-1}, d_{k-1}}(0) &= \psi_{x_k, v_k}(0) + \beta_k^{\text{DY}} \psi_{x_{k-1}, d_{k-1}}(t_{k-1}, d_{k-1}) \\ &\geq \varphi(DF(x_k)(v_k + \beta_k^{\text{DY}} \mathcal{T}^{k-1}(d_{k-1}))) \\ &= \varphi(DF(x_k) d_k) \\ &= \psi_{x_k, d_k}(0). \end{aligned}$$

If $\psi_{x_{k-1}, d_{k-1}}(0) < 0$, then it holds that

$$\beta_k^{\text{DY}} \leq \frac{\psi_{x_k, d_k}(0)}{\psi_{x_{k-1}, d_{k-1}}(0)}. \quad (4.3.14)$$

Proposition 4.3.6. Assume $0 \leq \beta_k \leq \beta_k^{\text{DY}}$ and that t_k satisfies the Wolfe conditions. If, for each $k \geq 0$, $v(x_k) \neq 0$, then the algorithm is well defined and the following two inequalities hold:

$$\psi_{x_k, d_k}(0) \leq 0, \quad (4.3.15)$$

$$\psi_{x_k, d_k}(0) \leq \psi_{x_k, d_k}(t_k d_k). \quad (4.3.16)$$

Proof. It can be proved by induction as in [120].

When $k = 0$, we have $\psi_{x_0, d_0}(0) = \psi_{x_0, v_0}(0) \leq 0$ and $\psi_{x_0, d_0}(t_0 d_0) \geq c_2 \psi_{x_0, d_0}(0) \geq \psi_{x_0, d_0}(0)$.

Assume these two inequalities still hold for arbitrary $k - 1$. Then $\beta_k^{\text{DY}} \leq 0$ holds, which means it is well defined. Recalling inequality (4.3.3), combined with the definition of β^{DY} , we can easily deduce

$$\begin{aligned} \psi_{x_k, d_k}(0) &\leq \psi_{x_k, v_k}(0) + \beta_k \psi_{x_{k-1}, d_{k-1}}(t_{k-1} d_{k-1}) \\ &= \psi_{x_k, v_k}(0) + \beta_k [\psi_{x_{k-1}, d_{k-1}}(t_{k-1} d_{k-1}) - \psi_{x_{k-1}, d_{k-1}}(0)] + \beta_k \psi_{x_{k-1}, d_{k-1}}(0) \\ &\leq \psi_{x_k, v_k}(0) + \beta_k^{\text{DY}} [\psi_{x_{k-1}, d_{k-1}}(t_{k-1} d_{k-1}) - \psi_{x_{k-1}, d_{k-1}}(0)] + \beta_k \psi_{x_{k-1}, d_{k-1}}(0) \\ &= \beta_k \psi_{x_{k-1}, d_{k-1}}(0) \\ &< 0. \end{aligned}$$

About the second inequality, considering Wolfe condition (4.2.4), we have

$$\psi_{x_k, d_k}(t_k d_k) \geq c_2 \psi_{x_k, d_k}(0) \geq \psi_{x_k, d_k}(0).$$

■

Theorem 4.3.4. Consider the NVRCG algorithm. Assume that Assumptions 4.3.1 and 4.3.2 hold and that

$$\sum_{k \geq 0} \frac{1}{\|d_k\|_{x_k}^2} = \infty. \quad (4.3.17)$$

If $0 \leq \beta_k \leq \beta_k^{\text{DY}}$ and t_k satisfies the strong Wolfe conditions, then $\liminf_{k \rightarrow \infty} \|v(x_k)\|_{x_k} = 0$ holds.

Proof. Assume by contradiction as discussed in Theorem 4.3.2 that the inequalities (4.3.7) and (4.3.11) still hold. Then we adjust the terms of (4.3.11) and use (4.3.14) to obtain

$$\begin{aligned} \psi_{x_k, d_k}^2(0) &\geq \sigma \psi_{x_k, v_k}^2(0) - \frac{\beta_k^2}{2} \psi_{x_{k-1}, d_{k-1}}^2(0) \\ &\geq \sigma \psi_{x_k, v_k}^2(0) - \frac{1}{2} \frac{\psi_{x_k, d_k}^2(0)}{\psi_{x_{k-1}, d_{k-1}}^2(0)} \psi_{x_{k-1}, d_{k-1}}^2(0) \\ &= \sigma \psi_{x_k, v_k}^2(0) - \frac{1}{2} \psi_{x_k, d_k}^2(0). \end{aligned}$$

Because v_k is a K -descent direction and x^k is not a K -critical point, we have $\varphi(\text{DF}(x_k)v_k) + \|v_k\|_{x_k}^2/2 < 0$, and hence $\frac{1}{4} \|v_k\|_{x_k}^4 < \psi_{x_k, v_k}^2(0)$. Therefore, it holds that

$$\psi_{x_k, d_k}^2(0) \geq \frac{2}{3} \sigma \psi_{x_k, v_k}^2(0) \geq \frac{1}{6} \sigma \|v_k\|_{x_k}^4.$$

Then we have

$$\frac{\psi_{x_k, d_k}^2(0)}{\|d_k\|_{x_k}^2} \geq \frac{\sigma \epsilon^4}{6 \|d_k\|_{x_k}^2} > 0,$$

which leads to a contradiction with Zoutendijk's condition. ■

4.3.4 NVRCG method with PRP, HS, and LS parameter

While the FR, DY, and CD parameters of conjugate gradient methods have been extended, the theoretical properties of the PRP, HS, and LS parameters remain less explored. Even in Euclidean spaces, these three conjugate gradient methods with the (strong) Wolfe conditions do not guarantee convergence [5]. Powell [115] demonstrated that the PRP and HS parameters combined with exact line searches can exhibit cycling behavior without approaching a solution point. As a result, various variants have been proposed and analyzed. Gilbert and Nocedal [71] proved that global convergence under inexact line search can be achieved for $\beta_k = \max\{\beta_k^{\text{PRP}}, 0\}$ and $\beta_k = \max\{\beta_k^{\text{HS}}, 0\}$. Dai et al. [49] showed that the positiveness restriction on β_k cannot be relaxed for the PRP parameter. There is a large variety of hybrid conjugate gradient methods whose purpose is to combine the properties of the standard ones in order to get new ones, rapidly convergent to a solution. The idea is to avoid jamming. For example, the method with FR parameter has strong convergence properties, but it may not perform well in computational experiments. On the other hand, although the method with PRP parameter may not generally converge, they often perform better than FR. Indeed, the method with PRP parameter possesses a built-in restart feature that directly addresses to jamming. When the difference between x_k and x_{k+1} is small enough, the numerator of β^{PRP} tends to zero. In this case, the search direction d_k is essentially the steepest descent direction just after a restart occurred. Therefore, the combination of these methods tries to exploit the attractive features of each one, which is the motivation for the hybrid computational scheme PRP–FR. The PRP-, HS-, and LS-type hybrid schemes of conjugate gradient methods to generate descent directions have been discussed in [5, 121]. Here, we follow the hybrid idea and simply propose the following algorithms.

Recall that the PRP parameter is defined as

$$\beta_k^{\text{PRP}} := \frac{-\psi_{x_k, v_k}(0) + \psi_{x_k, \mathcal{T}^{k-1}(v_k)}(0)}{-\psi_{x_{k-1}, v_{k-1}}(0)}.$$

Theorem 4.3.5. *Under the assumptions in Theorem 4.3.2, we set $\beta_k = \max\{0, \min\{\beta_k^{\text{FR}}, \beta_k^{\text{PRP}}\}\}$. Then $\liminf_{k \rightarrow \infty} \|v(x_k)\|_{x_k} = 0$ holds.*

Proof. Because $0 \leq \beta_k = \max\{0, \min\{\beta_k^{\text{FR}}, \beta_k^{\text{PRP}}\}\} \leq \beta_k^{\text{FR}}$, the result follows directly from Theorem 4.3.2. ■

Recall that the LS parameter is defined as

$$\beta_k^{\text{LS}} := \frac{-\psi_{x_k, v_k}(0) + \psi_{x_{k-1}, \mathcal{T}^{k-1}(v_k)}(0)}{-\psi_{x_{k-1}, d_{k-1}}(0)}.$$

Theorem 4.3.6. *Under the assumptions in Theorem 4.3.3, we set $\beta_k = \max\{0, \min\{\beta_k^{\text{LS}}, \beta_k^{\text{CD}}\}\}$. Then $\liminf_{k \rightarrow \infty} \|v(x_k)\|_{x_k} = 0$ holds.*

Proof. Because $0 \leq \beta_k = \max\{0, \min\{\beta_k^{\text{LS}}, \beta_k^{\text{CD}}\}\} \leq \beta_k^{\text{CD}}$, the result follows directly from Theorem 4.3.3. ■

Recall that the HS parameter is defined as

$$\beta_k^{\text{HS}} := \frac{-\psi_{x_k, v_k}(0) + \psi_{x_{k-1}, \mathcal{T}^{k-1}(v_k)}(0)}{\psi_{x_k, \mathcal{T}^{k-1}(d_{k-1})}(0) - \psi_{x_{k-1}, d_{k-1}}(0)}.$$

Theorem 4.3.7. *Under the assumptions in Theorem 4.3.4, we set $\beta_k = \max\{0, \min\{\beta_k^{\text{HS}}, \beta_k^{\text{DY}}\}\}$. Then $\liminf_{k \rightarrow \infty} \|v(x_k)\|_{x_k} = 0$ holds.*

Proof. Because $0 \leq \beta_k = \max\{0, \min\{\beta_k^{\text{HS}}, \beta_k^{\text{DY}}\}\} \leq \beta_k^{\text{DY}}$, the result follows directly from Theorem 4.3.4. ■

Remark 4.3.3. *In Theorems 4.3.2, 4.3.3, and 4.3.4, rather than assuming $\|T_{R_{x_k}(\lambda d_k)}(d_k)\|_{x_{k+1}} \leq \|d_k\|_{x_k}$ as assumed in [108] and discussed in [120], we assume that*

$$\sum_{k \geq 0} \frac{1}{\|d_k\|_{x_k}^2} = \infty.$$

We believe our assumption is more natural because, generally, $\|d_k\|_{x_k}^2$ is expected to be bounded according to the definitions of d_k and v_k .

4.4 Numerical experiments

In this section, we present the numerical results to illustrate the NVRCG algorithm for the different choices of parameter β on the sphere, which is a typical manifold.

We solve some simple problems with the steepest descent and conjugate gradient methods with FR, CD, and DY parameters. For each method, we applied the standard Wolfe conditions. We implemented the code Python 3.9.12 on an Apple MacBook Pro with Apple M1 Pro chip and 16 GB of RAM and calculated the retraction and vector transport with the ‘Pymanopt’ package.

4.4.1 Examples on Sphere manifold

Now consider problems in sphere manifolds $\mathcal{S}^{n-1} = \{x \in \mathbb{R}^n : \|x\| = 1\}$. The retraction and transport are defined as:

$$R_x(v) = \frac{x + v}{\|x + v\|_x},$$

$$\mathcal{T}_x(u) = \text{Proj}_x(u) = u - \langle x, u \rangle x,$$

where $x \in \mathcal{S}^{n-1}$ and $u, v \in \mathcal{T}_x \mathcal{S}^{n-1} = \{\eta \in \mathbb{R}^n : \langle x, \eta \rangle = 0\}$.

Example 4.4.1.

$$\begin{aligned} \min \quad & F(x) = (f_1(x), f_2(x))^\top, \\ \text{s.t.} \quad & x \in \mathcal{S}^1, \end{aligned} \quad (4.4.1)$$

where $f_1(x) = x^\top A x$, $f_2(x) = c x$ and $\mathcal{S}^1$ is a 1-dimensional sphere. Let $A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $A_2 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$, $A_3 = \begin{bmatrix} 1 & 2 \\ 2 & 2 \end{bmatrix}$, and $c = [1, 1]$.

We choose a random initial point x_0 on each manifold and terminate the algorithms when $\|v(x)\| \leq 10^{-4}$ is attained and run the algorithms 100 times for each method. Here we use three cases of A to show three different situations of convergence points. In case A_1 , convergence to a single optimal point is observed. In the case of A_2 , the Pareto stationary points are also Pareto optimal points. Case A_3 shows a more general result.

Although the results presented in Table 4.1 with A_2 and A_3 do not exhibit significant differences due to the limited number of iterations, we can still see that the conjugate gradient methods have fewer cycles than the steepest descent method. From the results provided in Table 4.1 with A_1 , it is evident that the conjugate gradient method outperforms the steepest descent method by significantly reducing the number of iterations. Particularly noteworthy is the remarkable performance of the algorithm when employing the DY parameter.

Furthermore, it is noteworthy that Figure 5.1(i) exhibits the presence of two distinct Pareto fronts, but clearly, the result of the lower one is superior to the upper one. This phenomenon arises due to the convergence of points that correspond to Pareto stationary points rather than Pareto optimal points. In many multiobjective optimization algorithms in Euclidean space, the prevailing assumption is the convexity of the objective functions, which ensures that the Pareto stable solutions coincide with the Pareto optimal solutions. However, the treatment of scenarios where convexity assumptions are relaxed, and the handling of stable solutions, remains an unresolved topic, particularly in the context of Riemannian optimization.

Table 4.1: Example (4.4.1) with $c_1 = 0.1$ and $c_2 = 0.6$

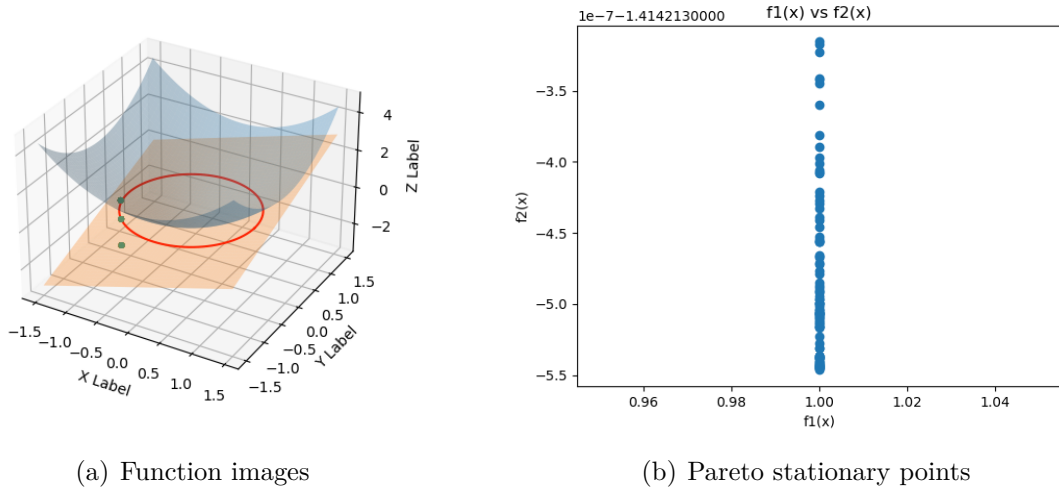
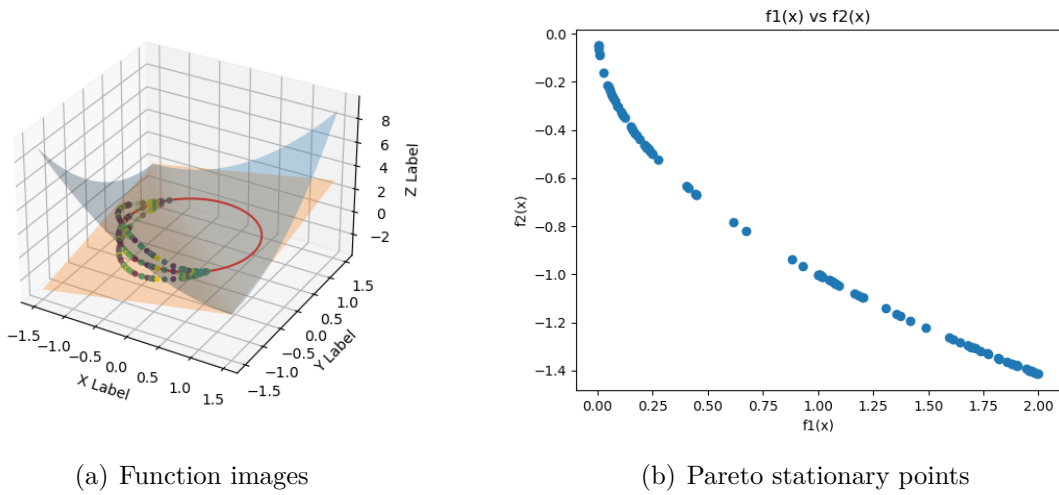
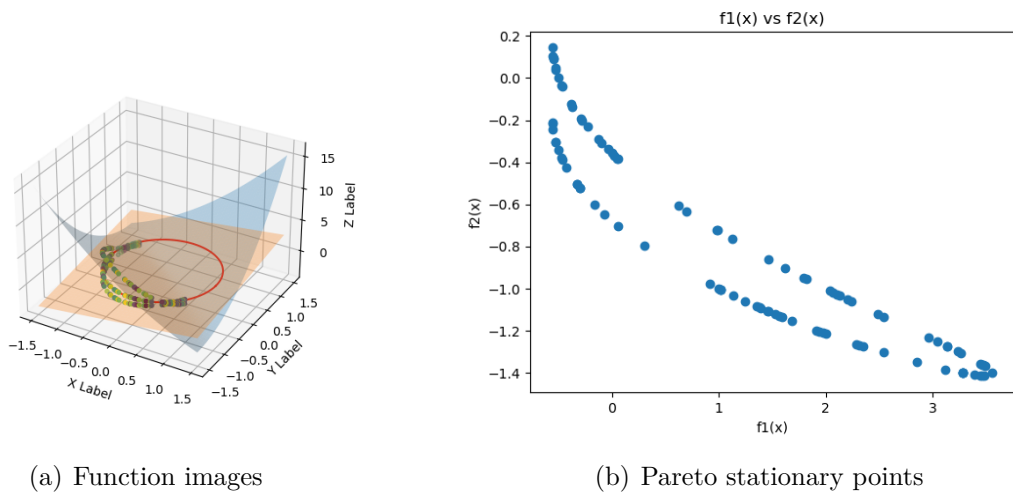
A	n	Times	Average iterations			
			FR	CD	DY	SD
A_1	2	100	12.17	17.73	6.95	29.47
A_2	2	100	1.85	2.21	1.79	3.87
A_3	2	100	2.50	1.60	1.91	1.83

Now consider another problem.

Example 4.4.2.

$$\begin{aligned} \min \quad & F(x) = (f_1(x), f_2(x))^\top, \\ \text{s.t.} \quad & x \in \mathcal{S}^{n-1}, \end{aligned} \quad (4.4.2)$$

where $f_i(x) = x^\top A_i x$ for all i and $\mathcal{S}^{n-1}$ is a sphere manifold.

Figure 4.1: Example (4.4.1) with A_1 and c .Figure 4.2: Example (4.4.1) with A_2 and c .Figure 4.3: Example (4.4.1) with A_3 and c .

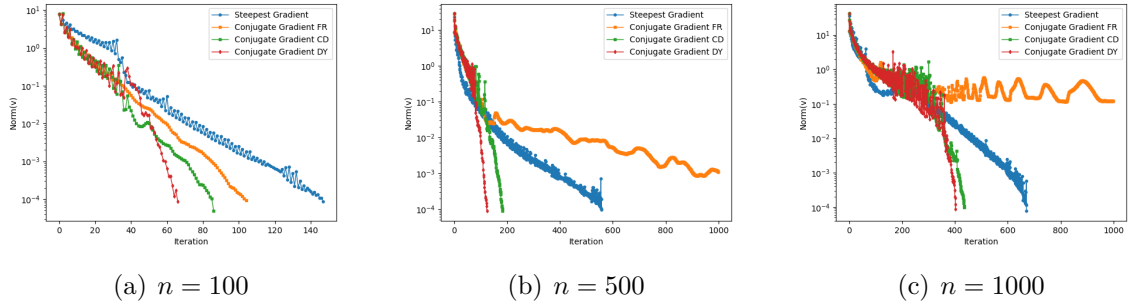


Figure 4.4: Example (4.4.2) with random A_1 and $A_2 = A_1 + E$, where E is All-ones matrix.

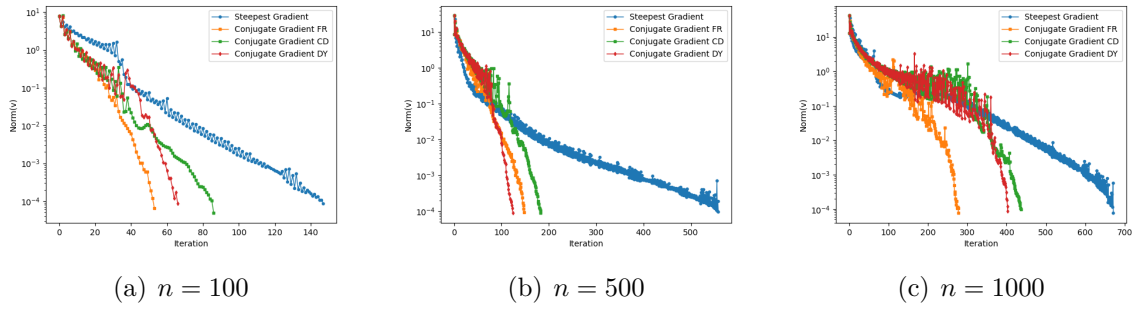


Figure 4.5: Example (4.4.2) with random A_1 and $A_2 = A_1 + E$, where E is All-ones matrix, and $0.9\beta^{\text{FR}}$

Considering the outcomes presented in Example (4.4.2) with $c_1 = 10^{-3}$, $c_2 = 0.6$, as illustrated in Figures 4.4 and 4.5, it is evident that NVRCG (Nonlinear Variable Reduction Conjugate Gradient) methods outperform the steepest descent method, particularly as the value of n increases. It is noteworthy that in Figure 4.4, the performance of the NVRCG method using the FR parameter is not optimal, possibly due to the somewhat stringent conditions of Proposition 4.3.4. To mitigate this, we adjust the parameter to $\beta = 0.9 \times \beta^{\text{FR}}$, which yields improved results over the steepest descent method as in [108].

4.4.2 Examples on Stiefel manifold

Now consider the problem in the Stiefel manifold $st(n, p) := \{X \in \mathbb{R}^{n \times p} : X^\top X = I_p\}$. The Retraction and transport are defined as:

$$R_X(V) = \text{qr}(X + V),$$

$$\mathcal{T}_X(U) = \text{Proj}_X(U) = U - X \frac{X^\top U + U^\top X}{2} = (I - XX^\top)U + X \frac{X^\top U - U^\top X}{2},$$

where $X \in st(n, p)$ and $U, V \in \mathcal{T}_X st(n, p) = \{V \in \mathbb{R}^{n \times p} : X^\top V + V^\top X = 0\}$.

We do the experiments with $c_1 = 10^{-4}$, $c_2 = 0.1$, and choose a random initial point x_0 on each manifold to see the performance of each method.

We examine the two-objective Examples 4.4.3 and 4.4.4.

Example 4.4.3.

$$\begin{aligned} \min \quad & F(x) = (\|A_1 X - X B_1\|_F^2, \|A_2 X - X B_2\|_F^2)^\top, \\ \text{s.t.} \quad & x \in st(n, p) := \{X \in \mathbb{R}^{n \times p} : X^\top X = I_p\} \end{aligned} \quad (4.4.3)$$

where A_i is $n \times n$ and B_i is a $p \times p$ constant matrix, $i = 1, 2$, and $st(n, p)$ is the Stiefel manifold.

Example 4.4.4.

$$\begin{aligned} \min \quad & F(x) = (\|N_1 - X X^\top N_1\|_F^2, \|N_2 - X X^\top N_2\|_F^2)^\top, \\ \text{s.t.} \quad & x \in st(n, p) := \{X \in \mathbb{R}^{n \times p} : X^\top X = I_p\} \end{aligned} \quad (4.4.4)$$

where N_i is an $n \times n$ constant matrix, $i = 1, 2$, and $st(n, p)$ is the Stiefel manifold.

For the convenience of numerical experiments, we conducted two experiments for Examples 4.4.3:

- (a) Randomly generate different symmetric matrices A_i and B_i , $i=1,2$.
- (b) Each time, randomly generate a different symmetric matrix A_1, B_1 and set $A_2 = A_1 + e, B_2 = B_1 + e$, where e is the identity matrix.

For Examples 4.4.3, we also conducted two experiments:

- (a) Randomly generate different symmetric matrices N_1 and N_2 .
- (b) Randomly generate N_1 and set $N_2 = N_1 + e$, where e is the identity matrix.

For each experiment in Examples 4.4.3 and 4.4.4, we tested with $\beta = 1$ and $\beta = 0.9$.

Observations reveal that the NVRCG method outperforms the Steepest Descent method, particularly in scenarios involving higher dimensions. This performance superiority becomes more pronounced as the dimensional complexity increases. Furthermore, it was observed that in some instances, setting the value of β to 0.9β appears to enhance the convergence speed. This insight presents an intriguing aspect for future analyses focused on optimizing convergence rates. In addition, with the increase of dimension and iteration, the step size found may become too small, thus affecting the efficiency of the algorithm. The specific relationship between it and the convergence speed is also worth considering.

4.5 Conclusion

In this chapter, we proposed the nonlinear conjugate gradient methods for vector optimization on Riemannian manifolds with retraction and vector transport. We gave the extension of FR, CD, DY, PRP, LS, and HS parameters. We proved the convergence with the vector extensions of the FR, CD, and DY parameters, then simply gave results of PRP, HS, and LS parameters. Under some assumptions, the sequence obtained by the algorithm can converge to a Pareto stationary point. Numerical experiments illustrating the practical behavior of the methods were presented.

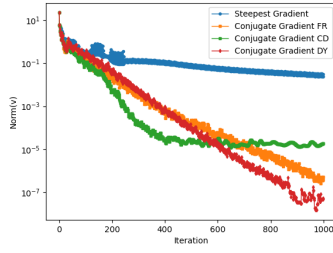
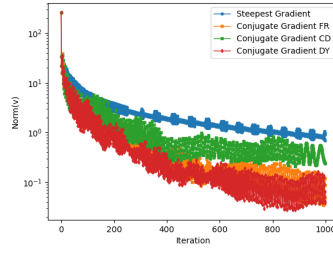
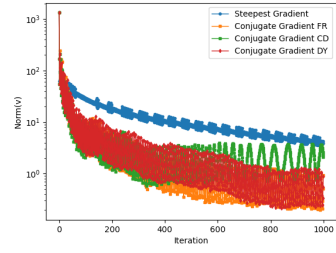
(a) $n = 10, p = 4$ (b) $n = 50, p = 5$ (c) $n = 100, p = 10$

Figure 4.6: Example (4.4.3) with random A_1, B_2 , and $A_2 = A_1 + e, B_2 = B_1 + e$, where e is identity matrix

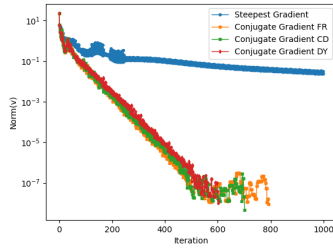
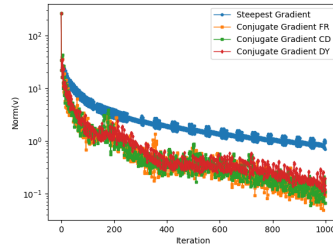
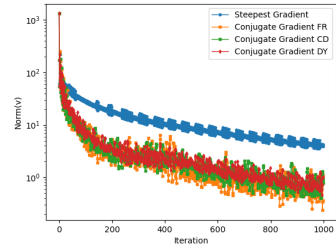
(a) $n = 10, p = 4$ (b) $n = 50, p = 5$ (c) $n = 100, p = 10$

Figure 4.7: Example (4.4.3) with random $A_1, B_2, A_2 = A_1 + e, B_2 = B_1 + e$, where e is identity matrix, and 0.9β .

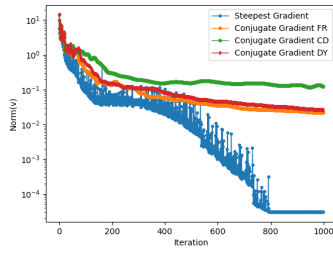
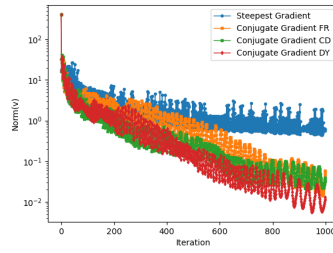
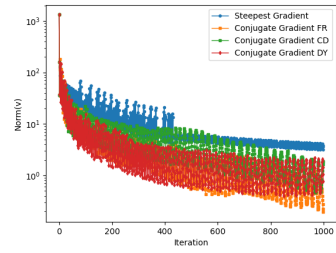
(a) $n = 10, p = 4$ (b) $n = 50, p = 5$ (c) $n = 100, p = 10$

Figure 4.8: Example (4.4.3) with random A and B

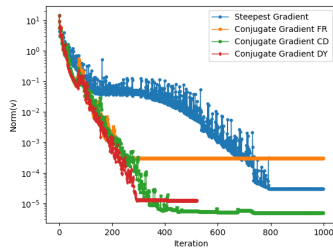
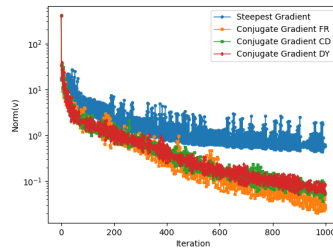
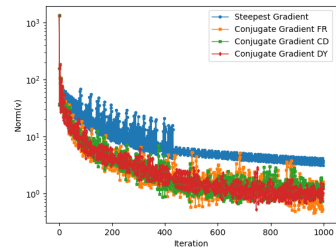
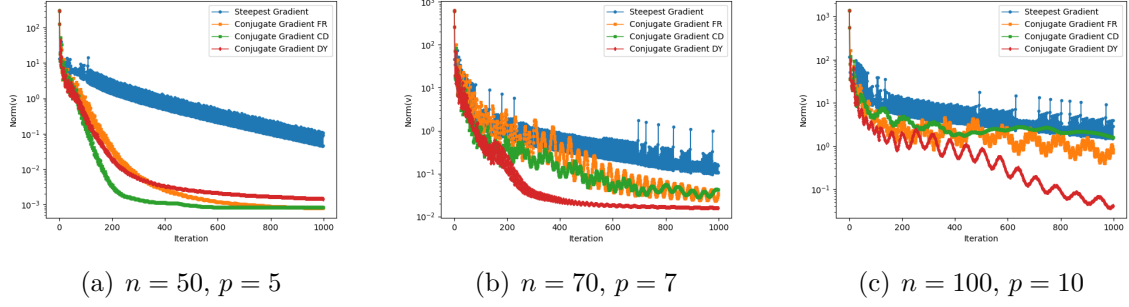
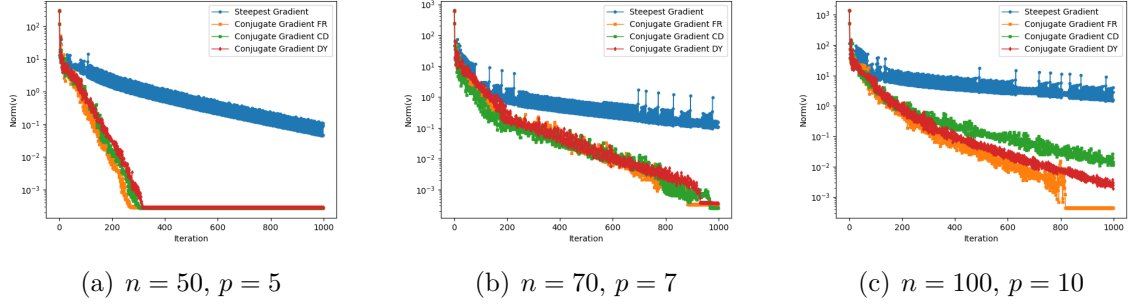
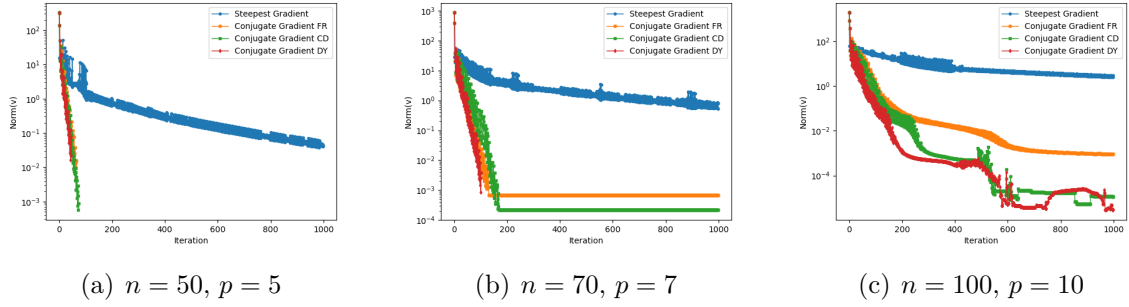
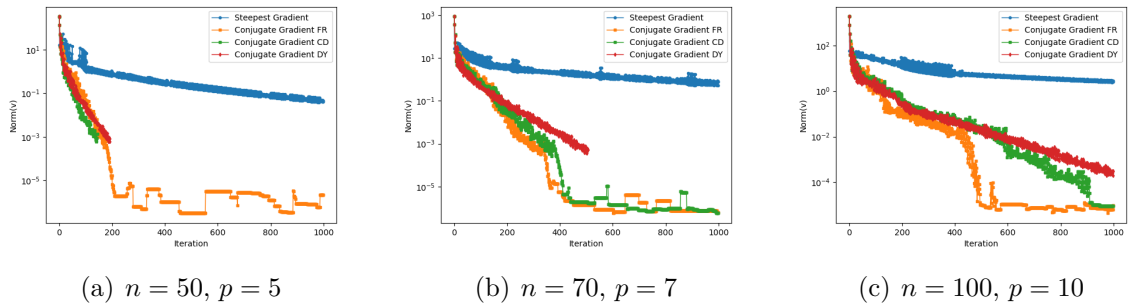
(a) $n = 10, p = 4$ (b) $n = 50, p = 5$ (c) $n = 100, p = 10$

Figure 4.9: Example (4.4.3) with random A, B and 0.9β .

Figure 4.10: Example (4.4.4) with random N_1 and N_2 .Figure 4.11: Example (4.4.4) with random N_1 , N_2 and 0.9β .Figure 4.12: Example (4.4.4) with random N_1 and $N_2 = N_1 + e$, where e is the identity matrix.Figure 4.13: Example (4.4.4) with random N_1 , $N_2 = N_1 + e$ and 0.9β , where e is the identity matrix.

Chapter 5

Riemannian conditional gradient methods for composite optimization problems

5.1 Introduction

As discussed in Chapter 1.3.3, the Conditional Gradient (CondG) method, or Frank-Wolfe algorithm, offers a projection-free alternative for constrained optimization, making it highly effective for high-dimensional problems. While its Euclidean variants [61, 89], Riemannian variants [137], and recent multi-objective extensions are well-established [8, 44], the extension of the Generalized Conditional Gradient (GCondG) method to Riemannian manifolds remains an active frontier. This is particularly relevant for composite problems of the form $F(x) := f(x) + g(x)$, where f is smooth and g is non-smooth, a structure ubiquitous in modern machine learning but challenging in non-Euclidean settings. Although Riemannian proximal gradient methods exist [43, 86], a fully developed Riemannian CondG framework for composite optimization is still lacking.

In this chapter, we address this gap by proposing Riemannian Generalized Conditional Gradient (RGCondG) methods. We systematically investigate three distinct step-size strategies: adaptive, diminishing, and an Armijo-based backtracking line search. A rigorous convergence analysis is provided for each strategy. Specifically, we establish a sublinear convergence rate of $\mathcal{O}(1/k)$ for the adaptive and diminishing step sizes. Furthermore, for the Armijo-based strategy, we derive an iteration complexity of $\mathcal{O}(1/\epsilon^2)$ to achieve an ϵ -approximate stationary point, which corresponds to a convergence rate of $\mathcal{O}(1/\sqrt{k})$. We also discuss the practical implementation of the subproblems and demonstrate the method's efficacy through numerical experiments.

The remainder of this chapter is structured as follows. Section 2 briefly reviews essential Riemannian concepts and formalizes the specific notions of convexity and smoothness relative to retractions. Section 5.3 formally introduces the RGCondG method and the three step-size strategies. The convergence analysis is detailed in Section 5.4, followed by a discussion on

subproblem solutions in Section 5.5. Section 5.6 explores accelerated variants, and Section 5.7 presents numerical results. Finally, Section 5.8 concludes the chapter.

5.2 Retraction-based convexity and smoothness

To analyze the convergence of the proposed Generalized Conditional Gradient methods, we require notions of convexity and smoothness that align with the use of general retractions.

We recall the standard concepts of geodesic convexity for sets and functions as detailed in Chapter 2 (Section 2.4.3). While geodesic convexity provides the theoretical foundation for Riemannian optimization, practical algorithms often employ general retractions R that may differ from the exponential map to reduce computational costs. Consequently, we introduce the generalized concept of R -convexity, which ensures that the objective function behaves convexly along the curves generated by the retraction.

Assumption 5.2.1 (R -Convexity). *Let $h : \mathcal{M} \rightarrow \mathbb{R}$ be a smooth function and let R be a retraction on $\mathcal{M}$. We assume that for any $x, y \in \mathcal{M}$ where the inverse retraction $R_x^{-1}(y)$ is well-defined, the function satisfies the following inequality for all $\lambda \in [0, 1]$:*

$$h(R_x(\lambda R_x^{-1}(y))) \leq (1 - \lambda)h(x) + \lambda h(y). \quad (5.2.1)$$

To distinguish it from geodesic convexity, we define a function h as R -convex if it satisfies the conditions outlined in Assumption 5.2.1. Note that the exponential map satisfies this assumption when h is geodesically convex. And not all retractions possess the R -convexity property, as it often depends on the specific function. For example, under the Euclidean retraction on the SPD manifold, the function $h(P) = -\log \det(P)$ is globally R -convex, whereas further examples of geodesically convex functions on the SPD manifold endowed with non-Euclidean metrics can be found in [54]. However, since a second-order retraction [29, Definition 5.42] approximates the geodesic more closely, a well-behaved function (e.g., a geodesically convex function) is more likely to satisfy R -convexity with respect to such a retraction, particularly in a local neighborhood. In the following discussion and throughout the development of our algorithms, we restrict our attention to retractions that satisfy this assumption.

Analogously, we define the notion of an R -convex set, which generalizes the feasible set constraint for our subproblems.

Definition 5.2.1 (R -Convex Set). *Let $\mathcal{M}$ be a Riemannian manifold equipped with a retraction R . A subset $\mathcal{S} \subseteq \mathcal{M}$ is said to be **R -convex** if for any $x, y \in \mathcal{S}$ such that $R_x^{-1}(y)$ is well-defined, the retraction curve joining them lies entirely within $\mathcal{S}$. Formally:*

$$R_x(tR_x^{-1}(y)) \in \mathcal{S}, \quad \forall t \in [0, 1]. \quad (5.2.2)$$

Finally, we state the definition of smoothness relative to a retraction, often referred to as L -retraction-smoothness [86]. This generalizes the standard Lipschitz gradient condition to the specific retraction used in our algorithm.

Definition 5.2.2 (*L*-Retraction-Smoothness). A function $h: \mathcal{M} \rightarrow \mathbb{R}$ is called *L*-smooth with respect to a retraction R if there exists a constant $L \geq 0$ such that for all $x, y \in \mathcal{M}$:

$$h(y) \leq h(x) + \langle \text{grad } h(x), R_x^{-1}(y) \rangle_x + \frac{L}{2} \|R_x^{-1}(y)\|_x^2. \quad (5.2.3)$$

This inequality serves as a Riemannian descent lemma adapted to the retraction geometry. A strong version of (5.2.3), as given below, is also used in [31, Assumption 2.6].

$$|h(y) - h(x) - \langle \text{grad } h(x), R_x^{-1}(y) \rangle_x| \leq \frac{L}{2} \|R_x^{-1}(y)\|_x^2 \quad \forall x, y \in \mathcal{M}.$$

In addition, if the retraction is chosen to be the exponential map, it follows that h is geodesically *L*-smooth, as defined in [137] and [29, Section 10.4].

5.3 Riemannian generalized conditional gradient method

In this chapter, we consider the following Riemannian optimization problem:

$$\begin{aligned} \min \quad & F(x) := f(x) + g(x) \\ \text{s.t.} \quad & x \in \mathcal{X}, \end{aligned} \quad (5.3.1)$$

where $\mathcal{X} \subseteq \mathcal{M}$ is a compact *R*-convex set and $F: \mathcal{M} \rightarrow \mathbb{R}$ is a composite function with $f: \mathcal{M} \rightarrow \mathbb{R}$ being continuously differentiable and $g: \mathcal{M} \rightarrow \mathbb{R}$ being a closed, *R*-convex and lower semicontinuous (possibly nonsmooth) function. Here, we can also consider F as an extended-valued function. All the analyses would still apply, but we omit the infinity for simplicity. Also, we make the following assumption.

Assumption 5.3.1. The set $\mathcal{X} \subseteq \mathcal{M}$ is an *R*-convex and complete set where the retraction R and its inverse R^{-1} are well-defined and smooth over $\mathcal{X}$. Additionally, the inverse retraction $R_x^{-1}(y)$ is continuous with respect to both x and y .

As discussed in [29, Section 10.2], for general retractions, over some domain of $\mathcal{M} \times \mathcal{M}$ which contains all pairs (x, x) , the map $(x, y) \mapsto R_x^{-1}(y)$ can be defined smoothly jointly in x and y . In this sense, we can say that the above assumption is reasonable.

Now, let us recall the CondG method in the Euclidean space. Consider minimizing a convex function $\tilde{f}: \mathbb{R}^n \rightarrow \mathbb{R}$ over a convex compact set $\mathcal{X} \subseteq \mathbb{R}^n$. At each iteration k , the method solves a linear approximation of the original problem by finding a direction s^k that minimizes the linearized objective function over the constraint set $\mathcal{X}$, i.e., $s^k = \arg \min_{s \in \mathcal{X}} \langle \nabla \tilde{f}(x^k), s \rangle$. Then, it updates the current point x^k using a step size λ_k along the direction $s^k - x^k$, i.e., $x^{k+1} = x^k + \lambda_k(s^k - x^k)$. For composite functions of the form $\tilde{f}(x) = \tilde{h}(x) + \tilde{g}(x)$, where $\tilde{h}$ is smooth and $\tilde{g}$ is convex, the CondG method can be adapted to handle the non-smooth term $\tilde{g}$. The update step becomes:

$$x^{k+1} = \arg \min_{s \in \mathcal{X}} \langle \nabla \tilde{h}(x^k), s \rangle + \tilde{g}(s). \quad (5.3.2)$$

In this case, $s \mapsto \langle \nabla \tilde{h}(x^k), s \rangle + \tilde{g}(s)$ is the objective function for the subproblem, which remains convex since it is the sum of the convex function $\tilde{g}$ and the linear term $\langle \nabla \tilde{h}(x^k), \cdot \rangle$. Moreover, the boundedness of $\mathcal{X}$ guarantees the existence of a solution to this subproblem. The step size λ_k can be chosen using various strategies. The CondG method is particularly useful when the constraint set $\mathcal{X}$ is not an easy set, making projection-based methods computationally expensive.

The exponential map, the parallel transport, and specific step size are used in the Riemannian CondG method proposed in [137]. Now we extend this approach to a more general framework for composite optimization problems, without specifying the retraction and the vector transport.

Algorithm 7: Riemannian generalized conditional gradient method (RGCondG)

Step 0. Initialization:

Choose an initial point $x^0 \in \mathcal{X}$ and set $k = 0$.

Step 1. Compute the search direction:

Compute an optimal solution $p(x^k)$ and the optimal value $\theta(x^k)$ as

$$p(x^k) = \arg \min_{u \in \mathcal{X}} \langle \text{grad} f(x^k), R_{x^k}^{-1}(u) \rangle_{x^k} + g(u) - g(x^k), \quad (5.3.3)$$

$$\theta(x^k) = \langle \text{grad} f(x^k), R_{x^k}^{-1}(p(x^k)) \rangle_{x^k} + g(p(x^k)) - g(x^k). \quad (5.3.4)$$

Define the search direction by $d(x^k) = R_{x^k}^{-1}(p(x^k))$.

Step 2. Compute the step size:

Compute the step size λ_k .

Step 3. Update the iterates:

Update the current iterate $x^{k+1} = R_{x^k}(\lambda_k d(x^k))$.

Step 4. Convergence check:

If a convergence criteria is met, stop; otherwise, set $k = k + 1$ and return to Step 1.

Our method is presented in Algorithm 7. In Step 1, we need to solve the subproblem (5.3.3), which, similarly to the Euclidean case, is related to the linear approximation of the objective functions. In Step 2, we compute the step size based on some strategies that we will mention later. In Step 3, we update the iterate by using some retraction. Note that if $u = x^k$ in (5.3.3), then $\langle \text{grad} f(x^k), R_{x^k}^{-1}(u) \rangle_{x^k} + g(u) - g(x^k) = 0$, which implies that $\theta(x^k) \leq 0$. And since $d(x^k) = R_{x^k}^{-1}(p(x^k))$, the subproblem is equivalent to

$$d(x^k) = \arg \min_{d \in T_{x^k} \mathcal{X}} \langle \text{grad} f(x^k), d \rangle_{x^k} + g(R_{x^k}(d)) - g(x^k). \quad (5.3.5)$$

We now consider three distinct, rigorously defined strategies, which are also discussed in [7] and elaborated below.

Armijo step size: Let $\zeta \in (0, 1)$ and $0 < \omega_1 < \omega_2 < 1$. The step size λ_k is determined using the following algorithm:

Step 0. Set $\lambda_{k_0} = 1$ and initialize $\ell \leftarrow 0$.

Step 1. If

$$F(R_{x^k}(\lambda_k d(x^k))) \leq F(x^k) + \zeta \lambda_{k_\ell} \theta(x^k),$$

then set $\lambda_k := \lambda_{k_\ell}$ and return to the main algorithm.

Step 2. Find $\lambda_{k_{\ell+1}} \in [\omega_1 \lambda_{k_\ell}, \omega_2 \lambda_{k_\ell}]$, update $\ell \leftarrow \ell + 1$, and return to Step 1.

Adaptive step size: For all k , set

$$\lambda_k := \min \left\{ 1, \frac{-\theta(x^k)}{L \text{dist}^2(p(x^k), x^k)} \right\} = \operatorname{argmin}_{\lambda \in (0, 1]} \left\{ \lambda \theta(x^k) + \frac{L}{2} \lambda^2 \text{dist}^2(p(x^k), x^k) \right\}.$$

Diminishing step size: For all k , set

$$\lambda_k := \frac{2}{k+2}.$$

5.4 Convergence analysis

In this section, we will provide the convergence analysis for three types of step sizes. The following assumptions will be used in the convergence results.

Assumption 5.4.1. *The function f is L -smooth, namely, there exists a constant L such that (5.2.3) holds.*

Since $\mathcal{X}$ is a compact set, we can, without loss of generality, assume that $\mathcal{X} = \text{dom}(g)$ is a compact set, its diameter is finite and can be defined by

$$\text{diam}(\mathcal{X}) := \sup_{x, y \in \mathcal{X}} \text{dist}(x, y), \quad (5.4.1)$$

where $\text{dist}(x, y)$ is defined in Section 2.

Assumption 5.4.2. *For $\{x^k\}$ generated by Algorithm 7, there exists $\Omega > 0$ such that $\Omega \geq \max \left\{ \max \left\{ \|R_{x^i}^{-1}(x^j)\|_{x^i} : x^i, x^j \in \{x^k\} \right\}, \text{diam}(\mathcal{X}) \right\}$.*

As k tends to infinity, we can assume without loss of generality that x^k lies within a compact neighborhood of x^* . Due to the smoothness of the inverse retraction, this ensures that $\|R_{x^i}^{-1}(x^j)\|_{x^i}$ remains bounded. Hence, Assumption 5.4.2 is reasonable.

Lemma 5.4.1. *Let $\{x^k\}$ be generated by Algorithm 7. Define $z^k = p(x^k)$ and recall that $x^{k+1} = R_{x^k}(\lambda R_{x^k}^{-1}(z^k))$. Then, we have*

$$F(x^{k+1}) \leq F(x^k) + \lambda \theta(x^k) + \frac{L \lambda^2}{2} \Omega^2. \quad (5.4.2)$$

Proof.

From the definition of F , we have

$$\begin{aligned}
F(x^{k+1}) &= f(x^{k+1}) + g(x^{k+1}) \\
&\leq f(x^k) + \langle \text{grad} f(x^k), R_{x^k}^{-1}(x^{k+1}) \rangle_{x^k} + \frac{L}{2} \|R_{x^k}^{-1}(x^{k+1})\|_{x^k}^2 + g(x^{k+1}) \\
&\leq f(x^k) + \langle \text{grad} f(x^k), R_{x^k}^{-1}(x^{k+1}) \rangle_{x^k} + \frac{L}{2} \|R_{x^k}^{-1}(x^{k+1})\|_{x^k}^2 + (1-\lambda)g(x^k) + \lambda g(z^k) \\
&= F(x^k) + \lambda \langle \text{grad} f(x^k), R_{x^k}^{-1}(z^k) \rangle_{x^k} - \lambda g(x^k) + \lambda g(z^k) + \frac{L}{2} \|R_{x^k}^{-1}(x^{k+1})\|_{x^k}^2 \\
&= F(x^k) + \lambda (\langle \text{grad} f(x^k), R_{x^k}^{-1}(z^k) \rangle_{x^k} - g(x^k) + g(z^k)) + \frac{L}{2} \|R_{x^k}^{-1}(x^{k+1})\|_{x^k}^2 \\
&= F(x^k) + \lambda \theta(x^k) + \frac{L\lambda^2}{2} \|R_{x^k}^{-1}(z^k)\|_{x^k}^2 \\
&\leq F(x^k) + \lambda \theta(x^k) + \frac{L\lambda^2}{2} \Omega^2,
\end{aligned}$$

where the first inequality comes from the L -smoothness of f and the second inequality is due to the R -convexity of g . Then, from the definition of θ in (5.3.4), and using the fact that $\|R_{x^k}^{-1}(x^{k+1})\|_{x^k}^2 = \lambda^2 \|R_{x^k}^{-1}(z^k)\|_{x^k}^2$ and Assumption 5.4.2, we obtain the desired result. ■

Additionally, we can derive the following proposition.

Proposition 5.4.1. *Let $x^* \in \mathcal{X}$ be an optimal point of (5.3.1), and assume that f is R -convex. Then, Algorithm 7 generates $\{x^k\}$ satisfying*

$$F(x^{k+1}) \leq F(x^*) + \pi_k (1 - \lambda_0) (F(x^0) - F(x^*)) + \sum_{s=0}^k \frac{\pi_k}{\pi_s} \frac{L\lambda_s^2}{2} \Omega^2, \quad (5.4.3)$$

where $\pi_k := \prod_{s=1}^k (1 - \lambda_s)$ with $\pi_0 = 1$.

Proof. From Lemma 5.4.1, we obtain

$$F(x^{k+1}) \leq F(x^k) + \lambda_k \theta(x^k) + \frac{L\lambda_k^2}{2} \Omega^2.$$

Define $\Delta_k := F(x^k) - F(x^*)$. Then, we get

$$\Delta_{k+1} \leq \Delta_k + \lambda_k \theta(x^k) + \frac{L\lambda_k^2}{2} \Omega^2.$$

From the convexity of f , we obtain

$$\begin{aligned}
F(x^*) - F(x^k) &\geq \langle \text{grad} f(x^k), R_{x^k}^{-1}(x^*) \rangle_{x^k} + g(x^*) - g(x^k) \\
&\geq \min_{u \in \mathcal{X}} \langle \text{grad} f(x^k), R_{x^k}^{-1}(u) \rangle_{x^k} + g(u) - g(x^k) = \theta(x^k).
\end{aligned}$$

Then we have $\Delta_k \leq -\theta(x^k)$, which gives

$$\Delta_{k+1} \leq (1 - \lambda_k)\Delta_k + \frac{L\lambda_k^2}{2}\Omega^2. \quad (5.4.4)$$

Now, let us expand Δ_k and gradually express Δ_k in terms of Δ_0 . Ultimately, for the recursive relation at step $k + 1$, we obtain:

$$\Delta_{k+1} \leq \prod_{s=0}^k (1 - \lambda_s)\Delta_0 + \sum_{s=0}^k \frac{L\lambda_s^2}{2} \prod_{j=s+1}^k (1 - \lambda_j)\Omega^2, \quad (5.4.5)$$

that is,

$$\Delta_{k+1} \leq \pi_k (1 - \lambda_0) \Delta_0 + \sum_{s=0}^k \frac{\pi_k}{\pi_s} \frac{L\lambda_s^2}{2} \Omega^2.$$

Rearranging the terms completes the proof. $\blacksquare$

Now we present the following lemma, which will be instrumental in our analysis of convergence and iteration complexity bounds for the CondG method.

Lemma 5.4.2. *[13, Lemma 13.13] Let $\{a_k\}$ and $\{b_k\}$ be nonnegative sequences of real numbers satisfying*

$$a_{k+1} \leq a_k - b_k\beta_k + \frac{A}{2}\beta_k^2, \quad k = 0, 1, 2, \dots,$$

where $\beta_k := 2/(k + 2)$ and A is a positive number. Suppose that $a_k \leq b_k$ for all k . Then

$$(1) \ a_k \leq \frac{2A}{k} \text{ for all } k = 1, 2, \dots$$

$$(2) \ \min_{\ell \in \{\lfloor \frac{k}{2} \rfloor + 2, \dots, k\}} b_\ell \leq \frac{8A}{k-2} \text{ for all } k = 3, 4, \dots, \quad \text{where } \lfloor k/2 \rfloor = \max\{n \in \mathbb{N} : n \leq k/2\}.$$

5.4.1 Analysis with the adaptive and the diminishing step sizes

Building upon the theoretical results established previously, we now present the main theorem regarding the convergence rate for two types of step sizes: the adaptive and the diminishing step sizes.

Theorem 5.4.1. *Let x^* be an optimal point of (5.3.1), and assume that f is R -convex. Then, the sequence of iterates $\{x^k\}$ generated by Algorithm 7 with the adaptive or diminishing step size satisfies $F(x^k) - F(x^*) = O(1/k)$. Specifically, we obtain*

$$\min_{\ell \in \{\lfloor \frac{k}{2} \rfloor + 2, \dots, k\}} -\theta(x^\ell) \leq \frac{8L\Omega^2}{k-2}.$$

Proof. From Lemma 5.4.1 and the diminishing step size $\lambda_k = \frac{2}{k+2} \in (0, 1)$, we have

$$F(x^{k+1}) - F(x^*) \leq F(x^k) - F(x^*) + \lambda_k \theta(x^k) + \frac{L\lambda_k^2}{2} \Omega^2. \quad (5.4.6)$$

Considering the adaptive step size $\eta_k = \operatorname{argmin}_{\lambda \in (0, 1]} \{ \lambda \theta(x^k) + \frac{L}{2} \lambda^2 d^2(p(x^k), x^k) \}$, we have $\eta_k \theta(x^k) + \frac{L\eta_k^2}{2} d^2(p(x^k), x^k) \leq \lambda_k \theta(x^k) + \frac{L\lambda_k^2}{2} d^2(p(x^k), x^k) \leq \lambda_k \theta(x^k) + \frac{L\lambda_k^2}{2} \operatorname{diam}(\mathcal{X})^2 \leq \lambda_k \theta(x^k) + \frac{L\lambda_k^2}{2} \Omega^2$, so the inequality (5.4.6) still holds.

Define $\Delta_k := F(x^k) - F(x^*)$. Then (5.4.6) is equivalent to

$$\Delta_{k+1} \leq \Delta_k + \lambda_k \theta(x^k) + \frac{L\lambda_k^2}{2} \Omega^2.$$

From the convexity of f , we obtain

$$\begin{aligned} F(x^*) - F(x^k) &\geq \langle \operatorname{grad} f(x^k), R_{x^k}^{-1}(x^*) \rangle_{x^k} + g(x^*) - g(x^k) \\ &\geq \min_{u \in \mathcal{X}} \langle \operatorname{grad} f(x^k), R_{x^k}^{-1}(u) \rangle_{x^k} + g(u) - g(x^k) = \theta(x^k). \end{aligned}$$

Then we have $0 \leq \Delta_k \leq -\theta(x^k)$. So setting $a_k = \Delta_k$ and $b_k = -\theta(x^k) \geq 0$, $\beta_k = \lambda_k$, and $A = L\Omega^2$ in Lemma 5.4.2 we obtain

$$F(x^k) - F(x^*) \leq \frac{2L}{k} \Omega^2,$$

and

$$\min_{\ell \in \{\lfloor \frac{k}{2} \rfloor + 2, \dots, k\}} -\theta(x^\ell) \leq \frac{8L\Omega^2}{k-2},$$

for all $k = 3, 4, \dots$, where $\lfloor k/2 \rfloor = \max\{n \in \mathbb{N} : n \leq k/2\}$. ■

The above theorem also indicates that the iteration complexity of the method is $\mathcal{O}(1/k)$. Note that a similar result can be obtained by following the proof of [139, Corollary 5].

5.4.2 Analysis with the Armijo step size

The following lemma establishes the existence of intervals of step sizes that satisfy the Armijo condition.

Lemma 5.4.3. *Let $\zeta \in (0, 1)$, $x^k \in \mathcal{X}$ and $\theta(x^k) \neq 0$. Then, there exists $0 < \bar{\lambda} \leq 1$ such that*

$$F(R_{x^k}(\lambda R_{x^k}^{-1}(p(x^k)))) \leq F(x^k) - \zeta \lambda |\theta(x^k)| \quad \forall \lambda \in (0, \bar{\lambda}]. \quad (5.4.7)$$

Proof. Since f is differentiable and g is R -convex, the following holds for all $\lambda \in (0, 1)$,

$$\begin{aligned} &F(R_{x^k}(\lambda R_{x^k}^{-1}(p(x^k)))) \\ &= g(R_{x^k}(\lambda R_{x^k}^{-1}(p(x^k)))) + f(R_{x^k}(\lambda R_{x^k}^{-1}(p(x^k)))) \\ &\leq (1 - \lambda)g(x^k) + \lambda g(p(x^k)) + f(x^k) + \lambda \langle \operatorname{grad} f(x^k), R_{x^k}^{-1}(p(x^k)) \rangle_{x^k} + o(\lambda) \\ &= F(x^k) + \lambda (\langle \operatorname{grad} f(x^k), R_{x^k}^{-1}(p(x^k)) \rangle_{x^k} + g(p(x^k)) - g(x^k)) + o(\lambda) \\ &= F(x^k) + \lambda \zeta \theta(x^k) + \lambda \left((1 - \zeta) \theta(x^k) + \frac{o(\lambda)}{\lambda} \right). \end{aligned}$$

The inequality arises from the Taylor expansion of f and the R -convexity of g . Since $\lim_{\lambda \rightarrow 0} \frac{o(\lambda)}{\lambda} = 0$, $\zeta \in (0, 1)$ and $\theta(x^k) < 0$ from the definition as in (5.3.4), there exists $\bar{\lambda} \in (0, 1]$ such that $(1 - \zeta)\theta(x^k) + \frac{o(\lambda)}{\lambda} \leq 0$, then (5.4.7) holds for all $\lambda \in (0, \bar{\lambda}]$. ■

In the subsequent results, we assume that the sequence generated by Algorithm 7 is infinite, which means that $\theta(x^k) < 0$ for all k .

Assumption 5.4.3. Define $\mathcal{L} = \{x \mid F(x) \leq F(x^0)\}$, and assume that all monotonically nonincreasing sequences in $F(\mathcal{L})$ are bounded from below.

Theorem 5.4.2. Assume that F satisfies Assumption 5.4.3 and the sequence $\{x^k\}$ is generated by Algorithm 7 with the Armijo step size. The following statements hold:

- (1) $\lim_{k \rightarrow \infty} \theta(x^k) = 0$;
- (2) Let x^* be an accumulation point of the sequence $\{x^k\}$. Then x^* is a stationary point of (5.3.1) and $\lim_{k \rightarrow \infty} F(x^k) = F(x^*)$.

Proof. (1) By the Armijo step size, the fact that $\theta(x^k) < 0$ for all $k \in \mathbb{N}$, and by Lemma 5.4.3, we have

$$F(x^k) - F(x^{k+1}) \geq \zeta \lambda_k |\theta(x^k)| > 0, \quad (5.4.8)$$

which implies that the sequence $\{F(x^k)\}$ is monotonically decreasing. On the other hand, since $\{x^k\} \subset \mathcal{X}$ and $\mathcal{X}$ is compact, there exists $\bar{x} \in \mathcal{X}$ that is a accumulation point of a subsequence of $\{x^k\}$. Without loss of generality, assume that $x^k \rightarrow \bar{x}$.

From (5.4.8), with $k = 0, 1, \dots, N$ for some N , we have $F(x^0) - F_{\min} \geq F(x^0) - F(x^N) \geq \zeta \sum_{k=0}^N \lambda_k |\theta(x^k)|$, where $F_{\min}$ exists from Assumption 5.4.3. Taking $N \rightarrow \infty$, we have $\sum_{k=0}^{\infty} \lambda_k |\theta(x^k)| < +\infty$. Therefore, we have $\lim_{k \rightarrow \infty} \lambda_k \theta(x^k) = 0$. If

$$\lim_{k \rightarrow \infty} \theta(x^k) = 0, \quad (5.4.9)$$

then the assertion holds. So we consider the case

$$\lim_{k \rightarrow \infty} \lambda_k = 0. \quad (5.4.10)$$

Assume that $\lim_{\mathbb{K}_1 \ni k \rightarrow \infty} \lambda_k = 0$ for some $\mathbb{K}_1 \subseteq \mathbb{N}$. Suppose by contradiction that $\theta(\bar{x}) < 0$. Then, there exist $\delta > 0$ and $\mathbb{K}_2 \subseteq \mathbb{K}_1$ such that

$$\theta(x^k) < -\delta \quad \forall k \in \mathbb{K}_2.$$

Without loss of generality, we assume the existence of $\bar{p} \in \mathcal{X}$ such that:

$$\lim_{\mathbb{K}_2 \ni k \rightarrow \infty} p(x^k) = \bar{p}.$$

Since $\lambda_k < 1$ for all $k \in \mathbb{K}_2$, by Armijo step size, there exists $\bar{\lambda}_k \in (0, \lambda_k/\omega_1)$ such that

$$F(R_{x^k}(\bar{\lambda}_k R_{x^k}^{-1}(p(x^k)))) > F(x^k) + \zeta \bar{\lambda}_k \theta(x^k) \quad \forall k \in \mathbb{K}_2,$$

Then, we have

$$\frac{F(R_{x^k}(\bar{\lambda}_k R_{x^k}^{-1}(p(x^k)))) - F(x^k)}{\bar{\lambda}_k} > \zeta \theta(x^k) \quad \forall k \in \mathbb{K}_2. \quad (5.4.11)$$

On the other hand, since $0 < \bar{\lambda}_k \leq 1$, and g is R -convex, then

$$g(R_{x^k}(\bar{\lambda}_k R_{x^k}^{-1}(p(x^k)))) \leq (1 - \bar{\lambda}_k)g(x^k) + \bar{\lambda}_k g(p(x^k)),$$

namely,

$$\frac{g(R_{x^k}(\bar{\lambda}_k R_{x^k}^{-1}(p(x^k)))) - g(x^k)}{\bar{\lambda}_k} \leq g(p(x^k)) - g(x^k) \quad \forall k \in \mathbb{K}_2. \quad (5.4.12)$$

Since f is differentiable and $\lim_{\mathbb{K}_2 \ni k \rightarrow \infty} \bar{\lambda}_k = 0$, for all $k \in \mathbb{K}_2$, we have

$$f(R_{x^k}(\bar{\lambda}_k R_{x^k}^{-1}(p(x^k)))) = f(x^k) + \bar{\lambda}_k \langle \text{grad } f(x^k), R_{x^k}^{-1}(p(x^k)) \rangle_{x^k} + o(\bar{\lambda}_k), \quad (5.4.13)$$

which, together with (5.4.12) and (5.4.13) and the definition of θ , gives

$$\begin{aligned} \theta(x^k) &= \langle \text{grad } f(x^k), R_{x^k}^{-1}(p(x^k)) \rangle_{x^k} + g(p(x^k)) - g(x^k) \\ &\geq \frac{f(R_{x^k}(\bar{\lambda}_k R_{x^k}^{-1}(p(x^k)))) - f(x^k) - o(\bar{\lambda}_k)}{\bar{\lambda}_k} + \frac{g(R_{x^k}(\bar{\lambda}_k R_{x^k}^{-1}(p(x^k)))) - g(x^k)}{\bar{\lambda}_k} \\ &\geq \frac{F(R_{x^k}(\bar{\lambda}_k R_{x^k}^{-1}(p(x^k)))) - F(x^k)}{\bar{\lambda}_k} - \frac{o(\bar{\lambda}_k)}{\bar{\lambda}_k} \quad \forall k \in \mathbb{K}_2. \end{aligned} \quad (5.4.14)$$

Combining (5.4.14) with (5.4.11), we obtain for all $k \in \mathbb{K}_2$,

$$\frac{F(R_{x^k}(\bar{\lambda}_k R_{x^k}^{-1}(p(x^k)))) - F(x^k)}{\bar{\lambda}_k \zeta} > \frac{F(R_{x^k}(\bar{\lambda}_k R_{x^k}^{-1}(p(x^k)))) - F(x^k)}{\bar{\lambda}_k} - \frac{o(\bar{\lambda}_k)}{\bar{\lambda}_k},$$

that is,

$$\left(\frac{1}{\zeta} - 1\right) \frac{F(R_{x^k}(\bar{\lambda}_k R_{x^k}^{-1}(p(x^k)))) - F(x^k)}{\bar{\lambda}_k} > -\frac{o(\bar{\lambda}_k)}{\bar{\lambda}_k}.$$

Then, we have

$$\frac{F(R_{x^k}(\bar{\lambda}_k R_{x^k}^{-1}(p(x^k)))) - F(x^k)}{\bar{\lambda}_k} > \left(\frac{-\zeta}{1 - \zeta}\right) \frac{o(\bar{\lambda}_k)}{\bar{\lambda}_k} \quad \forall k \in \mathbb{K}_2.$$

On the other hand, from the fact that $\theta(x^k) < -\delta$ and (5.4.14), we obtain

$$-\delta + \frac{o(\bar{\lambda}_k)}{\bar{\lambda}_k} > \frac{F(R_{x^k}(\bar{\lambda}_k R_{x^k}^{-1}(p(x^k)))) - F(x^k)}{\bar{\lambda}_k} \quad \forall k \in \mathbb{K}_2.$$

Combining the two results above, it holds that

$$-\delta + \frac{o(\bar{\lambda}_k)}{\bar{\lambda}_k} > \left(\frac{-\zeta}{1 - \zeta}\right) \frac{o(\bar{\lambda}_k)}{\bar{\lambda}_k} \quad \forall k \in \mathbb{K}_2.$$

Since $\frac{o(\bar{\lambda}_k)}{\bar{\lambda}_k} \rightarrow 0$ as $k \rightarrow \infty$, we obtain $\delta \leq 0$, which contradicts the fact that $\delta > 0$. Thus, $\theta(\bar{x}) = 0$.

(2) Assume that x^* , an accumulation point of the sequence $\{x^k\}$, is not a stationary point of F , namely $\theta(x^*) < 0$. By the definition of θ in (5.3.4), we have

$$\begin{aligned}\theta(x^k) &= \langle \text{grad} f(x^k), R_{x^k}^{-1}(p(x^k)) \rangle_{x^k} + g(p(x^k)) - g(x^k) \\ &\leq \langle \text{grad} f(x^k), R_{x^k}^{-1}(u) \rangle_{x^k} + g(u) - g(x^k) \quad \forall u \in \mathcal{X},\end{aligned}$$

and by taking the limit we obtain

$$\limsup_{k \rightarrow \infty} \theta(x^k) \leq \langle \text{grad} f(x^*), R_{x^*}^{-1}(u) \rangle_{x^*} + g(u) - g(x^*) \quad \forall u \in \mathcal{X},$$

since f is continuously differentiable, g is lower semicontinuous and Assumption 5.3.1 holds. Now taking the minimum over $x \in \mathcal{X}$ and because x^* is not stationary point of F ,

$$\limsup_{k \rightarrow \infty} \theta(x^k) \leq \min_{u \in \mathcal{X}} \langle \text{grad} f(x^*), R_{x^*}^{-1}(u) \rangle_{x^*} + g(u) - g(x^*) = \theta(x^*) < 0,$$

which contradicts to $\lim_{k \rightarrow \infty} \theta(x^k) = 0$, so x^* is a stationary point of F .

Furthermore, from the monotonicity of the sequence $\{F(x^k)\}$ given in (5.4.8), we have $\lim_{k \rightarrow \infty} F(x^k) = F(x^*)$, which completes the proof. ■

The following theorem provides the iteration complexity of Algorithm 7 with Armijo step size, indicating that after a finite number of steps, the iteration complexity of the method could be $\mathcal{O}(1/\epsilon^2)$.

Theorem 5.4.3. *Assume that F satisfies Assumptions 5.4.1 and 5.4.3. Then, there exists a finite integer $J \in \mathbb{N}$ such that Algorithm 7 with Armijo step size achieves an ϵ -optimal solution within $J + \mathcal{O}(1/\epsilon^2)$ iterations.*

Proof. First, we consider that $0 < \lambda_k < 1$. From the Armijo step size condition, there exists $0 < \bar{\lambda}_k \leq \min \left\{ 1, \frac{\lambda_k}{\omega_1} \right\}$ such that

$$F(R_{x^k} \bar{\lambda}_k (R_{x^k}^{-1}(p(x^k)))) > F(x^k) + \zeta \bar{\lambda}_k \theta(x^k).$$

Recall that from Lemma 5.4.1, we have

$$F(R_{x^k}(\bar{\lambda}_k(R_{x^k}^{-1}(p(x^k))))) < F(x^k) + \bar{\lambda}_k \theta(x^k) + \frac{L \bar{\lambda}_k^2}{2} \Omega^2.$$

Combining the above two inequalities, we obtain

$$\zeta \bar{\lambda}_k \theta(x^k) < \bar{\lambda}_k \theta(x^k) + \frac{L \bar{\lambda}_k^2}{2} \Omega^2,$$

namely,

$$0 < -\frac{2(1-\zeta)}{L\Omega^2} \theta(x^k) < \bar{\lambda}_k \leq \frac{\lambda_k}{\omega_1}.$$

Defining $\gamma = \frac{2\omega_1(1-\zeta)}{L\Omega^2}$, we get

$$0 < -\gamma\theta(x^k) < \lambda_k. \quad (5.4.15)$$

From Theorem 5.4.2, we have $\lim_{k \rightarrow \infty} \theta(x^k) = 0$. Then, without loss of generality, for sufficiently large J , when $k > J$, $|\theta(x^k)| \leq 1/\gamma$. Then (5.4.15) still holds for $\lambda_k = 1$. Taking it into (5.4.8), we conclude that

$$0 < \zeta\gamma |\theta(x^k)|^2 \leq F(x^k) - F(x^{k+1}). \quad (5.4.16)$$

Namely (5.4.16) holds for all $k > J$. By summing both sides of the second inequality in (5.4.16) for $k = J+1, \dots, N$, we have

$$\sum_{k=J+1}^N |\theta(x^k)|^2 \leq \frac{1}{\zeta\gamma} (F(x^{J+1}) - F(x^{N+1})) \leq \frac{1}{\zeta\gamma} (F(x^0) - F(x^*)),$$

where x^* is an accumulation point of $\{x^k\}$, as in Theorem 5.4.2.

This implies $\min_k \{|\theta(x^k)| \mid k = J+1, \dots, N\} \leq \sqrt{(F(x^0) - F(x^*)) / (\zeta\gamma(N-J))}$. Namely, Algorithm 7 generates an iterate x^k satisfying $|\theta(x^k)| \leq \epsilon$ in at most $J + (F(x^0) - F(x^*)) / (\zeta\gamma\epsilon^2)$ iterations. ■

Furthermore, we can obtain stronger results by making additional assumptions about the objective function g , similarly to [8, Section 3 (A4)].

Assumption 5.4.4. Assume the function g is Lipschitz continuous with constant $L_g > 0$ in $\mathcal{X}$, namely, $|g(x) - g(y)| \leq L_g \|R_x^{-1}(y)\|_x$.

Moreover, we define

$$\begin{aligned} \rho &:= \sup \{ \|\text{grad } f(x)\|_x \mid x \in \mathcal{X} \} \\ \gamma &:= \min \left\{ \frac{1}{(L_g + \rho)\Omega}, \frac{2\omega_1(1-\zeta)}{L\Omega^2} \right\}. \end{aligned} \quad (5.4.17)$$

Next, we introduce the following lemma, which is crucial for the subsequent convergence rate analysis.

Lemma 5.4.4. Let $\{x^k\}$ be a sequence generated by Algorithm 7 with Armijo step size. Assume that Assumptions 5.4.1, 5.4.2 and 5.4.4 hold. Then $\lambda_k \geq \gamma |\theta(x^k)| > 0$.

Proof. Since $\lambda_k \in (0, 1]$, we consider two cases: $\lambda_k = 1$ and $0 < \lambda_k < 1$. First, assume that $\lambda_k = 1$. From (5.3.4), we have

$$0 < -\theta(x^k) = g(x^k) - g(p(x^k)) - \langle \text{grad } f(x^k), R_{x^k}^{-1}(p(x^k)) \rangle_{x^k},$$

thus, it follows from Assumption 5.4.4 and the Cauchy-Schwarz inequality that

$$0 < -\theta(x^k) \leq L_g \|R_{x^k}^{-1}(p(x^k))\|_{x^k} + \|\text{grad } f(x^k)\|_{x^k} \|R_{x^k}^{-1}(p(x^k))\|_{x^k}.$$

Using (5.4.17), we have $0 < -\theta(x^k) \leq (L_g + \rho)\Omega$. Furthermore,

$$0 < \gamma|\theta(x^k)| = -\gamma\theta(x^k) \leq \frac{-\theta(x^k)}{(L_g + \rho)\Omega} \leq 1,$$

which demonstrates that the desired inequality holds.

Now, consider $0 < \lambda_k < 1$. From the Armijo step size, there exists $0 < \bar{\lambda}_k \leq \min \left\{ 1, \frac{\lambda_k}{\omega_1} \right\}$ such that

$$F(R_{x^k}(\bar{\lambda}_k R_{x^k}^{-1}(p(x^k)))) > F(x^k) + \zeta \bar{\lambda}_k \theta(x^k).$$

By using Lemma 5.4.1, we have

$$F(R_{x^k}(\bar{\lambda}_k R_{x^k}^{-1}(p(x^k)))) \leq F(x^k) + \bar{\lambda}_k \theta(x^k) + \frac{L}{2} \Omega^2 \bar{\lambda}_k^2.$$

From the above two inequalities, we have that

$$-\theta(x^k)(1 - \zeta) < \frac{L}{2} \Omega^2 \bar{\lambda}_k \leq \frac{L}{2} \Omega^2 \frac{\lambda_k}{\omega_1}.$$

Therefore, from (5.4.17), we get

$$0 < \gamma|\theta(x^k)| = -\gamma\theta(x^k) \leq -\frac{2\omega_1(1 - \zeta)}{L\Omega^2} \theta(x^k) < \lambda_k,$$

which concludes the proof. ■

Theorem 5.4.4. *Let $\{x^k\}$ be a sequence generated by Algorithm 7 with Armijo step size. Assume that Assumptions 5.4.1, 5.4.2 and 5.4.4 hold. Then $\lim_{k \rightarrow \infty} F(x^k) = F(x^*)$, for some $x^* \in \mathcal{X}$. Moreover, the following statements hold:*

- (1) $\lim_{k \rightarrow \infty} \theta(x^k) = 0$;
- (2) $\min \{ |\theta(x^k)| \mid k = 0, 1, \dots, N-1 \} \leq \sqrt{F(x^0) - F(x^*) / (\zeta \gamma N)}$.

Proof.

$$F(x^k) - F(x^{k+1}) \geq \zeta \lambda_k |\theta(x^k)| > 0. \quad (5.4.18)$$

It implies that the sequence $\{F(x^k)\}_{k \in \mathbb{N}}$ is monotone decreasing. Combined with Lemma 5.4.4, we have

$$0 < \zeta \gamma |\theta(x^k)|^2 \leq F(x^k) - F(x^{k+1}). \quad (5.4.19)$$

Since $\{x^k\}_{k \in \mathbb{N}} \subset \mathcal{X}$ and $\mathcal{X}$ is compact, there exists a limit point $x^* \in \mathcal{X}$ of the sequence $\{x^k\}_{k \in \mathbb{N}}$. Let $\{x^{k_j}\}_{j \in \mathbb{N}}$ denote a subsequence of $\{x^k\}_{k \in \mathbb{N}}$ such that $\lim_{j \rightarrow \infty} x^{k_j} = x^*$. Since $\{x^{k_j}\}_{j \in \mathbb{N}} \subset \mathcal{X}$ and g is Lipschitz continuous, we have

$$\|F(x^{k_j}) - F(x^*)\| = \|g(x^{k_j}) - g(x^*) + f(x^{k_j}) - f(x^*)\| \leq L_g \|x^{k_j} - x^*\| + \|f(x^{k_j}) - f(x^*)\|,$$

for all $j \in \mathbb{N}$. Considering that f is continuous and $\lim_{j \rightarrow \infty} x^{k_j} = x^*$, we conclude from the last inequality that $\lim_{j \rightarrow \infty} F(x^{k_j}) = F(x^*)$. Thus, from the monotonicity of the sequence $\{F(x^k)\}_{k \in \mathbb{N}}$, we have $\lim_{k \rightarrow \infty} F(x^k) = F(x^*)$. Therefore, from (5.4.19), $\lim_{k \rightarrow \infty} |\theta(x^k)| = 0$, which implies item (1).

By summing both sides of the second inequality in (5.4.19) for $k = 0, 1, \dots, N-1$, we conclude that

$$\sum_{k=0}^{N-1} |\theta(x^k)|^2 \leq \frac{1}{\zeta\gamma} (F(x^0) - F(x^N)) \leq \frac{1}{\zeta\gamma} (F(x^0) - F(x^*)),$$

which implies the item (2).

Therefore, x^k satisfies $|\theta(x^k)| \leq \epsilon$ in at most $(F(x^0) - F(x^*)) / (\zeta\gamma\epsilon^2)$ iterations. ■

Remark 5.4.1. *It is worth noting that our framework naturally recovers the classical setting of geodesic convexity. Specifically, when the retraction in this chapter is chosen to be the exponential map, our R -convexity requirement on the function g reduces to standard geodesic convexity. It follows that theoretical results (Lemma 5.4.1, Proposition 5.4.1, and Theorem 5.4.1) remain valid without Assumption 5.4.2, with the constant Ω in the convergence bounds being replaced by $\text{diam}(\mathcal{X})$.*

5.5 Solving the subproblem

In this section, we will propose an algorithm to solve the subproblem of the RGCondG method. Recall that in Algorithm 7, given the current iterate x , we need to compute $p(x)$ and $\theta(x)$ as

$$p(x) = \arg \min_{u \in \mathcal{X}} \langle \text{grad} f(x), R^{-1}(u) \rangle_x + g(u) - g(x), \quad (5.5.1)$$

$$\theta(x) = \langle \text{grad} f(x), R_x^{-1}(p(x)) \rangle_x + g(p(x)) - g(x). \quad (5.5.2)$$

Now define $\tilde{p}(x)$ as the descent direction on the tangent space restricted to a compact domain, i.e.,

$$\tilde{p}(x) = \arg \min_{d \in \mathcal{C}_x} \langle \text{grad} f(x), d \rangle_x + g(R_x(d)). \quad (5.5.3)$$

Here, the domain $\mathcal{C}_x$ is defined as the intersection of the inverse image of $\mathcal{X}$ and a bounded region in the tangent space:

$$\mathcal{C}_x = \{v \in T_x \mathcal{M} \mid R_x(v) \in \mathcal{X}\} \cap \{v \in T_x \mathcal{M} \mid \|v\|_x \leq \Delta\}, \quad (5.5.4)$$

where $\Delta > 0$ is a sufficiently large radius. Although the inverse image $R_x^{-1}(\mathcal{X})$ itself may be unbounded (due to the periodicity of the retraction), its intersection with the closed ball $\{v \mid \|v\|_x \leq \Delta\}$ ensures that $\mathcal{C}_x$ is compact. This guarantees that the linear subproblem is bounded and admits a solution. And in practice, Δ could be chosen to be large enough such

that this artificial boundary remains inactive during the optimization process, meaning it serves as a theoretical requirement rather than a practical restriction on the descent direction.

We also define

$$\ell_x(d) = \langle \text{grad} f(x), d \rangle_x + g(R_x(d)). \quad (5.5.5)$$

We observe that for both (5.5.1) and (5.5.3), the retraction is not necessarily a convex function, which implies that the resulting optimization problem is not necessarily convex. Non-convex problems are inherently more challenging to solve. To address this, we aim to transform the subproblem into a series of convex problems, similar to the general method for solving Riemannian proximal mappings proposed in [86]. To ensure that we can locally approximate and simulate the subproblems effectively, we consider the following assumption:

Assumption 5.5.1. *Assume that the manifold $\mathcal{M}$ is either an embedded submanifold of $\mathbb{R}^n$ or a quotient manifold whose total space is an embedded submanifold of $\mathbb{R}^n$.*

Let us now present a way to find the approximation of $\tilde{p}(x) = \arg \min_{d \in \mathcal{C}_x} \ell_x(d)$. Assume that $d^0 \in T_x \mathcal{M}$, and let d^k denote the current estimate of $\tilde{p}(x)$. Observe that

$$\begin{aligned} \ell_x(d^k + \tilde{\xi}_k) &= \left\langle \text{grad} f(x), d^k + \tilde{\xi}_k \right\rangle_x + g(R_x(d^k + \tilde{\xi}_k)) \\ &= \left\langle \text{grad} f(x), d^k \right\rangle_x + \left\langle \text{grad} f(x), \tilde{\xi}_k \right\rangle_x + g(R_x(d^k + \tilde{\xi}_k)), \end{aligned}$$

for any $\tilde{\xi}_k \in T_x \mathcal{M}$. Let $y_k = R_x(d^k)$ and $\xi_k = \mathcal{T}_{R_{d^k}}^{-1} \tilde{\xi}_k$. By the smoothness property of the retraction R , we obtain $R_x(d^k + \tilde{\xi}_k) = y_k + \xi_k + O(\|\xi_k\|_x^2)$, where $y = x + O(z)$ means $\limsup_{z \rightarrow 0} \|y - x\|/\|z\| < \infty$. Then, we obtain:

$$\begin{aligned} \ell_x(d^k + \tilde{\xi}_k) &= \left\langle \text{grad} f(x), d^k \right\rangle_x + \left\langle \text{grad} f(x), \mathcal{T}_{R_{d^k}}^{-1} \xi_k \right\rangle_x + g(y_k + \xi_k + O(\|\xi_k\|_x^2)) \\ &= \left\langle \text{grad} f(x), d^k \right\rangle_x + \left\langle \text{grad} f(x), \mathcal{T}_{R_{d^k}}^{-1} \xi_k \right\rangle_x + g(y_k + \xi_k) + O(\|\xi_k\|_x^2), \end{aligned}$$

where the second equality follows from Assumption 5.4.4, which states the Lipschitz continuity of g . Note that the expression

$$\left\langle \text{grad} f(x), \mathcal{T}_{R_{d^k}}^{-1} \xi_k \right\rangle_x + g(y_k + \xi_k) \quad (5.5.6)$$

can be viewed as a simple local model of $\ell_x(d^k + \mathcal{T}_{R_{d^k}}^{-1} \xi_k)$. Thus, to obtain a new estimate from d^k , we first determine a search direction ξ_k^* by minimizing (5.5.6) on $\mathcal{C}_{y_k}$. Then, we update d^k in the direction of $\mathcal{T}_{R_{d^k}}^{-1}(\xi_k^*)$. As described in Algorithm 8.

It is easy to find that Algorithm 8 is the application of the Riemannian GCondG method for problem (5.5.3) with a proper retraction. Moreover, note that (5.5.7) can be solved by the subgradient method or proximal gradient method.

Algorithm 8: Solving the subproblem

Require: An initial iterate $d^0 \in T_x \mathcal{M}$; a small positive constant σ ;

1: for $k = 0, 1, \dots$, do

2: $y_k = R_x(d^k)$

3: Compute ξ_k^* by solving

$$\xi_k^* = \arg \min_{\xi \in \mathcal{C}_{y_k}} \left\langle \mathcal{T}_{R_{d^k}}^{-\sharp}(\text{grad } f(x)), \xi \right\rangle_{y_k} + g(y_k + \xi); \quad (5.5.7)$$

4: $\alpha = 1$

5: while $\ell_x(d^k + \alpha \mathcal{T}_{R_{d^k}}^{-1} \xi_k^*) \geq \ell_x(d^k) - \sigma \alpha \|\xi_k^*\|_x^2$ do

6: $\alpha = \frac{1}{2} \alpha$;

7: end while

8: $d_{k+1} = d^k + \alpha \mathcal{T}_{R_{d^k}}^{-1} \xi_k^*$;

4: end for

5.6 Accelerated conditional gradient methods

Now we consider the accelerated version of the RCondG method. We noticed that based on the accelerated gradient methods, Li et al. [95] establish links between subproblems of conditional gradient methods and the notion of momentum. They proposed a momentum variant of CondG method and proved that it converges with a fast rate of $O(\frac{1}{k^2})$. However, the extension of this work to the Riemannian manifold case is not trivial, even though there are works on Nesterov accelerated methods on Riemannian manifolds [4]. Based on these existing works, here we propose an accelerated CondG method on Riemannian manifolds for composite function.

Algorithm 9: Riemannian accelerated GCondG method

Step 0. Initialization Choose $x^0 = p_0 \in \mathcal{X}$, $\tau^k = \frac{2}{k+3}$ and initialize $k = 0$,
 $\beta_0 = \mathbf{0}$, $d^0 = \mathbf{0}$.

Step 1. Compute

$$\begin{aligned} y_k &= R_{x^k}(\tau^k R_{x^k}^{-1}(d^k)) \\ \beta_{k+1} &= \mathcal{T}_{y_k}(1 - \tau^k)\beta_k + \lambda_k \text{grad}f(y_k) \end{aligned}$$

Step 2. Compute p_k and $\theta(x^k)$ as

$$p_{k+1} = \arg \min_{u \in \mathcal{X}} \langle \beta_{k+1}, R_{y_k}^{-1}(u) \rangle_{y_k} + g(u) - g(y_k), \quad (5.6.1)$$

$$d_{k+1} = R_{y_k}^{-1}(p_{k+1}).$$

Step 3. Stopping criteria If $\theta(x^k) = 0$, then stop.

Step 4. Compute the iterate Set

$$x^{k+1} = R_{x^k}(\lambda_k R_{x^k}^{-1}(p_{k+1})). \quad (5.6.2)$$

Step 5. Starting a new iteration Set $k = k + 1$ and return to Step 1.

Analyzing the convergence and convergence rate for the above accelerated method is challenging, primarily due to the inherent nonconvexity of the problem and the complexities introduced by transporting tangent vectors. A comprehensive analysis of the accelerated method is beyond the scope of this chapter. However, the performance of the accelerated method compared to the non-accelerated method is shown in the numerical experiments in Section 5.7.

5.7 Numerical experiments

Sparse principal component analysis (Sparse PCA) plays a crucial role in high-dimensional data analysis by enhancing the interpretability and efficiency of dimensionality reduction. Unlike traditional PCA, which often produces complex results in high-dimensional settings, Sparse PCA introduces sparsity to refine data representation and improve clarity. This is particularly relevant in fields such as bioinformatics, image processing, and signal analysis, where interpretability and reconstruction accuracy are essential. In this section, we consider Sparse PCA problems to check the validity of our proposed methods. We explore two models for sparse PCA as in [86]; see that reference for further analysis. Here, we set $\mathcal{X} = \mathcal{M}$ in each problem, and Assumptions 5.4.1, 5.4.3 and 5.4.4 hold.

5.7.1 Sparse PCA on the sphere manifold

Now consider problems in sphere manifolds $\mathbb{S}^{n-1} = \{x \in \mathbb{R}^n : \|x\| = 1\}$. The specific optimization problem is formulated as follows:

$$\min_{x \in \mathbb{S}^{n-1}} -x^\top A^\top A x + \lambda \|x\|_1, \quad (5.7.1)$$

where $\|\cdot\|_1$ denotes the L_1 -norm, and $\lambda > 0$ is a parameter. Here, we use the exponential mapping as the retraction, i.e.,

$$\text{Exp}_x(\eta_x) = x \cos(\|\eta_x\|) + \eta_x \sin(\|\eta_x\|) / \|\eta_x\|, \quad x \in \mathbb{S}^{n-1}, \quad \eta_x \in T_x \mathbb{S}^{n-1}.$$

The inverse exponential mapping can be defined as follows, as presented in [86]:

$$\log_x(y) = \text{Exp}_x^{-1}(y) = \frac{\cos^{-1}(x^\top y)}{\sqrt{1 - (x^\top y)^2}} (I_n - x x^\top) y, \quad x, y \in \mathbb{S}^{n-1}.$$

The subproblem (5.3.3) applied to (5.7.1), can be written as

$$\min_{y \in \mathbb{S}^{n-1}} u(y), \quad \text{where } u(y) = \frac{1}{\lambda} \langle \text{grad} f(x), \log_x(y) \rangle + \|y\|_1. \quad (5.7.2)$$

Let us not check how to compute its solution. Let $h(y) = \frac{1}{\lambda} \langle \text{grad} f(x), \log_x(y) \rangle$, and let y^k denote the current estimate of the minimizer of $u(y)$. Then solving the following problem to obtain the next estimate y^{k+1} :

$$\min_{y \in \mathbb{S}^{n-1}} h(y^k) + \nabla h(y^k)^\top (y - y^k) + \|y\|_1,$$

which is equal to

$$\min_{y \in \mathbb{S}^{n-1}} \nabla h(y^k)^\top y + \|y\|_1.$$

Actually, $h(y)$ is approximated by its first-order Taylor expansion around y^k . Note that since $\|y\| = 1$ for all $y \in \mathbb{S}^{n-1}$, the above problem is further equivalent to:

$$\min_{y \in \mathbb{S}^{n-1}} \frac{1}{2} \|y + \nabla h(y^k)\|^2 + \|y\|_1. \quad (5.7.3)$$

The above problem can be resolved directly using [86, Lemma D.1].

5.7.2 Sparse PCA on the Stiefel manifold

Now consider sparse PCA on the Stiefel manifold. The specific optimization problem is formulated as follows:

$$\min_{X \in \text{st}(p,n)} -\text{trace}(X^\top A^\top A X) + \lambda \|X\|_1. \quad (5.7.4)$$

where $\text{st}(p, n) = \{X \in \mathbb{R}^{n \times p} \mid X^\top X = I_p\}$ denotes the Stiefel manifold, and $\lambda > 0$. Moreover, the term $-\text{trace}(X^\top A^\top A X)$ aims to maximize the variance of the data, $\|X\|_1$ characterizes the sparsity of the principal components, and A is the original data matrix. The tangent space is defined as $T_X \text{st}(p, n) = \{\eta \in \mathbb{R}^{n \times p} \mid X^\top \eta + \eta^\top X = 0\}$.

Here we use the Riemannian metric defined as $\langle \eta_X, \xi_X \rangle_X = \text{trace}(\eta_X^\top \xi_X)$. And using the polar decomposition as the retraction, i.e.,

$$R_X(\eta_X) = (X + \eta_X) (I_p + \eta_X^\top \eta_X)^{-1/2}.$$

The vector transport is defined by

$$\mathcal{T}_{\eta_X} \xi_X = Y \Omega + (I_n - Y Y^\top) \xi_X (Y^\top (X + \eta_X))^{-1},$$

where $Y = R_X(\eta_X)$ and Ω is the solution of the Sylvester equation $(Y^\top (X + \eta_X)) \Omega + \Omega (Y^\top (X + \eta_X)) = Y^\top \xi_X - \xi_X^\top Y$, as given in [84, Lemma 10.2.1].

For ease of notation, we define the operator $\mathcal{A}_k$ by $\mathcal{A}_k(V) := V^\top X_k + X_k^\top V$ and rewrite the subproblem (5.5.7) as

$$\begin{aligned} V_k &:= \underset{V}{\operatorname{argmin}} \left\langle \mathcal{T}_{R_{d^k}}^{-\sharp} \operatorname{grad} f(X_k), V \right\rangle + g(X_k + V) \\ &\text{s.t. } \mathcal{A}_k(V) = 0. \end{aligned}$$

This can be solved by the subgradient method or the proximal gradient method. For the subgradient method, note that the subgradient of $g(X)$ can be calculate through function $m(X)$, which is defined as:

- $m_{ij}(X) = \lambda$ if $(X)_{ij} > 0$,
- $m_{ij}(X) = -\lambda$ if $(X)_{ij} < 0$,
- $m_{ij}(X) \in [-\lambda, \lambda]$ if $(X)_{ij} = 0$,

where $m_{ij}(X)$ is the element of $m(X)$. Then the subgradient of the above problem can be written as $\mathcal{T}_{R_{d^k}}^{-\sharp} \operatorname{grad} f(X_k) + \lambda \cdot \operatorname{sign}(X_k + V_k)$.

5.7.3 Numerical results

We implemented the algorithms in Python 3.9.12, and the experiments were carried out on a MacBook Pro with an Apple M1 Pro chip and 16 GB of RAM.

Sphere experiment

For the example (5.7.1), we conducted experiments on a n -dimensional problem, using a randomly generated $n \times n$ standardized A and a regularization parameter $\lambda = 0.1$. The experiment evaluates Armijo step size with parameters $\zeta = 0.1$, $\omega_1 = 0.05$, and $\omega_2 = 0.95$. Each experiment runs for the same original problem 10 times with different initial points

and stops if the norm of θ_{x^k} is less than 10^{-4} or if the difference between the latest function value and the function value from five iterations ago is less than 10^{-4} . In this example, our experiments were conducted on high-dimensional spheres, with dimensions set to 10, 100, and 1000, across ten runs for each method. In the numerical experiments, we observed that solving the subproblems typically converges after just one iteration, suggesting that the local model provides an effective approximation of the original subproblem.

As shown in Table 5.1 and Figure 5.1, the results demonstrated that the Armijo step-size strategy consistently outperformed the others in terms of computational time and required iterations to achieve function value convergence. Overall, our findings suggest that the Armijo step-size strategy offers significant advantages in efficiency and speed, particularly in high-dimensional settings.

We observe the performance of different step-size strategies (Armijo, adaptive, and diminishing) under varying values of λ . Across all values of λ in Table 5.2, the Armijo step size consistently outperforms the other methods in terms of both time and iterations. Specifically, at $\lambda = 0.1$, the Armijo method achieves the fastest convergence, taking an average of 0.3505 seconds and 53.7 iterations. As λ increases, the performance gap between Armijo and the other methods becomes more pronounced. For $\lambda = 0.5$, Armijo converges in 0.3153 seconds and 44.8 iterations, while the diminishing strategy requires 2.0290 seconds and 530.2 iterations, demonstrating its inefficiency for larger λ values. Similarly, the adaptive step size shows intermediate performance, but its time and iteration counts remain higher than Armijo. Overall, the results suggest that the Armijo step size is the most robust and efficient strategy for different values of λ , while the diminishing method struggles with larger problem sizes and regularization terms.

Comparison with the accelerated method

We consider the same problem and compare the performance of the accelerated algorithm with the non-accelerated version. To better observe the convergence rate and differences between various step sizes, we run 50 iterations. As shown in Figure 5.2, the results indicate that the accelerated algorithm with Armijo and diminishing step sizes generally outperforms the non-accelerated algorithm, particularly when compared with adaptive and diminishing step sizes. The results are similar to the method in [95], and it can indeed accelerate the decline at the beginning.

Table 5.3 summarizes the results of 10 random runs for the Sphere Compare Experiment with $\lambda = 0.1$. Across all dimensions ($n = 10, 100, 500, 1000$), the RGCondG method combined with Armijo step size consistently achieves the best performance, with the lowest computation time and iteration count. For instance, with $n = 1000$, RGCondG + Armijo takes 8.47 seconds and 346.8 iterations, outperforming the adaptive and diminishing strategies, which require more time and iterations.

The accelerated method shows similar trends, with Armijo step size yielding better results than adaptive and diminishing strategies. However, for higher dimensions, RGCondG with

Armijo step size remains faster overall. The diminishing step size, in both methods, leads to significantly longer runtimes and higher iteration counts, especially as n increases. For $n = 1000, 2000$, indicating that the accelerated method benefits more from the diminishing step size in higher dimensions. In summary, the Armijo step size proves to be the most efficient strategy, particularly when paired with the RGCondG method, making it the preferred choice for both low and high-dimensional problems.

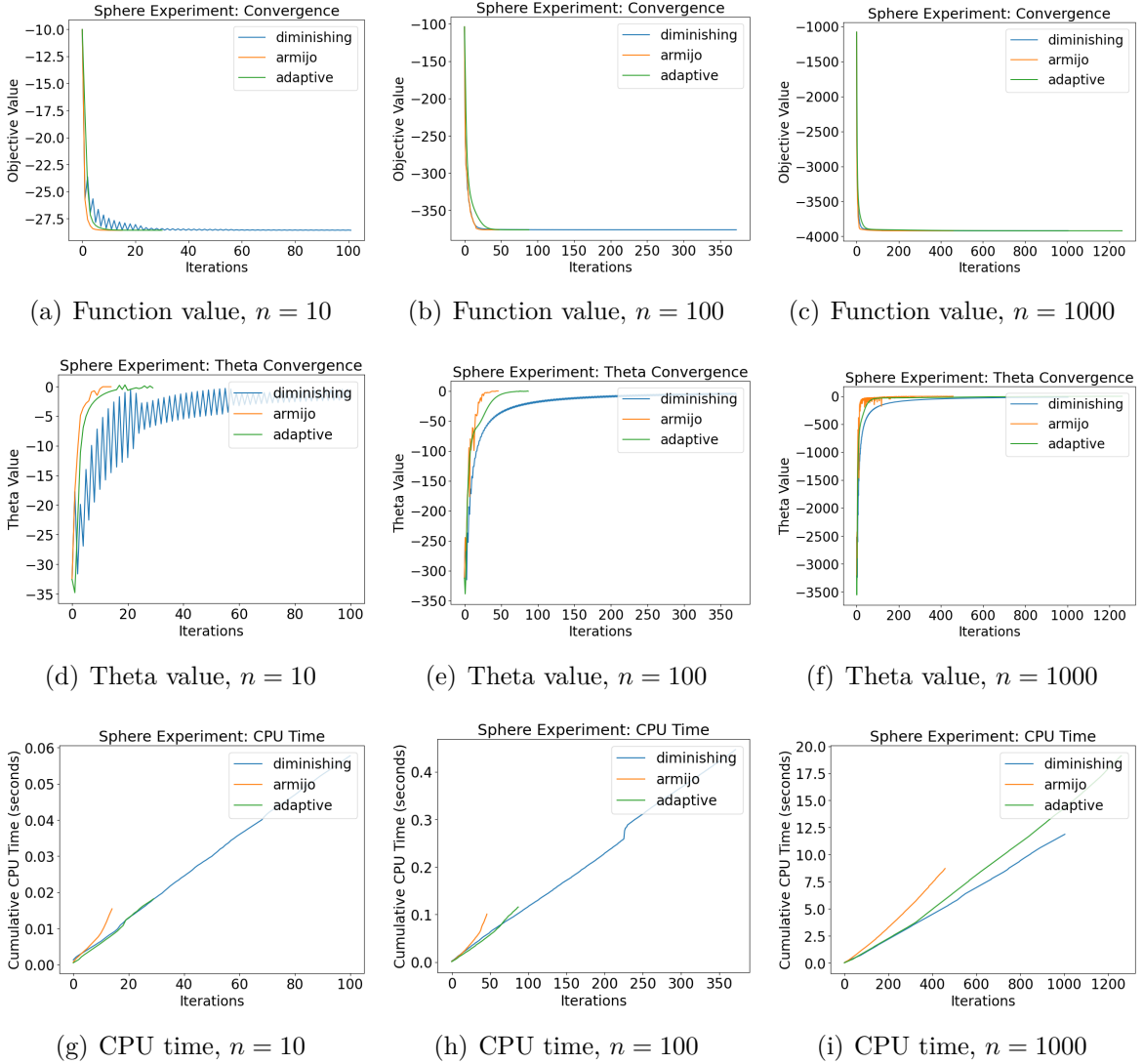


Figure 5.1: Example (5.7.1) with fixed A , different n and $\lambda = 0.1$

Stiefel experiment

For the example (5.7.4), we conduct experiments on a (n, p) -dimensional problem, using a randomly generated $n \times n$ standardized symmetric A and a regularization parameter $\lambda = 0.1$. As in the previous experiment, here we evaluate Armijo step size with parameters $\zeta = 0.1$,

Table 5.1: Average results of 10 random runs for problem (5.7.1) with $\lambda = 0.1$ and fixed A

n	Step size	Time (seconds)	Iterations
10	Armijo	0.0152	14.50
	Adaptive	0.0208	31.30
	Diminishing	0.0491	88.80
100	Armijo	0.0771	32.20
	Adaptive	0.1761	78.70
	Diminishing	0.3113	244.60
1000	Armijo	8.3842	344.60
	Adaptive	17.4690	921.00
	Diminishing	13.1272	929.70

$\omega_1 = 0.05$, and $\omega_2 = 0.95$. Each experiment runs for the same original problem 10 times with different initial points, and stops if the norm of θ or the difference of the function value is less than 10^{-4} . The maximum number of iterations allowed for solving subproblems is set to 2. The performance of the RGCondG method is analyzed by plotting the convergence behavior of the objective values and the value of θ over iterations and the table of average performance.

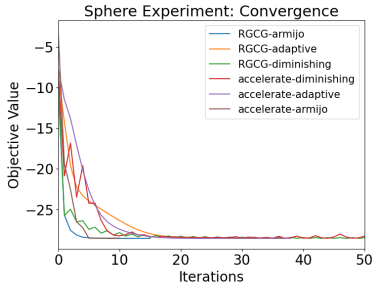
The experimental results for (5.7.4) on the Stiefel manifold (Table 5.4 and Figures 5.3) show clear differences in performance between the three step-size strategies across various problem sizes. For smaller problem dimensions (e.g., $n = 100$ and $n = 200$), the Armijo step size consistently achieves the fewest iterations, indicating its efficiency in reaching convergence faster. However, the diminishing step size, while stable, requires significantly more iterations and time, especially as the problem dimension increases.

For higher dimensions ($n = 500$ and $n = 1000$), the adaptive step size shows a competitive balance between time and iterations, especially compared to the diminishing one, which becomes increasingly less efficient. However, Armijo continues to outperform both in terms of iterations, making it the preferred choice for high-dimensional problems where computational efficiency is critical. Overall, the Armijo step size demonstrates superior performance in both time and iteration count, particularly for larger problem sizes. The diminishing step size, while robust, is less efficient for larger problems, indicating that it may not be suitable for high-dimensional optimization tasks.

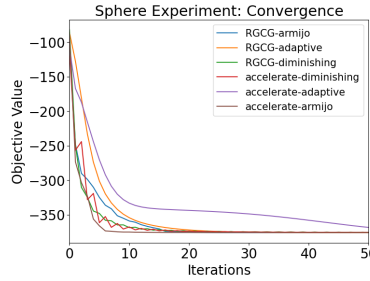
We see a similar trend with the Armijo step size consistently performing well across varying values of λ (Table 5.5). At $\lambda = 0.1$, Armijo converges in 0.6839 seconds and 103.3 iterations, slightly slower than the adaptive method in terms of time (0.5516 seconds) but significantly faster than the diminishing strategy (1.3375 seconds and 409.9 iterations). For different λ , Armijo maintains its competitive performance. For $\lambda = 1.0$, it achieves an average of 0.9943 seconds and 147.8 iterations, compared to 1.7331 seconds and 514.4 iterations for

Table 5.2: Average results of 10 random runs for problem (5.7.1) with varying λ and fixed A , $n = 500$

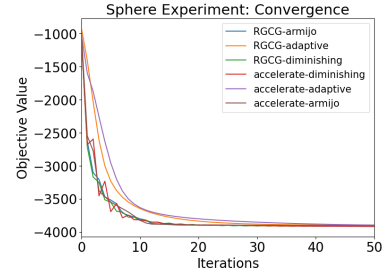
λ	Step size	Time (seconds)	Iterations
0.1	Armijo	0.3505	53.70
	Adaptive	0.7006	136.10
	Diminishing	2.2043	522.90
0.3	Armijo	0.3987	51.60
	Adaptive	0.6094	111.70
	Diminishing	1.6226	381.80
0.5	Armijo	0.3153	44.80
	Adaptive	0.5716	111.30
	Diminishing	2.0290	530.20
1.0	Armijo	0.3016	46.50
	Adaptive	0.5137	97.30
	Diminishing	1.8412	465.50



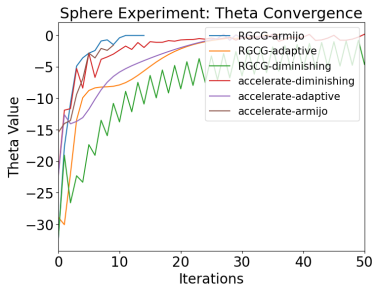
(a) Function value, $n = 10$



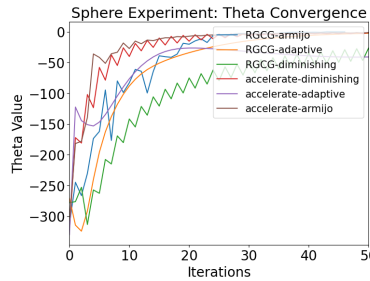
(b) Function value, $n = 100$



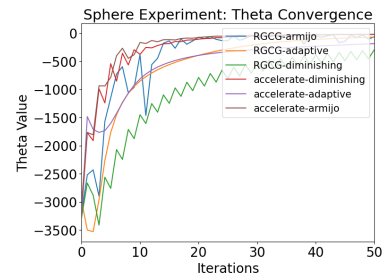
(c) Function value, $n = 1000$



(d) Theta value, $n = 10$



(e) Theta value, $n = 100$



(f) Theta value, $n = 1000$

Figure 5.2: Example (5.7.1) with fixed A , $\lambda = 0.1$ and different n for RGCondG and accelerated method

Table 5.3: Comparison of Accelerated and RGCondG Methods: Average Results of 10 Random Runs for problem (5.7.1) with $\lambda = 0.1$

n	Method + Step size	Time (seconds)	Iterations
10	RGCondG + Armijo	0.0173	15.30
	RGCondG + Adaptive	0.0288	36.10
	RGCondG + Diminishing	0.0796	109.50
10	Accelerated + Armijo	0.0496	36.80
	Accelerated + Adaptive	0.0507	61.30
	Accelerated + Diminishing	0.1137	107.30
100	RGCondG + Armijo	0.1069	32.40
	RGCondG + Adaptive	0.1670	83.00
	RGCondG + Diminishing	0.6486	295.50
100	Accelerated + Armijo	0.2629	85.30
	Accelerated + Adaptive	0.3232	169.30
	Accelerated + Diminishing	0.9453	368.50
500	RGCondG + Armijo	0.3750	56.70
	RGCondG + Adaptive	0.8563	160.50
	RGCondG + Diminishing	2.4350	605.30
500	Accelerated + Armijo	1.3190	166.00
	Accelerated + Adaptive	2.4168	385.60
	Accelerated + Diminishing	2.8703	417.10
1000	RGCondG + Armijo	8.4704	346.80
	RGCondG + Adaptive	17.8079	830.50
	RGCondG + Diminishing	14.0213	908.70
1000	Accelerated + Armijo	8.6065	255.30
	Accelerated + Adaptive	13.2967	687.70
	Accelerated + Diminishing	12.1202	445.60
2000	RGCondG + Armijo	14.6191	160.20
	RGCondG + Adaptive	21.9078	411.40
	RGCondG + Diminishing	50.3453	1106.70
2000	Accelerated + Armijo	19.0932	188.00
	Accelerated + Adaptive	47.8875	752.60
	Accelerated + Diminishing	25.3371	396.20

the diminishing step size. While the adaptive strategy generally outperforms the diminishing one, it remains less efficient than Armijo in terms of iterations.

Table 5.4: Average results of 10 random runs for problem (5.7.4) with $\lambda = 0.1$ and fixed A

(p, n)	Step size	Time (seconds)	Iterations
(10, 100)	Armijo	0.2344	104.7
	Adaptive	0.2287	233.0
	Diminishing	0.1775	192.4
(10, 200)	Armijo	0.2729	59.2
	Adaptive	0.4160	242.8
	Diminishing	0.5557	347.0
(10, 500)	Armijo	1.8151	115.8
	Adaptive	1.5056	229.5
	Diminishing	5.5653	827.5
(10, 1000)	Armijo	11.9895	219.0
	Adaptive	12.6799	459.3
	Diminishing	44.2100	1630.5

The overall results demonstrate that both the adaptive and Armijo step size methods outperform the diminishing step size in terms of convergence speed and computational time. Specifically, the adaptive strategy consistently exhibits the fastest convergence and the fewest iterations in most cases, while the Armijo step size method shows similar superiority in the majority of experiments. The performance of the algorithms is significantly influenced by different experimental parameters. Notably, as the number of rows and columns increases, the computational time and the number of iterations for all methods increase, but the adaptive and Armijo step sizes still perform better than the diminishing method. It was observed that the diminishing step size method experiences a significant slowdown in convergence speed as it approaches the stationary point. This indicates that while the method can initially reduce the objective function value quickly, its decreasing step size results in very small updates near the optimal solution, leading to a slower convergence rate. Although all step size strategies effectively converge for smaller problems, the choice of the appropriate step size becomes crucial for larger problems. These experiments suggest that in Stiefel manifold optimization, the adaptive and Armijo step size methods are generally more effective than the diminishing step size method, and their use is recommended in practical applications.

Through all the experiments, we observed that as the iteration point approaches the optimal solution, the descent slows down and requires more iterations to reach optimality. To address this issue, we might consider using a threshold or combining our approach with

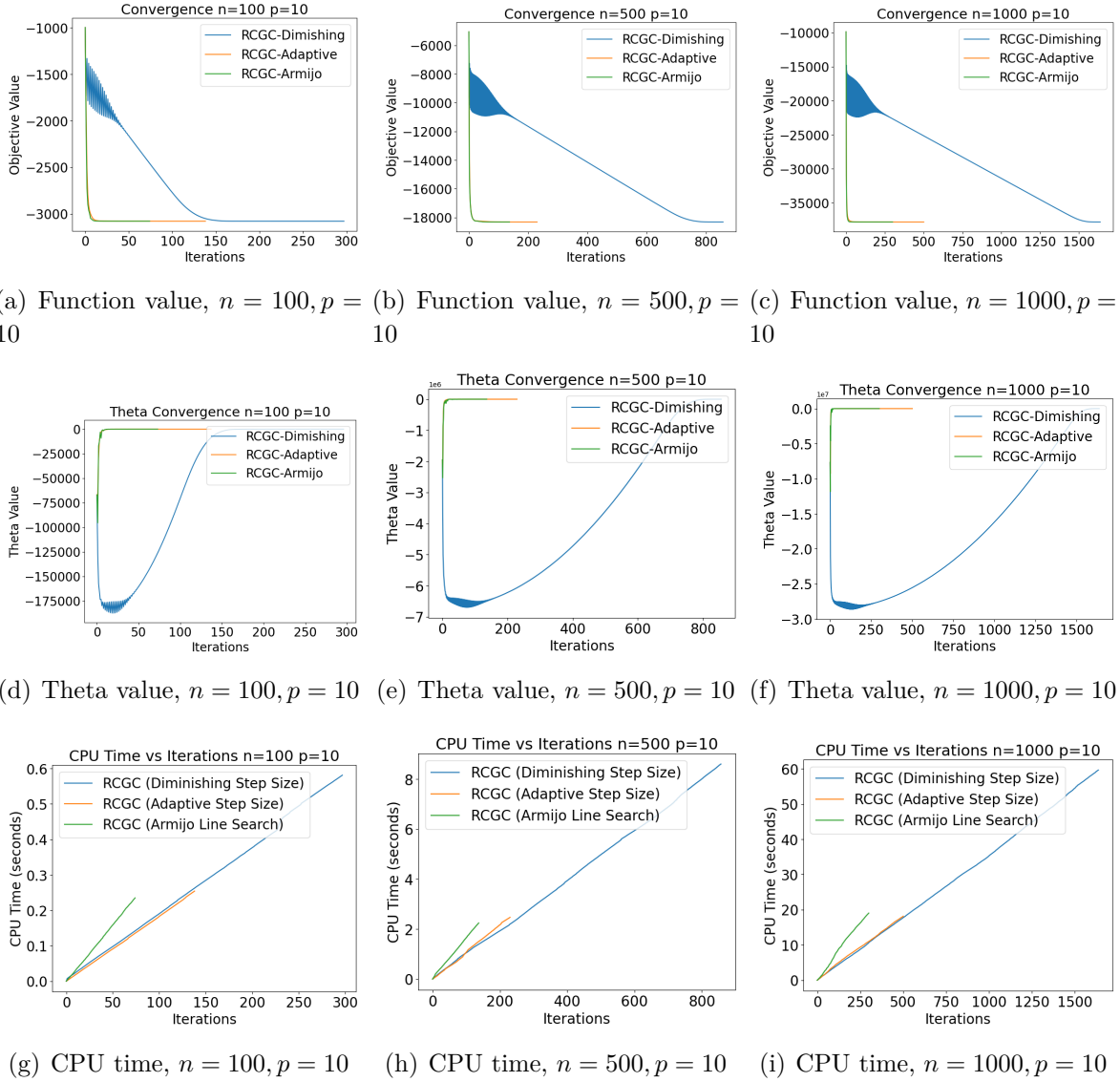


Figure 5.3: Example (5.7.4) with fixed A , $\lambda = 0.1$, different n and p

Table 5.5: Average results of 10 random runs for problem (5.7.4) with varying λ and fixed A , $n = 200$, $p = 10$

λ	Step size	Time (seconds)	Iterations
0.1	Armijo	0.6839	103.3
	Adaptive	0.5516	171.9
	Diminishing	1.3375	409.9
0.3	Armijo	1.0916	166.6
	Adaptive	0.6786	206.7
	Diminishing	1.4409	442.2
0.5	Armijo	0.9410	136.0
	Adaptive	0.5557	167.9
	Diminishing	1.3503	402.8
1.0	Armijo	0.9943	147.8
	Adaptive	0.5600	170.1
	Diminishing	1.7331	514.4

other methods to improve convergence.

5.8 Conclusion

In this chapter, we proposed novel Generalized Conditional Gradient (GCG) methods designed for composite optimization on Riemannian manifolds. We systematically investigated adaptive, diminishing, and Armijo-based step-size strategies, establishing their theoretical guarantees without relying on specific retraction or transport operations. Specifically, we proved that the adaptive and diminishing strategies achieve a sublinear convergence rate of $\mathcal{O}(1/k)$. For the Armijo strategy, we derived an iteration complexity of $J + \mathcal{O}(1/\epsilon^2)$, which improves to $\mathcal{O}(1/\epsilon^2)$ under the assumption that the non-smooth term g is Lipschitz continuous. This complexity corresponds to a convergence rate of $\mathcal{O}(1/\sqrt{k})$. Furthermore, we introduced a specialized algorithm to address non-convex subproblems, thereby significantly broadening the applicability of the proposed framework.

Our numerical experiments, particularly on the Stiefel manifold, showed that the adaptive step-size strategy consistently outperformed the diminishing one. While theoretical analysis often relies on diminishing steps to ensure convergence, our empirical results suggest that adaptive strategies are more efficient in practice. This highlights the need for future research to bridge the gap between theoretical guarantees and practical performance.

Chapter 6

Conclusions and future works

In this thesis, we developed efficient algorithms for optimization problems that cannot be solved by classical Euclidean methods. Specifically, we addressed problems involving multiple objectives and manifold constraints. We proposed a series of first-order methods, analyzed their theoretical convergence, and demonstrated their practical performance.

The main results obtained in this thesis are summarized as follows:

- (a) **Multi-Objective Optimization with Relative Smoothness:** In Chapter 3, we have proposed a multi-objective proximal gradient method incorporating Bregman distances. By utilizing the framework of relative smoothness, we successfully relaxed the standard Lipschitz continuity assumption on the gradients, which is often too restrictive for many practical applications. We proved the global convergence of the algorithm to Pareto stationary points under both constant and backtracking step-size strategies, providing a robust tool for non-Lipschitz multi-objective problems.
- (b) **Matrix-Free Methods for Riemannian Vector Optimization:** In Chapter 4, we extended nonlinear conjugate gradient methods to the domain of vector optimization on Riemannian manifolds. A pivotal contribution of this work was the generalization of the Wolfe and Zoutendijk conditions to the Riemannian vector-valued setting. We established the global convergence for vector extensions of the Fletcher–Reeves, conjugate descent, and Dai–Yuan parameters. This work bridges the gap between manifold optimization and vector optimization, offering a memory-efficient solution for geometric vector problems.
- (c) **Riemannian Conditional Gradient for Composite Optimization:** In Chapter 5, we addressed the computational challenges associated with minimizing composite functions on high-dimensional manifolds. We developed Riemannian Generalized Conditional Gradient methods that circumvent the need for computationally intensive operations such as proximal mappings. We provided a comprehensive complexity analysis, establishing an $\mathcal{O}(1/k)$ convergence rate for adaptive and diminishing step sizes, and $\mathcal{O}(1/\sqrt{k})$ for the Armijo-based strategy, thereby validating the method’s efficiency for structured constraints.

The research presented in this thesis has established a foundation for solving complex optimization problems across multi-objective and Riemannian settings. However, several avenues remain open for further investigation to enhance the scalability, theoretical depth, and practical efficiency of these methods. Drawing from the conclusions of the preceding chapters, we identify four primary directions for future research.

- (a) **Acceleration of Riemannian Algorithms:** While we have established convergence rates for our proposed methods, these rates are generally sublinear. A significant opportunity lies in developing accelerated variants to achieve faster convergence.
 - Specifically for the Riemannian conditional gradient method (Chapter 5), adapting momentum-guided techniques (as seen in Euclidean works like Li et al. [95]) or weighted average gradient approaches [141] to the manifold setting could notably improve performance.
 - Similarly, incorporating Nesterov-type acceleration into the multi-objective proximal gradient framework (Chapter 3) remains a promising direction to achieve optimal $\mathcal{O}(1/k^2)$ rates for convex vector optimization problems.
- (b) **Theoretical Characterization of Retractions and Parameters:** There are open theoretical questions regarding the underlying geometric structures and algorithmic parameters:
 - For the Riemannian conditional gradient method, our analysis relies on the concept of R -convexity. A crucial theoretical task is to fully characterize the class of retractions that satisfy this assumption beyond the exponential map. Investigating retractions defined on totally retractive neighborhoods or those that are local diffeomorphisms could expand the theoretical applicability of the method.
 - For the Riemannian nonlinear conjugate gradient method (Chapter 4), while we proved global convergence for the FR, CD, and DY parameters, the theoretical guarantees for other classical parameters such as PRP and HS remain challenging without strong assumptions. Developing rigorous convergence proofs for other variants on manifolds is a natural next step.
- (c) **Refining Step-Size Strategies:** Our experiments in Chapter 5 show that theoretical bounds do not always reflect practical performance. For example, while the diminishing step-size rule is easier to analyze, the adaptive strategy performed better on the Stiefel manifold. Future work should investigate geometry-aware adaptive step sizes, such as those using local curvature, to align theoretical analysis with practical efficiency.

Bibliography

- [1] P.-A. ABSIL, C. G. BAKER, AND K. A. GALLIVAN, *Trust-region methods on Riemannian manifolds*, Foundations of Computational Mathematics, 7 (2007), pp. 303–330.
- [2] 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, vol. 5, IEEE, 2006, pp. V–V.
- [3] P.-A. ABSIL, R. MAHONY, AND R. SEPULCHRE, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, Princeton, NJ, 2008.
- [4] F. ALIMISIS, A. ORVIETO, G. BECIGNEUL, AND A. LUCCHI, *Momentum improves optimization on Riemannian manifolds*, in International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 1351–1359.
- [5] N. ANDREI ET AL., *Nonlinear Conjugate Gradient Methods for Unconstrained Optimization*, Springer, 2020.
- [6] M. A. T. ANSARY AND J. DUTTA, *A proximal gradient method for multi-objective optimization problems using Bregman functions*. Optimization Online, 5 2022.
- [7] P. B. ASSUNÇÃO, O. P. FERREIRA, AND L. F. PRUDENTE, *Conditional gradient method for multiobjective optimization*, Computational Optimization and Applications, 78 (2021), pp. 741–768.
- [8] P. B. ASSUNÇÃO, O. P. FERREIRA, AND L. F. PRUDENTE, *A generalized conditional gradient method for multiobjective composite optimization problems*, Optimization, (2023), pp. 1–31.
- [9] F. BACH, *Duality between subgradient and conditional gradient methods*, SIAM Journal on Optimization, 25 (2015), pp. 115–129.
- [10] C. G. BAKER, *Riemannian manifold trust-region methods with applications to eigenproblems*, The Florida State University, 2008.
- [11] C. G. BAKER, P.-A. ABSIL, AND K. A. GALLIVAN, *An implicit trust-region method on Riemannian manifolds*, IMA Journal of Numerical Analysis, 28 (2008), pp. 665–689.

- [12] H. H. BAUSCHKE, J. BOLTE, AND M. TEBoulLE, *A descent lemma beyond Lipschitz gradient continuity: first-order methods revisited and applications*, Mathematics of Operations Research, 42 (2017), pp. 330–348.
- [13] A. BECK, *First-Order Methods in Optimization*, SIAM, 2017.
- [14] A. BECK AND M. TEBoulLE, *Mirror descent and nonlinear projected subgradient methods for convex optimization*, Operations Research Letters, 31 (2003), pp. 167–175.
- [15] —, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 183–202.
- [16] Y. BELLO-CRUZ, J. G. MELO, AND R. V. SERRA, *A proximal gradient splitting method for solving convex vector optimization problems*, Optimization, 71 (2022), pp. 33–53.
- [17] G. BENTO AND J. CRUZ NETO, *A subgradient method for multiobjective optimization on Riemannian manifolds*, Journal of Optimization Theory and Applications, 159 (2013), pp. 125–137.
- [18] G. BENTO, J. X. DA CRUZ NETO, AND P. SANTOS, *An inexact steepest descent method for multicriteria optimization on Riemannian manifolds*, Journal of Optimization Theory and Applications, 159 (2013), pp. 108–124.
- [19] G. BENTO, O. P. FERREIRA, AND P. R. OLIVEIRA, *Unconstrained steepest descent method for multicriteria optimization on Riemannian manifolds*, Journal of Optimization Theory and Applications, 154 (2012), pp. 88–107.
- [20] G. C. BENTO, J. X. CRUZ NETO, G. LÓPEZ, A. SOUBEYRAN, AND J. C. O. SOUZA, *The proximal point method for locally Lipschitz functions in multiobjective optimization with application to the compromise problem*, SIAM Journal on Optimization, 28 (2018), pp. 1104–1120.
- [21] G. D. C. BENTO, J. X. DA CRUZ NETO, AND L. V. DE MEIRELES, *Proximal point method for locally lipschitz functions in multiobjective optimization of hadamard manifolds*, Journal of Optimization Theory and Applications, 179 (2018), pp. 37–52.
- [22] G. D. C. BENTO, O. P. FERREIRA, AND Y. R. L. PEREIRA, *Proximal point method for vector optimization on Hadamard manifolds*, Operations Research Letters, 46 (2018), pp. 13–18.
- [23] D. P. BERTSEKAS, *Nonlinear Programming*, Athena Scientific, 1999.

-
- [24] M. BINDER, J. MOOSBAUER, J. THOMAS, AND B. BISCHL, *Multi-objective hyperparameter tuning and feature selection using filter ensembles*, in Proceedings of the 2020 Genetic and Evolutionary Computation Conference, Association for Computing Machinery, 2020, pp. 471–479.
- [25] J. BOLTE, S. SABACH, M. TEBoulLE, AND Y. VAISBOURD, *First order methods beyond convexity and Lipschitz gradient continuity with applications to quadratic inverse problems*, SIAM Journal on Optimization, 28 (2018), pp. 2131–2151.
- [26] S. BONNABEL, *Stochastic gradient descent on Riemannian manifolds*, IEEE Transactions on Automatic Control, 58 (2013), pp. 2217–2229.
- [27] H. BONNEL, A. N. IUSEM, AND B. F. SVAITER, *Proximal methods in vector optimization*, SIAM Journal on Optimization, 15 (2005), pp. 953–970.
- [28] N. BOUMAL, *An Introduction to Optimization on Smooth Manifolds*, Cambridge University Press, 2023.
- [29] —, *An Introduction to Optimization on Smooth Manifolds*, Cambridge University Press, 2023.
- [30] N. BOUMAL AND P.-A. ABSIL, *RTRMC: A Riemannian trust-region method for low-rank matrix completion*, Advances in Neural Information Processing systems, 24 (2011).
- [31] N. BOUMAL, P.-A. ABSIL, AND C. CARTIS, *Global rates of convergence for nonconvex optimization on manifolds*, IMA Journal of Numerical Analysis, 39 (2019), pp. 1–33.
- [32] R. BOT AND S.-M. GRAD, *Inertial forward-backward methods for solving vector optimization problems*, Optimization, 67 (2018), pp. 1–16.
- [33] L. M. BREGMAN, *The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming*, USSR Computational Mathematics and Mathematical Physics, 7 (1967), pp. 200–217.
- [34] R. W. BROCKETT, *Differential geometry and the design of gradient algorithms. differential geometry*, Partial Differential Equations on Manifolds, (1993), pp. 69–92.
- [35] G. A. CARRIZO, P. A. LOTITO, AND M. C. MACIEL, *Trust region globalization strategy for the nonconvex unconstrained multiobjective optimization problem*, Mathematical Programming, 159 (2016), pp. 339–369.
- [36] G. CHEN AND M. TEBoulLE, *Convergence analysis of a proximal-like minimization algorithm using Bregman functions*, SIAM Journal on Optimization, 3 (1993), pp. 538–543.

- [37] J. CHEN, Y. BAI, G. YU, X. OU, AND X. QIN, *A PRP type conjugate gradient method without truncation for nonconvex vector optimization*, Journal of Optimization Theory and Applications, 204 (2025), p. 13.
- [38] J. CHEN, L. TANG, AND X. YANG, *Convergence rates analysis of interior Bregman gradient method for vector optimization problems*, 6 2022.
- [39] K. CHEN AND E. H. FUKUDA, *Riemannian conditional gradient methods for composite optimization problems*, Computational and Applied Mathematics, 45 (2026), p. 151.
- [40] K. CHEN, E. H. FUKUDA, AND H. SATO, *Nonlinear conjugate gradient method for vector optimization on Riemannian manifolds with retraction and vector transport*, Applied Mathematics and Computation, 486 (2025), p. 129001.
- [41] K. CHEN, E. H. FUKUDA, AND N. YAMASHITA, *A proximal gradient method with Bregman distance in multi-objective optimization*. Master thesis, Kyoto University, 2022. Available at http://www.amp.i.kyoto-u.ac.jp/tecrep/ps_file/2022/2022-002.pdf.
- [42] K. CHEN, E. H. FUKUDA, AND N. YAMASHITA, *A proximal gradient method with Bregman distance in multi-objective optimization*, Pacific Journal of Optimization, 20 (2024), pp. 809–826.
- [43] S. CHEN, S. MA, A. MAN-CHO SO, AND T. ZHANG, *Proximal gradient method for nonsmooth optimization over the Stiefel manifold*, SIAM Journal on Optimization, 30 (2020), pp. 210–239.
- [44] W. CHEN, X. YANG, AND Y. ZHAO, *Conditional gradient method for vector optimization*, Computational Optimization and Applications, 85 (2023), pp. 857–896.
- [45] Z. CHEN, X. HUANG, AND X. YANG, *Generalized proximal point algorithms for multiobjective optimization problems*, Applicable Analysis, 90 (2011), pp. 935–949.
- [46] A. CHERIAN AND S. SRA, *Riemannian dictionary learning and sparse coding for positive definite matrices*, IEEE Transactions on Neural Networks and Learning Systems, 28 (2016), pp. 2859–2871.
- [47] G. COCCHI AND M. LAPUCCI, *An augmented Lagrangian algorithm for multi-objective optimization*, Computational Optimization and Applications, 77 (2020), pp. 29–56.
- [48] J. X. CRUZ NETO, G. J. P. SILVA, O. P. FERREIRA, AND J. O. LOPES, *A subgradient method for multiobjective optimization*, Computational Optimization and Applications, 54 (2013), pp. 461–472.
- [49] Y. DAI, J. HAN, G. LIU, D. SUN, H. YIN, AND Y.-X. YUAN, *Convergence properties of nonlinear conjugate gradient methods*, SIAM Journal on Optimization, 10 (2000), pp. 345–358.

- [50] N. DALMASSO, R. ZHAO, M. GHASSEMI, V. POTLURU, T. BALCH, AND M. VELOSO, *Efficient event series data modeling via first-order constrained optimization*, in Proceedings of the Fourth ACM International Conference on AI in Finance, 2023, pp. 463–471.
- [51] K. DEB, A. PRATAP, S. AGARWAL, AND T. MEYARIVAN, *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Transactions on Evolutionary Computation, 6 (2002), pp. 182–197.
- [52] A. DOUIK, F. SALEHI, AND B. HASSIBI, *A novel Riemannian optimization approach and algorithm for solving the phase retrieval problem*, in 2019 53rd Asilomar Conference on Signals, Systems, and Computers, IEEE, 2019, pp. 1962–1966.
- [53] S.-Q. DUAN, X.-F. DUAN, C.-M. LI, AND J.-F. LI, *Riemannian conjugate gradient method for low-rank tensor completion*, Advances in Computational Mathematics, 49 (2023), p. 41.
- [54] O. P. FERREIRA, M. S. LOUZEIRO, AND L. PRUDENTE, *Gradient method for optimization on Riemannian manifolds with lower bounded curvature*, SIAM Journal on Optimization, 29 (2019), pp. 2517–2541.
- [55] O. P. FERREIRA, M. S. LOUZEIRO, AND L. F. PRUDENTE, *Iteration-complexity of the subgradient method on Riemannian manifolds with lower bounded curvature*, Optimization, 68 (2019), pp. 713–729.
- [56] O. P. FERREIRA, L. R. LUCAMBIO PÉREZ, AND L. F. PRUDENTE. Personal communication, 2019.
- [57] R. FLECHER AND C. REEVES, *Function minimization by conjugate gradient*, Computer Journal, 7 (1964), pp. 149–154.
- [58] R. FLETCHER, *Practical Methods of Optimization: Unconstrained optimization*, John Wiley & Sons, 1980.
- [59] J. FLIEGE, L. M. GRAÑA DRUMMOND, AND B. F. SVAITER, *Newton’s method for multiobjective optimization*, SIAM Journal on Optimization, 20 (2009), pp. 602–626.
- [60] J. FLIEGE AND B. F. SVAITER, *Steepest descent methods for multicriteria optimization*, Mathematical Methods of Operations Research, 51 (2000), pp. 479–494.
- [61] M. FRANK AND P. WOLFE, *An algorithm for quadratic programming*, Naval Research Logistics, 3 (1956), pp. 95–110.
- [62] E. H. FUKUDA AND L. M. GRAÑA DRUMMOND, *On the convergence of the projected gradient method for vector optimization*, Optimization, 60 (2011), pp. 1009–1021.

- [63] —, *Inexact projected gradient method for vector optimization*, Computational Optimization and Applications, 54 (2013), pp. 473–493.
- [64] —, *A survey on multiobjective descent methods*, Pesquisa Operacional, 34 (2014), pp. 585–620.
- [65] M. FUKUSHIMA, *A modified Frank-Wolfe algorithm for solving the traffic assignment problem*, Transportation Research Part B: Methodological, 18 (1984), pp. 169–177.
- [66] M. FUKUSHIMA AND H. MINE, *A generalized proximal point algorithm for certain non-convex minimization problems*, International Journal of Systems Science, 12 (1981), pp. 989–1000.
- [67] O. GANEA, G. BÉCIGNEUL, AND T. HOFMANN, *Hyperbolic neural networks*, Advances in Neural Information Processing Systems, 31 (2018).
- [68] B. GAO AND P.-A. ABSIL, *A Riemannian rank-adaptive method for low-rank matrix completion*, Computational Optimization and Applications, 81 (2022), pp. 67–90.
- [69] A. G. GEBRIE AND E. H. FUKUDA, *Adaptive generalized conditional gradient method for multiobjective optimization*, Journal of Optimization Theory and Applications, 206 (2025), p. 13.
- [70] A. M. GEOFFRION, *Proper efficiency and the theory of vector maximization*, Journal of Mathematical Analysis and Applications, 22 (1968), pp. 618–630.
- [71] J. C. GILBERT AND J. NOCEDAL, *Global convergence properties of conjugate gradient methods for optimization*, SIAM Journal on optimization, 2 (1992), pp. 21–42.
- [72] D. S. GONÇALVES, M. L. GONÇALVES, AND J. G. MELO, *Improved convergence rates for the multiobjective frank-wolfe method*, Journal of Optimization Theory and Applications, 205 (2025), p. 20.
- [73] M. L. GONÇALVES, F. LIMA, AND L. PRUDENTE, *A study of Liu-Storey conjugate gradient methods for vector optimization*, Applied Mathematics and Computation, 425 (2022), p. 127099.
- [74] M. L. N. GONÇALVES AND L. F. PRUDENTE, *On the extension of the Hager-Zhang conjugate gradient method for vector optimization*, Computational Optimization and Applications, 76 (2020), pp. 889–916.
- [75] M. GONG, M. ZHANG, AND Y. YUAN, *Unsupervised band selection based on evolutionary multiobjective optimization for hyperspectral images*, IEEE Transactions on Geoscience and Remote Sensing, 54 (2016), pp. 544–557.
- [76] S.-M. GRAD, *A Survey on Proximal Point Type Algorithms for Solving Vector Optimization Problems*, Springer International Publishing, 2019, pp. 269–308.

-
- [77] L. M. GRAÑA DRUMMOND AND A. N. IUSEM, *A projected gradient method for vector optimization problems*, Computational Optimization and Applications, 28 (2004), pp. 5–29.
 - [78] L. M. GRAÑA DRUMMOND AND A. N. IUSEM, *A projected gradient method for vector optimization problems*, Computational Optimization and Applications, 28 (2004), pp. 5–29.
 - [79] L. M. GRAÑA DRUMMOND AND B. F. SVAITER, *A steepest descent method for vector optimization*, Journal of Computational and Applied Mathematics, 175 (2005), pp. 395–414.
 - [80] E. HAZAN AND S. KALE, *Projection-free online learning*, arXiv preprint arXiv:1206.4657, (2012).
 - [81] M. R. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems*, Journal of Research of the National Bureau of Standards, 49 (1952), pp. 409–435.
 - [82] M. R. HESTENES, E. STIEFEL, ET AL., *Methods of conjugate gradients for solving linear systems*, vol. 49, NBS Washington, DC, 1952.
 - [83] F. HUANG, J. LI, S. GAO, AND H. HUANG, *Enhanced bilevel optimization via Bregman distance*, Advances in Neural Information Processing Systems, 35 (2022), pp. 28928–28939.
 - [84] W. HUANG, *Optimization algorithms on Riemannian manifolds with applications*, PhD thesis, The Florida State University, 2013.
 - [85] W. HUANG, K. A. GALLIVAN, AND X. ZHANG, *Solving phase-lift by low-rank Riemannian optimization methods for complex semidefinite constraints*, SIAM Journal on Scientific Computing, 39 (2017), pp. B840–B859.
 - [86] W. HUANG AND K. WEI, *Riemannian proximal gradient methods*, Mathematical Programming, 194 (2022), pp. 371–413.
 - [87] Z. HUANG AND L. VAN GOOL, *A Riemannian network for SPD matrix learning*, in Proceedings of the AAAI conference on artificial intelligence, vol. 31, 2017.
 - [88] Z. HUANG, J. WU, AND L. VAN GOOL, *Building deep networks on Grassmann manifolds*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, 2018.
 - [89] M. JAGGI, *Revisiting Frank-Wolfe: Projection-free sparse convex optimization*, in International Conference on Machine Learning, PMLR, 2013, pp. 427–435.

- [90] J. JAHN, *Vector Optimization: Theory, Applications, and Extensions*, Springer, 01 2004.
- [91] D. S. KALHAN, A. S. BEDI, A. KOPPEL, K. RAJAWAT, H. HASSANI, A. K. GUPTA, AND A. BANERJEE, *Dynamic online learning via Frank-Wolfe algorithm*, IEEE Transactions on Signal Processing, 69 (2021), pp. 932–947.
- [92] S. LACOSTE-JULIEN, *Convergence rate of Frank-Wolfe for non-convex objectives*, arXiv preprint arXiv:1607.00345, (2016).
- [93] S. LACOSTE-JULIEN AND M. JAGGI, *On the global linear convergence of Frank-Wolfe optimization variants*, Advances in Neural Information Processing Systems, 28 (2015).
- [94] S. LACOSTE-JULIEN, M. JAGGI, M. SCHMIDT, AND P. PLETSCHER, *Block-coordinate Frank-Wolfe optimization for structural SVMs*, in International Conference on Machine Learning, PMLR, 2013, pp. 53–61.
- [95] B. LI, M. COUTINO, G. B. GIANNAKIS, AND G. LEUS, *A momentum-guided Frank-Wolfe algorithm*, IEEE Transactions on Signal Processing, 69 (2021), pp. 3597–3611.
- [96] P.-L. LIONS AND B. MERCIER, *Splitting algorithms for the sum of two nonlinear operators*, SIAM Journal on Numerical Analysis, 16 (1979), pp. 964–979.
- [97] Y. LIU AND C. STOREY, *Efficient generalized conjugate gradient algorithms, part 1: theory*, Journal of Optimization Theory and Applications, 69 (1991), pp. 129–137.
- [98] H. LU, R. M. FREUND, AND Y. NESTEROV, *Relatively smooth convex optimization by first-order methods, and applications*, SIAM Journal on Optimization, 28 (2018), pp. 333–354.
- [99] L. LUCAMBIO PÉREZ AND L. PRUDENTE, *Nonlinear conjugate gradient methods for vector optimization*, SIAM Journal on Optimization, 28 (2018), pp. 2690–2720.
- [100] L. R. LUCAMBIO PÉREZ AND L. F. PRUDENTE, *Nonlinear conjugate gradient methods for vector optimization*, SIAM Journal on Optimization, 28 (2018), pp. 2690–2720.
- [101] I. LUCHNIKOV, A. RYZHOV, S. FILIPPOV, AND H. OUERDANE, *QGOpt: Riemannian optimization for quantum technologies*, Scipost physics, 10 (2021), p. 079.
- [102] D. G. LUENBERGER, *The gradient projection method along geodesics*, Management Science, 18 (1972), pp. 620–631.
- [103] D. G. LUENBERGER, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1973.

- [104] G. MAVROTAS, *Effective implementation of the ε -constraint method in multi-objective mathematical programming problems*, Applied mathematics and computation, 213 (2009), pp. 455–465.
- [105] J. J. MOREAU, *Fonctions convexes duales et points proximaux dans un espace hilbertien*, Comptes rendus hebdomadaires des séances de l'Académie des sciences, 255 (1962), pp. 2897–2899.
- [106] J.-J. MOREAU, *Proximité et dualité dans un espace Hilbertien*, Bulletin de la Société mathématique de France, 93 (1965), pp. 273–299.
- [107] V. MOROVATI, H. BASIRZADEH, AND L. POURKARIMI, *Quasi-Newton methods for multiobjective optimization problems*, 4OR, 16 (2018), pp. 261–294.
- [108] S. NAJAFI AND M. HAJARIAN, *Multiobjective conjugate gradient methods on Riemannian manifolds*, Journal of Optimization Theory and Applications, 197 (2023), pp. 1229–1248.
- [109] Y. NESTEROV, *A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$* , in Dokl. Akad. Nauk. SSSR, vol. 269, 1983, p. 543.
- [110] Y. NISHIMURA, E. H. FUKUDA, AND N. YAMASHITA, *Monotonicity for multiobjective accelerated proximal gradient methods*, Journal of the Operations Research Society of Japan, 67 (2024), pp. 1–17.
- [111] G. B. PASSTY, *Ergodic convergence to a zero of the sum of monotone operators in hilbert space*, Journal of Mathematical Analysis and Applications, 72 (1979), pp. 383–390.
- [112] J.-W. PENG, J.-W. ZHANG, AND J.-C. YAO, *A novel hybrid conjugate gradient method for multiobjective optimization problems*, Optimization, (2024), pp. 1–20.
- [113] E. POLAK AND G. RIBIERE, *Note sur la convergence de méthodes de directions conjuguées*, Revue française d'informatique et de recherche opérationnelle. Série rouge, 3 (1969), pp. 35–43.
- [114] B. T. POLYAK, *The conjugate gradient method in extremal problems*, USSR Computational Mathematics and Mathematical Physics, 9 (1969), pp. 94–112.
- [115] M. J. POWELL, *Nonconvex minimization calculations and the conjugate gradient method*, in Numerical Analysis: Proceedings of the 10th Biennial Conference held at Dundee, Scotland, June 28–July 1, 1983, Springer, 1984, pp. 122–141.
- [116] Y. QI, X. MA, F. LIU, L. JIAO, J. SUN, AND J. WU, *MOEA/D with adaptive weight adjustment*, Evolutionary computation, 22 (2014), pp. 231–264.

- [117] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, 1970.
- [118] H. SAKAI AND H. IIDUKA, *Sufficient descent Riemannian conjugate gradient methods*, Journal of Optimization Theory and Applications, 190 (2021), pp. 130–150.
- [119] H. SAKAI, H. SATO, AND H. IIDUKA, *Global convergence of Hager–Zhang type Riemannian conjugate gradient method*, Applied Mathematics and Computation, 441 (2023), p. 127685.
- [120] H. SATO, *Riemannian Optimization and Its Applications*, Springer, 2021. OCLC: 1237966923.
- [121] ———, *Riemannian conjugate gradient methods: General framework and specific algorithms with convergence analyses*, SIAM Journal on Optimization, 32 (2022), pp. 2690–2717.
- [122] H. SATO AND T. IWAI, *A Riemannian optimization approach to the matrix singular value decomposition*, SIAM Journal on Optimization, 23 (2013), pp. 188–212.
- [123] ———, *A new, globally convergent Riemannian conjugate gradient method*, Optimization, 64 (2015), pp. 1011–1031.
- [124] H. SATO, H. KASAI, AND B. MISHRA, *Riemannian stochastic variance reduced gradient algorithm with retraction and vector transport*, SIAM Journal on Optimization, 29 (2019), pp. 1444–1472.
- [125] H. SATO AND K. SATO, *Riemannian trust-region methods for h^2 optimal model reduction*, in 2015 54th IEEE Conference on Decision and Control (CDC), IEEE, 2015, pp. 4648–4655.
- [126] K. SATO AND H. SATO, *Structure-preserving h^2 optimal model reduction based on the Riemannian trust-region method*, IEEE Transactions on Automatic Control, 63 (2017), pp. 505–512.
- [127] J. C. O. SOUZA, J. X. CRUZ NETO, J. O. LOPES, R. C. M. SILVA, AND S. D. B. BITAR, *The proximal point method with a vectorial Bregman regularization in multiobjective DC programming*, Optimization, 71 (2022), pp. 263–284.
- [128] A. SRIVASTAVA, E. KLASSEN, S. H. JOSHI, AND I. H. JERMYN, *Shape analysis of elastic curves in euclidean spaces*, IEEE transactions on pattern analysis and machine intelligence, 33 (2010), pp. 1415–1428.
- [129] H. TANABE, E. H. FUKUDA, AND N. YAMASHITA, *Proximal gradient methods for multiobjective optimization and their applications*, Computational Optimization and Applications, 72 (2019), pp. 339–361.

-
- [130] —, *Convergence rates analysis of a multiobjective proximal gradient method*, Optimization Letters, 17 (2023), pp. 333–350.
 - [131] —, *New merit functions for multiobjective optimization and their properties*, Optimization, (2023), p. to appear.
 - [132] M. TEBoulLE, *A simplified view of first order methods for optimization*, Mathematical Programming, 170 (2018), pp. 67–96.
 - [133] F. J. THEIS, T. P. CASON, AND P.-A. ABSIL, *Soft dimension reduction for ica by joint diagonalization on the Stiefel manifold*, in International Conference on Independent Component Analysis and Signal Separation, Springer, 2009, pp. 354–361.
 - [134] B. VANDEREYCKEN, *Low-rank matrix completion by Riemannian optimization*, SIAM Journal on Optimization, 23 (2013), pp. 1214–1236.
 - [135] S. VERMA, M. PANT, AND V. SNASEL, *A comprehensive review on NSGA-II for multi-objective combinatorial optimization problems*, IEEE access, 9 (2021), pp. 57757–57791.
 - [136] H. WANG AND A. BANERJEE, *Bregman alternating direction method of multipliers*, in Advances in Neural Information Processing Systems, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds., vol. 27, 2014.
 - [137] M. WEBER AND S. SRA, *Riemannian optimization via Frank-Wolfe methods*, Mathematical Programming, 199 (2023), pp. 525–556.
 - [138] J. YAHAYA, P. KUMAM, A. BELLO, AND K. SITTHITHAKERNGKIET, *On the Dai-Liao conjugate gradient method for vector optimization*, Optimization, 74 (2025), pp. 4137–4167.
 - [139] Y. YU, X. ZHANG, AND D. SCHUURMANS, *Generalized conditional gradient for sparse estimation*, Journal of Machine Learning Research, 18 (2017), pp. 1–46.
 - [140] Q. ZHANG AND H. LI, *Moea/d: A multiobjective evolutionary algorithm based on decomposition*, IEEE Transactions on evolutionary computation, 11 (2007), pp. 712–731.
 - [141] Y. ZHANG, B. LI, AND G. B. GIANNAKIS, *Accelerating Frank-Wolfe with weighted average gradients*, in ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2021, pp. 5529–5533.
 - [142] X. ZHU, *A Riemannian conjugate gradient method for optimization on the Stiefel manifold*, Computational Optimization and Applications, 67 (2017), pp. 73–110.
 - [143] X. ZHU AND C. SHEN, *Practical gradient and conjugate gradient methods on flag manifolds*, Computational Optimization and Applications, 88 (2024), pp. 491–524.