Large-Scale Optimization Methods: Theory and Applications

by

Nuri Denizcan Vanli

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2021

Author . . . .
Department of Electrical Engineering and Computer Science
August 27, 2021

Certified by . . . .
Asuman Ozdaglar
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by . . . .
Pablo A. Parrilo
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Large-Scale Optimization Methods: Theory and Applications

by

Nuri Denizcan Vanli

Submitted to the Department of Electrical Engineering and Computer Science
on August 27, 2021, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

Large-scale optimization problems appear quite frequently in data science and machine learning applications. In this thesis, we show the efficiency of coordinate descent (CD) and mirror descent (MD) methods in solving large-scale optimization problems.

First, we investigate the convergence rate of the CD method with different coordinate selection rules. We present certain problem classes, for which deterministic rules provably outperform randomized rules. We quantify the amount of improvement and the corresponding deterministic order that achieves the maximum improvement. We then show that for a certain subclass of problems, using any fixed deterministic rule yields a superior performance than using random permutations. Then, we illustrate the efficiency of the CD method on a constrained non-convex optimization problem that arise from semidefinite programming with diagonal constraints. We show that the proposed CD methods can recover the optimal solution when the rank of the factorization is sufficiently large, and establish the rate of convergence. When the rank of the factorization is small, we provide tight approximation bounds as a function of the rank.

Next, we study convergence properties of the continuous-time and discrete-time MD methods. We present a unified convergence theory for mirror descent and related methods. Then, we establish the implicit bias of the MD method with non-differentiable distance generating functions. Finally, we introduce the continuous-time MD method with non-differentiable and non-strictly convex distance generating functions. We show the existence and convergence of the solutions generated by the MD method and establish their implicit bias. We illustrate that the combinatorial algorithms resulting from this approach can be used to solve sparse optimization problems.

Thesis Supervisor: Asuman Ozdaglar
Title: Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Pablo A. Parrilo
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I would like to express my deepest gratitude to my advisers Asu Ozdaglar and Pablo Parrilo as well as my committee member Suvrit Sra.

Asu, I cannot thank you enough for your support throughout my doctoral studies. Thank you for always finding time to meet with me despite your busy schedule. Thank you for your enthusiasm, for exposing me to interesting research problems and teaching me how to communicate my research. Working with you has truly been a joy and I am grateful to have had the opportunity.

Pablo, working with you has been a wonderful experience. There has not been a single meeting that I was not impressed by your knowledge and intuition. Thank you for guiding my research by asking stimulating questions.

Suvrit, thank you for discussing research problems and directions with me, interacting with you has been invaluable.

Most of this thesis would not have come to fruition if it was not for my amazing collaborators: Mert and Murat. I am thankful that I have had the pleasure to work with you, you have taught me so much. I will always cheerish our fun conversations and I cannot thank you enough for your guidance on my career directions.

I would like to extend my gratitude to NSF and Draper for funding my research and to the LIDS staff, and in particular Roxana Hernandez, Jennifer Donovan, Lynne Dell and Brian Jones for their help in numerous administrative tasks. Special thanks to Roxana for always creating an opening in Asu's calendar for our meetings, it has remained a mystery to me how she managed to do so.

I am thankful to the LIDS/MIT community at large for making my time here so enjoyable. I have been fortunate to have had many friends at MIT and for that I am thankful to Asu's and Pablo's group, Dennis Shen, Matthew Staib, Matthew Brennan[1], Zhi Xu, Igor Kadota and so many more. Special thanks to Jason, James and Jackie for many fun

[1] Rest in peace.

memories[2].

I have been fortunate to have Seyhmus as my roommate for many years, who has made my life at Boston incredibly fun. Of course, the past several years would not have been nearly as enjoyable if it was not for the company of Ozge with whom I shared many great memories.

Throughout my life, I have been spoiled by the love and support of two amazing individuals that I look up to. Mom and Dad, this thesis is dedicated to you.

[2] I also extend thanks to Sazerac Company for producing Fireball, which may or may not have caused us to forget some great memories.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In the last few decades, machine learning has become increasingly prevalent to learn explanatory models of the world. Continuous optimization methods, as a tool for learning and solving machine learning models, usually determine the computational bottlenecks for the size of the models and problems that can be solved. With the ever-increasing amount of data available to process, the number of applications that are cast as large-scale optimization problems are growing dramatically. Therefore, there is a significant interest in developing simple continuous optimization methods that require low iteration cost and low memory storage, so that they scale better with the size of the problems. Although such methods need not converge to high-accuracy solutions fast, low-accuracy solutions are sufficient for machine learning applications as the data is usually noisy.

One of the most celebrated methods that is acknowledged for its simplicity is the coordinate descent (CD) method. The CD method is a classical optimization algorithm that has seen a revival of interest because of its competitive performance in machine learning applications. The CD method is particularly convenient for large-scale optimization problems since updating a single variable (or a block of variables in the case of block-CD) at each iteration of the algorithm is significantly cheaper than updating all variables simultaneously. Consequently, the CD method has been successfully applied to a variety of large-scale optimization problems in the literature, such as lasso, support vector machines and optimal transport.

Another method that is particularly well-suited for large-scale optimization problems is the mirror descent (MD) method. The MD method is a first-order optimization algorithm that generalizes the gradient descent method to non-Euclidean geometries via distance generating functions that are specific to the desired geometry. Thus, by changing the Euclidean geometry to a more pertinent geometry to the problem at hand, the MD method enjoys almost dimension-free convergence rates, which makes the MD method extremely useful in large-scale optimization problems.

In this thesis, we present several algorithmic and theoretic contributions to the CD and MD methods. Our main purposes are to obtain a better understanding of the performance of existing optimization methods and to develop algorithms that can efficiently solve certain large-scale optimization problems. There are five fundamental questions we ask, each of which we study in one corresponding section in the thesis:

1. When does the CD method with randomized coordinate selection rule outperform the CD method with deterministic coordinate selection rule?

2. Does randomly permuted coordinate selection fix the worst-case behavior of uniformly random coordinate selection in the CD method?

3. Can the CD method be efficiently applied to solve large-scale non-convex optimization problems of certain kind with precise convergence and approximation guarantees?

4. How does the MD method relate to existing optimization methods in the literature and is there a unified lens through which they can be understood and analyzed?

5. Can the MD method be extended to more general geometries that are not necessarily smooth?

1.2 Thesis Outline

The first part of the thesis focuses on the CD method. In Chapter 3, we provide a background on the CD method. In Chapter 4, which is based on [72], we study the CD method with deterministic and randomized update rules. We present problem classes for which the CD method with any cyclic order is faster than the CD method with randomized coordinate selection in terms of asymptotic worst-case convergence. Then in Chapter 5, which is based on [73], we show that using random permutations instead of random with-replacement sampling improves the performance of the CD method in the worst-case. In Chapter 6, which is based on [61], we consider applying the CD method to the non-convex optimization problem that arises from low-rank factorization to semidefinite programs with diagonal constraints. We establish global sublinear convergence and local linear convergence of the CD method. We then develop a method based on the CD and Lanczos methods that returns an approximately globally optimal solution.

The second part of the thesis focuses on the MD method. In Chapter 7, we provide a background on the MD method. In Chapter 8, we present a unified approach to analyze several optimization methods including MD, dual averaging, Bregman proximal gradient and Bregman proximal point. We apply the presented methodology to two problem classes and systematically recover the celebrated rate estimates for the aforementioned methods often under weaker assumptions. In Chapter 9, we develop the continuous-time MD method that generalize the existing MD method to non-smooth geometries. We investigate the convergence properties of the corresponding continuous-time inclusion and discuss how to discretize it. Finally, we show the efficiency of the resulting method for a few celebrated problems.

Chapter 2

Background

In this section, we provide some definitions and basic results on convex analysis. Our presentation largely follows [128, 130, 10] and we refer to these books for a more detailed treatment of convex analysis, variational analysis and set-valued analysis, respectively.

2.1 Notation

Unless stated otherwise, all vectors are column vectors and represented by lowercase letters. Matrices are represented by uppercase letters, scalars are represented by lowercase Greek letters, and sets are represented by uppercase Greek letters. Superscripts are used to represent iteration counters, whereas subscripts are used to represent coordinates for a vector and columns for a matrix. $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}_{\geq}$ denotes the set of non-negative real numbers and $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$ denotes the set of extended real numbers. $[n]$ denotes the set of positive integers up to and including n.

For a vector x, $\|x\|_p$ represents its ℓ_p-norm. For matrices A, B, we write $\langle A, B\rangle = \text{trace}(AB^\top)$ for the inner product associated to the Frobenius norm $\|A\|_{\text{F}} = \sqrt{\langle A, A\rangle}$. For a matrix A, $\|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^n |A_{ij}|$ represents its 1-norm, $\|A\|_{1,1} = \sum_{i,j=1}^n |A_{ij}|$ represents its $L_{1,1}$-norm, and $\|A\|_* = \text{tr}(\sqrt{A^\top A})$ represents its nuclear norm. $\mathcal{B}_{2,n} = \{x \in \mathbb{R}^n : \|x\|_2 \leq 1\}$ denotes the Euclidean unit ball in $\mathbb{R}^n$. For matrices, $\geq$ and $\leq$ are entry-wise operators. Matrices I and 0 denote the identity matrix and the zero matrix respectively and their dimensions can be understood from the context.

2.2 Convex Sets

The indicator function of a set $\mathcal{X}$ is denoted by

$$\iota_{\mathcal{X}}(x) = \begin{cases} 0 & \text{if } x \in \mathcal{X}, \\ +\infty & \text{otherwise.} \end{cases}$$

A subset $\mathcal{X}$ of $\mathbb{R}^n$ is said to be

- *convex* if $(1-\lambda)x + \lambda y \in \mathcal{X}$ whenever $x \in \mathcal{X}$, $y \in \mathcal{X}$ and $0 < \lambda < 1$.
- *closed* if it contains all its limit points.

For a set $\mathcal{X} \subset \mathbb{R}^n$:

- its *normal cone* is denoted by $\mathcal{N}_X$ and defined as

$$\mathcal{N}_{\mathcal{X}}(x) = \begin{cases} \{y \in \mathbb{R}^n : \langle y, u - x \rangle \leq 0, \, \forall u \in \mathcal{X}\} & \text{if } x \in \mathcal{X} \\ \emptyset & \text{if } x \notin \mathcal{X} \end{cases}$$

- its *affine hull* is denoted by $\operatorname{aff} \mathcal{X}$ and defined as

$$\operatorname{aff} \mathcal{X} = \left\{ \sum_{i=1}^{m} \lambda_i x_i \,\middle|\, m > 0, \, x_i \in \mathcal{X}, \, \lambda_i \in \mathbb{R}, \, \sum_{i=1}^{m} \lambda_i = 1 \right\}.$$

- its *closure* is denoted by $\operatorname{cl} \mathcal{X}$ and defined as

$$\operatorname{cl} \mathcal{X} = \bigcap \{\mathcal{X} + \epsilon \mathcal{B}_{2,n} : \epsilon > 0\}.$$

- its *interior* is denoted by $\operatorname{int}\mathcal{X}$ and defined as

$$\operatorname{int}\mathcal{X} = \{x : \exists\epsilon > 0,\, x + \epsilon\mathcal{B}_{2,n} \subset \mathcal{X}\}.$$

- its *relative interior* is denoted by $\operatorname{ri}\mathcal{X}$ and defined as

$$\operatorname{ri}\mathcal{X} = \{x \in \operatorname{aff}\mathcal{X} : \exists\epsilon > 0,\, (x + \epsilon\mathcal{B}_{2,n}) \cap (\operatorname{aff}\mathcal{X}) \subset \mathcal{X}\}.$$

2.3 Convex Functions and Conjugates

Let $f : \mathcal{X} \to \bar{\mathbb{R}}$ be a function where $\mathcal{X} \subset \mathbb{R}^n$. The *graph* of f is defined to be the set:

$$\operatorname{graph} f = \{(x, y) \in \mathcal{X} \times \bar{\mathbb{R}} \,:\, y = f(x)\}.$$

The *epigraph* of f is defined to be the set:

$$\operatorname{epi} f = \{(x, y) \in \mathcal{X} \times \bar{\mathbb{R}} \,:\, y \geq f(x)\}.$$

A function f is said to be *convex* on $\mathcal{X}$ if $\operatorname{epi} f$ is convex as a subset of $\bar{\mathbb{R}}^{n+1}$. The *effective domain* of a convex function f, denoted by $\operatorname{dom} f$, is the projection of $\operatorname{epi} f$ on $\mathbb{R}^n$:

$$\operatorname{dom} f = \{x \,:\, \exists y,\, (x, y) \in \operatorname{epi} f\} = \{x \,:\, f(x) < +\infty\}.$$

Convex functions are continuous on $\operatorname{int}\operatorname{dom} f$ and differentiable on $\operatorname{int}\operatorname{dom} f$ except for a set of measure zero. The closure of a convex function f, $\operatorname{cl} f$, is the function whose epigraph is the closure of the epigraph of f.

A convex function f is said to be

- *proper* if $\operatorname{dom} f$ is non-empty and $f(x) > -\infty$ for every $x \in \operatorname{dom} f$,

- *closed* if $\operatorname{cl} f = f$,

- *lower semicontinuous* if the sublevel set $\{x : f(x) \leq \alpha\}$ is closed for every $\alpha \in \mathbb{R}$.

For proper convex functions, closedness is the same as lower semicontinuity.

For a function $f : \mathcal{X} \to \bar{\mathbb{R}}$, its *convex conjugate* is defined as

$$f^*(y) = \sup_{x \in \mathcal{X}} \{\langle x, y \rangle - f(x)\}.$$

If f is convex, then f^* is a closed convex function, and proper if and only if f is proper. For any proper convex function f and its convex conjugate f^*, *Fenchel's inequality* holds:

$$\langle x, y \rangle \leq f(x) + f^*(y), \quad \forall x \in \operatorname{dom} f \text{ and } \forall y \in \operatorname{dom} f^*.$$

A proper convex function f with effective domain $\mathcal{X}$ is:

- *convex* if and only if

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y), \quad \forall x, y \in \mathcal{X} \text{ and } \lambda \in (0,1), \tag{2.3.1}$$

- called *strictly convex* if the inequality (2.3.1) is strict,

- called *strongly convex* with constant $\mu > 0$ with respect to the norm $\|\cdot\|$ if the following holds:

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) - \frac{\lambda(1-\lambda)\mu}{2}\|x - y\|^2, \quad \forall x, y \in \mathcal{X} \text{ and } \lambda \in (0,1).$$

- called *essentially strictly convex* if f is strictly convex on every convex subset of $\operatorname{dom} \partial f$ (see the next section for the definition of ∂f).

- called *essentially smooth* if it satisfies the following three conditions for $\mathcal{C} = \operatorname{int}(\operatorname{dom} f)$:

  1. $\mathcal{C} \neq \emptyset$;
  2. f is differentiable on $\mathcal{C}$;

3. $\lim_{k\to\infty} \|\nabla f(x^k)\| = +\infty$ for any sequence $\{x^k\}$ in $\mathcal{C}$ converging to a boundary point x of $\mathcal{C}$.

2.4 Subgradients

A vector g is said to be a *subgradient* of a convex function f at point x if

$$f(y) \geq f(x) + \langle g, y - x\rangle, \quad \forall z \in \operatorname{dom} f.$$

The set of all subgradients of f at x is called the *subdifferential* of f at x and is denoted by $\partial f(x)$. The *effective domain* of ∂f is given by

$$\operatorname{dom} \partial f = \{x : \partial f(x) \neq \emptyset\},$$

and satisfies

$$\operatorname{ri}(\operatorname{dom} f) \subseteq \operatorname{dom} \partial f \subseteq \operatorname{dom} f.$$

The *range* of ∂f is given by

$$\operatorname{rge} \partial f = \bigcup \{\partial f(x) : x \in \mathbb{R}^n\},$$

and satisfies

$$\operatorname{ri}(\operatorname{dom} f^*) \subseteq \operatorname{rge} \partial f \subseteq \operatorname{dom} f^*.$$

Chain rule holds for subdifferentials under mild conditions: For any proper convex function f, we have $\partial(\lambda f)(x) = \lambda \partial f(x)$. For any proper convex functions f and g, we have $\partial(f+g)(x) = \partial f(x) + \partial g(x)$ if $\operatorname{ri}(\operatorname{dom} f) \cap \operatorname{ri}(\operatorname{dom} g) \neq \emptyset$. Let $f(x) = h(Ax)$, where h is a proper convex function on $\mathbb{R}^m$ and A is a linear transformation from $\mathbb{R}^n$ to $\mathbb{R}^m$, then $\partial f(x) = A^\top \partial h(Ax)$ if $\operatorname{rge} A \cap \operatorname{ri}(\operatorname{dom} h) \neq \emptyset$ or if h is polyhedral and $\operatorname{rge} A \cap \operatorname{dom} h \neq \emptyset$.

For any closed proper convex function f, the following are equivalent (known as the conjugate subgradient theorem):

1. $f(x) + f^*(g) = \langle x, g \rangle$;

2. $g \in \partial f(x)$;

3. $x \in \partial f^*(g)$.

This implies ∂f^* is the inverse of ∂f in the sense of set-valued maps.

For a closed proper convex function f, ∂f is a single-valued mapping if and only if f is essentially smooth. In this case, $\partial f(x) = \{\nabla f(x)\}$, for all $x \in \operatorname{int}(\operatorname{dom} f)$, whereas $\partial f(x) = \emptyset$ for all $x \notin \operatorname{int}(\operatorname{dom} f)$. Furthermore, ∂f is injective if and only if f is strictly convex on $\operatorname{ri}(\operatorname{dom} f)$ and essentially smooth.

2.5 Set-Valued Maps

A set-valued map $F : \mathcal{E} \rightrightarrows \mathcal{E}^*$ is a map that associates with any $x \in \mathcal{E}$ a subset $F(x)$ of $\mathcal{E}^*$. The subset $\operatorname{dom} F = \{x : F(x) \neq \emptyset\}$ is called *domain* of F. The *image* of a set $\mathcal{X} \subset \mathcal{E}$ under F is the set

$$\mathcal{F}(\mathcal{X}) = \bigcup_{x \in \mathcal{X}} F(x).$$

The *range* of F is the image of $\mathcal{E}$. The *graph* of F is given by

$$\operatorname{graph} F = \{(x, y) \in \mathcal{E} \times \mathcal{E}^* \ : \ y \in F(x)\}.$$

A set-valued map $F : \mathcal{E} \rightrightarrows \mathcal{E}^*$ is said to be

- *closed* (or has *closed graph*) if its graph is a closed subset of $\mathcal{E} \times \mathcal{E}^*$.

- *closed-valued* (or has *closed values*) if $F(x)$ is a closed set for each x. The terms, *open-valued*, *compact-valued* and *convex-valued* are defined similarly.

- *locally bounded at* x if for some neighborhood $\mathcal{M}$ of x, the set $F(\mathcal{M})$ is bounded. F is said to be *locally bounded* if it is so at every $x \in \mathcal{E}$. F is said to be *bounded* if $\operatorname{rge} F$ is a bounded subset of $\mathcal{E}^*$.

For a set-valued map $F : \mathcal{E} \rightrightarrows \mathcal{E}^*$:

- the *upper inverse* F^{u} of a subset $\mathcal{Y}$ of $\mathcal{E}^*$ is defined by

$$F^{\mathrm{u}}(\mathcal{Y}) = \{x \in \mathcal{E} : F(x) \subset \mathcal{Y}\},$$

- the *lower inverse* F^{ℓ} of a subset $\mathcal{Y}$ of $\mathcal{E}^*$ is defined by

$$F^{\ell}(\mathcal{Y}) = \{x \in \mathcal{E} : F(x) \cap \mathcal{Y} \neq \emptyset\}.$$

A set-valued map $F : \mathcal{E} \rightrightarrows \mathcal{E}^*$ is called:

- *upper semi-continuous at* $x \in \mathcal{E}$ if for any open $\mathcal{N}$ containing $F(x)$, there exists a neighborhood $\mathcal{M}$ of x such that $F(\mathcal{M}) \subset \mathcal{N}$ (equivalently, the upper inverse image $F^{\mathrm{u}}(\mathcal{N})$ contains a neighborhood of x in $\mathcal{E}$). We say that F is *upper semi-continuous* if it is so at every $x \in \mathcal{E}$.

- *lower semi-continuous at* $x \in \mathcal{E}$ if for any $y \in F(x)$ and any neighborhood $\mathcal{N}$ of y, there exists a neighborhood $\mathcal{M}$ of x such that $F(x') \cap \mathcal{N} \neq \emptyset$ for all $x' \in \mathcal{M}$ (equivalently, the lower inverse image $F^{\ell}(\mathcal{N})$ contains a neighborhood of x). We say that F is *lower semi-continuous* if it is so at every $x \in \mathcal{E}$.

- *continuous at* $x \in \mathcal{E}$ if it is both upper and lower semi-continuous at x. We say that F is *continuous* if it is so at every $x \in \mathcal{E}$.

- *monotone* if it has the property that

$$\langle y_1 - y_2, x_1 - x_2 \rangle \geq 0, \quad \forall y_1 \in F(x_1) \text{ and } y_2 \in F(x_2).$$

- *maximal monotone* if no enlargement of its graph is possible without destroying monotonicity, i.e., if for every $(x_1, y_1) \in (\mathcal{E} \times \mathcal{E}^*) \setminus \operatorname{graph} F$, there exists $(x_2, y_2) \in \operatorname{graph} F$ such that $\langle y_1 - y_2, x_1 - x_2 \rangle < 0$.

Part I

Coordinate Descent Method

Chapter 3

An Overview

Consider the unconstrained optimization problem

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is smooth and convex. The CD method is an iterative algorithm that performs global minimizations with respect to a single coordinate (or several coordinates in the case of block-CD) at each iteration. Specifically, at iteration k, an index $i_k \in \{1, 2, \ldots, n\}$ is chosen and the decision variable is updated to minimize the objective function in the i_k-th coordinate direction [21, 22]. The steps of this method are summarized in Algorithm 1. The integer $k = \ell n + j$ keeps track of the total number of iterations consisting of outer iterations indexed by ℓ and inner iterations indexed by the counter j. Each outer iteration is called a "cycle" or an "epoch" of the algorithm.

The CD method can be implemented with various coordinate selection schemes, both deterministic and stochastic, for choosing the coordinate i_k to be updated at iteration k. Prominent schemes include the following.

- Cyclic CD (CCD): The index $i(\ell, j)$ is chosen in a cyclic fashion over the elements in the set $\{1, 2, \ldots, n\}$ satisfying $i(\ell, j) = j + 1$.

- Randomized CD (RCD): The index $i(\ell, j)$ is chosen randomly with replacement from

Algorithm 1: Coordinate Descent (CD)

```
Choose initial point $x^0 \in \mathbb{R}^n$
for $\ell = 0, 1, \dots$ do
  for $j = 0, 1, \dots, n-1$ do
    Set $k = \ell n + j$
    Choose index $i_k = i(\ell, j) \in \{1, 2, \dots, n\}$
    $x_i^{k+1} = x_i^k$ for all $i \neq i_k$, and $x_{i_k}^{k+1} \in \arg\min_{\xi \in \mathbb{R}} f(x_1^k, \dots, x_{i_k-1}^k, \xi, x_{i_k+1}^k, \dots, x_n^k)$
  end for
end for
```

the set $\{1, 2, \dots, n\}$. Unless otherwise stated, each coordinate has the same probability of being chosen.

- Random Permutations CD (RPCD): At the beginning of each epoch ℓ, a permutation of $\{1, 2, \dots, n\}$ is chosen, denoted by π_ℓ, uniformly at random over all permutations. Then, the index $i(\ell, j)$ is chosen as the $(j+1)$-th element of π_ℓ. Each permutation π_ℓ is independent of the permutations used at all previous and later epochs. This approach amounts to sampling indices from the set $\{1, 2, \dots, n\}$ without replacement for each epoch.

The CD method in Algorithm 1 is an exact-minimization scheme along each chosen coordinate. When the exact-minimization of the objective function in the i_k-th coordinate is costly, it is often replaced with an approximate-minimization scheme, e.g., by moving along the direction pointed by the negative coordinate gradient, see Algorithm 2. This method is called as the coordinate *gradient* descent (CGD) method.

3.1 Existing Convergence Results

The CD method has a long history in optimization and have been used in many applications. The convergence of the CD method has been studied extensively in the literature (cf. [23, 95, 96, 110]). It is known that when f is continuously differentiable but possibly non-

Algorithm 2: Coordinate Gradient Descent (CGD)

```
Choose initial point $x^0 \in \mathbb{R}^n$
for $\ell = 0, 1, 2, \ldots$ do
  for $j = 0, 1, 2, \ldots, n-1$ do
    Set $k = \ell n + j$
    Choose index $i_k = i(\ell, j) \in \{1, 2, \ldots, n\}$
    Choose stepsize $\eta_k > 0$
    $x^{k+1} \leftarrow x^k - \eta_k [\nabla f(x^k)]_{i_k} e_{i_k}$, where $[\nabla f(x^k)]_{i_k} = e_{i_k}^\top \nabla f(x^k)$
  end for
end for
```

convex, if each subproblem has a unique solution and f is monotonically non-increasing between the current iterate and the minimizer of the subproblem, then every limit point of the sequence $\{x^k\}$ generated by the CD method is a stationary point. When one of these conditions do not hold, the CD method does not necessarily converge to a stationary point of non-convex problems as shown by Powell [119]. When f is convex and its level sets are compact, the CD method converges even when the subproblems do not have unique solutions [69]. In [122], it has been shown that each subproblem can be solved inexactly, by optimizing a certain surrogate function, and the resulting inexact CD method converges.

The rate of convergence of the CD method, even when f is convex, is difficult to establish [108]. To the best of our knowledge, the global rate of convergence is not established in the general case, whereas in [95] it is shown that when f is twice-differentiable and strongly convex, the CD method with almost cyclic rule or Gauss-Southwell rule[1] converges linearly to a minimizer of f, although no rate estimate is given. In [132], the authors consider a special composite optimization problem, where f is the composition of a convex function with Lipschitz continuous gradients and the ℓ_1-norm of the variable. They show that the CD method with cyclic rule converges faster than the CGD method with cyclic rule when a certain isotonicity condition holds, whereas the difference in the rate of convergence is not quantified. Note that in the case of 2-block updates, the CD method is equivalent to the

[1] Gauss-Southwell rule corresponds to choosing the "best" coordinate to update.

alternating minimization method, for which sublinear convergence is proven for composite optimization problems in [17].

The existing theory on the convergence of the CGD method is significantly richer compared to the one of the CD method. Of course the rate of convergence of the CGD method is significantly affected by the choice of coordinate selection and stepsize rules. This is extensively studied in the literature and we present a non-exhaustive overview of the existing results in Table 3.1. Let $\{x^k\}$ denote the sequence generated by the CGD method with some coordinate selection rule. Linear and sublinear rates of convergence in Table 3.1 represent an upper bound on either $f(x^{\ell n}) - f(x^*)$ or $\|x^{\ell n} - x^*\|^2$, where ℓ represents the epoch counter and for randomized coordinate selection rules, expectation of these values are considered. Most of the existing work analyze the rate of convergence of the CGD method under the assumption that ∇f is coordinate-wise Lipschitz continuous, i.e., for every coordinate $i = 1, \dots, n$, there exist a constant $0 < L_i < \infty$ such that

$$|[\nabla f(x + te_i)]_i - [\nabla f(x)]_i| \leq L_i |h|, \quad \forall\, x \in \mathbb{R}^n,\, t \in \mathbb{R}, \tag{3.1.1}$$

where $\{e_i\}$ denote the standard basis vectors. The maximum, minimum and average of such constants are respectively denoted by

$$L_{\max} = \max_{i \in [n]} L_i, \quad L_{\min} = \min_{i \in [n]} L_i \quad \text{and} \quad L_{\text{avg}} = \frac{1}{n} \sum_{i=1}^{n} L_i.$$

Let L denote the global Lipschitz constant of ∇f, that is

$$\|\nabla f(x) - \nabla f(y)\|_* \leq L \|x - y\|, \quad \forall\, x, y \in \mathbb{R}^n.$$

By using the relationships between norm and trace of a symmetric matrix, we can observe that

$$1 \leq \frac{L}{L_{\max}} \leq n.$$

Indeed, the lower bound holds trivially and the upper bound is achieved when $\nabla f(x) = Ax$,

Table 3.1: Rate of convergence of the CGD method when f is convex continuously differentiable and ∇f is coordinate-wise Lipschitz continuous, see (3.1.1). ℓ denotes the iteration counter such that $k = \ell n$. The CGD method is implemented with a constant stepsize rule specific to each coordinate, i.e., there exists constants $\eta(i) > 0$ for all $i \in [n]$ such that $\eta_k = \eta(j)$ when $i_k = j$. When presenting the linear convergence rate of CGD methods with randomized rules, it is assumed that $\mu \ll L_{\max}$ and the approximation $(1-c/n)^{n\ell} \approx (1-c)^{\ell}$ is used for simplicity. Dependence on constants are ignored for clarity.

Paper	Coordinate Selection Rule	Stepsize	Sublinear Rate	Linear Rate
[108]	Randomized: $p_i = \frac{L_i}{nL_{\text{avg}}}$	$\frac{1}{L_i}$	$\frac{L_{\text{avg}}}{\ell}$	$\left(1 - \frac{\mu}{L_{\text{avg}}}\right)^{\ell}$
[19]	Cyclic	$\frac{1}{L_i}$	$\frac{L_{\max}^3 n^3}{L_{\min}^2 \ell}$	$\left(1 - \frac{L_{\min}^2 \mu}{L_{\max}^3 n^3}\right)^{\ell}$
[19]	Cyclic	$\frac{1}{L}$	$\frac{Ln}{\ell}$	$\left(1 - \frac{\mu}{Ln}\right)^{\ell}$
[137]	Cyclic	$\frac{1}{L_{\max}}$	$\frac{\min(L^2 n, L_{\text{avg}}^2 n^2)}{L_{\max}\ell}$	×

where A is the matrix of ones. Whenever f is assumed to be strongly convex, we denote the strong convexity constant by $\mu > 0$.

In [108], sublinear and linear convergence rates of the CGD method with randomized coordinate selection rule is presented, see the first row of Table 3.1. Following this work, sublinear and linear convergence rates of the CGD method with cyclic update rule is presented in [19]. The sublinear rates presented in [19] are improved in [137], where the authors also presented tighter converge rates for quadratic optimization problems. In order to compare these rates of convergence, let us first recall the convergence rate of the gradient descent method. The gradient descent method when applied to smooth convex functions enjoy a sublinear convergence rate of L/ℓ and when the function is additionally μ-strongly convex it enjoys a linear convergence rate of $(1 - \mu/L)^{\ell}$, where ℓ is the iteration counter. Note that in general an iteration of gradient descent requires as many flops as an epoch of coordinate gradient descent, and hence these rates can be fairly compared with the ones in Table 3.1. The rate results in Table 3.1 indicate that the randomized CGD method converges faster (in expectation) than the gradient descent method. In particular, these rates suggest that the randomized CGD method can be $\mathcal{O}(n)$ times faster than the gradient descent method since L/L_{avg} can be as large as n. On the other hand, the CGD method with a cyclic update rule does not enjoy this feature, which can be observed by inspecting Table 3.1: the cyclic CGD method converges much slower than the gradient descent method. When $L_{\min} = L_{\max}$, it can be observed that the cyclic CGD method with stepsize $1/L_i$ can be $\mathcal{O}(n^2)$ times slower than the gradient descent method, whereas this gap reduces to $\mathcal{O}(n)$ for more conservative stepsize rules.

While the convergence of the CGD method with cyclic and randomized coordinate selection rules are relatively well-studied, there is limited understanding of the effects of random permutations in CGD methods, with the exception of a few recent papers that focus on special quadratic problems [138, 88, 89, 113]. Among these, Oswald and Zhou [113] studies the effects of random permutations on the convergence rate of the successive over-relaxation (SOR) method (that is used to solve linear systems) and presents a convergence rate on the expected function value of the iterates generated by the SOR method. The cyclic CGD method, when applied to quadratic minimization problems with stepsize $1/L_i$, is equivalent to the SOR method (applied to the linear system that represents the first-order optimality condition of the quadratic problem) when the relaxation parameter is chosen as $\omega = 1$. Therefore, the convergence rate results in [113] readily extend to the CGD method with random permutations for quadratic problems. In [138], the authors construct a quadratic problem, for which the distance of the iterates (to the optimal solution) for the cyclic CGD method decays $\mathcal{O}(n^2)$ times slower than the distance of the expected iterates for the CGD method with randomized and randomly permuted coordinate selections. Lee and Wright [88] consider the same problem and present that the expected function values of the iterates generated by the CGD method with randomized and randomly permuted coordinate selections decay with similar rates, while the asymptotic convergence rate of with random permutations is shown to be slightly better than for random with-replacement. In a following paper [89], the results in [88] are generalized to a larger class of quadratic problems through a more elaborate analysis.

3.2 CD and CGD Methods for Quadratic Problems

In order to clarify the rate of convergence comparison between cyclic and randomized coordinate selection rules, we consider quadratic optimization problems:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}x^\top Ax - b^\top x, \tag{3.2.1}$$

where A is a positive semidefinite matrix. For this problem, the coordinate Lipschitz constant L_i is equal to the corresponding diagonal entry of the matrix A, that is $L_i = A_{ii}$. Furthermore the strong convexity and the smoothness constants are respectively given by $\mu = \lambda_{\min}(A)$ and $\mu = \lambda_{\max}(A)$.

Starting from an initial point $x^0 \in \mathbb{R}^n$, the CD method, at each iteration k, picks a coordinate of x, say i_k, and updates the decision vector by performing exact minimization along the i_k-th coordinate:

$$x^{k+1} = x^k - \frac{A_{i_k}^\top x^k - b_{i_k}}{A_{i_k,i_k}} e_{i_k}.$$

The CGD method on the other hand performs the following update:

$$x^{k+1} = x^k - \eta_k [Ax^k - b]_{i_k} e_{i_k}.$$

It is easy to see that the CD update rule is equivalent to the CGD update rule with stepsize $1/A_{i_k,i_k}$. Therefore, for quadratic optimization problems, we will refer to the CGD method with stepsize $1/A_{i_k,i_k}$ as the CD method in order to highlight the choice of stepsize.

To observe the the convergence rate difference between different variants of CD methods for quadratic problems, let us consider the case $L_i = A_{ii} = 1$ for all $i \in [n]$ and $L = \mathcal{O}(n)$. Then, the RCD method enjoys [108] a sublinear convergence rate of $1/\ell$ and a linear convergence rate of $(1-\mu)^\ell$. On the other hand, [19] shows that the CCD method enjoys a sublinear convergence rate of n^3/ℓ and a linear convergence rate of $(1-\mu/n^3)^\ell$. It can be observed from Table 3.1 that more conservative choice of stepsizes yield a sublinear convergence rate of n^2/ℓ and a linear convergence rate of $(1 - \mu/n^2)^\ell$ for the cyclic CGD method. It is shown in [138] that these rate results also hold for when stepsize is chosen more greedily. In particular, the authors show that the CCD method enjoys a sublinear convergence rate of n^2/ℓ and a linear convergence rate of $(1-\mu/n^2)^\ell$. Let us recall that the gradient descent method enjoys a sublinear convergence rate of n/ℓ and a linear convergence rate of $(1 - \mu/n)^\ell$ for this particular setting. This performance gap between the gradient descent, CCD and RCD methods is investigated in [138], where the authors constructed a quadratic problem for which the cyclic CGD method is $\mathcal{O}(n)$ times slower than the gradient descent method, which in turn is $\mathcal{O}(n)$ times slower than the randomized CGD method.

Establishing tight convergence rate estimates for the RPCD method is significantly more difficult compared to the CCD and RCD methods. There are only a few papers in the literature that characterize rate estimates for the RPCD method even for quadratic problems. Among these, Oswald and Zhou [113] studies the convergence rate of the successive overrelaxation (SOR) method (that is used to solve linear systems) with random permutations and presents a convergence rate on the expected function value of the iterates generated by the SOR method. The CD method, when applied to quadratic minimization problems, is equivalent to the SOR method (applied to the linear system that represents the first-order optimality condition of the quadratic problem) when the relaxation parameter is chosen as $\omega = 1$. Therefore, the rate estimates in [113] readily extend to the RPCD method for quadratic problems, yielding a linear convergence rate estimate of $(1-\mu/L^2)^\ell$. This implies that the RPCD method can be as slow as the CCD method in the worst-case. However, this conservative rate estimate is rarely observed in practice. In particular, in [138] and [88], the authors consider the quadratic problem for which the CCD method is $\mathcal{O}(n^2)$ times slower than the RCD method, and show that the RPCD method attains a slightly better asymptotic rate of convergence compared to the RCD method. In [89], the results in [88] are generalized to a larger class of quadratic problems through a more elaborate analysis.

3.3 Summary of Contributions

In the remainder of this part, we present several contributions on the convergence of the CD method as detailed below.

In Chapter 4, we investigate the performance gap between the CCD and RCD methods for quadratic problems. The existing rate estimates suggest the RCD method performs better than the CCD method, whereas numerical experiments do not provide clear justification for this comparison. We address this problem by establishing the efficiency of CCD over RCD on three problem classes:

1. A is an M-matrix, i.e., the off-diagonal entries of A are nonpositive. These matrices arise in a large number of applications. A notable example is problems that consider minimization of quadratic forms of graph Laplacians (where $A = D - W$ and W denoes the weighted adjacency graph and D is a diagonal matrix given by $D_{i,i} = \sum_j W_{i,j}$), e.g., for spectral partitioning and semisupervised learning.

2. A is a non-frustrated matrix, i.e., off-diagonal entries of $-A$ does not contain any cycles with an odd number of negative edge weights. This set of matrices naturally extend M-matrices as any non-frustrated matrix is sign-similar to an M-matrix.

3. A is a 2-cyclic matrix, i.e., the graph induced by A is bipartite.

We build on the seminal works of Young [153] and Varga [147] on the analysis of Gauss-Seidel method for solving linear systems of equations (with matrices satisfying certain properties) and provide a novel analysis that allows us to compare the asymptotic worst-case convergence rate of CCD and RCD for the aforementioned class of problems and establish the faster performance of CCD with any deterministic order. Furthermore, we provide lower and upper bounds on the amount of improvement on the rate of CCD relative to RCD. We also provide a characterization of the best cyclic order (that leads to the maximum improvement in convergence rate) in terms of the combinatorial properties of the Hessian matrix of the objective function.

In Chapter 5, we study the convergence rate of RPCD for a special class of quadratic optimization problems with a diagonally dominant, permutation invariant Hessian matrix of $A = (1+\alpha)I - \alpha\mathbf{1}\mathbf{1}^\top$, and compare its performance to that of RCD and CCD. In particular, we first provide an exact worst-case convergence rate comparison between RPCD, RCD, and CCD in terms of the distance of the expected iterates to the optimal solution, as a function of a parameter that represents the extent of diagonal dominance of the Hessian matrix. Our results show that, on this problem, CCD is always faster than RPCD, which in turn is always faster than RCD. Furthermore, we show that the relative convergence rate of RPCD to RCD goes to infinity as the Hessian matrix becomes more diagonally dominant. On the other extreme, as the Hessian matrix becomes less diagonally dominant, the ratio of convergence rates converges to a value in $[3/2, e-1)$, with the upper bound $e-1$ achieved in the limit as $n \to \infty$. Our second set of results compares the convergence rates of RPCD and RCD with respect to two other criteria that are widely used in the literature: the expected distance of the iterates to the solution and the expected function values of the iterates. For these criteria, we show that RPCD is faster than RCD in terms of the tightest upper bounds we obtain, and the amount of improvement increases as the matrices become more diagonally dominant.

In Chapter 6, we provide the first local and global convergence rate guarantees for the CD method applied to a particular non-convex problem. This problem arises from a low-rank factorization to semidefinite programs with diagonal constraints. As discussed above, establishing precise rate estimates for the CD method is notoriously difficult. However, we exploit the special manifold structure of the corresponding problem and characterize convergence rate estimates for the CD method. First, we establish the global sublinear convergence of the CD method without any assumptions on the problem. We then show that the CD method enjoys a linear convergence rate around a neighborhood of any local minimum when the objective function satisfies a quadratic growth condition. We then prove that this quadratic growth condition generically holds, i.e., the set of problems for which this condition does not hold has measure zero. Next, we propose an algorithm by incorporating the

CD and Lanczos methods. We show that this algorithm returns an approximately globally optimal solution to the corresponding non-convex problem. We conclude the chapter by validating our theoretical results via numerical examples and presenting the efficiency of the CD method compared to the state-of-the-art manifold optimization methods.

Chapter 4

When Does CCD outperform RCD?

In this chapter, we investigate problem classes for which the CCD method is faster than the RCD method in terms of asymptotic worst-case convergence. Our main focus will be on quadratic problems:

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^\top A x, \tag{4.0.1}$$

where A is a positive definite matrix. Although we do not include a linear term in (4.0.1), it is only for ease of presentation and all our results directly extend to quadratic problems of the type $\frac{1}{2}x^\top Ax - b^\top x$ for any $b \neq 0$.

In order to compactly represent the CCD and RCD iterations in a matrix form, we introduce the following decomposition on the Hessian matrix:

$$A = D - N - N^\top,$$

where D is the diagonal part of A and $-N$ is the strictly lower triangular part of A. The update rule (over an epoch) of the CCD method with cyclic order $1, \dots, n$ (i.e., $i_k = k \pmod{n} + 1$), is then given by:

$$x_{\text{CCD}}^{(\ell+1)n} = C\, x_{\text{CCD}}^{\ell n}, \quad \text{where} \quad C = (D - N)^{-1} N^\top. \tag{4.0.2}$$

Note that the update rule in (4.0.2) is equivalent to one iteration of the Gauss-Seidel (GS) method applied to the first-order optimality condition of (4.0.1), i.e., applied to the linear system $Ax = 0$ [150].

We next consider the RCD method, where i_k is chosen at random among $\{1, \ldots, n\}$ with probabilities $\{p_1, \ldots, p_n\}$ independently at each iteration k. Given the k-th iterate generated by the RCD algorithm x^k_{RCD}, we have

$$\mathbb{E}_k\big[x^{k+1}_{\text{RCD}} \mid x^k_{\text{RCD}}\big] = \big(I - PD^{-1}A\big)x^k_{\text{RCD}},$$

where $P = \operatorname{diag}(p_1, \ldots, p_n)$ and the conditional expectation $\mathbb{E}_k$ is taken over the random variable i_k given x^k_{RCD}. Using the nested property of the expectations, the RCD iterations in expectation over an epoch satisfy

$$\mathbb{E}x^{(\ell+1)n}_{\text{RCD}} = R\, \mathbb{E}x^{\ell n}_{\text{RCD}}, \quad \text{where} \quad R = \big(I - PD^{-1}A\big)^n. \tag{4.0.3}$$

In Section 4.1, we present the notion of asymptotic convergence rate to compare the CCD and RCD methods and provide a motivating example on which CCD converges faster than RCD. In Section 4.2, we present classes of problems for which the asymptotic convergence rate of CCD is faster than the one of RCD. We conclude in Section 4.3 by providing numerical experiments that validates our theoretical results on the performance of the CCD and RCD methods.

4.1 An Asymptotic Rate Comparison Metric

In the following section, we define our basis of comparison for rates of CCD and RCD methods. To measure the performance of these methods, we use the notion of the average worst-case asymptotic rate that has been studied extensively in the literature for characterizing the rate of iterative algorithms [147]. In Section 4.1.2, we construct an example, for which the rate of CCD is more than twice the rate of RCD. This raises the question whether the best known convergence rates of CCD in the literature are tight or whether there exist a class of problems for which CCD provably attains better convergence rates than the best known rates for RCD, a question which we will answer positively in Section 4.2.

4.1.1 Asymptotic Converge Rate for Iterative Algorithms

Consider an iterative method with update rule $x^{(\ell+1)n} = Cx^{\ell n}$ (e.g., the CCD method). The reduction in the distance to the optimal solution of the iterates generated by this algorithm after ℓ epochs is given by

$$\frac{\|x^{\ell n} - x^*\|}{\|x^0 - x^*\|} = \frac{\|C^\ell(x^0 - x^*)\|}{\|x^0 - x^*\|}. \tag{4.1.1}$$

Note that the right hand side of (4.1.1) can be as large as $\|C^\ell\|$, hence in the worst-case, the average decay of distance at each epoch of this algorithm is $\|C^\ell\|^{1/\ell}$. Over any finite epochs $\ell \geq 1$, we have $\|C^\ell\|^{1/\ell} \geq \rho(C)$ and $\|C^\ell\|^{1/\ell} \to \rho(C)$ as $\ell \to \infty$. Thus, we define the *asymptotic worst-case convergence rate* of an iterative algorithm (with iteration matrix C) as follows

$$\text{Rate(CCD)} := \lim_{\ell\to\infty} \sup_{x^0\in\mathbb{R}^n} -\frac{1}{\ell}\log\left(\frac{\|x^{\ell n} - x^*\|}{\|x^0 - x^*\|}\right) = -\log(\rho(C)). \tag{4.1.2}$$

We emphasize that this notion has been used extensively for studying the performance of iterative methods such as GS and Jacobi methods [23, 110, 147, 153]. Note that according to our definition in (4.1.2), larger rate means faster algorithm and we will use these terms interchangably in throughout the chapter.

Analogously, for a randomized method with expected update rule $\mathbb{E}x^{(\ell+1)n} = R\,\mathbb{E}x^{\ell n}$ (e.g., the RCD method), we consider the asymptotic convergence of the expected iterate error $\|\mathbb{E}(x^{\ell n}) - x^*\|$ and define the asymptotic worst-case convergence rate as

$$\text{Rate(RCD)} := \lim_{\ell\to\infty} \sup_{x^0\in\mathbb{R}^n} -\frac{1}{\ell}\log\left(\frac{\|\mathbb{E}(x^{\ell n}) - x^*\|}{\|x^0 - x^*\|}\right) = -\log(\rho(R)), \tag{4.1.3}$$

Note that in (4.1.3), we use the distance of the expected iterates $\|\mathbb{E}x^{\ell n} - x^*\|$ as our convergence criterion. One can also use the expected distance (or the squared distance) of the iterates $\mathbb{E}\|x^{\ell n} - x^*\|$ as the convergence criterion, which is a stronger convergence criterion than the one in (4.1.3). This follows since $\mathbb{E}\|x^{\ell n} - x^*\| \geq \|\mathbb{E}x^{\ell n} - x^*\|$ by Jensen's inequality and any convergence rate on $\mathbb{E}\|x^{\ell n} - x^*\|$ immediately implies at least the same convergence rate on $\|\mathbb{E}x^{\ell n} - x^*\|$ as well. Since we consider the reciprocal case, i.e., obtain a convergence rate on $\|\mathbb{E}x^{\ell n} - x^*\|$ and show that it is slower than that of CCD, our results naturally imply that the convergence rate on $\mathbb{E}\|x^{\ell n} - x^*\|$ is also slower than that of CCD.

4.1.2 A Motivating Example

In this section, we provide an example for which the (asymptotic worst-case convergence) rate of CCD is better than the one of RCD and building on this example, in Section 4.2, we construct a class of problems for which CCD attains a better rate than RCD. For some positive integer $n \geq 1$, consider the $2n \times 2n$ symmetric matrix

$$A = I - N - N^\top, \quad \text{where} \quad N = \frac{1}{n^2}\begin{bmatrix} 0_{n\times n} & 0_{n\times n} \\ \mathbf{1}_{n\times n} & 0_{n\times n} \end{bmatrix}, \tag{4.1.4}$$

and $\mathbf{1}_{n\times n}$ is the $n \times n$ matrix with all entries equal to 1 and $0_{n\times n}$ is the $n \times n$ zero matrix. Noting that A has a special structure (A is equal to the sum of the identity matrix and the rank-two matrix $-N-N^\top$), it is easy to check that $1-1/n$ and $1+1/n$ are eigenvalues of A with the corresponding eigenvectors $\begin{bmatrix} \mathbf{1}_{1\times n} & \mathbf{1}_{1\times n} \end{bmatrix}^\top$ and $\begin{bmatrix} \mathbf{1}_{1\times n} & -\mathbf{1}_{1\times n} \end{bmatrix}^\top$. The remaining $2n-2$ eigenvalues of A are equal to 1.

The iteration matrix of the CCD algorithm when applied to the problem in (4.0.1) with the matrix (4.1.4) can be found as

$$C = \begin{bmatrix} 0_{n\times n} & \frac{1}{n^2}\mathbf{1}_{n\times n} \\ 0_{n\times n} & \frac{1}{n^3}\mathbf{1}_{n\times n} \end{bmatrix}.$$

The eigenvalues of C are all zero except the eigenvalue of $1/n^2$ with the corresponding eigenvector $[n\mathbf{1}_{1\times n}, \mathbf{1}_{1\times n}]^\top$. Therefore, $\rho(C) = 1/n^2$ and $\text{Rate(CCD)} = -\log(\rho(C)) = 2\log n$. On the other hand, the spectral radius of the expected iteration matrix of RCD can be found as

$$\rho(R) = \left(1 - \frac{\lambda_{\min}(A)}{n}\right)^n \geq 1 - \lambda_{\min}(A) = \frac{1}{n},$$

which yields $\text{Rate(RCD)} = -\log(\rho(R)) \leq \log n$. Thus, we conclude

$$\frac{\text{Rate(CCD)}}{\text{Rate(RCD)}} \geq 2, \quad \forall\, n \geq 1.$$

That is, CCD is at least twice as fast as RCD in terms of the the asymptotic rate. This motivates us to investigate if there exists a more general class of problems for which the asymptotic worst-case rate of CCD is larger than that of RCD. The answer to this question turns out to be positive as we describe in the following section.

4.2 Deterministic Orders Provably Outperform Randomized Sampling

In this section, we present special classes of problems (of the form (4.0.1)) for which the asymptotic worst-case rate of CCD is larger than that of RCD. We begin our discussion by highlighting the main assumption we will use in this section.

Assumption 1. *Hessian matrix A has the following properties:*

(i) A is a symmetric positive definite matrix with smallest eigenvalue $\mu > 0$.

(ii) The diagonal entries of A are 1.

Given any positive definite matrix A with diagonals $D \neq I$, the diagonal entries of the preconditioned matrix $D^{-1/2}AD^{-1/2}$ are 1. Therefore, part *(ii)* of Assumption 1 is mild. The relationship between the smallest eigenvalue of the original matrix and the preconditioned matrix are as follows. Let $\sigma > 0$ and $L_{\max}$ denote the smallest eigenvalue and the largest diagonal entry of the original matrix, respectively. Then, the smallest eigenvalue of the preconditioned matrix satisfies $\mu \geq \sigma / L_{\max}$.

Remark 4.1. *For the RCD algorithm, the coordinate index $i_k \in \{1, \ldots, n\}$ (at iteration k) can be chosen using different probability distributions $\{p_1, \ldots, p_n\}$. The most widely used distributions (due to their simplicity) have the form $p_i = \frac{A_{i,i}^{\alpha}}{\sum_{J=1}^{N} A_{j,j}^{\alpha}}$ for a choice of $\alpha \geq 0$ as discussed in [108]. Since by Assumption 1, the diagonal entries of A are 1, we have $p_i = \frac{1}{n}$ for all $i \in \{1, \ldots, n\}$ and $\alpha \geq 0$. Therefore, in the rest of the chapter, we consider the RCD algorithm with uniform and independent coordinate selection at each iteration.*

In the following lemma, we characterize the spectral radius of the RCD method.

Lemma 4.2. *Suppose Assumption 1 holds. Then, the spectral radius of the expected iteration matrix R of the RCD algorithm (defined in (4.0.3)) is given by*

$$\rho(R) = \left(1 - \frac{\mu}{n}\right)^n. \tag{4.2.1}$$

Proof By Assumption 1, $\mu > 0$ and $\operatorname{tr} A = n$, which implies all eigenvalues of the matrix A/n are in the interval $(0, 1)$. Therefore, we have

$$\rho(R) = \lambda_{\max}\left(\left(I - \frac{1}{n}A\right)^n\right) = \left(1 - \frac{1}{n}\lambda_{\min}(A)\right)^n = \left(1 - \frac{\mu}{n}\right)^n.$$

□

In the following sections, we present classes of problems for which CCD attains better convergence rates than RCD.

4.2.1 Convergence Rate of CCD for 2-Cyclic Matrices

In this section, we introduce the class of 2-cyclic matrices and show that the asymptotic worst-case convergence rate of CCD is more than two times faster than that of RCD.

Definition & Properties

Definition 4.3 (2-Cyclic Matrix)**.** *A matrix H is 2-cyclic if there exists a permutation matrix P such that*

$$PHP^\top = D + \begin{bmatrix} 0 & B_1 \\ B_2 & 0 \end{bmatrix}, \tag{4.2.2}$$

where the diagonal null submatrices are square and D is a diagonal matrix.

This definition can be interpreted as follows. Let H be a 2-cyclic matrix, i.e., H satisfies (4.2.2). Then, the graph induced by the matrix $H - D$ is bipartite. The definition in (4.2.2) is first introduced in [153], where it had an alternative name, called *Property A*. A generalization of this property is later introduced by Varga to the class of p-cyclic matrices [147] where $p \geq 2$ can be arbitrary.

We next introduce the following definition that will be useful in Theorem 4.13 and explicitly identify the class of matrices that satisfy this definition.

Definition 4.4 (Consistently Ordered Matrix)**.** *For a matrix H, let $H = H_D - H_L - H_U$ be its decomposition such that H_D is a diagonal matrix, H_L (and H_U) is a strictly lower (and upper) triangular matrix. If the eigenvalues of the matrix $\alpha H_L + \alpha H_U - \gamma H_D$ are independent of α for any $\gamma \in \mathbb{R}$ and $\alpha \neq 0$, then H is said to be consistently ordered.*

In the next lemma, we highlight the connection between Definitions 4.3 and 4.4.

Lemma 4.5 ([153, Theorem 4.5])**.** *A matrix H is 2-cyclic if and only if there exists a permutation matrix P such that $PHP^\top$ is consistently ordered.*

This lemma shows that in order for the lower bounds in Theorem 4.13 to hold with equality, it is necessary and sufficient that the lower triangular part of A can be written as $N = \begin{bmatrix} 0 & 0 \\ B & 0 \end{bmatrix}$, for a real matrix B where the diagonal null submatrices are square matrices of appropriate dimension. However, in Theorem 4.13, we assume that A is an M-matrix, i.e., $N \geq 0$. In the following theorem, we prove that a similar spectral radius equality to Theorem 4.13 holds for consistently ordered 2-cyclic matrices under less restrictive assumptions (by removing the assumption that the off-diagonal entries are non-positive).

Convergence Rates

In the next theorem, we characterize the convergence rate of CCD algorithm applied to a 2-cyclic matrix. Since $\rho(R) \geq 1 - \mu$ by Lemma 4.2, the following theorem indicates that the spectral radius of the CCD iteration matrix is smaller than $\rho^2(R)$.

Theorem 4.6. *Suppose Assumption 1 holds and A is a consistently ordered 2-cyclic matrix. Then, the spectral radius of the CCD method is given by*

$$\rho(C) = (1-\mu)^2.$$

Proof The eigenvalues of C are the roots of the polynomial

$$\phi_C(\lambda) = \det(\lambda I - C) = 0.$$

As $I - N$ is non-singular and $\det(I - N) = 1$, we have

$$\begin{aligned}\phi_C(\lambda) &= \det(I-N)\det(\lambda I - C)\\ &= \det(\lambda I - \lambda N - N^\top)\\ &= \sqrt{\lambda}\det\left(\sqrt{\lambda} I - \left(\sqrt{\lambda}N + \frac{1}{\sqrt{\lambda}}N^\top\right)\right).\end{aligned}$$

Therefore, if $\sqrt{\lambda}$ is an eigenvalue of the matrix $\sqrt{\lambda}N + \frac{1}{\sqrt{\lambda}}N^\top$, then λ is an eigenvalue of C. Furthermore, since the eigenvalues of the matrix $\sqrt{\lambda}N + \frac{1}{\sqrt{\lambda}}N^\top$ are independent of λ as A is a consistently ordered matrix by definition, then $\sqrt{\lambda}$ is an eigenvalue of $N + N^\top$ as well. Consequently, we have $\rho(C) = \rho^2(N + N^\top) = \rho^2(I - A) = (1 - \mu)^2$. □

Remark 4.7. *Note that our motivating example given by* (4.1.4) *in Section 4.1.2 is an example of a consistently ordered 2-cyclic matrix where Theorem 4.6 is directly applicable. In fact, for* (4.1.4)*, we can apply Theorem 4.6 with $\mu = 1 - 1/n$ leading to $\rho(C) = 1/n^2$, which coincides exactly with our previous computations of $\rho(C)$ in Section 4.1.2. We also give an example in Section 4.5.2 where CCD is twice faster from any arbitrary initialization with probability one.*

The following corollary states that the asymptotic worst-case convergence rate of CCD is more than twice larger than that of RCD for quadratic problems whose Hessian is a 2-cyclic matrix. This corollary directly follows by Theorem 4.6 and definitions (4.1.2)-(4.1.3).

Corollary 4.8. *Suppose Assumption 1 holds and A is a consistently ordered 2-cyclic matrix. Then, for the constant $\nu_n > 1$ as defined in* (4.2.10)*, the asymptotic worst-case rate of CCD and RCD satisfies*

$$\frac{\text{Rate(CCD)}}{\text{Rate(RCD)}} = 2\nu_n, \quad \textit{where} \quad \nu_n := \frac{\log(1-\mu)}{n\log\left(1-\frac{\mu}{n}\right)}. \tag{4.2.3}$$

In the following remark, we highlight several properties of the constant ν_n.

Remark 4.9. *ν_n is a monotonically increasing function of n over the interval $[1, \infty)$, where $\nu_1 = 1$ and $\lim_{n\to\infty} \nu_n = \frac{-\log(1-\mu)}{\mu} > 1$. Furthermore, $\lim_{\mu\to 0^+} \nu_n = 1$.*

4.2.2 Convergence Rate of CCD for Irreducible M-Matrices

In this section, we first define the class of M-matrices and then present the convergence rate of the CCD algorithm applied to quadratic problems whose Hessian is an M-matrix.

Definition & Properties

Definition 4.10 (M-matrix)**.** *A real matrix A with $A_{i,j} \leq 0$ for all $i \neq j$ is an M-matrix if A has the decomposition $A = sI - B$ such that $B \geq 0$ and $s \geq \rho(B)$.*

We emphasize that M-matrices arise in a variety of applications such as belief propagation over Gaussian graphical models [99] and distributed control of positive systems [121], and has been used to analyze performance of various algorithms in the literature [23, 132, 144]. Furthermore, graph Laplacians are M-matrices, therefore solving linear systems with M-matrices (or equivalently solving (4.0.1) for an M-matrix A) arise in a variety of applications for analyzing random walks over graphs and distributed optimization and consensus problems over graphs (cf. [79] for a survey). For quadratic problems, the Hessian is an M-matrix if and only if the gradient descent mapping is an isotone operator [23, 132] and in Gaussian graphical models, M-matrices are often referred as attractive models [99].

In the following lemma, we highlight a property of non-singular M-matrices, which we will use in the following section to characterize the convergence rate of the CCD method applied to quadratic problems whose Hessian is an M-matrix.

Lemma 4.11 ([117, Theorem 2])**.** *A is a nonsingular M-matrix if and only if A^{-1} exists and $A^{-1} \geq 0$.*

Before concluding this section, we introduce the following lemma, which is presented in variuos papers (e.g., [147, Lemma 4.12], [109, Corollary 1.2], [78, Theorem 1]) to analyze the spectral radii of nonnegative matrices. Particularly, this lemma states that if the matrix $e^{\alpha}N+e^{-\alpha}N^{\top}$ is not consistently ordered (where $N \geq 0$ is a strictly lower triangular matrix), then its spectral radius is strictly log-convex in α. The proof of this lemma is presented in Section 4.5.1 for completeness.

Lemma 4.12. *Let $B_{\alpha} = e^{\alpha}N + e^{-\alpha}N^{\top}$, where $N \geq 0$ is a strictly lower triangular matrix and $\alpha \in \mathbb{R}$. Then, either $\rho(B_{\alpha})$ is strictly log-convex in α with $\rho(B_{\alpha}) > \rho(B_0)$ for all $\alpha \neq 0$ or $\rho(B_{\alpha})$ is constant for all $\alpha \in \mathbb{R}$ (i.e., B_{α} is a consistently ordered matrix).*

Convergence Rates

In the following theorem, we provide lower and upper bounds on the spectral radius of the iteration matrix of CCD for quadratic problems whose Hessian matrix is an irreducible

M-matrix. In particular, we show that the spectral radius of the iteration matrix of CCD is strictly smaller than the one of RCD for irreducible M-matrices. Note that the Hessian matrix in our motivating example (in Section 4.1.2) is an irreducible M-matrix.

Theorem 4.13. *Suppose Assumption 1 holds, A is an irreducible M-matrix and $n \geq 2$. Then, the iteration matrix of the CCD algorithm $C = (I - N)^{-1}N^\top$ satisfies the following inequality*

$$(1-\mu)^2 \leq \rho(C) \leq \frac{1-\mu}{1+\mu}, \tag{4.2.4}$$

where the inequality on the left holds with equality if and only if A is a consistently ordered matrix.

Proof Since A is an M-matrix, $I - N$ is an M-matrix as well. Consequently, $(I-N)^{-1} \geq 0$, which implies $C = (I - N)^{-1}N^\top \geq 0$ by Lemma 4.11. Then, by Perron-Frobenius Theorem, there exists a real eigenvalue of C, denoted by λ, and the corresponding unit-norm eigenvector $z \geq 0$ satisfying $\lambda = \rho(C) \geq 0$ and

$$Cz = \lambda z.$$

Multiplying both sides of the above equality by $I - N$ from the left, we obtain

$$N^\top z = \lambda(I - N)z,$$

and rearranging terms yields

$$(\lambda N + N^\top)z = \lambda z. \tag{4.2.5}$$

Therefore, λ is an eigenvalue of the matrix $\lambda N + N^\top$. We then observe that $\lambda N + N^\top$ is an irreducible matrix as A is irreducible as the indices of the nonzero entries of both matrices are the same. Since $\lambda N + N^\top$ is nonnegative and irreducible and z is nonnegative, then by Perron-Frobenius Theorem, z is the eigenvector corresponding to the spectral radius of

$\lambda N + N^\top$. Therefore,

$$\lambda = \rho(\lambda N + N^\top) = \sqrt{\lambda}\,\rho\Big(\sqrt{\lambda}N + \frac{1}{\sqrt{\lambda}}N^\top\Big). \tag{4.2.6}$$

In order to obtain a lower bound on the right-hand side of (4.2.6), we use Lemma 4.12 (note that $\lambda < 1$ by Definition 4.10) and conclude that

$$\lambda \geq \sqrt{\lambda}\,\rho\big(N + N^\top\big), \tag{4.2.7}$$

with equality if and only if A is a consistently ordered matrix. Since $\lambda = \rho(C)$, (4.2.7) yields

$$\rho(C) \geq \rho^2\big(N + N^\top\big) = \rho^2(I - A) = (1-\mu)^2,$$

with equality if and only if A is a consistently ordered matrix, which concludes the proof of the lower bound in (4.2.4). In order to obtain an upper bound on $\rho(C)$, we turn our attention back to (4.2.5) and multiply both sides by $z^\top$ from the left. This yields

$$\lambda z^\top N z + z^\top N^\top z = \lambda,$$

since $\|z\| = 1$. Noting that $z^\top N z = z^\top N^\top z$ and defining $\beta = z^\top N z$, we obtain

$$\lambda = \frac{\beta}{1-\beta}. \tag{4.2.8}$$

Since $\rho(N + N^\top) = \rho(I - A) = 1 - \mu$, then for any $\|y\| = 1$, we have $y^\top(N + N^\top)y \leq 1 - \mu$. Picking $y = z$ in this inequality yields $2\beta \leq 1 - \mu$ and combining this with (4.2.8) and noting $\lambda = \rho(C)$ imply the upper bound in (4.2.4). □

An immediate consequence of Theorem 4.13 is that for quadratic problems whose Hessian is an irreducible M-matrix, the best cyclic order that should be used in CCD can be characterized as follows.

Remark 4.14. *Throughout the text, we considered the CCD method that follows the standard cyclic order* $(1, 2, \ldots, n)$*. However, we can construct a CCD method that follows an alternative deterministic order by considering a permutation* π *of* $\{1, 2, \ldots, n\}$*, and choosing the coordinates according to the order* $(\pi(1), \pi(2), \ldots, \pi(n))$ *instead. For any given order* π*, (4.0.1) can be reformulated as follows*

$$\min_{x_\pi \in \mathbb{R}^n} \frac{1}{2} x_\pi^\top A_\pi x_\pi, \quad \textit{where} \quad A_\pi := P_\pi A P_\pi^\top \quad \textit{and} \quad x_\pi = P_\pi x,$$

where P_π *is the corresponding permutation matrix of* π*. Supposing that Assumption 1 holds, the corresponding CCD iterations for this problem can be written as follows*

$$x_\pi^{(\ell+1)n} = C_\pi \, x_\pi^{\ell n}, \quad \textit{where} \quad C_\pi = (I - N_\pi)^{-1} N_\pi^\top \quad \textit{and} \quad N_\pi = P_\pi L P_\pi.$$

If A *is an irreducible M-matrix and satisfies Assumptions 1, then so does* A_π*. Consequently, Theorem 4.13 yields the same upper and lower bounds (in (4.2.4)) on* $\rho(C_\pi)$ *as well, i.e., the spectral radius of the iteration matrix of CCD with any cyclic order* π *satisfies*

$$(1 - \mu)^2 \leq \rho(C_\pi) \leq \frac{1 - \mu}{1 + \mu}, \tag{4.2.9}$$

where the inequality on the left holds with equality if and only if A_π *is a consistently ordered matrix. Therefore, if a consistent order* π^* *exists, then the CCD method with the consistent order* π^* *attains the smallest spectral radius (or equivalently, the fastest asymptotic worst-case convergence rate) among the CCD methods with any cyclic order.*

Remark 4.15. *The irreducibility of* A *is essential to derive the lower bound in (4.2.4) of Theorem 4.13. However, the upper bound in (4.2.4) holds even when* A *is a reducible matrix.*

We next compare the spectral radii bounds for CCD (given in Theorem 4.13) and RCD (given in Lemma 4.2). Since $\mu > 0$, the right-hand side of (4.2.4) can be relaxed to $(1 - \mu)^2 \leq \rho(C) < 1 - \mu$. A direct consequence of this inequality is the following corollary, which states that the asymptotic worst-case rate of CCD is strictly better than that of RCD at least by a factor that is strictly greater than 1.

Corollary 4.16. *Suppose Assumption 1 holds, A is an irreducible M-matrix and $n \geq 2$. Then, the asymptotic worst-case rate of CCD and RCD satisfies*

$$1 < \nu_n < \frac{\text{Rate(CCD)}}{\text{Rate(RCD)}} \leq 2\nu_n, \quad \textit{where} \quad \nu_n := \frac{\log(1-\mu)}{n\log\big(1-\frac{\mu}{n}\big)}, \tag{4.2.10}$$

and the inequality on the right holds with equality if and only if A is a consistently ordered matrix.

In the following corollary, we highlight that as the smallest eigenvalue of A goes to zero, the asymptotic worst-case rate of the CCD algorithm becomes twice the asymptotic worst-case rate of the RCD algorithm.

Corollary 4.17. *Suppose Assumption 1 holds, A is an irreducible M-matrix and $n \geq 2$. Then, we have*

$$\lim_{\mu \to 0^+} \frac{\text{Rate(CCD)}}{\text{Rate(RCD)}} = 2.$$

Proof By Theorem 4.13, we have the following worst-case asymptotic rate bounds for the CCD algorithm

$$-\log(1-\mu) + \log(1+\mu) \leq \text{Rate(CCD)} \leq -2\log(1-\mu).$$

Dividing both sides of the above inequality by $-\log(1-\mu)$, we obtain

$$1 - \frac{\log(1+\mu)}{\log(1-\mu)} \leq \frac{\text{Rate(CCD)}}{-\log(1-\mu)} \leq 2.$$

Taking limit of both sides as $\mu \to 0^+$ yields

$$\lim_{\mu \to 0^+} \frac{\text{Rate(CCD)}}{-\log(1-\mu)} = 2. \tag{4.2.11}$$

By Lemma 4.2, we have the following asymptotic worst-case rate for the RCD algorithm

$$\text{Rate(RCD)} = -n \log\Big(1 - \frac{\mu}{n}\Big).$$

Dividing both sides of the above inequality by $-\log(1-\mu)$ and taking limit of both sides as $\mu \to 0^+$, we get

$$\lim_{\mu \to 0^+} \frac{\text{Rate(RCD)}}{-\log(1-\mu)} = 1. \tag{4.2.12}$$

Combining (4.2.11) and (4.2.12) concludes the proof. □

4.2.3 Convergence Rate of CCD for Non-frustrated Matrices

In this section, we define the class of non-frustrated matrices and present the convergence rate of the CCD algorithm applied to quadratic problems whose Hessian is a non-frustrated matrix.

Definition & Properties

Definition 4.18. *A real matrix A is called a non-frustrated matrix if A has the decomposition $A = I - B$ such that B does not contain any frustrated cycles, i.e., cycles with an odd number of negative edge weights.*

The class of non-frustrated matrices are highly related to the class of M-matrices as we highlight in the following lemma. It states that any non-frustrated matrix is sign-similar to an M-matrix.

Lemma 4.19 ([59])**.** *Let $S(B)$ be the signed digraph of B, i.e., $S(B) = \text{sign}(B)$. If B is irreducible and all cycles of $S(B)$ are positive, then B is sign-similar to a non-negative matrix, i.e., $B = DED^{-1}$, where $E \geq 0$ and D is a diagonal matrix with entries ± 1.*

Convergence Rates

Using Lemma 4.19 and Theorem 4.13, we show that the same convergence rate guarantees in Theorem 4.13 (for M-matrices) hold for the non-frustrated matrices as well.

Theorem 4.20. *Suppose Assumption 1 holds, A is an irreducible non-frustrated matrix and $n \geq 2$. Then, the iteration matrix of the CCD algorithm $C = (I - N)^{-1}N^\top$ satisfies the following inequality*

$$(1-\mu)^2 \leq \rho(C) \leq \frac{1-\mu}{1+\mu},$$

where the inequality on the left holds with equality if and only if A is a consistently ordered matrix.

Proof Since A is assumed to be an irreducible non-frustrated matrix, then by Definition 4.18 and Lemma 4.19, A is sign-similar to $\bar{A}$ (i.e., $A = D\bar{A}D^{-1}$ for some diagonal matrix D whose entries are ±1.), where $\bar{A}$ is the comparison matrix of A defined as

$$\bar{A}_{i,j} = \begin{cases} A_{i,j} & \text{, if } i = j \\ -|A_{i,j}| & \text{, else.} \end{cases} \tag{4.2.13}$$

Let $\bar{A} = I - \bar{N} - \bar{N}^\top$ be the decomposition of $\bar{A}$ such that $\bar{N}$ is a strictly lower triangular matrix. Then, by Theorem 4.13, we conclude that

$$(1-\mu)^2 \leq \rho(\bar{C}) \leq \frac{1-\mu}{1+\mu},$$

where the inequality on the left holds with equality if and only if $\bar{A}$ is a consistently ordered matrix. To conclude the proof, we claim that C is sign-similar to $\bar{C}$, which follows since

$$\begin{aligned} C &= (I - N)^{-1} N^\top \\ &= (D(I - \bar{N})D)^{-1}(D\bar{N}D)^\top \\ &= D(I - \bar{N})^{-1} D^2 \bar{N}^\top D \\ &= D(I - \bar{N})^{-1} \bar{N}^\top D \\ &= D\bar{C}D, \end{aligned}$$

where the equalities follow since D is a Householder matrix, i.e., $D = D^{-1}$ and $D^2 = I$. Hence, C is sign-similar to $\bar{C}$ and consequently $\rho(C) = \rho(\bar{C})$, which concludes the proof. □

4.3 Numerical Validation

In this section, we compare the performance of CCD and RCD through numerical examples. First, we consider the quadratic optimization problem in (4.0.1), where A is an $n \times n$ matrix defined as follows

$$A = I - N - N^\top, \quad \text{where} \quad N = \frac{1}{n}\begin{bmatrix} 0 & 0 \\ \mathbf{1}_{\frac{n}{2}\times\frac{n}{2}} & 0 \end{bmatrix}, \tag{4.3.1}$$

and $\mathbf{1}_{\frac{n}{2}\times\frac{n}{2}}$ is the $\frac{n}{2} \times \frac{n}{2}$ matrix with all entries equal to 1. Here, it can be easily checked that A is a consistently ordered 2-cyclic matrix. By Theorem 4.6 and Corolloary 4.8, the worst-case convergence rate of CCD on this example is

$$2\nu_m = 2\frac{\log(1-\mu)}{m\log\left(1 - \frac{\mu}{m}\right)} = \frac{\log(0.5)}{50\log\left(1 - \frac{1}{200}\right)} \approx 2.77$$

times faster than the convergence rate of RCD asymptotically. This is illustrated on the left panel of Figure 4-1, where the distance to the optimal solution is plotted in a logarithmic

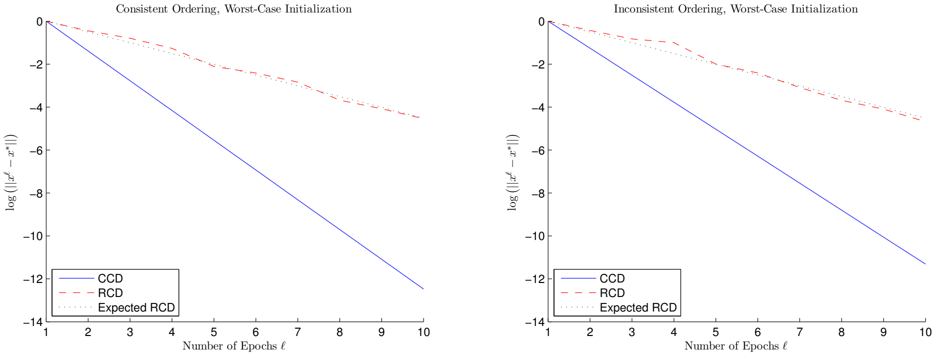

Figure 4-1: Distance to the optimal solution of the iterates of CCD and RCD for the cyclic matrix in (4.3.1) (left figure) and a randomly permuted version of the same matrix (right figure) where the y-axis is on a logarithmic scale. The left (right) panel corresponds to the consistent (inconsistent) ordering for the same quadratic optimization problem.

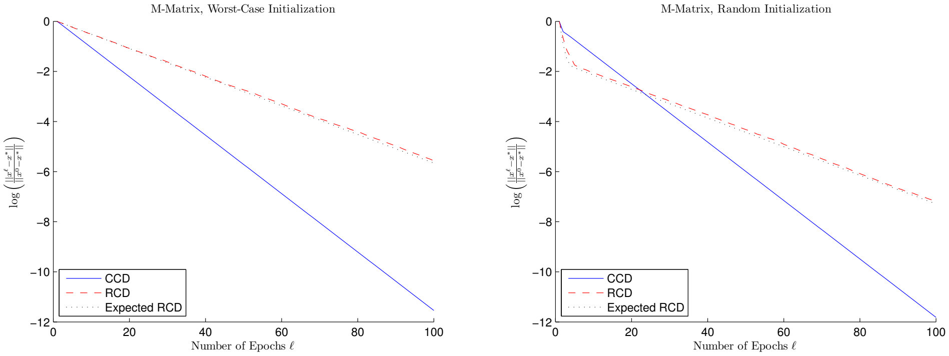

Figure 4-2: Distance to the optimal solution of the iterates of CCD and RCD for the M-matrix matrix in (4.3.2) for the worst-case initialization (left figure) and a random initialization (right figure).

scale over epochs. Note that even if our results our asymptotic, we see the same difference in performances on the early epochs (for small ℓ). On the other hand, when the matrix A is not consistently ordered, according to Theorem 4.13, CCD is still faster but the difference in the convergence rates decreases with respect to the consistent ordering case. To illustrate this, we need to generate an inconsistent ordering of the matrix A. For this goal, we generate a random permutation matrix P and replace A with $A_P := PAP^\top$ in the optimization problem (4.0.1). The right panel in Figure 4-1 shows that for this inconsistent ordering CCD is still faster compared to RCD, but not as fast (the slope of the decay of error line in blue marker is less steep) predicted by our theory.

We next consider the case that A is an irreducible positive definite M-matrix. In particular, we consider the matrix

$$A = (1+\delta)I - \delta \mathbf{1}_{n\times n}, \tag{4.3.2}$$

where $\mathbf{1}_{n\times n}$ is the $n \times n$ matrix with all entries equal to 1 as before and $\delta = \frac{1}{n+5}$. We set $n = 100$ and plot the performance of CCD and RCD methods for the quadratic problem defined by this matrix. In Figure 4-2, we compare the convergence rate of CCD and RCD for an initial point that corresponds to a worst-case (left figure) and for a random choice of an initial point (right figure). We conclude that the asymptotic rate of CCD is faster than that of RCD demonstrating our results in Theorem 4.13 and Corolloary 4.16.

4.4 Discussion

In this chapter, we compared the CCD and RCD methods on a class of quadratic problems. We showed by a novel analysis that for this problem class, the CCD method is always faster than the RCD method in terms of the worst-case asymptotic rate. We also gave a characterization of the best cyclic order to follow in the CCD method. We showed that using the best cyclic order the CCD method can converge more than twice as fast as the RCD method. Finally, we verified the tightness of our results through numerical experiments.

4.5 Additional Proofs

4.5.1 Proof of Lemma 4.12

Suppose the largest eigenvalue of B_α has a multiplicity of 1. Then,

$$\rho(B_\alpha) = \lim_{t\to\infty} [\mathrm{tr}\,(B_\alpha)^t]^{1/t}. \tag{4.5.1}$$

In order to find the diagonal entries of $(B_\alpha)^t$, we consider the graph generated by the matrix B_α and define the weight of a walk as the product of the weights of the corresponding edges in the walk. We then observe that the ith diagonal of the matrix $(B_\alpha)^t$ can be written as the summation of weights of all closed walks of length t (from the ith node to itself). In particular, consider a valid closed walk w that contains edges $(i_s, i_{s+1})_{s=0}^{t-1}$ such that $i_0 = i_t = i$ and $[B_\alpha]_{i_s,i_{s+1}} > 0$ for all s. Then, we can define a symmetric walk w' with edges $(i_{s+1}, i_s)_{s=0}^{t-1}$ and the ith diagonal entry of $(B_\alpha)^t$ contains the weights of both w and w' as summands. Furthermore, the weight of the walk w can be written as $\phi_\alpha(w) = e^{c_w \alpha}\phi_0(w)$, for some integer c_w, where

$$\phi_0(w) = \prod_{s=0}^{t-1} [B_0]_{i_s,i_{s+1}}.$$

The weight of the symmetric walk w' is then found by $\phi_\alpha(w') = e^{-c_w\alpha}\phi_0(w)$ since B_0 is symmetric. Therefore, the ith diagonal entry of $(B_\alpha)^t$ can be found as follows

$$[(B_\alpha)^t]_{i,i} = \sum_{\text{all valid walks } w} \frac{e^{c_w\alpha} + e^{-c_w\alpha}}{2} \phi_0(w).$$

It is easy to observe that $\cosh(c_w\alpha) = \frac{e^{c_w\alpha}+e^{-c_w\alpha}}{2}$ is a strictly log-convex function of α for any $c_w \neq 0$. Thus, if there exists a walk w for which $c_w \neq 0$, then $\operatorname{tr}(B_\alpha)^t$ is a strictly log-convex function of α since $\phi_0(w) > 0$ for all valid walks. On the other hand, $\operatorname{tr}(B_\alpha)^t$ is constant in α if and only if $c_w = 0$ for all valid walks, which implies that the graph is bipartite since starting from an arbitrary node i it is not possible to return back to node i in odd number of steps. This together with (4.5.1) imply the statement of the lemma.

For the case the largest eigenvalue of B_α has a multiplicity of at least 2, we consider the matrix $\tilde{B}_\alpha(\epsilon) = B_\alpha + \epsilon I$, whose largest eigenvalue has a multiplicity of 1 for any $\epsilon > 0$. Using the same arguments as above, we can conclude that the statement of the lemma holds for any $\tilde{B}_\alpha(\epsilon)$ with $\epsilon > 0$ and taking the limit as $\epsilon \to 0^+$ concludes the proof of the lemma.

4.5.2 An Example Achieving Lower and Upper Bounds

Consider solving the linear system $Ax = 0$ where A is defined as follows

$$A = \begin{bmatrix} 1 & -\delta \\ -\delta & 1 \end{bmatrix}$$

for some $\delta \in (0, 1)$. The CCD algorithm applied to this problem has the following iteration matrix

$$C = \begin{bmatrix} 0 & \delta \\ 0 & \delta^2 \end{bmatrix},$$

whereas the expected RCD iteration matrix is

$$R = \left(I - \frac{A}{2}\right)^2 = \begin{bmatrix} 1/2 & \delta/2 \\ \delta/2 & 1/2 \end{bmatrix}^2 = \frac{1}{4}\begin{bmatrix} 1+\delta^2 & 2\delta \\ 2\delta & 1+\delta^2 \end{bmatrix}.$$

The eigendecomposition of this matrix can be found as follows

$$R = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1+\delta}{2} & 0 \\ 0 & \frac{1-\delta}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}^{-1}.$$

Therefore, after ℓ epochs the distance of the iterates generated by RCD starting from the initial point $x^0 = [a, b]^\top$ becomes

$$\begin{aligned} \mathbb{E}\|x^\ell - x^*\| = \mathbb{E}\|x^\ell\| \geq \|\mathbb{E}x^\ell\| = \|R^\ell x^0\| = \| &\begin{bmatrix} \left(\frac{1+\delta}{2}\right)^\ell a \\ \left(\frac{1-\delta}{2}\right)^\ell b \end{bmatrix} \| \\ &= \sqrt{\left(\frac{1+\delta}{2}\right)^{2\ell} a^2 + \left(\frac{1-\delta}{2}\right)^{2\ell} b^2}. \\ &\geq \left(\frac{1+\delta}{2}\right)^\ell |a| \\ &\geq \delta^\ell |a|. \end{aligned}$$

Therefore, in order to achieve a solution in the ϵ-neighborhood of the optimal solution $x^* = 0$, i.e., to attain $\|x^\ell - x^*\| = \epsilon$, the RCD method requires

$$N_R(\epsilon) \geq \frac{\log \epsilon}{\log \delta} - \frac{\log |a|}{\log \delta}$$

epochs, for any $a \neq 0$.

On the other hand, for the CCD algorithm, we have

$$C^\ell = \begin{bmatrix} 0 & \delta^{2\ell-1} \\ 0 & \delta^{2\ell} \end{bmatrix},$$

and consequently the suboptimality of the iterates generated by the CCD algorithm is

$$\|C^\ell x_0\| = \delta^{2\ell}\sqrt{b^2 + \frac{1}{\delta^2}b^2}.$$

Therefore, in order to achieve a solution in the ϵ-neighborhood of the optimal solution $x^* = 0$, i.e., to attain $\|x^\ell - x^*\| = \epsilon$, the CCD method requires

$$N_C(\epsilon) = \frac{\log \epsilon}{2\log \delta} - \frac{\log\left(b^2 + \frac{1}{\delta^2}b^2\right)}{4\log \delta}$$

epochs.

Note that for small ϵ the first terms in the expression of $N_J(\epsilon)$ and $N_C(\epsilon)$ are dominant. In particular we have,

$$\lim_{\epsilon \to 0^+} \frac{N_R(\epsilon)}{N_C(\epsilon)} =\geq \frac{2\log \delta}{\log \delta} = 2, \tag{4.5.2}$$

for any $a \neq 0$.

Chapter 5

Randomness and Permutations in the CD Method

In this chapter, we investigate the convergence rate of the RPCD method for a special class of quadratic problems we studied in Chapter 4. Interest in the RPCD method is motivated by both empirical observations and practical implementation: In many machine learning applications, RPCD is observed numerically to outperform its with-replacement sampling counterpart RCD [105, 124]. Moreover, without-replacement sampling-based algorithms (such as RPCD and random reshuffling [71, 146]) are often easier to implement efficiently than their with-replacement counterparts (such as RCD and stochastic gradient descent) [88, 124] as it requires sequential data access, in contrast to the random data access required by with-replacement sampling (see e.g. [29, 134]).

The organization of this chapter is as follows. In Section 5.1, we discuss the CCD, RCD and RPCD algorithms in more detail and describe the three criteria that are used for analyzing convergence throughout the chapter. In Section 5.3, we survey known results on the convergence rate of RPCD. We analyze the convergence rates of CCD, RCD, and RPCD with respect to the first convergence criterion in Section 5.4.1 and the behavior of RCD and RPCD with respect to the second and third convergence criteria in Section 5.4.2. We validate our theoretical results via numerical experiments in Section 5.5 and conclude the chapter in Section 5.6.

5.1 Preliminaries

In this chapter, we consider the quadratic problem in (4.0.1). Similar to the previous chapter, the update rule of the CCD method (with update order $1, \ldots, n$) over an epoch is given by

$$x_{\text{CCD}}^{(\ell+1)n} = B_{\text{CCD}}\, x_{\text{CCD}}^{\ell n}, \quad \text{where} \quad B_{\text{CCD}} = (D-N)^{-1}N^T, \tag{5.1.1}$$

and $A = D - N - N^T$ with D representing the diagonal part of A and $-N$ representing the strictly lower triangular part of A.

We next consider the CCD method with a given order π. We let P_π denote the permutation matrix corresponding to order π and split the permuted Hessian matrix as follows:

$$A_\pi = P_\pi^T A P_\pi = D_\pi - N_\pi - N_\pi^T, \tag{5.1.2}$$

where $-N_\pi$ is a strictly lower triangular matrix and D_π is a diagonal matrix. Then, similar to (5.1.1), we have

$$x_{\text{CCD-}\pi}^{(\ell+1)n} = B_{\text{CCD-}\pi}\, x_{\text{CCD-}\pi}^{\ell n}, \quad \text{where} \quad B_{\text{CCD-}\pi} = (D_\pi - N_\pi)^{-1}N_\pi^T. \tag{5.1.3}$$

Note that B_{CCD} and $B_{\text{CCD-}\pi}$ are not symmetric matrices as the first column of both matrices are zero, whereas the first row contains nonzero entries.

For the RCD method, the indices i_k are chosen independently at random at each iteration k. Denoting by x_{RCD}^k the k-th iterate generated by RCD, the update rule for RCD over a single iteration can be written as

$$x_{\text{RCD}}^{k+1} = B_{\text{RCD-}k}\, x_{\text{RCD}}^{k}, \quad \text{where} \quad B_{\text{RCD-}k} = I - \frac{1}{A_{i_k i_k}} e_{i_k} e_{i_k}^T A. \tag{5.1.4}$$

The expectation of $B_{\text{RCD-}k}$ with respect to the random variable i_k is denoted as follows:

$$B_{\text{RCD}} = \mathbb{E}_k B_{\text{RCD-}k}, \tag{5.1.5}$$

where we note that B_{RCD} is a symmetric matrix, by symmetry of A and uniform distribution of i_k.

For the RPCD algorithm, each coordinate is processed exactly once in each epoch according to a uniformly and independently chosen order. Recalling that π_ℓ denotes the permutation of coordinates used in epoch ℓ and using the iteration matrix corresponding to CCD-π_ℓ (see (5.1.3)), epoch ℓ of RPCD can be written as

$$x_{\text{RPCD}}^{(\ell+1)n} = B_{\text{RPCD-}\ell}\, x_{\text{RPCD}}^{\ell n}, \quad \text{where} \quad B_{\text{RPCD-}\ell} = P_{\pi_\ell} B_{\text{CCD-}\pi_\ell} P_{\pi_\ell}^T. \tag{5.1.6}$$

We introduce the following notation for the expected value of $B_{\text{RPCD-}\ell}$ with respect to permutation π_ℓ:

$$B_{\text{RPCD}} = \mathbb{E}_\ell B_{\text{RPCD-}\ell}, \tag{5.1.7}$$

where we note that B_{RPCD} is a symmetric matrix since π_ℓ is chosen uniformly at random over all permutations (see Lemma 5.5).

5.2 Convergence Rate Criteria

We next discuss how to measure and compare the convergence rates of different variants of CD. Three different improvement sequences have been used to measure the performance of CD methods in the literature:

(i) $\mathcal{I}_1(x_{\text{CD}}^k) = \|\mathbb{E}x_{\text{CD}}^k - x^*\|,$ (Distance of expected iterates)

(ii) $\mathcal{I}_2(x_{\text{CD}}^k) = \mathbb{E}\|x_{\text{CD}}^k - x^*\|^2,$ (Expected distance of iterates)

(iii) $\mathcal{I}_3(x_{\text{CD}}^k) = \mathbb{E}f(x_{\text{CD}}^k) - f(x^*).$ (Expected function value)

(see e.g. [19, 72, 108, 126, 137, 138, 150]). While these three measures can be related to each other (Jensen's inequality yields $\mathcal{I}_1^2 \leq \mathcal{I}_2$ and strong convexity enables lower and upper bounding $\mathcal{I}_3$ between constant positive multiples of $\mathcal{I}_2$), we will provide different analyses for each of the measures to obtain the tightest estimates.

In the above definitions, expectations can be removed for deterministic algorithms such as CCD. By Jensen's inequality, we have that $\mathcal{I}_1^2(x_{\text{CD}}^k) \leq \mathcal{I}_2(x_{\text{CD}}^k)$ for all k. For a strongly convex function f, $\mathcal{I}_3$ can be lower and upper bounded between constant positive multiples of $\mathcal{I}_2$.

To study convergence rate of the CCD, RCD and RPCD methods with respect to improvement sequence $\mathcal{I}_1$, we use the operators derived in the previous section that represent one iterate or one epoch. The iteration matrices of CCD and RPCD are defined over an epoch (see (5.1.1) for CCD, (5.1.6) and (5.1.7) for RPCD). Therefore, using the generic subscript "CD" to represent the cases $B_{\text{CD}} = B_{\text{CCD}}$ for CCD and $B_{\text{CD}} = B_{\text{RPCD}}$ for RPCD, we have the following update rule

$$\mathbb{E}_\ell x_{\text{CD}}^{(\ell+1)n} = B_{\text{CD}}\, x_{\text{CD}}^{\ell n},$$

where $\mathbb{E}_\ell$ denotes the expectation with respect to the random variables in epoch ℓ given $x_{\text{CD}}^{\ell n}$. Note that the random variables in each epoch are independent and identically distributed across different epochs for RPCD (and RCD). Therefore, by using the law of iterated expectations, we obtain

$$\mathbb{E} x_{\text{CD}}^{(\ell+1)n} = B_{\text{CD}}^\ell\, x^0,$$

where $\mathbb{E}$ here denotes the expectation with respect to *all* random variables arising in the algorithm. Hence, the *worst-case convergence rate* with respect to $\mathcal{I}_1$ can be expressed as

$$\sup_{x^0 \in \mathbb{R}^n} \left(\frac{\|\mathbb{E} x_{\text{CD}}^{\ell n}\|}{\|x^0\|} \right)^{1/\ell} = \sup_{x^0 \in \mathbb{R}^n} \left(\frac{\|B_{\text{CD}}^\ell\, x^0\|}{\|x^0\|} \right)^{1/\ell} = \|B_{\text{CD}}^\ell\|^{1/\ell}. \tag{5.2.1}$$

When B_{CD} is a symmetric matrix (as in RPCD), we have $\|B_{\text{CD}}^\ell\|^{1/\ell} = \rho(B_{\text{CD}})$. Hence, (5.2.1) yields a *per-epoch* worst-case convergence rate of $\rho(B_{\text{RPCD}})$ for RPCD. When B_{CD} is asymmetric (which is the case for CCD), we have by Gelfand's formula $\lim_{\ell\to\infty}\|B_{\text{CD}}^{\ell}\|^{1/\ell} = \rho(B_{\text{CD}})$. Thus, $\rho(B_{\text{CCD}})$ represents an *asymptotic* worst-case convergence rate measure for CCD.

For RCD, a similar derivation involving a single iteration (rather than one epoch) yields from (5.1.4) and (5.1.5) that

$$\mathbb{E}_k x_{\text{RCD}}^{k+1} = B_{\text{RCD}}\, x_{\text{CCD}}^{k}.$$

Similar reasoning to the above yields a *per-iteration* worst-case convergence rate of $\rho(B_{\text{RCD}})$, or equivalently a per-epoch rate of $\rho(B_{\text{RCD}})^n$, for RCD. (Note that, because B_{RCD} is symmetric, we have $\rho(B_{\text{RCD}}) = \|B_{\text{RCD}}\|$.)

In our analysis of convergence rate of RCD with respect to improvement sequence $\mathcal{I}_2$, it follows from (5.1.4) that

$$\begin{aligned}\mathbb{E}\|x_{\text{RCD}}^{k+1}\|^2 &= (x_{\text{RCD}}^{k})^T\mathbb{E}\big[(B_{\text{RCD-}k})^T B_{\text{RCD-}k}\big]x_{\text{RCD}}^{k}\\ &\le \|\mathbb{E}\big[(B_{\text{RCD-}k})^T B_{\text{RCD-}k}\big]\|\|x_{\text{RCD}}^{k}\|^2.\end{aligned}$$

For RPCD, we have similarly from (5.1.6) that

$$\begin{aligned}\mathbb{E}\|x_{\text{RPCD}}^{(\ell+1)n}\|^2 &= (x_{\text{RPCD}}^{\ell n})^T\mathbb{E}\big[(B_{\text{RPCD-}\ell})^T B_{\text{RPCD-}\ell}\big]x_{\text{RPCD}}^{\ell n}\\ &\le \|\mathbb{E}\big[(B_{\text{RPCD-}\ell})^T B_{\text{RPCD-}\ell}\big]\|\|x_{\text{RPCD}}^{\ell n}\|^2.\end{aligned}$$

The matrices $\mathbb{E}\big[(B_{\text{RCD-}k})^T B_{\text{RCD-}k}\big]$ and $\mathbb{E}\big[(B_{\text{RPCD-}\ell})^T B_{\text{RPCD-}\ell}\big]$ are both symmetric. Convergence rates be obtained from $\rho\big(\mathbb{E}\big[(B_{\text{RCD-}k})^T B_{\text{RCD-}k}\big]\big)$ and $\rho\big(\mathbb{E}\big[(B_{\text{RPCD-}\ell})^T B_{\text{RPCD-}\ell}\big]\big)$ (or equivalently from the norms of these matrices), the first being a per-iteration convergence rate for RCD under criterion $\mathcal{I}_2$, and the second being a per-epoch rate for RPCD under the same criterion. Results along these lines appear in Section 5.4.2.

Finally, in our analysis of convergence rate of RCD with respect to $\mathcal{I}_3$, iteration (5.1.4) yields

$$\begin{aligned}\mathbb{E}f(x_{\text{RCD}}^{k+1}) &= (x_{\text{RCD}}^{k})^T\mathbb{E}_k\big[(B_{\text{RCD-}k})^TAB_{\text{RCD-}k}\big]x_{\text{RCD}}^{k}\\ &= (A^{1/2}x_{\text{RCD}}^{k})^T\mathbb{E}_k\big[A^{-1/2}(B_{\text{RCD-}k})^TAB_{\text{RCD-}k}A^{-1/2}\big]A^{1/2}x_{\text{RCD}}^{k}\\ &\le \|\mathbb{E}_k\big[A^{-1/2}(B_{\text{RCD-}k})^TAB_{\text{RCD-}k}A^{-1/2}\big]\|\|A^{1/2}x_{\text{RCD}}^{k}\|^2.\end{aligned}$$

A similar analysis applied to the RPCD update formula (5.1.6) yields

$$\mathbb{E}f(x_{\text{RPCD}}^{(\ell+1)n}) \le \|\mathbb{E}_\ell\big[A^{-1/2}(B_{\text{RPCD-}\ell})^TAB_{\text{RPCD-}\ell}A^{-1/2}\big]\|\|A^{1/2}x_{\text{RPCD}}^{\ell n}\|^2.$$

We will show that the matrices in these two bounds are symmetric. Thus, our convergence rate characterizations for RCD and RPCD with respect to $\mathcal{I}_3$ (see Section 5.4.2) will involve the norms (equivalently, the spectral radii) of these two matrices.

Remark 5.1. *Note that for improvement sequence $\mathcal{I}_1$, the asymptotic worst-case convergence rate of the algorithm can be simply computed as the spectral radius of the expected iteration matrix. Furthermore, this bound is tight in the sense that there can be no smaller contraction rate c_1, for which an inequality of the type $\mathcal{I}_1(x_{CD}^{\ell n}) \le c_1^\ell\,\mathcal{I}_1(x^0)$ asymptotically holds for all $x^0 \in \mathcal{R}^n$. Therefore, in Section 5.4.1, we compare the worst-case convergence rates of CCD, RCD and RPCD with respect to $\mathcal{I}_1$ through a tight analysis (in Proposition 5.8). We analyze the ratio of the convergence rates of RCD and RPCD in Proposition 5.7. On the other hand, for improvement sequences $\mathcal{I}_2$ and $\mathcal{I}_3$, we consider per-iteration and per-epoch upper bounds that are not necessarily asymptotically tight. Using a similar argument to (5.2.1), we can formulate the worst-case contraction factors for $\mathcal{I}_2$ and $\mathcal{I}_3$, but they would involve computation of powers of matrices (e.g., $\mathbb{E}\big[(B_{CD\text{-}k}^\ell)^TB_{CD\text{-}k}^\ell\big]$ and $\mathbb{E}\big[A^{-1/2}(B_{CD\text{-}k}^\ell)^TAB_{CD\text{-}k}^\ell A^{-1/2}\big]$), which does not admit a closed form characterization. Hence, in Section 5.4.2, we compare the convergence rates of RCD and RPCD based on per-iteration and per-epoch improvement rates, as has been done previously in the literature [88, 89, 138].*

5.3 Prior work on the RPCD method

In this section, we survey the known results on the performance of RPCD. There are several recent works that study the effects of random permutations in the convergence behavior of CD methods [88, 89, 113, 138]. To unify the randomization parameters (in RCD and RPCD) and the component-wise Lipschitz constants in different papers, throughout this chapter we consider that Assumption 1 holds. Consequently, there exists a unique solution to (4.0.1), which is denoted by x^* throughout the chapter.

Oswald and Zhou [113] analyzed the effects of random permutations for the successive over-relaxation (SOR) method, which is equivalent to the CD method with exact line search for a particular choice of algorithm parameter. They consider quadratic problems whose Hessian matrix is positive semidefinite and present convergence guarantees for SOR iterations with random permutations, which implies the following guarantee on the performance of the RPCD method.

Theorem 5.2 ([113, Theorem 4])**.** *Let f be a quadratic function of the form (4.0.1) and suppose Assumption 1 holds. Then the RPCD algorithm enjoys the following guarantee*

$$\mathbb{E}f(x_{RPCD}^{\ell n}) - f(x^*) \leq \left(1 - \frac{\mu}{(1+L)^2}\right)^{\ell} \big(f(x^0) - f(x^*)\big) \tag{5.3.1}$$

Theorem 5.2 provides a convergence rate guarantee on the performance of RPCD for general quadratic functions. Under the same assumptions in Theorem 5.2, the best known upper bound on the performance of RCD is given by [108, Theorem 5]:

$$\begin{aligned}\mathbb{E}\left[\frac{1}{2}\|x_{\mathrm{RCD}}^k - x^*\|^2 + f(x_{\mathrm{RCD}}^k) - f(x^*)\right]& \\ \leq \left(1 - \frac{2\mu}{n(1+\mu)}\right)^k &\left(\frac{1}{2}\|x^0 - x^*\|^2 + f(x^0) - f(x^*)\right).\end{aligned} \tag{5.3.2}$$

This shows that the the upper bound on the performance of RCD per-epoch is approximately $\left(1 - \frac{2\mu}{n(1+\mu)}\right)^n \approx 1 - \frac{2\mu}{1+\mu}$, whereas it follows from (5.3.1) that the upper bound on the performance of RPCD can be as large as $1 - \frac{\mu}{(1+n)^2}$ since $L \leq \operatorname{tr} A = n$. These bounds suggest that RPCD may require $\mathcal{O}(n^2)$ times more iterations than RCD to return an ϵ-optimal solution. However, empirical results show that RPCD often outperforms RCD in machine learning applications [124, 28]. Furthermore, it has been conjectured that the expected performance of RPCD should be no worse than the expected performance of RCD [124] (see also [74, 154] for related work on this conjecture). This motivates to derive tight bounds for the convergence rate of RPCD and compare them with the known bounds on the convergence rate of RCD.

A similar phenomenon has been observed for CCD in comparison to RCD. In particular, the tightest known convergence rate results on the performance of CCD (see [19, 138, 137]) suggest that CCD may require $\widetilde{\mathcal{O}}(n^2)$ times more iterations than RCD to guarantee an ϵ-optimal solution. To understand this gap in the convergence rate bounds, Sun and Ye [138] focused on the quadratic problem in (4.0.1) with the following permutation invariant[1] Hessian matrix

$$A = \delta I + (1-\delta)\mathbf{1}\mathbf{1}^T, \quad \text{where} \quad \delta \in (0, n/(n-1)). \tag{5.3.3}$$

In particular, the authors considered a worst-case initialization and the case when δ is close to 0, for which $L = \mathcal{O}(n)$.[2] For this problem, they showed that CCD with the worst-case initialization indeed requires $\mathcal{O}(n^2)$ times more iterations than RCD to return an ϵ-optimal solution. They also provided rate comparisons between RPCD and CCD without providing a comparison between RPCD and RCD, which is presented in the following theorem.

Theorem 5.3 ([138, Proposition 3.4])**.** *Let $K_{\mathrm{CCD}}(\epsilon)$, $K_{\mathrm{RCD}}(\epsilon)$ and $K_{\mathrm{RPCD}}(\epsilon)$ be the minimum number of epochs for CCD, RCD and RPCD (respectively) to achieve (expected) relative error*

$$\frac{\|\mathbb{E}(x^k_{CD}) - x^*\|}{\|x^0 - x^*\|} \leq \epsilon,$$

for initial point $x^0 \in \mathbb{R}^n$ (for CCD, the expectation operator can be ignored). There exists a

[1] A is a permutation invariant matrix if $PAP^T = A$, for any permutation matrix P.

[2] Since A has two eigenvalues: $\delta + n(1-\delta)$ with multiplicity 1 and δ with multiplicity $n-1$, the Lipschitz constant becomes $L = \delta + n(1-\delta)$, for $\delta \leq 1$; and as $\delta \to 0$, $L \to n$.

quadratic problem, whose Hessian matrix A satisfies (5.3.3) *for some δ around zero, such that*

$$\frac{K_{\text{CCD}}(\epsilon)}{K_{\text{RCD}}(\epsilon)} \geq \frac{n^2}{2\pi^2} \approx \frac{n^2}{20}, \tag{5.3.4a}$$

$$\frac{K_{\text{CCD}}(\epsilon)}{K_{\text{RPCD}}(\epsilon)} \geq \frac{n(n+1)}{2\pi^2} \approx \frac{n(n+2)}{20}. \tag{5.3.4b}$$

Theorem 5.3 shows that the worst-case performance (in improvement sequence $\mathcal{I}_1$) of RPCD and RCD is $\mathcal{O}(n^2)$ times faster than that of CCD. In a follow-up work, Lee and Wright [88] considered the same problem as [138] (see (5.3.3)) for the small δ case and presented asymptotic and non-asymptotic analyses of RPCD with respect to improvement sequence $\mathcal{I}_3$, presented in the following theorem.

Theorem 5.4 ([88, Theorem 3.3])**.** *Consider the quadratic problem* (4.0.1) *with the Hessian matrix A given by* (5.3.3)*, where $\delta \in (0, 0.4)$ and $n \geq 10$. For any $x^0 \in \mathbb{R}^n$, RPCD has the following non-asymptotic convergence guarantee*

$$\mathbb{E}f(x^{\ell n}_{RPCD}) - f(x^*) \leq (1 - 2\delta + 4\delta^2)^{\ell} R_0,$$

where R_0 is a constant depending on x_0 and δ. Furthermore, RPCD iterates enjoy an asymptotic convergence rate of

$$\lim_{\ell \to \infty} \left(\mathbb{E}f(x^{\ell n}_{RPCD}) - f(x^*)\right)^{1/\ell} = 1 - 2\delta - \frac{2\delta}{n} + 2\delta^2 + \mathcal{O}\left(\frac{\delta^2}{n}\right) + \mathcal{O}\big(\delta^3\big).$$

Theorem 5.4 shows that for the particular class of quadratic problems whose Hessian matrix satisfies (5.3.3), the convergence rate (in improvement sequence $\mathcal{I}_3$) of RPCD is faster than that of RCD in (5.3.2) in terms of the best known upper bounds (note that the convergence rate of RCD is approximately $1 - 2\delta/(1+\delta)$ for this case, see (5.3.2)). This is the first theoretical evidence that supports the empirical results showing RPCD often outperforms RCD [124]. In a follow-up work [89], Lee and Wright generalize the results of

Theorem 5.4 to quadratic problems, whose Hessian matrix satisfies

$$A = \delta I + (1-\delta)uu^T, \quad \text{where} \quad \delta \in (0, n/(n-1)), \tag{5.3.5}$$

where $u \in \mathbb{R}^n$ is a vector with elements of size $\mathcal{O}(1)$ (this generalizes (5.3.3) that corresponds to $u = \mathbf{1}$). The conclusions are similar to [88], but the analysis is different because A is no longer a permutation-invariant matrix.

5.4 Performance of RPCD vs RCD on a class of diagonally dominant matrices

As described in the previous section, the existing works [88, 138] analyze the performance of the RPCD method for quadratic problems, whose Hessian satisfies (5.3.3) for small δ. Here, we consider the other extreme, i.e., the $\delta > 1$ case, and provide tight convergence rate comparisons between RPCD, RCD and CCD with respect to all there improvement sequences defined in Section 5.2. In deriving convergence rate guarantees, we do not resort to the tools that are used in the earlier works on RPCD [88, 89, 138]. Instead, we present a novel analysis based on Perron-Frobenius theory that enables us to compute convergence rate bounds for all three criteria. For notational simplicity, we introduce the reformulation $\alpha = \delta - 1$, which yields

$$A = (1+\alpha)I - \alpha\mathbf{1}\mathbf{1}^T, \quad \text{where} \quad \alpha \in (0, 1/(n-1)). \tag{5.4.1}$$

It is simple to check that A has one eigenvalue at $1-(n-1)\alpha$ with the corresponding eigenvector $\mathbf{1}$ and other $n-1$ eigenvalues equal to $1+\alpha$. In particular, as α goes to zero, the condition number of A gets smaller and in the limit A is the identity matrix. On the other hand, as $\alpha \to \frac{1}{n-1}$, the matrix gets ill-conditioned. Therefore, the parameter

$$t := \max_i \frac{\sum_{j \neq i} A_{ij}}{A_{ii}} = \alpha(n-1) \in (0,1) \tag{5.4.2}$$

is a measure of diagonal dominance. In the remainder of this section, we analyze the performance of RPCD, RCD and CCD in improvement sequence $\mathcal{I}_1$ and the performance of RPCD and RCD in improvement sequences $\mathcal{I}_2$ and $\mathcal{I}_3$ with respect to this diagonal dominance measure.

5.4.1 Convergence rates of RPCD, RCD and CCD in improvement sequence $\mathcal{I}_1$

In this section, we compare convergence rates of RPCD, RCD and CCD, where improvement sequence $\mathcal{I}_1(x^k) = \|\mathbb{E}x^k - x^*\|$ is chosen as the convergence criterion (as in Theorem 5.3). As we highlighted in Section 5.2, we first compute the expected iteration matrices of the RPCD and RCD algorithms, and show that they are symmetric. Then, we compute their spectral radii to conclude the per-epoch worst-case convergence rate of RPCD and RCD, and analyze their ratio in Proposition 5.7. We also show that the asymptotic worst-case convergence rate of CCD is faster than that of RPCD and RCD in Proposition 5.8.

We begin our discussion by writing the expected RPCD iterates (see (5.1.6) and (5.1.7)) as follows

$$\mathbb{E}_\ell x_{\text{RPCD}}^{(\ell+1)n} = B_{\text{RPCD}}\, x_{\text{RPCD}}^{\ell n}. \tag{5.4.3}$$

Note that since the Hessian matrix A is permutation invariant, the iteration matrix of the CCD-π algorithm for any cyclic order π is equal to the iteration matrix of the standart CCD algorithm, i.e., $B_{\text{CCD}} = B_{\text{CCD-}\pi}$ for all orders π. Therefore, we have $B_{\text{RPCD}} = \mathbb{E}_\pi[P_\pi B_{\text{CCD}} P_\pi^T] = \mathbb{E}_P[P B_{\text{CCD}} P^T]$, where we drop the subscript π from the matrices for notational simplicity. In order to obtain a formula for B_{RPCD}, we first reformulate the

CCD iteration matrix in (5.1.1) as follows

$$B_{\mathrm{CCD}} = (I - N)^{-1} N^T = I - (I - N)^{-1}(I - N - N^T) = I - \Gamma^{-1} A,$$

where $\Gamma = I - N$. Using this reformulation, the expected iteration matrix of RPCD can computed as follows

$$B_{\mathrm{RPCD}} = \mathbb{E}_P\big[P B_{\mathrm{CCD}} P^T\big] = \mathbb{E}_P\big[P(I - \Gamma^{-1}A)P^T\big] = I - \mathbb{E}_P\big[P\Gamma^{-1}P^T\big]A,$$

where we used the fact that $PP^T = I$ and $AP^T = P^T A$. For the case the Hessian matrix A satisfies (5.4.1), Γ^{-1} can be explicitly computed as

$$\Gamma^{-1} = \mathrm{toeplitz}(c, r), \tag{5.4.4}$$

where toeplitz(c, r) denotes the Toeplitz matrix with the first column c and the first row r, which are given by

$$c = \Big[1, \quad \alpha, \quad \alpha(1+\alpha), \quad \alpha(1+\alpha)^2, \quad \ldots, \quad \alpha(1+\alpha)^{n-2}\Big]^T, \quad r = [1, 0, 0, \ldots, 0].$$

In order to compute $\mathbb{E}_P\big[P\Gamma^{-1}P^T\big]$, we use the following lemma, which states that expectation over all permutations separately averages the diagonal and off-diagonal entries of the permuted matrix.

Lemma 5.5 ([88, Lemma 3.1])**.** *Given any matrix $Q \in \mathcal{R}^{n\times n}$ and permutation matrix P selected uniformly at random from the set of all permutations, we have*

$$\mathbb{E}_P[PQP^T] = \tau_1 I + \tau_2 \mathbf{1}\mathbf{1}^T,$$

where

$$\tau_2 = \frac{\mathbf{1}^T Q \mathbf{1} - \mathrm{tr}(Q)}{n(n-1)} \quad \textit{and} \quad \tau_1 = \frac{\mathrm{tr}(Q)}{n} - \tau_2. \tag{5.4.5}$$

Letting $Q = \Gamma^{-1}$ in Lemma 5.5, we observe that the matrix $\mathbb{E}_P[P\Gamma^{-1}P^T]$ has diagonals equal to one and all the off-diagonal entries equal to each other:

$$\mathbb{E}_P[P\Gamma^{-1}P^T] = (1-\gamma)I + \gamma\mathbf{1}\mathbf{1}^T, \tag{5.4.6}$$

where γ can be found as the average of the off-diagonal entries of Γ^{-1}. The following lemma (whose proof is given in Section 5.7.1) provides an explicit expression for γ.

Lemma 5.6. *For any $\alpha \in (0, 1/(n-1))$, we have*

$$\gamma = \frac{(1+\alpha)^n - \alpha n - 1}{\alpha n(n-1)},$$

where γ denotes the off-diagonal entries of $\mathbb{E}_P[P\Gamma^{-1}P^T]$ in (5.4.6).

Using Lemma 5.6, it follows from the definition of A in (5.4.1) and equation (5.4.6) that

$$B_{\text{RPCD}} = I - \mathbb{E}_P[P\Gamma^{-1}P^T]A = ((n-1)\gamma - \beta)I + \beta\mathbf{1}\mathbf{1}^T,$$

where

$$\beta = \alpha - \gamma + \alpha\gamma(n-2).$$

Since B_{RPCD} is a symmetric matrix, then by (5.2.1), it suffices to compute the spectral radius of B_{RPCD} to obtain the worst-case performance of RPCD with respect to improvement sequence $\mathcal{I}_1$. To this end, we note that for any $\alpha \in (0, 1/(n-1))$, $B_{\text{RPCD}} > 0$ since $B_{\text{RPCD}} = \mathbb{E}_P[PB_{\text{CCD}}P^T]$ and $B_{\text{CCD}} \geq 0$ with at least one strictly positive entry in both the diagonal and off-diagonal parts (see also (5.7.13) for an explicit formula of B_{CCD}). Then, by the Perron-Frobenius Theorem [147, Lemma 2.8], we have

$$\begin{aligned}\rho(B_{\text{RPCD}}) &= \sum_{j=1}^{n} [B_{\text{RPCD}}]_{ij}, \quad \text{for all } i \in [n] \\ &= (n-1)(\gamma\alpha + \beta) \\ &= (n-1)(\alpha - \gamma + \alpha\gamma(n-1)) \\ &= 1 - [(1-\alpha(n-1))(1+\gamma(n-1))].\end{aligned}$$

Substituting the formula for γ from Lemma 5.6 above, we obtain the spectral radius of the RPCD iteration matrix as follows

$$\rho(B_{\text{RPCD}}) = 1 - (1-\alpha(n-1))\frac{(1+\alpha)^n - 1}{\alpha n} = 1 - \frac{1-t}{n}\left(\frac{\left(1+\frac{t}{n-1}\right)^n - 1}{\frac{t}{n-1}}\right), \tag{5.4.7}$$

where $t = \alpha(n-1)$ denotes the diagonal dominance factor (as defined in (5.4.2)).

For the RCD algorithm, on the other hand, we have (by (5.1.4) and (5.1.5)) the following expected iterates

$$\mathbb{E}_k x^{k+1}_{\text{RCD}} = B_{\text{RCD}}\, x^k_{\text{RCD}}, \quad \text{where} \quad B_{\text{RCD}} = I - \frac{1}{n}A.$$

Since A is a symmetric matrix, then by (5.2.1), the per-epoch worst-case asymptotic rate of RCD with respect to improvement sequence $\mathcal{I}_1$ can be found as

$$\rho(B_{\text{RCD}})^n = \left(1 - \frac{1}{n}\lambda_{\min}(A)\right)^n = \left(1 - \frac{1-t}{n}\right)^n.$$

In Proposition 5.7, we compare the performance of RPCD and RCD with respect to improvement sequence $\mathcal{I}_1$. To this end, we define

$$s(t,n) = \frac{-\log \rho(B_{\text{RPCD}})}{-\log \rho(B_{\text{RCD}})^n}, \tag{5.4.8}$$

(where log denotes the natural logarithm), which is equal to the ratio between the number

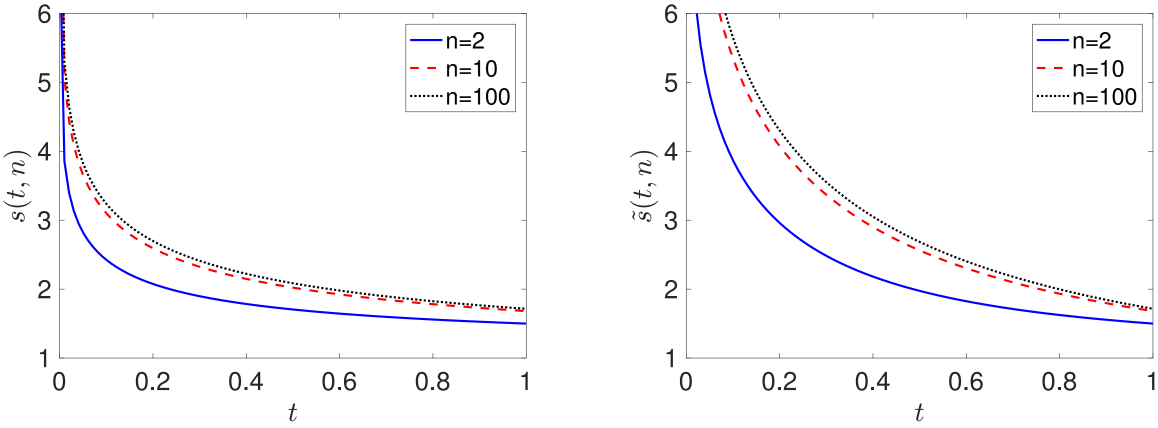

Figure 5-1: Plot of $s(t,n)$ and $\tilde{s}(t,n)$ versus $t \in (0,1)$ for different values of n.

of epochs required to guarantee $\|\mathbb{E}x^{\ell n} - x^*\| \leq \epsilon$ for RCD and RPCD algorithms. In particular $s(t,n) > 1$ implies RPCD has a faster worst-case convergence rate than RCD. In the following theorem, we show that RPCD is faster than RCD for any $t \in (0,1)$ and $n \geq 2$, and quantify the rate of improvement.

Proposition 5.7. *The following statements are true:*

(*i*) *The function $s(t,n)$ is strictly decreasing in t over $(0,1)$.*

(*ii*) $\lim_{t\to 0} s(t,n) = \infty$.

(*iii*) *Let $g(n) := \lim_{t\to 1} s(t,n)$. We have $g(n) \in [3/2, e-1)$, for any $n \geq 2$. Furthermore, $g(n)$ is strictly increasing in $n \geq 2$ satisfying*

$$g(2) = 3/2 \quad \textit{and} \quad \lim_{n\to\infty} g(n) = e - 1.$$

A consequence of Proposition 5.7 is that RPCD is faster than RCD in the worst-case, for every $t \in (0,1)$ by a factor $s(t,n) > 1$. Furthermore, the amount of acceleration $s(t,n)$ goes to infinity as $\alpha \to 0$ for any n fixed. This shows that as the matrix A becomes more and more well-conditioned (as $\alpha \to 0$), the amount of speed-up $s(t,n)$ we obtain with

RPCD with respect to RCD goes to infinity. This is consistent with the observation that cyclic orders work well for diagonal-like matrices that are well-conditioned (see e.g. [147]). Proposition 5.7 is illustrated in Figure 5-1 (left panel), where we plot the parameter $s(t, n)$ as a function of t for different values of n.

We next compare the convergence rate of CCD with respect to RPCD and RCD. To this end, as we discuss in Section 5.2 (cf. (5.2.1)), we use $\rho(B_{\text{CCD}})$ as the asymptotic per epoch worst-case convergence rate of CCD, whereas for comparison to RCD, we use a per-epoch rate of $\rho(B_{\text{RCD}})^n$. Note that as discussed in (5.4.3), $B_{\text{CCD}} = B_{\text{CCD-}\pi}$ for all π, and hence $\rho(B_{\text{CCD}}) = \rho(B_{\text{CCD-}\pi})$ for all π. Although, explicit calculation of $\rho(B_{\text{CCD}})$ appears to be challenging, we prove that the known upper bounds [72, Theorem 4.12] on $\rho(B_{\text{CCD}})$ is tighter than $\rho(B_{\text{RPCD}})$, which together with Proposition 5.7 imply the following result.

Proposition 5.8. *Let f be a quadratic function of the form (4.0.1), whose Hessian matrix given by (5.4.1). Then, the expected iteration matrices of CCD, RPCD and RCD satisfy*

$$\rho(B_{CCD}) < \rho(B_{RPCD}) < \rho(B_{RCD})^n, \tag{5.4.9}$$

for any $\alpha \in (0, 1/(n-1))$ and $n \geq 2$.

5.4.2 Convergence rates of RPCD and RCD in improvement sequences $\mathcal{I}_2$ & $\mathcal{I}_3$

In this section, we compare the rate of RPCD and RCD with respect to improvement sequences $\mathcal{I}_2$ and $\mathcal{I}_3$. When the Hessian matrix A satisfies (5.4.1), the smallest eigenvalue of A can be found as follows

$$\mu = 1 - t = 1 - \alpha(n-1). \tag{5.4.10}$$

Plugging this value in the convergence guarantee of RCD in (5.3.2), we can obtain a convergence guarantee on both improvement sequences $\mathcal{I}_2$ and $\mathcal{I}_3$ as the left hand-side of (5.3.2)

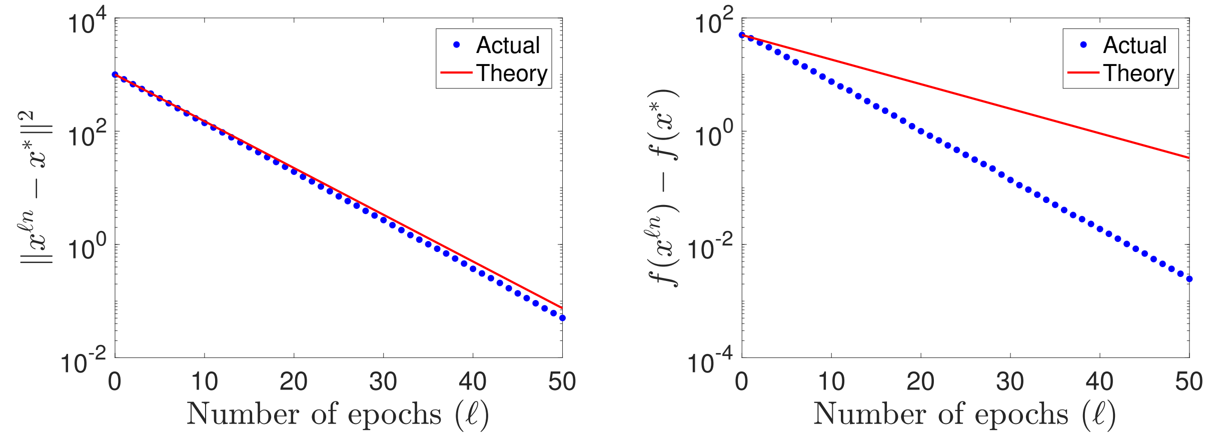

Figure 5-2: Tightness of the bounds in Proposition 5.9 when $n = 1000$ and $\alpha = \frac{0.9}{n-1}$: Left figure for (5.4.11) and right figure for (5.4.12).

upper bounds both $2\mathcal{I}_2$ and $\mathcal{I}_3$. However, for the particular problem class we consider in this chapter, we derive a tighter convergence rate guarantee for RCD in the next proposition, whose proof is deferred to Section 5.7.4.

Proposition 5.9. *Let f be a quadratic function of the form (4.0.1), whose Hessian matrix given by (5.4.1). Then, RCD iterations satisfy*

$$\mathbb{E}\|x_{RCD}^k - x^*\|^2 \le \left(1 - \frac{2\mu}{n} + \frac{\mu^2}{n}\right)^k \|x^0 - x^*\|^2, \tag{5.4.11}$$

and

$$\mathbb{E}\big(f(x_{RCD}^k) - f(x^*)\big) \le \left(1 - \frac{\mu}{n}\right)^k \big(f(x^0) - f(x^*)\big). \tag{5.4.12}$$

Remark 5.10. *We observe that the upper bound in (5.4.11) is smaller (tighter) than the upper bound in (5.3.2) for any $\alpha \in (0, 1/(n-1))$ because*

$$1 - \frac{2\mu}{n} + \frac{\mu^2}{n} < 1 - \frac{2\mu}{n} + \frac{2\mu^2}{n} = 1 - \frac{2\mu(1-\mu)}{n} = 1 - \frac{2\mu(1-\mu^2)}{n(1+\mu)} < 1 - \frac{2\mu}{n(1+\mu)},$$

where the inequalities are due to the fact that $\mu = 1 - \alpha(n-1) \in (0,1)$.

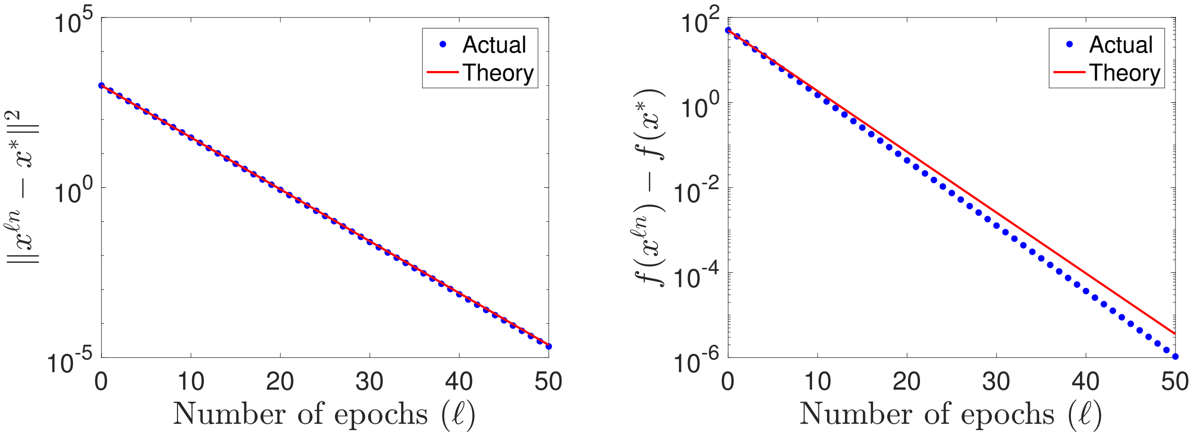

Figure 5-3: Tightness of the bounds in Proposition 5.11 when $n = 1000$ and $\alpha = \frac{0.9}{n-1}$: Left figure for (5.4.13) and right figure for (5.4.14).

We next analyze the performance of RPCD in the following proposition and show that the convergence rate guarantee of RPCD is tighter than the convergence rate guarantee of RCD in Proposition 5.9. The proof of Proposition 5.11 is given in Section 5.7.5.

Proposition 5.11. *Let f be a quadratic function of the form* (4.0.1)*, whose Hessian matrix given by* (5.4.1)*. Then, RPCD iterations satisfy*

$$\mathbb{E}\|x_{RPCD}^{\ell n} - x^*\|^2 \leq \left(1 - \frac{2\mu}{n}\left(\frac{(1+\alpha)^n - 1}{\alpha}\right) + \frac{\mu^2}{n}\left(\frac{(1+\alpha)^{2n} - 1}{\alpha(\alpha+2)}\right)\right)^{\ell} \|x^0 - x^*\|^2, \quad (5.4.13)$$

and

$$\mathbb{E}f(x_{RPCD}^{\ell n}) - f(x^*) \leq \left(1 - \frac{\mu}{n}\left(\frac{(1+\alpha)^{2n} - 1}{\alpha(\alpha+2)}\right)\right)^{\ell} \big(f(x^0) - f(x^*)\big). \quad (5.4.14)$$

We next compare the convergence rates we derive for the RCD and RPCD algorithms. In particular, we consider the convergence rate of both algorithms in improvement sequence $\mathcal{I}_2$ since we obtain tighter upper bounds for it. Comparing the convergence rate bounds for RCD and RPCD in (5.4.11) and (5.4.13), respectively, we can observe that RPCD is faster (in terms of the best known rate guarantees) than RCD by a factor of

$$\tilde{s}(t,n) := \frac{-\log\left(1-\frac{2\mu}{n}\left(\frac{(1+\alpha)^n-1}{\alpha}\right)+\frac{\mu^2}{n}\left(\frac{(1+\alpha)^{2n}-1}{\alpha(\alpha+2)}\right)\right)}{-n\log\left(1-\frac{2\mu}{n}+\frac{\mu^2}{n}\right)},$$

which is plotted in Figure 5-1 (right panel) in the interval $t \in (0,1)$ for different values of n. We observe from this figure that the convergence rate bound for RPCD is better than than the one for RCD for all $t \in (0,1)$ and $n \geq 2$. Furthermore, the difference in convergence rate bounds increases as t gets smaller, i.e., as the Hessian matrix becomes more diagonally dominant. We can also show that $\tilde{s}(t,n)$ behaves similar to $s(t,n)$ as $t \to 1$, where the limiting values can be found in Proposition 5.7.

5.5 Numerical Validation

Here we compare the performance of CCD, RPCD, and RCD for the quadratic problem (4.0.1) with Hessian matrix (5.4.1). In Figure 5-4, we use a worst-case initialization $x^0 = \mathbf{1}$, for $n \in \{1000, 10000\}$ and $\alpha \in \left\{\frac{0.01}{n-1}, \frac{0.50}{n-1}, \frac{0.99}{n-1}\right\}$. We observe that CCD is the faster than RPCD, which is faster than RCD. This behavior is in accordance with the theoretical results in Propositions 5.8-5.11. Furthermore, as α decreases, we can see that the ratio between the convergence rates of RPCD and RCD increases, consistent with Proposition 5.7 (see also Figure 5-1). We can also observe from the right column in Figure 5-4 that when α is close to $1/(n-1)$, the ratio between the convergence rates of RPCD and RCD is close to the theoretical limits obtained in Proposition 5.7 (see part *(iii)*, which shows that the ratio is in the interval $[3/2,\, e-1)$). Figure 5-5 plots similar results to Figure 5-4, but for a random initialization rather than worst-case initialization. Convergence rates depicted in Figure 5-5 are similar to those of Figure 5-4, due to the fact that $x^{\ell n}$ becomes colinear with the vector of ones as ℓ increases (as $\mathbf{1}$ is the leading eigenvector of the expected iteration matrix), so that the worst-case convergence rate dictates the performance of the algorithms.

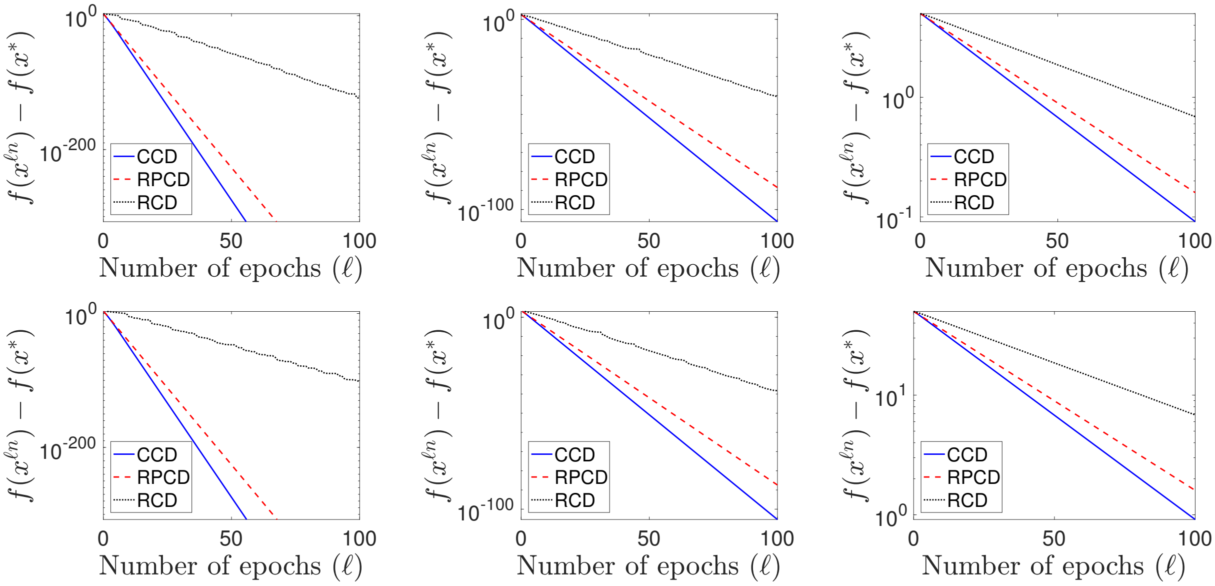

Figure 5-4: CCD vs RPCD vs RCD with worst-case initialization for $n = 1000$ (top row) and $n = 10000$ (bottom row): $\alpha = \frac{0.01}{n-1}$ in the left column, $\alpha = \frac{0.50}{n-1}$ in the middle column, and $\alpha = \frac{0.99}{n-1}$ in the right column.

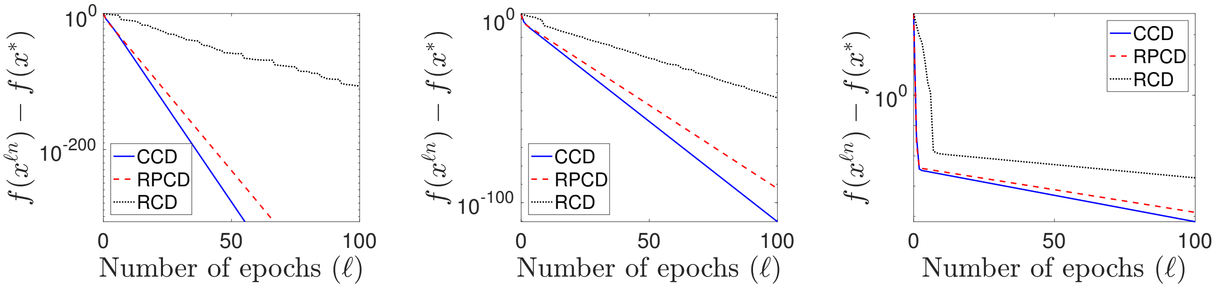

Figure 5-5: CCD vs RPCD vs RCD with random initialization for $n = 1000$: $\alpha = \frac{0.01}{n-1}$ (left figure), $\alpha = \frac{0.50}{n-1}$ (middle figure), and $\alpha = \frac{0.99}{n-1}$ (right figure).

5.6 Discussion

In this chapter, we surveyed the known results on the performance of RPCD for special cases of strongly convex quadratic objectives and add to these results by presenting a class of convex quadratic problems with diagonally dominant Hessians. Using the distance of the expected iterates to the optimal solution as the convergence criterion, we compared the ratio between the performances of RPCD and RCD with respect to a parameter that represents the extent of diagonal dominance. We illustrated that as the Hessian matrix becomes more diagonally dominant, this ratio goes to infinity, whereas as it gets smaller it goes to a constant in the interval $[3/2, e-1)$. We also showed that CCD outperforms both RPCD and RCD for this class of problems. When expected distance of the iterates or expected function value of the iterates is used as the convergence criterion, we presented that the worst-case convergence rate bounds derived for RPCD are tighter compared to the ones for RCD. This is in accordance with our first set of results, i.e., when distance of the expected iterates is used as the convergence criterion. Computational experiments validate our theoretical results, which fill a gap between the theoretical guarantees for RPCD and its empirical performance.

5.7 Additional Proofs

5.7.1 Proof of Lemma 5.6

Applying Lemma 5.5 with $Q = \Gamma^{-1}$, where Γ^{-1} is defined in (5.4.4), we get

$$\begin{aligned}\gamma &= \frac{\sum_{j=0}^{n-2}(n-1-j)\alpha(1+\alpha)^j}{n(n-1)} = \frac{\alpha}{n}\sum_{j=0}^{n-2}(1+\alpha)^j - \frac{\alpha}{n(n-1)}\sum_{j=0}^{n-2} j(1+\alpha)^j \\ &= \frac{(1+\alpha)^{n-1}-1}{n} - \frac{(1+\alpha)^{n-1}}{n} + \frac{(1+\alpha)^n - 1 - \alpha}{\alpha n(n-1)} = \frac{(1+\alpha)^n - \alpha n - 1}{\alpha n(n-1)},\end{aligned}$$

where the third equality follows by the following lemma. This completes the proof.

Lemma 5.12. *For any real scalar $\eta \neq 1$ and integer $k \geq 0$, we have*

$$\sum_{j=0}^{k} j\eta^j = (k+1)\frac{\eta^{k+1}}{\eta - 1} - \frac{(\eta^{k+1}-1)\eta}{(\eta-1)^2}.$$

Proof Consider the cumulative sums $u_k(\eta) := \sum_{j=0}^{k} \eta^j = \frac{\eta^{k+1}-1}{\eta-1}$. It is easy to see that $\sum_{j=0}^{k} j\eta^j = \eta u_k'(\eta)$ where $u_k'(\eta)$ is the derivative of $u_k(\eta)$. Differentiating the right-hand side of the formula for u_k yields the result. □

5.7.2 Proof of Proposition 5.7

Proof of Part (i)

Defining $h(t,n) = \frac{\left(1+\frac{t}{n-1}\right)^n - 1}{\frac{t}{n-1}}$, where $t \in (0,1)$ and $n \geq 1$ is an integer, we have by the definition in (5.4.8) that $s(t,n) = \rho_1(t,n)/\rho_2(t,n)$, where

$$\rho_1(t,n) = -\log\left(1 - \frac{1-t}{n} h(t,n)\right) \quad \text{and} \quad \rho_2(t,n) = -n \log\left(1 - \frac{1-t}{n}\right).$$

Throughout the rest of the proof, for simplicity, whenever the dependence of h, ρ_1 and ρ_2 on n is clear, we will abbreviate them by $h(t)$, $\rho_1(t)$ and $\rho_2(t)$, respectively. Similarly, whenever the dependence on t is also clear, we will abbreviate them by h, ρ_1 and ρ_2, respectively. In order to prove statement *(i)* of Proposition 5.7, it suffices to show that the partial derivative satisfies

$$\partial_t s(t,n) = \frac{\partial_t(\rho_1)\rho_2 - \rho_1 \partial_t(\rho_2)}{\rho_2^2} < 0,$$

for all $t \in (0,1)$. This holds if and only if

$$\frac{\partial_t(\rho_1)}{\rho_1} < \frac{\partial_t(\rho_2)}{\rho_2} \quad \Leftrightarrow \quad \partial_t(\log \rho_1) < \partial_t(\log \rho_2), \tag{5.7.1}$$

for all $t \in (0,1)$, where we used the fact that ρ_1 and ρ_2 are positive for $t \in (0,1)$. We can compute these partial derivatives in the right-hand side as follows

$$\partial_t(\log \rho_1) = \frac{1}{\rho_1}\partial_t(\rho_1) = \frac{-1}{\rho_1}\left(\frac{1}{1-\frac{1-t}{n}h(t)}\right)\left(\frac{h(t)+h'(t)(t-1)}{n}\right),$$

and similarly

$$\partial_t(\log \rho_2) = \frac{1}{\rho_2}\partial_t(\rho_2) = \frac{-1}{\rho_2}\left(\frac{1}{1-\frac{1-t}{n}}\right).$$

Hence, in order to prove (5.7.1), it is sufficient to show that

$$\frac{1}{\rho_1}\left(\frac{1}{1-\frac{1-t}{n}h(t)}\right)q(t) > \frac{1}{\rho_2}\left(\frac{1}{1-\frac{1-t}{n}}\right), \quad \text{where} \quad q(t) := \frac{h(t)+h'(t)(t-1)}{n},$$

which, after inserting the formulas for ρ_1 and ρ_2, is equivalent to

$$-n\log\left(1-\frac{1-t}{n}\right)\left(1-\frac{1-t}{n}\right)q(t) > -\log\left(1-\frac{1-t}{n}h(t)\right)\left(1-\frac{1-t}{n}h(t)\right), \quad (5.7.2)$$

for $t \in (0,1)$. The main ingredients to prove this inequality is to approximate the non-linear functions q and h with piecewise linear functions, which are easier to deal with, in other words, linearizing q and h above leads to simpler expressions for the derivatives of both sides of this inequality. In order to approximate q, we first write a binomial expansion for $h(t)$ as follows

$$h(t) = \frac{\left(1+\frac{t}{n-1}\right)^n - 1}{\frac{t}{n-1}} = \sum_{i=1}^{n}\binom{n}{i}\left(\frac{t}{n-1}\right)^{i-1}.$$

This implies that $q(t)$ is of the form $q(t) = \frac{1}{2} + \frac{2}{3}t + \sum_{j=2}^{n-1} c_j t^j$, where $c_2 > 0$ and $c_j \geq 0$, for all $j \in \{3,\ldots,n-1\}$. Therefore, the first and second derivatives of q are positive over $t \in (0,1)$ and q is strictly convex. We then consider linearizations of $q(t)$ at $t=0$ and $t=1$, which are given by

$$q_0(t) = \frac{1}{2} + \frac{2}{3}t \quad \text{and} \quad q_1(t) = \frac{h(1) - 2(n-1)(1-t)}{n}.$$

(Note that in the special case $n = 2$, $q(t)$ is linear so that $q_0(t) = q_1(t)$ for all t. However, for $n > 2$, $q_0 \neq q_1$). In particular, it can be checked that $q_0(\hat{t}) = q_1(\hat{t})$, for $\hat{t} = 1 - \frac{6h(1)-7n}{4(2n-3)}$. Since $q(t)$ is convex,

$$q(t) \geq \underline{q}(t) = \max(q_0(t), q_1(t)) = \begin{cases} q_0(t), & \text{if } t \in [0, \hat{t}), \\ q_1(t), & \text{if } t \in [\hat{t}, 1]. \end{cases} \tag{5.7.3}$$

The right-hand side of (5.7.2) is of the form

$$z(t) = -\log(y(t))y(t) = E(y(t)), \quad \text{where} \quad y(t) = 1 - \frac{1-t}{n}h(t), \quad E(y) = -\log(y)y. \tag{5.7.4}$$

As h is convex, we have the bounds

$$\overline{h}(t) = (1-t)h(0) + th(1) \geq h(t) \quad \text{and} \quad y(t) \geq \overline{y}(t) = 1 - \frac{1-t}{n}\overline{h}(t), \quad t \in (0,1). \tag{5.7.5}$$

Using the facts that the function $E(\cdot)$ has a maximum of $1/e$ over the interval $[0,1]$ and is strictly decreasing over the interval $(1/e, 1]$, it follows from (5.7.5) that

$$E(y(t)) = z(t) \leq \overline{z}(t) := \begin{cases} E(\bar{y}(t)) & \text{if} \quad \overline{y} \in (1/e, 1] \Leftrightarrow t \in (t_*, 1] \\ 1/e & \text{if} \quad \overline{y} \in [0, 1/e] \Leftrightarrow t \in [0, t_*] \end{cases} \tag{5.7.6}$$

where t_* is the largest $t \in (0,1)$ such that $\overline{y}(t) = 1/e$ and admits the formula

$$t_* = -\frac{1}{2}\frac{2n - h(1)}{h(1) - n} + \frac{1}{2}\sqrt{\left(\frac{2n - h(1)}{h(1) - n}\right)^2 + \frac{4}{e}\frac{n}{h(1) - n}}.$$

Combining the lower bound (5.7.3) on $q(t)$ and the upper bound (5.7.6) on $z(t)$, a sufficient condition for (5.7.2) is to show that the following relaxed inequality holds

$$-n\log\left(1 - \frac{1-t}{n}\right)\left(1 - \frac{1-t}{n}\right)\underline{q}(t) - \overline{z}(t) > 0, \quad \text{for all} \quad t \in (0,1). \tag{5.7.7}$$

The left-hand side is a piecewise continuously differentiable function (pieces defined by the intervals $[0, \hat{t}]$, $(\hat{t}, t_*]$ and $(t_*, 1])$) and it is positive at $t = 0$. The rest of the proof is about showing that the left-hand side in (5.7.7) stays positive for $t \in (0, 1)$, this is achieved by computing and lower bounding the first order derivatives of the left-hand side. The details are skipped due to space considerations and follows from standard calculus techniques.

Proof of Part (ii)

Since $\lim_{t \to 0^+} \rho_2(t) = -n \log(1 - 1/n)$, whereas $\lim_{t \to 0^+} \rho_1(t) = -\log(1 - h(0)/n) = \infty$ as $h(0) = n$, we obtain $\lim_{t \to 0^+} s(t, n) = \lim_{t \to 0}(\rho_1(t)/\rho_2(t)) = \infty$.

Proof of Part (iii)

We observe that $g(n) = \lim_{t \to 1^-} \frac{\rho_1(t)}{\rho_2(t)} = \lim_{t \to 1^-} \frac{\rho_1'(t)}{\rho_2'(t)}$, since $\lim_{t \to 1^-} \rho_1(t) = \lim_{t \to 1^-} \rho_2(t) = 0$. The derivatives of $\rho_1(t)$ and $\rho_2(t)$ with respect to t are given by

$$\rho_1'(t) = -\frac{h(t) + h'(t)(t-1)}{n - (1-t)h(t)} \quad \text{and} \quad \rho_2'(t) = -\frac{n}{n - (1-t)}.$$

Therefore, we obtain

$$g(n) = \lim_{t \to 1^-} \frac{\frac{h(t) + h'(t)(t-1)}{n - (1-t)h(t)}}{\frac{n}{n-(1-t)}} = \frac{h(1)}{n} = \left(1 + \frac{1}{n-1}\right)^{n-1} + \frac{1}{n} - 1.$$

In order to show that $g(n)$ is strictly increasing in n, consider the extension of g to the positive real line, i.e., consider the function $\bar{g}(z) = \left(1 + \frac{1}{z}\right)^z + \frac{1}{z+1} - 1$, where $z \geq 0$. Taking its derivative with respect to z, we get

$$\bar{g}'(z) = \left(\log\left(1 + \frac{1}{z}\right) - \frac{1}{z+1}\right)\left(1 + \frac{1}{z}\right)^z - \frac{1}{(z+1)^2}.$$

Using the lower bounds $\log(1 + y) \geq \frac{2y}{2+y}$ for $y \geq 0$ and $(1 + 1/y)^y \geq 2$ for $y \geq 1$, we obtain

$$\bar{g}'(z) \geq 2\left(\frac{2}{2z+1} - \frac{1}{z+1}\right) - \frac{1}{(z+1)^2} = \frac{1}{(z+1)(z+1/2)} - \frac{1}{(z+1)^2} > 0,$$

for any $z \geq 1$. Consequently, $g(n)$ is strictly increasing in $n \geq 2$. Furthermore, it follows directly from the definition that $g(2) = 3/2$ and since $\lim_{n\to\infty}(1 + 1/n)^n = e$, we get $\lim_{n\to\infty} g(n) = e - 1$. This completes the proof of part (iii).

5.7.3 Proof of Proposition 5.8

The proof of $\rho(B_{\text{RPCD}}) < \rho(B_{\text{RCD}})^n$ follows by Proposition 5.7, hence is omitted. Since the off-diagonal entries of A are nonpositive and A is a positive definite matrix, then it follows by [72, Theorem 4.12] that $\rho(B_{\text{CCD}}) \leq \frac{1-\mu}{1+\mu} = 1 - \frac{2\mu}{1+\mu}$, where $\mu = 1 - (n-1)\alpha$. On the other hand, from (5.4.7), we have $\rho(B_{\text{RPCD}}) = 1 - \mu\frac{(1+\alpha)^n - 1}{n\alpha}$. Hence, in order to show that $\rho(B_{\text{CCD}}) < \rho(B_{\text{RPCD}})$, for all $\alpha \in (1, 1/(n-1))$ and $n \geq 2$, it suffices to show

$$\frac{2}{1+\mu} > \frac{(1+\alpha)^n - 1}{n\alpha} \quad \Leftrightarrow \quad \frac{1}{1 - \frac{(n-1)\alpha}{2}} > \frac{(1+\alpha)^n - 1}{n\alpha}.$$

Since $\alpha \in (1, 1/(n-1))$, it is sufficient to show that

$$n\alpha > \left(1 - \frac{(n-1)\alpha}{2}\right)((1+\alpha)^n - 1). \tag{5.7.8}$$

Using the Binomial expansion $(1+\alpha)^n = \sum_{j=0}^n \binom{n}{j}\alpha^j$, we get

$$\begin{aligned}\left(1 - \frac{(n-1)\alpha}{2}\right)((1+\alpha)^n - 1) &= \sum_{j=1}^n \binom{n}{j}\alpha^j - \frac{n-1}{2}\sum_{j=1}^n \binom{n}{j}\alpha^{j+1} \\ &< \sum_{j=1}^n \binom{n}{j}\alpha^j - \frac{n-1}{2}\sum_{j=1}^{n-1} \binom{n}{j}\alpha^{j+1} \\ &= n\alpha + \sum_{j=2}^n \left(\binom{n}{j} - \frac{n-1}{2}\binom{n}{j-1}\right)\alpha^j,\end{aligned}$$

where the inequality follows since we omit the last term of the second sum and the last equality follows by peeling out the first entry of the first sum. We can observe that

$$\binom{n}{j} - \frac{n-1}{2}\binom{n}{j-1} = \left(\frac{n+1-j}{j} - \frac{n-1}{2}\right)\binom{n}{j-1} = \left(\frac{(n+1)(2-j)}{2j}\right)\binom{n}{j-1} \leq 0,$$

for all $j \in \{2, \dots, n\}$. This proves (5.7.8), which concludes the proof.

5.7.4 Proof of Proposition 5.9

RCD iterations can be written (by (5.1.4)) as follows

$$x_{\mathrm{RCD}}^{k+1} = \big(I - e_{i_k} e_{i_k}^T A\big) x_{\mathrm{RCD}}^k,$$

where i_k is drawn uniformly at random from the set $\{1, 2, \dots, n\}$. Letting $\mathbb{E}_k$ denote the expectation with respect to i_k given x_k and taking norm squares of both sides, we obtain

$$\begin{aligned}
\mathbb{E}_k \|x_{\mathrm{RCD}}^{k+1}\|^2 &= (x_{\mathrm{RCD}}^k)^T \, \mathbb{E}_k \big[\big(I - A^T e_{i_k} e_{i_k}^T\big)\big(I - e_{i_k} e_{i_k}^T A\big)\big] x_{\mathrm{RCD}}^k \\
&= (x_{\mathrm{RCD}}^k)^T \left(\frac{1}{n} \sum_{i=1}^n \big(I - A^T e_i e_i^T - e_i e_i^T A + A^T e_i e_i^T A\big) \right) x_{\mathrm{RCD}}^k \\
&= (x_{\mathrm{RCD}}^k)^T \left(I - \frac{2A}{n} + \frac{A^2}{n} \right) x_{\mathrm{RCD}}^k \le \|Q\| \|x_{\mathrm{RCD}}^k\|^2 \text{ with } Q := I - \frac{2A}{n} + \frac{A^2}{n},
\end{aligned}$$

where we used the fact that $A = A^T$ and $\sum_{i=1}^n e_i e_i^T = I$. Using this recursion and noting that $x^* = 0$, we get

$$\mathbb{E}\|x_{\mathrm{RCD}}^{k+1} - x^*\|^2 \le \|Q\|^k \, \|x^0 - x^*\|^2. \tag{5.7.9}$$

The eigenvalues of Q are of the form $1 - 2\lambda/n + \lambda^2/n$, where λ is an eigenvalue of A. Since Q is symmetric and A has only two distinct eigenvalues that are equal to $\mu = (1 - \alpha(n-1))$ and $L = 1 + \alpha$, we obtain

$$\|Q\| = \max\{1 - 2\mu/n + \mu^2/n, 1 - 2L/n + L^2/n\} = 1 - 2\mu/n + \mu^2/n. \tag{5.7.10}$$

Using (5.7.10) in (5.7.9) concludes the proof of (5.4.11). The proof of (5.4.12) can be done by following similar lines to the above proof as follows

$$\begin{aligned} f(x^{k+1}_{\text{RCD}}) &= (x^k_{\text{RCD}})^T\,\mathbb{E}_k\big[\big(I - A^T e_{i_k} e^T_{i_k}\big) A \big(I - e_{i_k} e^T_{i_k} A\big)\big] x^k_{\text{RCD}} \\ &= (x^k_{\text{RCD}})^T\,\mathbb{E}_k\big[A - A^T e_{i_k} e^T_{i_k} A - A e_{i_k} e^T_{i_k} A + A^T e_{i_k} e^T_{i_k} A e_{i_k} e^T_{i_k} A\big] x^k_{\text{RCD}} \\ &= (x^k_{\text{RCD}})^T\,\mathbb{E}_k\big[A - A e_{i_k} e^T_{i_k} A\big] x^k_{\text{RCD}} \\ &= (x^k_{\text{RCD}})^T \left(A - \frac{A^2}{n}\right) x^k_{\text{RCD}} \leq \|I - \frac{A}{n}\| f(x^k_{\text{RCD}}) = \left(1 - \frac{\mu}{n}\right) f(x^k_{\text{RCD}}), \end{aligned}$$

where in the third equality, we use the fact that $A = A^T$ and $e_i^T A e_i = 1$, for all $i \in [n]$, and in the fourth equality, we use $\sum_{i=1}^n e_i e_i^T = I$, respectively. This concludes the proof.

5.7.5 Proof of Proposition 5.11

RPCD iterations can be written (by (5.1.6)) as follows

$$x^{(\ell+1)n}_{\text{RPCD}} = P_{\pi_\ell} B_{\text{CCD}} P^T_{\pi_\ell}\, x^{\ell n}_{\text{RPCD}}.$$

Considering improvement sequence $\mathcal{I}_2$, this yields

$$\mathbb{E}_\ell \|x^{(\ell+1)n}_{\text{RPCD}}\|^2 = (x^{\ell n}_{\text{RPCD}})^T \mathbb{E}_P[P B^T_{\text{CCD}} B_{\text{CCD}} P^T] x^{\ell n}_{\text{RPCD}} \leq \|S\| \|x^{\ell n}_{\text{RPCD}}\|^2,$$

where $S = \mathbb{E}_P[P B^T_{\text{CCD}} B_{\text{CCD}} P^T]$. Using this recursion, we obtain

$$\mathbb{E}\|x^{\ell n}_{\text{RPCD}}\|^2 \leq \|S\|^\ell \big\|x^0_{\text{RPCD}}\big\|^2.$$

The contraction factor $\|S\|$ can be computed by applying Lemma 5.5 with $Q = B^T_{\text{CCD}} B_{\text{CCD}}$, which yields

$$S = \mathbb{E}_P[P B^T_{\text{CCD}} B_{\text{CCD}} P^T] = \tau_1 I + \tau_2 \mathbf{1}\mathbf{1}^T, \tag{5.7.11}$$

where

$$\tau_2 = \frac{\mathbf{1}^T B_{\text{CCD}}^T B_{\text{CCD}} \mathbf{1} - \operatorname{tr}(B_{\text{CCD}}^T B_{\text{CCD}})}{n(n-1)} \quad \text{and} \quad \tau_1 = \frac{\operatorname{tr}(B_{\text{CCD}}^T B_{\text{CCD}})}{n} - \tau_2.$$

Since S is a symmetric matrix, we have $\|S\| = \rho(S)$. Furthermore, we can observe that $B_{\text{CCD}}^T B_{\text{CCD}}$ has strictly positive entries both in its diagonals and off-diagonals, consequently we have $S > 0$. Then, by Perron-Frobenius Theorem [147, Lemma 2.8], we have

$$\|S\| = \rho(S) = \tau_1 + n\tau_2 = \frac{1}{n}\mathbf{1}^T S \mathbf{1}. \tag{5.7.12}$$

In order to compute (5.7.12), we first compute the matrix B_{CCD} as follows

$$B_{\text{CCD}} = I - \Gamma^{-1} A = \begin{cases} \alpha((1+\alpha)^{i-1} - (1+\alpha)^{i-j}), & \text{if} \quad i \geq j, \\ \alpha(1+\alpha)^{i-1}, & \text{if} \quad i < j. \end{cases} \tag{5.7.13}$$

Combining (5.7.12) and (5.7.13), we obtain

$$\|S\| = \frac{1}{n}\mathbf{1}^T B_{\text{CCD}}^T B_{\text{CCD}} \mathbf{1} = \frac{1}{n}\|B_{\text{CCD}}\mathbf{1}\|^2 = \frac{1}{n}\sum_{i=1}^{n} ((B_{\text{CCD}}\mathbf{1})_i)^2,$$

where

$$(B_{\text{CCD}}\mathbf{1})_i = 1 - \mu(1+\alpha)^{i-1}. \tag{5.7.14}$$

This yields

$$\begin{aligned} \|S\| &= \frac{1}{n}\sum_{i=1}^{n} \big(1 - 2\mu(1+\alpha)^{i-1} + \mu^2(1+\alpha)^{2(i-1)}\big) \\ &= 1 - \frac{2\mu}{n}\left(\frac{(1+\alpha)^n - 1}{\alpha}\right) + \frac{\mu^2}{n}\left(\frac{(1+\alpha)^{2n} - 1}{\alpha(\alpha+2)}\right), \end{aligned}$$

which proves (5.4.13).

We next prove the results regarding the function suboptimality in (5.4.14). To this end, we consider the expected function suboptimality (note that $f(x^*) = 0$), which yields

$$\begin{aligned}
\mathbb{E}_\ell f(x_{\text{RPCD}}^{(\ell+1)n}) &= (x_{\text{RPCD}}^{\ell n})^T \mathbb{E}_P[PB_{\text{CCD}}^T P^T A P B_{\text{CCD}} P^T] x_{\text{RPCD}}^{\ell n} \\
&= (x_{\text{RPCD}}^{\ell n})^T \mathbb{E}_P[PB_{\text{CCD}}^T A B_{\text{CCD}} P^T] x_{\text{RPCD}}^{\ell n} \\
&\le \|\mathbb{E}_P[A^{-1/2} P B_{\text{CCD}}^T A B_{\text{CCD}} P^T A^{-1/2}]\| \|A^{1/2} x_{\text{RPCD}}^{\ell n}\|^2 \\
&= \|\mathbb{E}_P[A^{-1/2} P B_{\text{CCD}}^T A B_{\text{CCD}} P^T A^{-1/2}]\| \, f(x_{\text{RPCD}}^{\ell n}) \\
&= \|\mathbb{E}_P[P A^{-1/2} B_{\text{CCD}}^T A B_{\text{CCD}} A^{-1/2} P^T]\| \, f(x_{\text{RPCD}}^{\ell n}) \\
&= \|G\| \, f(x_{\text{RPCD}}^{\ell n}),
\end{aligned}$$

where $G := \mathbb{E}_P[PA^{-1/2}B_{\text{CCD}}^T A B_{\text{CCD}} A^{-1/2} P^T]$ and the equalities follow since A and $A^{-1/2}$ are symmetric permutation invariant matrices, i.e., $PAP^T = A$ and $PA^{-1/2}P^T = A^{-1/2}$. It can be shown that $A^{1/2}B_{\text{CCD}}A^{-1/2}$ is a non-negative matrix, hence applying Lemma 5.5 to the matrix $Q = A^{-1/2}B_{\text{CCD}}^T A B_{\text{CCD}} A^{-1/2}$, it can be shown (similar to the previous proof) that

$$\|G\| = \rho(G) = \frac{1}{n}\|A^{1/2}B_{\text{CCD}}A^{-1/2}\mathbf{1}\|^2 = \frac{1}{n}\|\mathbf{1} - A^{1/2}\Gamma^{-1}A^{1/2}\mathbf{1}\|^2, \tag{5.7.15}$$

where $A^{1/2} = \gamma I - \sigma \mathbf{1}\mathbf{1}^T$ with $\gamma = \sqrt{1+\alpha}$ and $\sigma = (\gamma - \sqrt{\mu})/n$. This yields $A^{1/2}\mathbf{1} = (\gamma - n\sigma)\mathbf{1} = \sqrt{\mu}\mathbf{1}$. Multiplying both sides of the above equality by Γ^{-1} from the left, we obtain

$$\Gamma^{-1}A^{1/2}\mathbf{1} = \sqrt{\mu}\, c, \tag{5.7.16}$$

where it follows from (5.4.4) that

$$c = \begin{bmatrix} 1 \\ 1+\alpha \\ 1+\alpha+\alpha(1+\alpha) \\ \vdots \\ 1+\alpha+\alpha(1+\alpha)+\cdots+\alpha(1+\alpha)^{n-2} \end{bmatrix} = \begin{bmatrix} 1 \\ 1+\alpha \\ (1+\alpha)^2 \\ \vdots \\ (1+\alpha)^{n-1} \end{bmatrix}.$$

Multiplying (5.7.16) from the left by $A^{1/2}$, we get

$$A^{1/2}\Gamma^{-1}A^{1/2}\mathbf{1} = \sqrt{\mu}(\gamma c - \sigma\|c\|_1\mathbf{1}), \quad \text{where} \quad \|c\|_1 = \frac{(1+\alpha)^n - 1}{\alpha}. \tag{5.7.17}$$

Using (5.7.17) in (5.7.15), we obtain

$$\begin{aligned}
\|G\| &= \frac{1}{n}\sum_{i=1}^{n}(1-\sqrt{\mu}(\gamma c_i - \sigma\|c\|_1))^2 = 1 - \frac{2\sqrt{\mu}}{n}\sum_{i=1}^{n}(\gamma c_i - \sigma\|c\|_1) + \frac{\mu}{n}\sum_{i=1}^{n}(\gamma c_i - \sigma\|c\|_1)^2 \\
&= 1 - \frac{2\sqrt{\mu}}{n}(\gamma - n\sigma)\|c\|_1 + \frac{\mu}{n}\sum_{i=1}^{n}\big(\gamma^2 c_i^2 - 2\gamma\sigma\|c\|_1 c_i + \sigma^2\|c\|_1^2\big) \\
&= 1 - \frac{2\mu}{n}\|c\|_1 + \frac{\mu}{n}\big(\gamma^2\|c\|_2^2 - 2\gamma\sigma\|c\|_1^2 + n\sigma^2\|c\|_1^2\big),
\end{aligned} \tag{5.7.18}$$

where

$$\|c\|_2^2 = \frac{(1+\alpha)^{2n}-1}{\alpha(\alpha+2)} \quad \text{and} \quad \|c\|_1^2 = \frac{(1+\alpha)^{2n} - 2(1+\alpha)^n + 1}{\alpha^2}.$$

Modifying the terms in (5.7.18), we get

$$\begin{aligned}
\|G\| &= 1 - \frac{2\mu}{n}\|c\|_1 + \frac{\mu}{n}\big(\gamma^2\|c\|_2^2 - \gamma\sigma\|c\|_1^2 + \sigma(n\sigma - \gamma)\|c\|_1^2\big) \\
&= 1 - \frac{2\mu}{n}\|c\|_1 + \frac{\mu}{n}\left((1+\alpha)\|c\|_2^2 - \frac{1+\alpha-(1-\alpha(n-1))}{n}\|c\|_1^2\right) \\
&= 1 - \frac{2\mu}{n}\|c\|_1 + \frac{\mu}{n}\big((1+\alpha)\|c\|_2^2 - \alpha\|c\|_1^2\big) \\
&= 1 - \frac{2\mu}{n}\|c\|_1 + \frac{\mu}{n}\left((1+\alpha)\frac{(1+\alpha)^{2n}-1}{\alpha(\alpha+2)} - \frac{(1+\alpha)^{2n}-2(1+\alpha)^n+1}{\alpha}\right) \\
&= 1 - \frac{\mu}{n}\left(\frac{(1+\alpha)^{2n}-1}{\alpha(\alpha+2)}\right),
\end{aligned}$$

which concludes the proof of Proposition 5.11.

Chapter 6

Convergence Rate of the CD Method for Solving Large SDPs via Burer-Monteiro Approach

In this chapter, we study the convergence rate of the CD method on a certain non-convex problem that arises from semidefinite programming (SDP) with diagonal constraints:

$$\begin{aligned} \text{maximize} \quad & \langle A, X\rangle && \text{(CVX)} \\ \text{subject to} \quad & X_{ii} = 1, \ \text{for } i \in [n], \\ & X \succeq 0, \end{aligned}$$

where $A, X \in \mathrm{Sym}_n$ (real symmetric matrices of size $n \times n$). This problem appears as a convex relaxation to the celebrated Max-Cut problem [67], graphical model inference [60], community detection problems [13], and group synchronization [101].

Although SDPs serve as reliable relaxations to many combinatorial problems, the resulting convex problem is still computationally challenging. Interior point methods can solve SDPs to arbitrary accuracy in polynomial-time, but they do not scale well with the problem dimension n. A popular approach to remedy these limitations is to introduce a low-rank factorization $X = \sigma\sigma^\top$, where $\sigma \in \mathbb{R}^{n\times r}$ with r denoting the rank. This reformulation removes the positive semidefinite cone constraint in (CVX) since $X = \sigma\sigma^\top$ is guaranteed to be a positive semidefinite matrix, and choosing $r \ll n$ provides computational efficiency as well as storage benefits. This method is often referred to as Burer-Monteiro approach [40]. Denoting i-th row of σ by σ_i, i.e., $\sigma = [\sigma_1, \sigma_2, ..., \sigma_n]^\top$, the resulting non-convex problem can be written as follows

$$\begin{aligned} &\text{maximize} \;\; \langle A, \sigma\sigma^\top\rangle \\ &\text{subject to} \;\; \|\sigma_i\| = 1, \;\text{for } i \in [n]. \end{aligned} \tag{Non-CVX}$$

We consider solving (Non-CVX) using the CD method. To describe the update rule of the CD method, we let $f : \mathbb{R}^{n\times r} \to \mathbb{R}$ denote the objective function:

$$f(\sigma) = \langle A, \sigma\sigma^\top\rangle.$$

Given the current iterate σ^k, the CD method chooses a row $i_k \in [n]$ of the matrix σ^k and maximizes the following objective

$$f(\sigma^k) = \sum_{i=1}^{n} \langle \sigma_i^k, g_i^k\rangle, \quad \text{where} \quad g_i^k := \sum_{j\neq i} A_{ij}\, \sigma_j^k,$$

over the block $\sigma_{i_k}^k \in \mathcal{S}^{r-1}$. More formally, we can write the update rule of the algorithm as follows

$$\begin{aligned} \sigma_{i_k}^{k+1} &= \arg\max_{\|\zeta\|=1} f(\sigma_1^k, \ldots, \sigma_{i_k-1}^k, \zeta, \sigma_{i_k+1}^k, \ldots, \sigma_n^k), \\ &= \arg\max_{\|\zeta\|=1} 2\langle \zeta, g_{i_k}^k\rangle + \sum_{i\neq i_k}\sum_{j\neq i,i_k} A_{ij}\langle \sigma_i^k, \sigma_j^k\rangle, \qquad (6.0.1) \\ &= \arg\max_{\|\zeta\|=1} \langle \zeta, g_{i_k}^k\rangle = \frac{g_{i_k}^k}{\|g_{i_k}^k\|}, \qquad (6.0.2) \end{aligned}$$

with the convention that $\sigma_{i_k}^{k+1} = \sigma_{i_k}^k$ when $\|g_{i_k}^k\| = 0$. Blocks $\sigma_{i_k}^k$ to be updated at each

Algorithm 3: The CD Method

```
Initialize $\sigma^0 \in \mathbb{R}^{n\times r}$ and calculate $g_i^0 = \sum_{j\neq i} A_{ij}\sigma_j^0$, for all $i \in [n]$.
for $k = 0, 1, 2, \ldots$ do
    Choose block $i_k = i$ using one of the coordinate selection rules.
    $\sigma_{i_k}^{k+1} \leftarrow g_{i_k}^k / \|g_{i_k}^k\|$.
    $g_i^{k+1} \leftarrow g_i^k - A_{ii_k}\sigma_{i_k}^k + A_{ii_k}\sigma_{i_k}^{k+1}$, for all $i \neq i_k$.
end for
```

iteration can be chosen through any deterministic or randomized rule, and we focus on three coordinate selection rules:

- Uniform sampling: $i_k = i$ with probability $p_i = 1/n$.
- Importance sampling: $i_k = i$ with probability $p_i = \|g_i^k\| / \sum_{j=1}^n \|g_j^k\|$.
- Greedy coordinate selection: $i_k = \arg\max_{i\in[n]}(\|g_i^k\| - \langle \sigma_i^k, g_i^k\rangle)$.

In the following sections, we analyze the convergence of the CD method with these coordinate selection rules.

The remainder of this chapter is organized as follows. In Section 6.1, we prove the global sublinear convergence and local linear convergence of the CD method with explicit rate estimates. In Section 6.2, we introduce a second-order method based on the CD and Lanczos methods that is guaranteed to return solutions with global optimality guarantees. We also provide a global sublinear convergence rate estimate for this algorithm. We perform numerical experiments to validate our theoretical results in Section 6.4 and conclude the chapter in Section 6.5.

6.1 Convergence Rate of the CD Method

Throughout the chapter, we use the following notation. For a function h, ∇h and grad$h$ represent its Euclidean and Riemannian gradients, respectively. Similarly, $\nabla^2 h$ and Hess$h$ represent its Euclidean and Riemannian Hessians, respectively. We let $\mathcal{S}^{m-1}$ denote the unit sphere in $\mathbb{R}^m$. For a vector y, $\mathrm{Diag}(y)$ represents the diagonal matrix whose i-th diagonal entry is y_i. Similarly for a matrix A, $\mathrm{diag}(A)$ represents the vector whose i-th entry is A_{ii}.

Before discussing the convergence of the CD method, we first assume without loss of generality that A is a symmetric matrix and $A_{ii} = 0$, for all $i \in [n]$ (the latter assumption is removed in Section 6.2 to keep our presentation consistent with the existing works in the literature). Indeed, if A is not symmetric, then we can replace A by $(A + A^\top)/2$, which is a symmetric matrix, and the objective value (Non-CVX) remains the same for all $\sigma \in \mathbb{R}^{n\times r}$ since $\sigma\sigma^\top$ is symmetric. Similarly, replacing the diagonal entries of A by zeros decreases the objective value by the constant $\operatorname{tr} A$ for all feasible σ, since the diagonal entries of $\sigma\sigma^\top$ are equal to 1.

In order to analyze the convergence rate of the CD method, we require certain tools from the manifold optimization literature, which are highlighted in Section 6.1.1. We refer to [2, Section 5.4] for a more detailed treatment of this topic. In Section 6.1.2, we present a global sublinear rate estimate for the CD method and in Section 6.1.3, we present a local linear rate estimate under quadratic decay condition. In Section 6.1.4, we show that this condition generically holds.

6.1.1 Riemannian Geometry of the Problem

We define the following submanifold of matrices $\mathbb{R}^{n\times r}$ that corresponds to the Riemannian geometry induced by the constraints of the problem (Non-CVX) in the Euclidean space:

$$\mathcal{M}_r := \big\{\sigma = (\sigma_1, \dots, \sigma_n)^\top \in \mathbb{R}^{n\times r} : \|\sigma_i\| = 1,\ \forall i \in [n]\big\}.$$

This manifold represents the Cartesian product of n unit spheres in $\mathbb{R}^r$. For any given point $\sigma \in \mathcal{M}_r$, its tangent space can be found by taking the differential of the equality constraints as follows

$$T_\sigma \mathcal{M}_r := \big\{ u = (u_1, \ldots, u_n)^\top \in \mathbb{R}^{n \times r} : \langle u_i, \sigma_i \rangle = 0, \ \forall i \in [n] \big\}.$$

The Riemannian gradient of f on this manifold can be computed by the projection of its Euclidean gradient onto the tangent bundle. In particular, let $\mathcal{P}_\sigma^\perp : \mathbb{R}^{n \times r} \to T_\sigma \mathcal{M}_r$ denote the projection operator from the Euclidean space to the tangent space of σ. When applied to a given matrix $w = (w_1, \ldots, w_n)^\top \in \mathbb{R}^{n \times r}$, this projection operator yields

$$\begin{aligned} \mathcal{P}_\sigma^\perp(w) &= (w_1 - \langle \sigma_1, w_1 \rangle \sigma_1, \ldots, w_n - \langle \sigma_n, w_n \rangle \sigma_n)^\top, \\ &= w - \operatorname{Diag}(\operatorname{diag}(w \sigma^\top))\, \sigma. \end{aligned}$$

Therefore, the Riemannian gradient of f at σ can be computed as follows

$$\operatorname{grad} f(\sigma) = \mathcal{P}_\sigma^\perp(\nabla f(\sigma)) = 2(A - \Lambda)\sigma,$$

where $\Lambda = \operatorname{Diag}(\operatorname{diag}(A\sigma\sigma^\top))$. Or equivalently, the Riemannian gradient of f at σ can be explicitly expressed as follows

$$\operatorname{grad} f(\sigma) = 2\,(g_1 - \langle \sigma_1, g_1 \rangle \sigma_1, \ldots, g_n - \langle \sigma_n, g_n \rangle \sigma_n)^\top,$$

and its magnitude is given by

$$\|\operatorname{grad} f(\sigma)\|_{\mathrm{F}}^2 = 2 \sum_{i=1}^{n} \|g_i - \langle \sigma_i, g_i \rangle \sigma_i\|^2 = 2 \sum_{i=1}^{n} \big(\|g_i\|^2 - \langle \sigma_i, g_i \rangle^2 \big). \tag{6.1.1}$$

Using the same approach, we can calculate the Riemannian Hessian of f at σ along the direction of a vector $u \in T_\sigma \mathcal{M}_r$ by projecting the directional derivative of the gradient vector field onto the tangent space of σ as follows

$$\operatorname{Hess} f(\sigma)[u] = \mathcal{P}^\perp(\mathrm{D}\, \operatorname{grad} f(\sigma)[u]),$$

where $\mathrm{D}\,\mathrm{grad}f(\sigma)[u]$ denotes the directional gradient of $\mathrm{grad}f(\sigma)$ along the direction u. This yields

$$\mathrm{Hess}f(\sigma)[u] = \mathcal{P}^{\perp}\big(2(A-\Lambda)u - 2\mathrm{Diag}(\mathrm{diag}(A\sigma u^{\top} + Au\sigma^{\top}))\sigma\big) = \mathcal{P}^{\perp}(2(A-\Lambda)u), \tag{6.1.2}$$

and in particular, for any $u \in T_{\sigma}\mathcal{M}_r$, we have

$$\langle u, \mathrm{Hess}f(\sigma)[u]\rangle = 2\langle u, (A-\Lambda)u\rangle. \tag{6.1.3}$$

The geodesics $t \mapsto \sigma(t)$ (i.e., curves of shortest path with zero acceleration) can be expressed as a function of $\sigma = \sigma(0) \in \mathcal{M}_r$ and $u \in T_{\sigma}\mathcal{M}_r$ as follows

$$\sigma_i(t) = \sigma_i \cos(\|u_i\|t) + \frac{u_i}{\|u_i\|}\sin(\|u_i\|t). \tag{6.1.4}$$

This geodesic can be thought as the curve on the manifold that are obtained by moving from $\sigma \in \mathcal{M}_r$ towards the direction pointed by $u \in T_{\sigma}\mathcal{M}_r$. According to this definition, the exponential map $\mathrm{Exp}_{\sigma} : T_{\sigma}\mathcal{M}_r \to \mathcal{M}_r$ corresponds to evaluating the point at $t = 1$ on the geodesic function, i.e., letting $\sigma' = \mathrm{Exp}_{\sigma}(u)$, where $u \in T_{\sigma}\mathcal{M}_r$, we have

$$\sigma_i' = \sigma_i \cos(\|u_i\|) + \frac{u_i}{\|u_i\|}\sin(\|u_i\|).$$

According to this geodesic map, we can also define the following geodesic distance between two points σ and σ' on the manifold:

$$\mathrm{dist}(\sigma, \sigma') = \left(\sum_{i=1}^{n}(\arccos\langle\sigma_i, \sigma_i'\rangle)^2\right)^{1/2}. \tag{6.1.5}$$

More specifically, letting $\sigma' = \mathrm{Exp}_{\sigma}(u)$, we obtain

$$\mathrm{dist}(\sigma, \sigma') = \left(\sum_{i=1}^{n}(\arccos\langle\sigma_i, \sigma_i\cos\|u_i\|\rangle)^2\right)^{1/2} = \|u\|_{\mathrm{F}}.$$

Projection to the tangent space $T_\sigma\mathcal{M}_r$ at σ	$\mathcal{P}_\sigma^\perp(w) = \boldsymbol{w} - \mathrm{Diag}(\mathrm{diag}(w\sigma^\top))\,\sigma$
Riemannian gradient at σ	$\mathrm{grad}f(\sigma) = 2(A-\Lambda)\sigma$
Riemannian Hessian at σ along $u \in T_\sigma\mathcal{M}_r$	$\mathrm{Hess}f(\sigma)[u] = \mathcal{P}^\perp(2(A-\Lambda)u)$
Geodesic $t \to \sigma(t)$	$\sigma_i(t) = \sigma_i \cos(\|u_i\|t) + \frac{u_i}{\|u_i\|}\sin(\|u_i\|t)$
Exponential map $\sigma' = \mathrm{Exp}_\sigma(u)$	$\sigma_i' = \sigma_i \cos(\|u_i\|) + \frac{u_i}{\|u_i\|}\sin(\|u_i\|)$

Table 6.1: Summary of certain definitions stated in Section 6.1.1.

Similarly, the distance between a point σ and a non-empty, closed and (geodesically) convex set Ω can be found as

$$\mathrm{dist}(\sigma, \Omega) = \min_{\sigma' \in \Omega} \mathrm{dist}(\sigma, \sigma').$$

6.1.2 Global Rate of Convergence

In this section, we show that the CD method is globally convergent to a first-order stationary point of the problem (Non-CVX) with a sublinear rate. As a first step to prove the convergence of the CD method, we observe that the function values of the iterates generated by the CD method is a non-decreasing sequence. The increase in the function value per iteration (before reaching to stationarity) can be explicitly computed as we present in the following lemma.

Lemma 6.1. *Suppose at the k-th iteration of the CD method, i_k-th block is chosen (with some coordinate selection rule). Then, the CD method yields the following ascent on the objective value:*

$$f(\sigma^{k+1}) - f(\sigma^k) = 2\big(\|g_{i_k}^k\| - \langle \sigma_{i_k}^k, g_{i_k}^k\rangle\big) \geq 0.$$

Proof According to the decomposition in (6.0.1), we can compute the objective function as follows:

$$\begin{aligned} f(\sigma^{k+1}) &= 2\langle \sigma_{i_k}^{k+1}, g_{i_k}^{k+1}\rangle + \sum_{i \neq i_k} \sum_{j \neq i, i_k} A_{ij} \langle \sigma_i^{k+1}, \sigma_j^{k+1}\rangle, \\ &= 2\langle \sigma_{i_k}^{k+1}, g_{i_k}^{k}\rangle + \sum_{i \neq i_k} \sum_{j \neq i, i_k} A_{ij} \langle \sigma_i^{k}, \sigma_j^{k}\rangle, \end{aligned} \tag{6.1.6}$$

where the latter equality follows since $g_{i_k}^{k+1} = g_{i_k}^k$ and all the terms in the sum are independent of $\sigma_{i_k}^{k+1}$. After adding and subtracting $2\langle \sigma_{i_k}^k, g_{i_k}^k\rangle$ to the right-hand side of (6.1.6), we obtain

$$f(\sigma^{k+1}) = f(\sigma^k) + 2\big(\langle \sigma_{i_k}^{k+1}, g_{i_k}^k\rangle - \langle \sigma_{i_k}^k, g_{i_k}^k\rangle\big).$$

By the update rule of the algorithm, we have $\sigma_{i_k}^{k+1} = g_{i_k}^k / \|g_{i_k}^k\|$, and plugging this value in the above equation concludes the proof. □

In the following theorem, we consider the CD method with greedy coordinate selection and show that its functional ascent (see Lemma 6.1) can be related to the norm of the Riemannian gradient of the function evaluated at the current iterate. By doing so, we prove that the CD method returns a solution with arbitrarily small Riemannian gradient.

Theorem 6.2. *Let $f^* = \max_{\|\sigma_i\|=1, \forall i \in [n]} f(\sigma)$. Then, for any $K \geq 1$, CD with greedy coordinate selection yields the following guarantee*

$$\min_{k \in [K-1]} \|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 \leq \frac{2n\|A\|_1 (f^* - f(\sigma^0))}{K}. \tag{6.1.7}$$

Proof From Lemma 6.1, we have

$$f(\sigma^{k+1}) - f(\sigma^k) = 2\left(\|g_{i_k}^k\| - \langle \sigma_{i_k}^k, g_{i_k}^k\rangle\right) = 2 \max_{i \in [n]} \big(\|g_i^k\| - \langle \sigma_i^k, g_i^k\rangle\big),$$

where the latter equality follows by the greedy coordinate selection rule. We can rewrite this equation as follows:

$$\begin{aligned} f(\sigma^{k+1}) - f(\sigma^k) &= \max_{i\in[n]} \frac{2\|g_i^k\|\big(\|g_i^k\| - \langle \sigma_i^k, g_i^k\rangle\big)}{\|g_i^k\|}, \\ &\geq \max_{i\in[n]} \frac{\|g_i^k\|^2 - \langle \sigma_i^k, g_i^k\rangle^2}{\|g_i^k\|}, \end{aligned}$$

where the inequality follows since $\|g_i^k\| \geq \langle \sigma_i^k, g_i^k\rangle$ for all $\sigma_i^k \in \mathbb{R}^r$. Lower bounding the maximum with the mean of its arguments, we get

$$f(\sigma^{k+1}) - f(\sigma^k) \geq \frac{1}{n}\sum_{i=1}^{n} \frac{\|g_i^k\|^2 - \langle \sigma_i^k, g_i^k\rangle^2}{\|g_i^k\|}. \tag{6.1.8}$$

The $\|g_i^k\|$ term in the denominator in (6.1.8) can be upper bounded as follows

$$\|g_{i_k}^k\| \leq \sum_{j\neq i_k} |A_{i_k j}|\|\sigma_j^k\| \leq \|A\|_1. \tag{6.1.9}$$

Using this bound in (6.1.8), we get

$$f(\sigma^{k+1}) - f(\sigma^k) \geq \frac{1}{n\|A\|_1} \sum_{i=1}^{n} \big(\|g_i^k\|^2 - \langle \sigma_i^k, g_i^k\rangle^2\big) = \frac{\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2}{2n\|A\|_1}. \tag{6.1.10}$$

In order to conclude (6.1.7), we assume the contrary that $\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 > \epsilon$ for all $k \in [K-1]$. Then, using the boundedness of f, we observe that

$$f^* - f(\sigma^0) \geq f(\sigma^K) - f(\sigma^0) = \sum_{k=0}^{K-1} \big[f(\sigma^{k+1}) - f(\sigma^k)\big].$$

Using the functional ascent bound of CD in (6.1.10), we get

$$f^* - f(\sigma^0) \geq \sum_{k=0}^{K-1} \frac{\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2}{2n\|A\|_1} > \frac{K\epsilon}{2n\|A\|_1},$$

where the latter inequality follows by the assumption. Then, by contradiction, the algorithm returns a solution with $\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 \leq \epsilon$, for some $k \in [K-1]$, provided that

$$K \geq \frac{2n\|A\|_1(f^* - f(\sigma^0))}{\epsilon}.$$

□

Using a similar approach to Theorem 6.2, we show in the following corollary that the CD method with uniform and importance sampling attains a similar sublinear convergence rate in expectation. The proof of this corollary follows similar lines to the proof of Theorem 6.2, hence is deferred to Section 6.6.1.

Corollary 6.3. *Let $f^* = \max_{\|\sigma_i\|=1, \forall i \in [n]} f(\sigma)$. Then, for any $K \geq 1$, randomized CD yields the following guarantee*

$$\min_{k \in [K-1]} \mathbb{E}\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 \leq \frac{2L(f^* - f(\sigma^0))}{K}, \tag{6.1.11}$$

where

$$L = \begin{cases} n\|A\|_1, & \textit{for uniform sampling,} \\ \|A\|_{1,1}, & \textit{for importance sampling.} \end{cases} \tag{6.1.12}$$

We can observe from (6.1.7), (6.1.11) and (6.1.12) that the CD method with uniform sampling attains the same sublinear rate as the CD method with greedy coordinate selection in expectation as they both require at most $\lceil (2n\|A\|_1(f^* - f(\sigma^0)))/\epsilon \rceil$ iterations to return a solution σ satisfying $\|\mathrm{grad} f(\sigma)\|_{\mathrm{F}}^2 \leq \epsilon$. On the other hand, we see that the CD method with importance sampling enjoys a tighter convergence rate compared to the CD method with uniform sampling, as $\|A\|_{1,1} \leq n\|A\|_1$ for all $A \in \mathbb{R}^{n \times n}$.

6.1.3 Local Rate of Convergence

Although the CD method enjoys the sublinear convergence rates presented in Section 6.1.2, it is numerically observed that the rate of convergence is linear when σ^k is close to a local maximum [75, 148]. In this section, we investigate this behavior and prove that indeed CD attains a linear convergence rate around a local maximum under the quadratic decay condition on the objective function, which is classically defined as follows [6, 26]: Consider the unconstrained maximization problem: $\max_x \varphi(x)$, and let $\Omega_{\bar{x}}$ denote the set of local maximizers with objective value $\varphi(\bar{x})$. Then, the quadratic decay condition is said to be satisfied at $\bar{x}$ for φ, if there exists constants $\mu, \delta > 0$ such that $\varphi(x) \leq \varphi(\bar{x}) - \mu \operatorname{dist}^2(x, \Omega_{\bar{x}})$, for all x such that $\|x - \bar{x}\| \leq \delta$, where dist measures the distance between point x and set $\Omega_{\bar{x}}$.

For the constrained optimization problem that we are considering in (Non-CVX), this definition needs to be slightly reworked. In particular, let σ be a local maximum of (Non-CVX) and consider the Taylor expansion of $\operatorname{Exp}_\sigma(u)$ around σ:

$$f(\operatorname{Exp}_\sigma(u)) = f(\sigma) + \frac{1}{2}\langle u, \operatorname{Hess} f(\sigma)[u]\rangle + \mathcal{O}\big(\|u\|_{\mathrm{F}}^3\big),$$

where the first-order term is zero as σ is a local maximum. Then, for a sufficiently small neighborhood of σ, the quadratic decay condition is satisfied if and only if there exists a constant $\mu > 0$ such that $\langle u, \operatorname{Hess} f(\sigma)[u]\rangle \leq -\mu \operatorname{dist}^2(\operatorname{Exp}_\sigma(u), \Omega_\sigma)$, for all $\operatorname{Exp}_\sigma(u)$ sufficiently close to σ, where Ω_σ is the set on which f has constant value $f(\sigma)$. Assume for the sake of simplicity that σ is a strict local maximum, i.e., $\Omega_\sigma = \{\sigma\}$. Then, the distance between $\operatorname{Exp}_\sigma(u)$ and σ can be found as the norm of the tangent vector that connects these two points via the geodesic curve, i.e., $\operatorname{dist}(\operatorname{Exp}_\sigma(u), \sigma) = \|u\|_{\mathrm{F}}$. Therefore, the quadratic decay condition is satisfied if and only if there exists a constant $\mu > 0$ such that $\langle u, \operatorname{Hess} f(\sigma)[u]\rangle \leq -\mu\|u\|_{\mathrm{F}}^2$ for all $u \in T_\sigma\mathcal{M}_r$, where we note that the condition that $\operatorname{Exp}_\sigma(u)$ is sufficiently close to σ is dropped considering the limit as $u \to \mathbf{0}$.

Unfortunately, no local maximum is a strict local maximum for the problem (Non-CVX). To observe this, let $\mathrm{O}(r) = \{Q \in \mathbb{R}^{r\times r} : Q^\top Q = QQ^\top = I\}$ denote the orthogonal group in dimension r. Then, it can be observed that $f(\sigma Q) = \langle A, \sigma\, QQ^\top\sigma^\top\rangle = \langle A, \sigma\sigma^\top\rangle = f(\sigma)$, for any $Q \in \mathrm{O}(r)$. Therefore, in order to measure the distance between $\operatorname{Exp}_\sigma(u)$ and Ω_σ, we define the following equivalence relation $\sim$:

$$\sigma \sim \sigma' \quad \Longleftrightarrow \quad \exists Q \in O(r) : \sigma = \sigma' Q. \tag{6.1.13}$$

This equivalence relation induces a quotient space denoted by $\mathcal{M}_r / \sim$ and we let $[\sigma]$ denote the equivalence class of a given matrix $\sigma \in \mathcal{M}_r$. According to this definition, f has constant value of $f(\sigma)$ on the set $[\sigma]$, i.e., $\Omega_\sigma = [\sigma]$. We let $\mathcal{V}_\sigma \subset T_\sigma \mathcal{M}_r$ denote the tangent space to the equivalence class $[\sigma]$, which can be found as $\mathcal{V}_\sigma = \{\sigma B : B \in \mathbb{R}^{r \times r} \text{ and } B^\top = -B\}$.[1] Therefore, $\operatorname{dist}(\operatorname{Exp}_\sigma(u), [\sigma]) = \|u\|_\mathrm{F}$ if the closest point to $\operatorname{Exp}_\sigma(u)$ in $[\sigma]$ is σ, or equivalently $\operatorname{dist}(\operatorname{Exp}_\sigma(u), [\sigma]) = \|u\|_\mathrm{F}$ if $u \in T_\sigma \mathcal{M}_r \setminus \mathcal{V}_\sigma$. Consequently, we say that quadratic decay is satisfied at σ for f if $\operatorname{Hess} f(\sigma)$ is negative definite on the orthogonal complement of $\mathcal{V}_\sigma$ in $T_\sigma \mathcal{M}_r$. The formal statement of this definition is as follows.

Definition 6.4 (Quadratic Decay)**.** *Let σ be a local maximum of* (Non-CVX)*. Quadratic decay condition is said to be satisfied at σ for f if there exists a constant $\mu > 0$ such that*

$$\langle u, \operatorname{Hess} f(\sigma)[u] \rangle \leq -\mu \|u\|_\mathrm{F}^2, \quad \textit{for all} \;\; u \in T_\sigma \mathcal{M}_r \setminus \mathcal{V}_\sigma, \tag{6.1.14}$$

where $\mathcal{V}_\sigma$ is the tangent space to the equivalence class $[\sigma]$.

In the following theorem, we present the linear convergence rate of the CD method under the quadratic decay condition. We defer the validity of this condition to Section 6.1.4 where we show that quadratic decay generically (over the set of matrices A) holds for f when r is sufficiently large.

Theorem 6.5. *Let $\bar{\sigma}$ be a limit point of the CD method and assume that $\bar{\sigma}$ is a local maximum that satisfies the quadratic decay condition. If σ^0 is sufficiently close to the equivalent class $[\bar{\sigma}]$, then the iterates generated by the CD method with greedy coordinate*

[1] Note that the dimension of $\mathcal{V}_\sigma$ depends on the rank of σ, and hence the quotient space is not a manifold.

selection enjoy the following linear convergence rate

$$f(\bar{\sigma}) - f(\sigma^{k+1}) \leq \left(1 - \frac{\mu}{4n^2\|A\|_1}\right)\big(f(\bar{\sigma}) - f(\sigma^k)\big). \tag{6.1.15}$$

Proof We first discuss the outline of the proof for clarity. By (6.1.10), we have the following functional ascent bound on the iterates of the algorithm

$$f(\sigma^{k+1}) - f(\sigma^k) \geq \frac{\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2}{2n\|A\|_1}. \tag{6.1.16}$$

In order to prove linear convergence, our aim is to show that $\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 \geq c(f(\bar{\sigma}) - f(\sigma^k))$ for some positive constant c such that $c < 2n\|A\|_1$, in a neighborhood around the limit points of the iterates generated by the algorithm. To prove this, we consider the Taylor approximation of $\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2$ and $f(\sigma^k)$ around $\sigma \in [\bar{\sigma}]$, where σ is the closest point to σ^k in the set $\bar{\sigma}$. In the remainder of this proof, we show that the desired inequality holds by relating the most significant terms in these Taylor expansions. We defer bounding the higher-order terms to Section 6.6.2 in order not to distract the reader from the content.

Let $\bar{\sigma}$ be the limit point of a subsequence $\{\sigma^{k_\ell}\}_{k_\ell \geq 0}$ that contains σ^k. Then, we consider the solution $\sigma \in [\bar{\sigma}]$ such that σ is the projection of σ^k onto $[\bar{\sigma}]$, i.e., $\mathrm{dist}(\sigma, \sigma^k) \leq \mathrm{dist}(\sigma', \sigma^k)$ for all $\sigma' \in [\bar{\sigma}]$. Then, by construction there exists $\bar{u} \in T_\sigma\mathcal{M}_r \setminus \mathcal{V}_\sigma$ such that $\mathrm{Exp}_\sigma(\bar{u}) = \sigma^k$. For ease of presentation, we let $u = \bar{u}/\|\bar{u}\|_{\mathrm{F}}$ denote the normalized tangent vector and consider the following geodesic to describe σ^k:

$$\sigma_i^k = \sigma_i \cos(\|u_i\|t) + \frac{u_i}{\|u_i\|}\sin(\|u_i\|t), \tag{6.1.17}$$

where it can be observed that $t = \|\bar{u}\|_{\mathrm{F}}$ recovers the original exponential map $\sigma^k = \mathrm{Exp}_\sigma(\bar{u})$. The second order Taylor approximation to (6.1.17) yields (note that $t = \|\bar{u}\|_{\mathrm{F}} < 1$, when σ and σ^k are sufficiently close):

$$\sigma_i^k = \sigma_i + tu_i - \frac{t^2}{2}\|u_i\|^2\sigma_i + \mathcal{O}\big(t^3\big),$$

and using this approximation, we obtain

$$g_i^k = g_i + tv_i - \frac{t^2}{2}\tilde{g}_i + \mathcal{O}\big(t^3\big),$$

where

$$v_i^k = \sum_{j \neq i} A_{ij} u_j \quad \text{and} \quad \tilde{g}_i = \sum_{j \neq i} A_{ij} \|u_j\|^2 \sigma_j.$$

This yields the following Taylor approximation to $\|\text{grad} f(\sigma^k)\|_{\text{F}}^2$:

$$\begin{aligned}
\|\text{grad} f(\sigma^k)\|_{\text{F}}^2 &= 2\sum_{i=1}^{n} \Big(\|g_i^k\|^2 - \langle \sigma_i^k, g_i^k \rangle^2 \Big) \\
&= 2\sum_{i=1}^{n} \left(\|g_i + tv_i - \frac{t^2}{2}\tilde{g}_i\|^2 - \langle \sigma_i + tu_i - \frac{t^2}{2}\|u_i\|^2\sigma_i,\, g_i + tv_i - \frac{t^2}{2}\tilde{g}_i \rangle^2 \right) + \mathcal{O}\big(t^3\big), \\
&= 2\sum_{i=1}^{n} \Big\{ \|g_i\|^2 + 2t\langle g_i, v_i \rangle - t^2 \langle g_i, \tilde{g}_i \rangle + t^2 \|v_i\|^2 \\
&\qquad - \Big(\langle \sigma_i, g_i \rangle + t\langle \sigma_i, v_i \rangle - \frac{t^2}{2}\langle \sigma_i, \tilde{g}_i \rangle + t\langle u_i, g_i \rangle + t^2 \langle u_i, v_i \rangle - \frac{t^2}{2}\|u_i\|^2 \langle \sigma_i, g_i \rangle \Big)^2 \Big\} \\
&\qquad + \mathcal{O}\big(t^3\big).
\end{aligned}$$

Observe that as σ is a local maximum, we have $\sigma_i = g_i/\|g_i\|$ for all $i \in [n]$. This follows since the first-order stationarity condition implies $\sigma_i = \pm g_i/\|g_i\|$ for all $i \in [n]$; and having $\sigma_i = -g_i/\|g_i\|$ for some $i \in [n]$ conflicts with the assumption that σ is a local maximum as replacing σ_i with any other feasible point on the sphere increases the objective function. We also have that $\langle \sigma_i, u_i \rangle = 0$ for all $i \in [n]$, as $u \in T_\sigma \mathcal{M}_r$. Using these facts in the above equality, we get

$$\begin{aligned}
\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 &= 2\sum_{i=1}^n \Bigg[\|g_i\|^2 + 2t\|g_i\|\langle\sigma_i, v_i\rangle - t^2\|g_i\|\langle\sigma_i, \tilde{g}_i\rangle + t^2\|v_i\|^2 \\
&\qquad - \Bigg(\|g_i\| + t\langle\sigma_i, v_i\rangle - \frac{t^2}{2}\langle\sigma_i, \tilde{g}_i\rangle + t^2\langle u_i, v_i\rangle - \frac{t^2}{2}\|u_i\|^2\|g_i\|\Bigg)^2\Bigg] + \mathcal{O}\big(t^3\big), \\
&= 2\sum_{i=1}^n \Bigg[\|g_i\|^2 + 2t\|g_i\|\langle\sigma_i, v_i\rangle - t^2\|g_i\|\langle\sigma_i, \tilde{g}_i\rangle + t^2\|v_i\|^2 \\
&\qquad - \Bigg(\|g_i\|^2 + 2t\|g_i\|\langle\sigma_i, v_i\rangle - t^2\|g_i\|\langle\sigma_i, \tilde{g}_i\rangle + 2t^2\|g_i\|\langle u_i, v_i\rangle \\
&\qquad\qquad - t^2\|u_i\|^2\|g_i\|^2 + t^2\langle\sigma_i, v_i\rangle^2\Bigg)\Bigg] + \mathcal{O}\big(t^3\big), \\
&= 2t^2\sum_{i=1}^n \big(\|v_i\|^2 - \langle\sigma_i, v_i\rangle^2 - 2\|g_i\|\langle u_i, v_i\rangle + \|u_i\|^2\|g_i\|^2\big) + \mathcal{O}\big(t^3\big). \qquad (6.1.18)
\end{aligned}$$

Since $\langle\sigma_i, u_i\rangle = 0$ for all $i \in [n]$, we have by the Pythagorean theorem that

$$\|v_i\|^2 - \langle\sigma_i, v_i\rangle^2 - \langle\frac{u_i}{\|u_i\|}, v_i\rangle^2 \geq 0.$$

Using this inequality in (6.1.18), we get

$$\begin{aligned}
\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 &\geq 2t^2\sum_{i=1}^n \Bigg(\langle\frac{u_i}{\|u_i\|}, v_i\rangle^2 - 2\|g_i\|\langle u_i, v_i\rangle + \|u_i\|^2\|g_i\|^2\Bigg) + \mathcal{O}\big(t^3\big), \\
&= 2t^2\sum_{i=1}^n \Bigg(\|u_i\|\|g_i\| - \langle\frac{u_i}{\|u_i\|}, v_i\rangle\Bigg)^2 + \mathcal{O}\big(t^3\big). \qquad (6.1.19)
\end{aligned}$$

In order to lower bound (6.1.19) by $c(f(\sigma) - f(\sigma^k))$, we consider the second order Taylor approximation of $f(\sigma^k)$, which can be written as follows

$$\begin{aligned}
f(\sigma^k) &= \sum_{i=1}^n \langle \sigma_i^k, g_i^k\rangle, \\
&= \sum_{i=1}^n \langle \sigma_i + tu_i - \frac{t^2}{2}\|u_i\|^2\sigma_i,\; g_i + tv_i - \frac{t^2}{2}\tilde{g}_i\rangle + \mathcal{O}\big(t^3\big), \\
&= \sum_{i=1}^n \Big(\langle \sigma_i, g_i\rangle + t\langle \sigma_i, v_i\rangle - \frac{t^2}{2}\langle \sigma_i, \tilde{g}_i\rangle + t\langle u_i, g_i\rangle + t^2\langle u_i, v_i\rangle - \frac{t^2}{2}\|u_i\|^2\langle \sigma_i, g_i\rangle\Big) + \mathcal{O}\big(t^3\big).
\end{aligned}$$

Similar to the previous derivations, using the fact that $\sigma_i = g_i/\|g_i\|$ and $\langle \sigma_i, u_i\rangle = 0$ for all $i \in [n]$, we obtain

$$\begin{aligned}
f(\sigma^k) &= f(\sigma) + \sum_{i=1}^n \Big(t\langle \sigma_i, v_i\rangle - \frac{t^2}{2}\langle \sigma_i, \tilde{g}_i\rangle + t^2\langle u_i, v_i\rangle - \frac{t^2}{2}\|u_i\|^2\langle \sigma_i, g_i\rangle\Big) + \mathcal{O}\big(t^3\big), \\
&= f(\sigma) + \sum_{i=1}^n \Big(t\sum_{j\neq i} A_{ij}\langle \sigma_i, u_j\rangle - \frac{t^2}{2}\sum_{j\neq i} A_{ij}\|u_j\|^2\langle \sigma_i, \sigma_j\rangle + t^2\langle u_i, v_i\rangle - \frac{t^2}{2}\|u_i\|^2\langle \sigma_i, g_i\rangle\Big) \\
&\qquad + \mathcal{O}\big(t^3\big), \\
&= f(\sigma) + t\sum_{j=1}^n\sum_{i\neq j} A_{ji}\langle \sigma_i, u_j\rangle - \frac{t^2}{2}\sum_{j=1}^n\sum_{i\neq j} A_{ji}\|u_j\|^2\langle \sigma_i, \sigma_j\rangle \\
&\qquad + t^2\sum_{i=1}^n \Big(\langle u_i, v_i\rangle - \frac{1}{2}\|u_i\|^2\langle \sigma_i, g_i\rangle\Big) + \mathcal{O}\big(t^3\big),
\end{aligned}$$

where the last line follows since A is symmetric. Using the definition $g_j = \sum_{i\neq j} A_{ji}\sigma_i$ and $\sigma_i = g_i/\|g_i\|$ in the above inequality yields

$$\begin{aligned}
f(\sigma^k) &= f(\sigma) + t\sum_{j=1}^n \langle g_j, u_j\rangle - \frac{t^2}{2}\sum_{j=1}^n \|u_j\|^2\langle g_j, \sigma_j\rangle + t^2\sum_{i=1}^n\Big(\langle u_i, v_i\rangle - \frac{1}{2}\|u_i\|^2\langle \sigma_i, g_i\rangle\Big) \\
&\qquad + \mathcal{O}\big(t^3\big), \\
&= f(\sigma) + t^2\sum_{i=1}^n \big(\langle u_i, v_i\rangle - \|u_i\|^2\|g_i\|\big) + \mathcal{O}\big(t^3\big). \qquad (6.1.20)
\end{aligned}$$

Reorganizing terms, we get

$$f(\bar{\sigma}) - f(\sigma^k) = f(\sigma) - f(\sigma^k) = t^2 \sum_{i=1}^{n} \big(\|u_i\|^2 \|g_i\| - \langle u_i, v_i \rangle \big) + \mathcal{O}\big(t^3\big). \tag{6.1.21}$$

Turning back our attention to (6.1.19), we can lower bound the right-hand side as follows

$$\begin{aligned} \|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 &\geq 2t^2 \sum_{i=1}^{n} \frac{1}{\|u_i\|^2} \big(\|u_i\|^2 \|g_i\| - \langle u_i, v_i \rangle \big)^2 + \mathcal{O}\big(t^3\big), \\ &\geq 2t^2 \sum_{i=1}^{n} \big(\|u_i\|^2 \|g_i\| - \langle u_i, v_i \rangle \big)^2 + \mathcal{O}\big(t^3\big), \\ &\geq \frac{2t^2}{n} \left(\sum_{i=1}^{n} \big(\|u_i\|^2 \|g_i\| - \langle u_i, v_i \rangle \big) \right)^2 + \mathcal{O}\big(t^3\big), \end{aligned}$$

where the second inequality follows since $\|u_i\|^2 \leq \|u\|_{\mathrm{F}}^2 = 1$ and the last inequality follows since $\big(\sum_{i=1}^n a_i\big)^2 \leq n \sum_{i=1}^n a_i^2$, for all $a_i \in \mathbb{R}$, $i \in [n]$. Using the second order approximation derived in (6.1.21) in the above inequality, we obtain

$$\begin{aligned} \|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 &\geq \frac{f(\bar{\sigma}) - f(\sigma^k)}{n} \sum_{i=1}^{n} 2\big(\|u_i\|^2 \|g_i\| - \langle u_i, v_i \rangle \big) + \mathcal{O}\big(t^3\big), \\ &= \frac{2\langle u, (\Lambda - A)u \rangle}{n} \big(f(\bar{\sigma}) - f(\sigma^k) \big) + \mathcal{O}\big(t^3\big), \end{aligned}$$

where $\Lambda = \mathrm{Diag}(\|g_1\|, \ldots, \|g_n\|)$. Since we have $2\langle u, (A - \Lambda)u \rangle \leq -\mu \|u\|_{\mathrm{F}}^2$ by the quadratic decay condition, we conclude that

$$\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 \geq \frac{\mu}{n} \big(f(\bar{\sigma}) - f(\sigma^k) \big) + \mathcal{O}\big(t^3\big). \tag{6.1.22}$$

This implies that whenever σ^k is sufficiently close to σ, i.e., whenever t is sufficiently small (cf. (6.1.17)), the remainder in the Taylor approximation, i.e., the $\mathcal{O}(t^3)$ terms, will be dominated by $\frac{\mu}{n}\big(f(\bar{\sigma}) - f(\sigma^k)\big)$. In particular, if σ^0 is sufficiently close to $\bar{\sigma}$ to satisfy $\mathcal{O}(t^3) \geq -\frac{\mu}{2n}\big(f(\bar{\sigma}) - f(\sigma^k)\big)$ in the above inequality (see Section 6.6.2 for a proof of this), we then have

$$\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 \geq \frac{\mu}{2n}\big(f(\bar{\sigma}) - f(\sigma^k)\big). \tag{6.1.23}$$

Combining this inequality with (6.1.16), we get

$$f(\sigma^{k+1}) - f(\sigma^k) \geq \frac{\mu}{4n^2\|A\|_1}\big(f(\bar{\sigma}) - f(\sigma^k)\big). \tag{6.1.24}$$

Rearranging terms in the above inequality concludes the proof. □

The linear convergence rate of the CD method with greedy coordinate selection in Theorem 6.5 can be extended for importance sampling and uniform sampling as we highlight in the following (its proof follows similar lines to the proofs of Theorem 6.5 and Corollary 6.3 and hence is omitted).

Corollary 6.6. *Let the conditions in Theorem 6.5 hold. Then, the iterates generated by the CD method enjoys the local linear convergence rate*

$$f(\bar{\sigma}) - \mathbb{E}f(\sigma^k) \leq (1-\rho)^k\big(f(\bar{\sigma}) - f(\sigma^0)\big),$$

where $\rho = \frac{\mu}{4n\|A\|_{1,1}}$ *for importance sampling and* $\rho = \frac{\mu}{4n^2\|A\|_1}$ *for uniform sampling.*

6.1.4 Quadratic Decay Condition Holds Generically

In this section, we consider the quadratic decay condition, which is a condition on (Non-CVX), and relate it to a condition on the original problem in (CVX). In particular, we characterize sufficient conditions on (CVX) for quadratic decay to hold. We first provide some background on semidefinite programming (see for example [3], for a more detailed treatment of this topic). Consider the SDP in (CVX):

$$\begin{aligned} \text{maximize} \quad & \langle A, X\rangle \\ \text{subject to} \quad & X_{ii} = 1, \text{ for } i \in [n], \\ & X \succeq 0, \end{aligned}$$

and its dual:

$$\begin{aligned} \text{minimize} \quad & \langle 1, y\rangle \\ \text{subject to} \quad & Z = \operatorname{Diag}(y) - A, \\ & Z \succeq 0, \end{aligned}$$

where 1 is the vector of ones of appropriate size. Let X^* and (y^*, Z^*) denote the primal and dual optimal solutions, respectively, and let r^* denote the rank of X^*. Then, there exists a $Q \in O(n)$ such that

$$\begin{aligned} X^* &= Q \operatorname{Diag}(\lambda_1, \ldots, \lambda_{r^*}, 0, \ldots, 0)\, Q^\top, \\ Z^* &= Q \operatorname{Diag}(0, \ldots, 0, \omega_{r^*+1}, \ldots, \omega_n)\, Q^\top. \end{aligned}$$

We say that *strict complementarity* holds if $\lambda_i > 0$ for $i = 1, \ldots, r^*$ and $\omega_j > 0$ for $j = r^* + 1, \ldots, n$. Furthermore, let $Q_1 \in \mathbb{R}^{n \times r^*}$ and $Q_2 \in \mathbb{R}^{n \times (n-r^*)}$ respectively denote the first r^* columns and the last $n - r^*$ columns of Q and let q_i denote the ith row of Q_1, i.e., $Q_1 = [q_1, q_2, \ldots, q_n]^\top$. Then, (y^*, Z^*) is *dual nondegenerate* if and only if $\{q_1 q_1^\top, \ldots, q_n q_n^\top\}$ spans Sym_{r^*}, i.e., the set of real symmetric $r^* \times r^*$ matrices [3, Theorem 3]. Strict complementarity and dual nondegeneracy are known to hold generically (over the set of possible cost matrices $A \in \mathbb{R}^{n \times n}$, i.e., they fail to hold only on a subset of measure zero of $\mathbb{R}^{n \times n}$) as proven in [3, Lemma 2]. Using these definitions, we show in the next theorem that strict complementarity and dual nondegeneracy are sufficient for quadratic decay to hold at the maximizer of (Non-CVX).

Theorem 6.7. *Suppose that* $X^* = \sigma\sigma^\top$ *and* $(y^*, Z^*) = (\mathrm{diag}(\Lambda), \Lambda - A)$ *are respectively primal and dual optimal solutions satisfying strict complementarity and dual nondegeneracy, where* $\Lambda = \mathrm{Diag}(\|g_1\|, \ldots, \|g_n\|)$. *If* $r \geq \mathrm{rank}(X^*)$, *then quadratic decay is satisfied for* f *at all* $\bar{\sigma}$ *such that* $\bar{\sigma}\bar{\sigma}^\top = X^*$.

Proof Suppose $\mathrm{rank}(X^*) = r^* \leq r$, then by strict complementarity, we have $\mathrm{rank}(Z^*) = n - r^*$ and kernel of Z^* is equal to the column space of X^*, i.e., $\ker(Z^*) = \mathrm{col}(X^*)$. Since $X^* = \sigma\sigma^\top$ and $Z^* = \Lambda - A$, we equivalently have $\ker(\Lambda - A) = \mathrm{col}(\sigma)$. As Z^* is feasible for the dual, then $Z^* = \Lambda - A \succeq 0$, and consequently $\langle u, (\Lambda - A)u \rangle \geq 0$, for all $u \in \mathbb{R}^{n \times r}$.

Now consider the quadratic form $h(u) := \langle u, (\Lambda - A)u \rangle$ over $u \in T_\sigma\mathcal{M}_r$. First, we show that $h(u) = 0$ if and only if $u \in \mathcal{V}_\sigma$. The *if* direction of the proof is straightforward, i.e., $(\Lambda - A)\sigma = 0$ and $u = \sigma B$ for some skew-symmetric matrix B directly imply $h(u) = 0$ for all $u \in \mathcal{V}_\sigma$. To show the *only if* direction, let $u \in T_\sigma\mathcal{M}_r$ such that $h(u) = 0$, or equivalently $\mathrm{tr}((\Lambda - A)uu^\top) = 0$. As both $\Lambda - A$ and $uu^\top$ are positive semidefinite matrices, this implies $(\Lambda - A)u = 0$. Therefore, columns of u are in $\ker(\Lambda - A) = \mathrm{col}(\sigma)$, which implies there exists $B \in \mathbb{R}^{r \times r}$ such that $u = \sigma B$ (note that it is not possible to make this claim without strict complementarity). As $u \in T_\sigma\mathcal{M}_r$, then $\langle \sigma_i, u_i \rangle = \langle \sigma_i, B^\top \sigma_i \rangle = \langle \sigma_i\sigma_i^\top, B \rangle = 0$, for all $i \in [n]$. Without loss of generality, assume that the last $r - r^*$ columns of σ are equal to zero. Then, by dual nondegeneracy of the SDP, the principal submatrices of dimension $r^* \times r^*$ of $\{\sigma_i\sigma_i^\top\}_{i=1}^n$ spans $\mathcal{S}^{r^*}$. Consider the decomposition

$$B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix},$$

where $B_{11} \in \mathbb{R}^{r^* \times r^*}$ and $B_{22} \in \mathbb{R}^{(r-r^*)\times(r-r^*)}$. Then, the dual nondegeneracy implies that B_{11} is a skew-symmetric matrix, i.e., $B_{11}^\top = -B_{11}$. Furthermore, as the last $r - r^*$ columns of σ are equal to zero, then $u = \sigma B$ does not depend on B_{21} and B_{22}. Therefore, we can pick $B_{21} = -B_{12}^\top$ and $B_{22} = 0$ such that B is a skew-symmetric matrix and observe that $u \in \mathcal{V}_\sigma$. The same argument can be extended for all $\bar{\sigma}$ such that $\bar{\sigma}\bar{\sigma}^\top = X^*$ using parallel transport.

To conclude the proof, we let $\{u^\ell\}_{\ell=1}^{n(r-1)}$ be an orthogonal basis to $T_\sigma\mathcal{M}_r$ such that $\{u^\ell\}_{\ell=1}^{s}$ is a basis for $\mathcal{V}_\sigma$. Let $M \in \mathbb{R}^{n(r-1)\times n(r-1)}$ such that $M_{ij} = \langle u^i, (\Lambda - A)u^j\rangle$. Consider the function $\bar{h} : \mathbb{R}^{n(r-1)} \to \mathbb{R}^{n(r-1)}$ such that $\bar{h}(v) = v^\top M v$ and observe that $\bar{h}(\mathrm{vec}(u)) = h(u)$. Let $L = [\mathrm{vec}(u^1), \ldots, \mathrm{vec}(u^s)]^\top \in \mathbb{R}^{s\times n(r-1)}$, then $v^\top M v > 0$ for all v such that $Lv = 0$ and $v \neq \mathbf{0}$. Then, by Finsler's Lemma, $L_\perp^\top M L_\perp \succ 0$, where $L_\perp$ is any basis of the right null-space of L. Equivalently, there exists $\mu > 0$ such that $h(u) \geq \mu\|u\|_{\mathrm{F}}^2$ for all $u \in T_\sigma\mathcal{M}_r \setminus \mathcal{V}_\sigma$. □

Remark 6.8. *Finsler's Lemma [36, Lemma C.11.2] also yields that $\mu = \lambda_{\min}(L_\perp^\top M L_\perp)$.*

This theorem states that quadratic decay holds for all global maxima of (Non-CVX) provided that the rank of the factorization is large enough so that the global maximum values of (CVX) and (Non-CVX) are equal to one another. For this case, the set of all global maxima is an equivalence class corresponding to a solution since strict complementarity and dual nondegeneracy imply that the primal solution of (CVX) is unique. On top of this, when $r \geq \sqrt{2n}$, it is known that (see [32, Theorem 2]) any local maximum is global generically (i.e., for almost all cost matrices A). As strict complementarity and dual nondegeneracy also hold generically for (CVX), then consequently, when $r \geq \sqrt{2n}$, quadratic decay holds for all local maxima generically as we highlight in the following corollary.

Corollary 6.9. *If $r \geq \sqrt{2n}$, then quadratic decay holds for all local maxima generically.*

6.2 Approximately Achieving the Maximum Value of (CVX)

Our results in Section 6.1 show that the CD method converges with a sublinear rate to a first-order stationary solution and with a linear rate to a local maximum when initialized sufficiently close to it. In this section, we incorporate a second-order oracle to the CD method in order to obtain an algorithm, which we refer as CD2, that returns an approximate second-order stationary point. More specifically, at the current iteration of the algorithm, if the norm of the gradient is large, we take a CD step. Otherwise, we run a subroutine (e.g., Lanczos method) to find the leading eigenvector of the Hessian. The main motivation for designing such an algorithm is that the approximate second-order stationary solutions provide $\mathcal{O}(1/r)$ approximation to (CVX). In particular, call σ an *ε-approximate concave point* if $\langle u, \mathrm{Hess} f(\sigma)[u]\rangle \leq \varepsilon \langle u, u\rangle$, for all $u \in T_\sigma \mathcal{M}_r$. Then, the following theorem provides an approximation ratio between the approximate concave points of (Non-CVX) and the maximum value of (CVX).

Theorem 6.10 ([101, Theorem 1]). *Let $\sigma \in \mathcal{M}_r$ be an ε-approximate concave point. Then, for any positive semidefinite A, the following approximation ratio holds:*

$$f(\sigma) \geq \left(1 - \frac{1}{r-1}\right) \mathrm{SDP}(A) - \frac{n}{2}\varepsilon, \tag{6.2.1}$$

where $\mathrm{SDP}(A)$ is the maximum value of (CVX).

This approximation ratio follows due to a generalization of the randomized rounding approach (most famously presented by [67]) applied to an ε-approximate concave point. In fact, it can be shown that it is not possible to find a better approximation ratio (in terms of the dependence on the rank of the factorization r) for all problems A. This result is highlighted in the following theorem.

Theorem 6.11 ([37, Theorems 1 & 3]). *Let $\mathrm{SDP}(A)$ be the maximum value of (CVX) and $\mathrm{SDP}_r(A)$ be the maximum value of (Non-CVX). Then, for all positive semidefinite matrices A, the following approximation ratio holds:*

$$1 \geq \frac{\mathrm{SDP}_r(A)}{\mathrm{SDP}(A)} \geq \gamma(r) = \frac{2}{r}\left(\frac{\Gamma((r+1)/2)}{\Gamma(r/2)}\right)^2 = 1 - \Theta(1/r), \tag{6.2.2}$$

where $\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx$ is the Gamma function. Furthermore, under the unique games conjecture, there is no polynomial-time algorithm that approximates $\mathrm{SDP}_r(A)$ with an approximation ratio greater than $\gamma(r) + \varepsilon$ for any $\varepsilon > 0$.

Algorithm 4: CD2

```
Initialize $\sigma^0 \in \mathbb{R}^{n\times r}$ and calculate $g_i^0 = \sum_{j\neq i} A_{ij}\sigma_j^0$, for all $i \in [n]$.
for $k = 0, 1, 2, \dots$ do
    Compute $\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 = 2\sum_{i=1}^n (\|g_i^k\|^2 - \langle \sigma_i^k, g_i^k\rangle^2)$.
    if $\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 > \varepsilon^3/(1350\|A\|_1)$ then
        $i_k \leftarrow \arg\max_{i\in[n]} (\|g_i^k\| - \langle \sigma_i^k, g_i^k\rangle)$
        $\sigma_{i_k}^{k+1} \leftarrow g_{i_k}^k / \|g_{i_k}^k\|$.
        $g_i^{k+1} \leftarrow g_i^k - A_{ii_k}\sigma_{i_k}^k + A_{ii_k}\sigma_{i_k}^{k+1}$, for all $i \neq i_k$.
    else
        Find a direction $u^k \in T_{\sigma^k}\mathcal{M}_r$ such that $\langle u^k, \mathrm{Hess} f(\sigma^k)[u^k]\rangle \geq \lambda_{\max}(\mathrm{Hess} f(\sigma^k))/2$,
        $\langle u^k, \mathrm{grad} f(\sigma^k)\rangle \geq 0$, and $\|u^k\|_{\mathrm{F}} = 1$.
        $\sigma_i^{k+1} \leftarrow \sigma_i^k \cos(\|u_i^k\|t) + \frac{u_i^k}{\|u_i^k\|}\sin(\|u_i^k\|t)$, for all $i \in [n]$, where $t = \varepsilon/(15\|A\|_1)$.
        $g_i^{k+1} \leftarrow \sum_{j\neq i} A_{ij}\sigma_j^{k+1}$, for all $i \in [n]$.
    end if
end for
```

These results provide motivation to design algorithms with second-order guarantees to solve (Non-CVX) and for this reason, we propose the CD2 algorithm (see Algorithm 4), which can be described as follows: When the Frobenius norm of the Riemannian gradient is at least as large as $\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 > \epsilon^3/(1350\|A\|_1)$, we use the CD method to update the current solution. Otherwise, we assume that there is a second-order oracle that returns an update direction $u^k \in T_{\sigma^k}\mathcal{M}_r$ such that $\langle u^k, \mathrm{Hess} f(\sigma^k)[u^k]\rangle \geq \lambda_{\max}(\mathrm{Hess} f(\sigma^k))/2$, $\langle u^k, \mathrm{grad} f(\sigma^k)\rangle \geq 0$, and $\|u^k\|_{\mathrm{F}} = 1$. Notice that finding a tangent vector u^k that satisfy $\langle u^k, \mathrm{Hess} f(\sigma^k)[u^k]\rangle \geq \lambda_{\max}(\mathrm{Hess} f(\sigma^k))/2$ and $\|u^k\|_{\mathrm{F}} = 1$ is an eigenpair problem and can be solved efficiently using the Lanczos method. The condition $\langle u^k, \mathrm{grad} f(\sigma^k)\rangle \geq 0$, on the other hand, can always be satisfied by switching the sign of u^k. It is a straightforward exercise to explicitly construct such a vector and it can be found in [30, Lemma 11]. Once the update direction $u^k \in T_{\sigma^k}\mathcal{M}_r$ is obtained, we take a step towards this direction using the geodesics on the manifold. When the step size is carefully chosen, it can be shown that the objective value of the iterates generated by this procedure is a monotonically increasing sequence until the approximate second-order stationary condition is satisfied. This property is presented in the following lemma.

Lemma 6.12. *Let $u^k \in T_{\sigma^k}\mathcal{M}_r$ such that $\langle u^k, \mathrm{Hess} f(\sigma^k)[u^k]\rangle \geq \varepsilon/2$, $\langle u^k, \mathrm{grad} f(\sigma^k)\rangle \geq 0$ and $\|u^k\|_{\mathrm{F}} = 1$. Consider the update rule given by the exponential map $\sigma^{k+1} = \mathrm{Exp}_{\sigma^k}(tu^k)$, i.e.,*

$$\sigma_i^{k+1} = \sigma_i^k \cos(\|u_i^k\|t) + \frac{u_i^k}{\|u_i^k\|}\sin(\|u_i^k\|t), \qquad \text{for all } i \in [n], \tag{6.2.3}$$

where $t = \frac{\varepsilon}{15\|A\|_1}$ is the step size. These iterates satisfy the following ascent in the function value:

$$f(\sigma^{k+1}) - f(\sigma^k) \geq \frac{\varepsilon^3}{2700\|A\|_1^2}.$$

Proof The Taylor expansion of σ^{k+1} around σ^k is given by

$$\begin{aligned}\sigma_i^{k+1} &= \sigma_i^k \sum_{\ell=0}^{\infty} \frac{(-1)^\ell}{(2\ell)!}(\|u_i^k\|t)^{2\ell} + u_i^k \sum_{\ell=0}^{\infty} \frac{(-1)^\ell}{(2\ell+1)!}(\|u_i^k\|t)^{2\ell+1}, \\ &= \sigma_i^k + tu_i^k - \frac{t^2}{2}\|u_i^k\|^2\sigma_i^k - \frac{t^3}{6}\|u_i^k\|^2 u_i^k + \ldots,\end{aligned}$$

and using this, we can compute the Taylor expansion of $f(\sigma^{k+1})$ as follows

$$\begin{aligned}f(\sigma^{k+1}) &= \sum_{i=1}^{n}\sum_{j\neq i} A_{ij}\langle \sigma_i^{k+1}, \sigma_j^{k+1}\rangle, \\ &= \sum_{i=1}^{n}\sum_{j\neq i} A_{ij}\Big[\langle \sigma_i^k, \sigma_j^k\rangle + t\big(\langle \sigma_i^k, u_j^k\rangle + \langle u_i^k, \sigma_j^k\rangle\big) \\ &\qquad + \frac{t^2}{2}\big(-\|u_j^k\|^2\langle \sigma_i^k, \sigma_j^k\rangle + 2\langle u_i^k, u_j^k\rangle - \|u_i^k\|^2\langle \sigma_i^k, \sigma_j^k\rangle\big)\Big] - t^3\beta,\end{aligned}$$

where β represents the third and higher-order terms. Using the definitions of $f(\sigma^k)$ and its derivatives, the above equality can be written as follows

$$f(\sigma^{k+1}) = f(\sigma^k) + t\langle u^k, \mathrm{grad} f(\sigma^k)\rangle + \frac{t^2}{2}\langle u^k, \mathrm{Hess} f(\sigma^k)[u^k]\rangle - t^3\beta. \tag{6.2.4}$$

Here, our aim is to upper bound the magnitude of the remainder term corresponding to the third and higher-order terms. To this end, we upper bound the higher-order terms using the Cauchy-Schwarz inequality for each term individually. This yields

$$|\beta| \leq \sum_{i=1}^{n} \sum_{j \neq i} |A_{ij}| \left(\sum_{\ell=3}^{\infty} \frac{t^{\ell-3}}{\ell!} (\|u_i^k\| + \|u_j^k\|)^{\ell} \right).$$

As $t < 1$ and A is a symmetric matrix, we can upper bound the right hand-side of the above inequality as follows

$$|\beta| \leq \|A\|_1 \sum_{i=1}^{n} \left(\sum_{\ell=3}^{\infty} \frac{2^{\ell}}{\ell!} \|u_i^k\|^{\ell} \right).$$

Since $\|u_i\| \leq 1$ for all $i \in [n]$, we consequently have

$$|\beta| \leq \|A\|_1 \left(\sum_{i=1}^{n} \|u_i^k\|^2 \right) \left(\sum_{\ell=3}^{\infty} \frac{2^{\ell}}{\ell!} \right) = \|A\|_1 \sum_{\ell=3}^{\infty} \frac{2^{\ell}}{\ell!}.$$

where the latter equality follows since $\|u^k\|_{\mathrm{F}} = 1$. Using $\sum_{\ell=3}^{\infty} \frac{2^{\ell}}{\ell!} = e^2 - 5 \leq 5/2$ above and plugging this bound back in (6.2.4), we obtain

$$f(\sigma^{k+1}) \geq f(\sigma^k) + t\langle u^k, \mathrm{grad} f(\sigma^k)\rangle + \frac{t^2}{2}\langle u^k, \mathrm{Hess} f(\sigma^k)[u^k]\rangle - \frac{5\|A\|_1}{2} t^3. \tag{6.2.5}$$

Since we are given that $\langle u^k, \mathrm{grad} f(\sigma^k)\rangle \geq 0$ and $\langle u^k, \mathrm{Hess} f(\sigma^k)[u^k]\rangle \geq \varepsilon/2$, (6.2.5) yields

$$f(\sigma^{k+1}) - f(\sigma^k) \geq \frac{\varepsilon}{4} t^2 - \frac{5\|A\|_1}{2} t^3.$$

Choosing $t = \frac{\varepsilon}{15\|A\|_1}$ maximizes the right-hand side of the above inequality and concludes the proof. □

Using this ascent lemma, we next analyze the global convergence of Algorithm 4 in Theorem 6.13, where we assume that we have access to a subroutine that solves the eigenpair problem to the desired accuracy. We then implement the subroutine using the Lanczos algorithm (presented in Algorithm 5) and present its convergence in Theorem 6.17. In particular, we have the following theorem for the former case.

Theorem 6.13. *Suppose that in Algorithm 4, the CD method is used at iteration k when $\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 \geq \varepsilon^3/(1350\|A\|_1)$ and a second-order step (see lines 9-11 of Algorithm 4) is taken otherwise. Let K_{CD} denote the number of CD epochs made and let K_{H} denote the number of second-order oracle iterations made such that $K = nK_{\mathrm{CD}} + K_{\mathrm{H}}$. Then, as soon as*

$$K_{\mathrm{CD}} + K_{\mathrm{H}} = \left\lceil \frac{675n\|A\|_1^2}{\varepsilon^2} \right\rceil, \tag{6.2.6}$$

Algorithm 4 is guaranteed to return a solution σ^K that satisfies

$$f(\sigma^K) \geq \left(1 - \frac{1}{r-1}\right)\mathrm{SDP}(A) - \frac{n}{2}\varepsilon, \tag{6.2.7}$$

where SDP(A) *is the maximum value of* (CVX).

Proof As we have proven previously in (6.1.10), each iteration of CD yields the following functional ascent

$$f(\sigma^{k+1}) - f(\sigma^k) \geq \frac{\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2}{2n\|A\|_1} \geq \frac{\varepsilon^3}{2700n\|A\|_1^2}, \tag{6.2.8}$$

where the latter inequality holds since the CD method is applied at iteration k of Algorithm 4 if $\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 \geq \frac{\varepsilon^3}{1350\|A\|_1}$. Similarly, by Lemma 6.12, each iteration of the second-order oracle yields the following functional ascent

$$f(\sigma^{k+1}) - f(\sigma^k) \geq \frac{\varepsilon^3}{2700\|A\|_1^2}. \tag{6.2.9}$$

Hence, an epoch (n iterations) of CD yields the same amount of function value improvement as an iteration of the second-order oracle. Let

$$f^* = \left(1 - \frac{1}{r-1}\right)\mathrm{SDP}(A)$$

denote the desired approximation ratio and consider the approximation gap of the solution

σ with respect to f^* that is given by

$$h(\sigma) = f^* - f(\sigma). \tag{6.2.10}$$

The aim of the algorithm is to find a solution σ that satisfy $h(\sigma) \leq \epsilon$ for some $\epsilon > 0$. Consider that the CD2 method runs K_{CD} epochs of CD and K_{H} iterations of the second-order oracle such that a total of $K = nK_{\mathrm{CD}} + K_{\mathrm{H}}$ iterations are made. Let $\mathcal{G} = \{0 \leq k \leq K-1 : \|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 \geq \frac{\varepsilon^3}{1350\|A\|_1}\}$ be the set of iterations at which CD step is taken and let $\mathcal{H} = \{0 \leq k \leq K-1\} \setminus \mathcal{G}$ be the set of iterations at which a second-order oracle step is taken. Then, the approximation gap decreases at each iteration by the following amount:

$$h(\sigma^k) - h(\sigma^{k+1}) \geq \frac{\varepsilon^3}{2700\|A\|_1^2}\delta_k, \tag{6.2.11}$$

where, for notational simplicity, we introduced

$$\delta_k = \begin{cases} \frac{1}{n}, & \text{if } k \in \mathcal{G}, \\ 1, & \text{if } k \in \mathcal{H}. \end{cases} \tag{6.2.12}$$

By Theorem 6.10, we are given that any ε-approximate concave point σ satisfies

$$h(\sigma) \leq \frac{n}{2}\varepsilon. \tag{6.2.13}$$

Hence, the right-hand side of (6.2.11) can be lower bounded as follows

$$h(\sigma^k) - h(\sigma^{k+1}) \geq \frac{2\delta_k}{675n^3\|A\|_1^2} h^3(\sigma^k). \tag{6.2.14}$$

Considering the reciprocal of the approximation gap, we observe that

$$\begin{aligned}\frac{1}{h^2(\sigma^{k+1})} - \frac{1}{h^2(\sigma^k)} &= \frac{\big(h(\sigma^k) - h(\sigma^{k+1})\big)\big(h(\sigma^k) + h(\sigma^{k+1})\big)}{h^2(\sigma^{k+1})h^2(\sigma^k)}, \\ &\geq \frac{2\delta_k}{675n^3\|A\|_1^2}\frac{h(\sigma^k)\big(h(\sigma^k) + h(\sigma^{k+1})\big)}{h^2(\sigma^{k+1})}, \end{aligned} \tag{6.2.15}$$

where the inequality follows by (6.2.14). As the right-hand side of (6.2.14) is lower bounded by zero, we have $h(\sigma^k) \geq h(\sigma^{k+1})$. Thus, we can lower bound the right-hand side of (6.2.15) as follows

$$\frac{1}{h^2(\sigma^{k+1})} - \frac{1}{h^2(\sigma^k)} \geq \frac{4\delta_k}{675n^3\|A\|_1^2}. \tag{6.2.16}$$

Summing (6.2.16) over $k = 0, 1, \ldots, K-1$, we get

$$\frac{1}{h^2(\sigma^K)} - \frac{1}{h^2(\sigma^0)} \geq \sum_{k=0}^{K-1} \frac{4\delta_k}{675n^3\|A\|_1^2} = \frac{4}{675n^3\|A\|_1^2}(K_{\text{CD}} + K_{\text{H}}).$$

Given that σ^0 is not an ε-approximate concave point (or else, there is nothing to prove), we have

$$\frac{1}{h^2(\sigma^K)} \geq \frac{4}{675n^3\|A\|_1^2}(K_{\text{CD}} + K_{\text{H}}). \tag{6.2.17}$$

Since by (6.2.13), we know that $\frac{1}{h(\sigma)} \geq \frac{2}{n\varepsilon}$ for any ε-approximate concave point, then as soon as

$$K_{\text{CD}} + K_{\text{H}} \geq \frac{675n\|A\|_1^2}{\varepsilon^2} \tag{6.2.18}$$

iterations made, the CD2 method is guaranteed to return an ε-approximate concave point, i.e., there exists a solution σ^k for some $1 < k < K$ such that $h(\sigma^k) \leq \frac{n}{2}\varepsilon$. Since $\{h(\sigma^k)\}_{k\geq 0}$ is a nonincreasing sequence (as we have already shown in (6.2.14)), then the final iterate of the algorithm σ^K is guaranteed to satisfy $h(\sigma^K) \leq \frac{n}{2}\varepsilon$, i.e., σ^K is an ε-approximate concave point. □

In Theorem 6.13, $K_{\mathrm{CD}} + K_{\mathrm{H}}$ represents the total number of epochs to guarantee (6.2.7), whereas the iteration counter of the algorithm is given in terms of $K = nK_{\mathrm{CD}} + K_{\mathrm{H}}$. This is due to the fact that, at each iteration of the CD method, a single row of σ is updated and consequently n iterations of the CD method add up to an epoch. On the other hand, at each iteration of the second-order step, all entries of σ are updated, and hence each second-order iteration is an epoch. In terms of the computational cost, an iteration of CD requires $\mathcal{O}(nr)$ operations and consequently an epoch of CD requires $\mathcal{O}(n^2 r)$ operations, whereas the second-order direction of update is typically found approximately via a few iterations of the power method or the Lanczos method (see Theorem 6.17 for a more rigorous treatment of this statement), which require $\mathcal{O}(n^2 r)$ operations. Therefore, an epoch of Algorithm 4 typically has a computational complexity of $\mathcal{O}(n^2 r)$. Furthermore, by Theorem 6.13, we observe that in at most $\mathcal{O}(n\|A\|_1^2/\varepsilon^2)$ epochs, Algorithm 4 returns a solution that achieves the optimal approximation ratio up to an accuracy of $\mathcal{O}(n\varepsilon)$. In particular, picking $\varepsilon = 2\,\mathrm{SDP}(A)/(n(r-1))$, we obtain the following corollary.

Corollary 6.14. *Consider the setup of Theorem 6.13 and set $\varepsilon = 2\,\mathrm{SDP}(A)/(n(r-1))$. Then, as soon as*

$$K = \left\lceil \frac{675 n^3 (r-1)^2 \|A\|_1^2}{4(\mathrm{SDP}(A))^2} \right\rceil, \tag{6.2.19}$$

Algorithm 4 is guaranteed to return a solution σ^K that satisfies

$$f(\sigma^K) \geq \left(1 - \frac{2}{r-1}\right)\mathrm{SDP}(A).$$

Remark 6.15. *In order to understand the total running time of CD2, consider the following example. Let A be the adjacency matrix of a random Erdos-Rényi graph on n nodes and $\lfloor cn \rfloor$ edges. The size of the maximum cut in this graph normalized by the number of nodes can be bounded between $[c/2 + 0.4\sqrt{c}, c/2 + 0.6\sqrt{c}]$ with high probability as n increases, for all sufficiently large c [65]. Since the maximum value of (CVX) is within 0.878 of the maximum cut [67], we can then conclude that $\mathrm{SDP}(A)/n = \mathcal{O}(c)$ with high probability. We can also observe that for this graph, the degree of a node approximately follows a Poisson* *distribution with mean $2c$, which can be approximated by a normal distribution with mean $2c$ and variance $\sqrt{2c}$, for large c [65]. Then, we have $\|A\|_1 = \mathcal{O}(c \log n)$ with high probability. Therefore, for this problem, Corollary 6.14 states that in $\widetilde{\mathcal{O}}(nr^2)$ iterations (where tilde is used to hide the logarithmic dependences), Algorithm 4 returns a $\mathcal{O}(1/r)$-optimal solution with high probability. Per iteration computational cost of the algorithm is $\mathcal{O}(nrc)$, which results in a total running time of $\widetilde{\mathcal{O}}(n^2r^3c)$. In comparison, Klein-Lu method (see [82, Lemma 4]) requires $\widetilde{\mathcal{O}}(n^2r^3c)$ running time and the matrix multiplicative weights method (see [7, Theorem 3]) requires $\widetilde{\mathcal{O}}(n^2r^{3.5}/c)$ running time to return a $1/r$-optimal solution.*

Remark 6.16. *It has been shown in [120, Theorem 3.1] and [50, Theorem 3.5] that an exactly feasible approximately second-order stationary point to* (Non-CVX) *is also approximately optimal for* (CVX)*. Our CD2 method returns such a solution and in light of these results, we can conclude that it finds a high-quality solution to* (CVX) *with high probability whenever $r \geq \sqrt{2n}$. See Figure 6-4 for an empirical validation of this result.*

In the description of Algorithm 4 (see line 9), we assumed that we have access to a vector in the tangent space of the current iterate, which satisfies certain second-order conditions. In Algorithm 5, we describe an efficient subroutine to find this desired tangent vector based on the Lanczos method. In particular, the Lanczos method returns a tridiagonal real symmetric matrix whose diagonal entries are $\{\alpha_\ell\}_{\ell \geq 1}$ and off-diagonal entries are $\{\beta_\ell\}_{\ell \geq 2}$, where ℓ denotes the iteration counter in Algorithm 5. The entire spectrum of such a

Algorithm 5: Lanczos Method

```
Given $\sigma$, define $H[u] = \mathrm{Hess} f(\sigma)[u] + 4\|A\|_1 u$. Initialize $u_1 \in T_\sigma \mathcal{M}_r$ such that
$\|u_1\|_\mathrm{F} = 1$. Let $\alpha_1 = \langle u_1, H[u_1] \rangle$ and $r_1 = H[u_1] - \alpha_1 u_1$.
for $\ell \geq 2$ do
    $\beta_\ell = \|r_{\ell-1}\|_\mathrm{F}$
    $u_\ell = r_{\ell-1}/\beta_\ell$ (If $\beta_\ell = 0$, pick $u_\ell \perp \mathrm{span}(u_1, \ldots, u_{\ell-1})$ arbitrarily)
    $\alpha_\ell = \langle u_\ell, H[u_\ell] \rangle$
    $r_\ell = H[u_\ell] - \alpha_\ell u_\ell - \beta_\ell u_{\ell-1}$
end for
```

symmetric tridiagonal matrix can be efficiently computed in almost linear time in the dimension of the matrix [51]. Consequently, letting y denote the leading eigenvector of this tridiagonal matrix, we can construct the desired tangent vector in Algorithm 4 as $u^k = \sum_{\ell \geq 1} y_\ell u_\ell$. It is well-known that after $n(r-1)$ iterations, the Lanczos method constructs the leading eigenvector exactly (since order-$n(r-1)$ Krylov subspace spans the entire tangent space). Furthermore, it is also possible to analyze the performance of the Lanczos method with early termination [84]. Building on these ideas, we characterize the quality of the solution returned by Algorithms 4+5 in the following theorem, whose proof can be found in Section 6.6.3.

Theorem 6.17. *Suppose in Algorithm 5, we initialize u_1 uniformly at random over $T_\sigma \mathcal{M}_r$. Let*

$$\ell^* = \left\lceil \left(\frac{1}{2} + 2\sqrt{\frac{\|A\|_1}{\varepsilon}} \right) \log \left(\frac{\left\lceil \frac{675 n \|A\|_1^2}{\varepsilon^2} \right\rceil 1.648 \sqrt{n(r-1)}}{\delta} \right) \right\rceil,$$

and consider that Algorithm 5 is run for $\min(\ell^*, n(r-1))$ iterations at each call from Algorithm 4. Then, after K iterations (defined as in (6.2.19)), Algorithm 4 returns a solution σ^K that satisfies

$$f(\sigma^K) \geq \left(1 - \frac{1}{r-1}\right) \mathrm{SDP}(A) - \frac{n}{2}\varepsilon,$$

with probability at least $1 - \delta$.

6.3 Related Work

Landscape Results

There are numerous papers that analyze the landscape of the solution space of (Non-CVX). In particular, it is known that (CVX) admits an optimal solution of rank r such that $r(r+1)/2 \leq n$ [14, 114]. Using this observation, it has been shown in [40, 41, 76] that when $r \geq \sqrt{2n}$, if σ is a rank deficient second-order stationary point of (Non-CVX), then

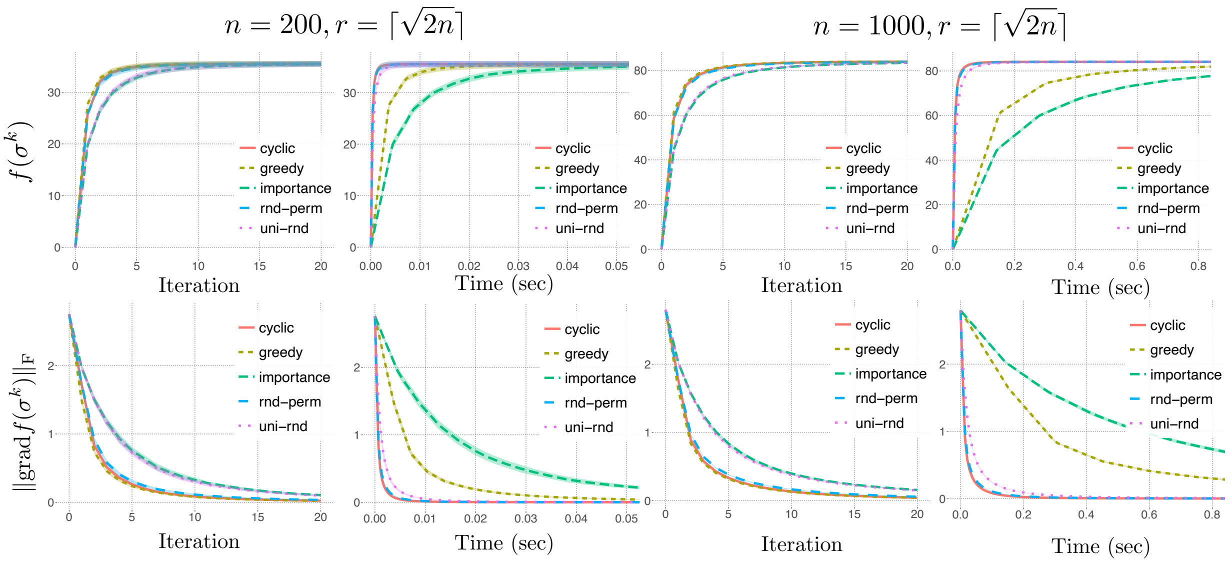

Figure 6-1: Comparisons of different randomization schemes for $n \in \{200, 1000\}$ with $r = \lceil\sqrt{2n}\rceil$.

σ is a global maximum for (Non-CVX) and $X = \sigma\sigma^\top$ is a global maximum for (CVX). In [33], the authors showed that when $r \geq \sqrt{2n}$, for almost all A, every σ that is a first-order stationary point is rank deficient. In [50], the authors showed that the Burer-Monteiro method can solve SDPs to any desired accuracy in polynomial time, in the setting of smooth analysis. For arbitrary rank r, it is shown that all local maxima are within a $n\|A\|_2/\sqrt{r}$ gap from the optimum of (CVX) [103], and any ε-approximate concave point is within a $\mathrm{Rg}(\text{Non-CVX})/(r-1) + n\varepsilon/2$ gap from the optimum of (CVX) [101], where Rg(Non-CVX) is the range of the problem (Non-CVX), i.e., the difference between the maximum and the minimum values of the objective in (Non-CVX).

Algorithms to Solve (Non-CVX)

Javanmard *et al.* [75] showed that when applied to solve (Non-CVX), Riemannian gradient ascent and block-coordinate maximization methods provide excellent numerical results, yet no convergence guarantee is provided. Similar experimental results are also observed in [148] for the block-coordinate maximization algorithm and in [101] for the Riemannian gradient ascent algorithm. Concurrent to this work, in [148], the authors analyzed the convergence of the deterministic block-coordinate maximization algorithm. In particular, they showed that the deterministic block-coordinate maximization algorithm is asymptotically convergent (see [148, Theorem 3.2]) and enjoys a local liner convergence with no explicit rate estimates (see [148, Theorem 3.5]). They also proved that the deterministic block-coordinate maximization approach converges to a local maximum generically under random initialization using the center-stable manifold theorem similar to [90]. These results hold under the assumption that the iterates generated by the algorithm satisfy a certain condition that is seemingly impossible to verify without actually running the algorithm. To alleviate this issue, the authors suggested using a coordinate ascent method with a sufficiently small step size, for which the aforementioned convergence results hold without this precarious assumption. In [30], the authors provided a global sublinear convergence rate for the Riemannian trust-region method for general non-convex problems and these results have been used in [32, 101] for the non-convex Burer-Monteiro approach. Augmented Lagrangian methods have been proposed to solve (Non-CVX) as well [40, 41], however these methods do not benefit from separability of the manifold constraints, and hence are usually slower [33].

Algorithms to Solve (CVX)

There also exist methods that solve (CVX) by exploiting its special structure [7, 66, 82, 136]. In particular, [82] reduces (CVX) to a sequence of approximate eigenpair computations that is efficiently solved using the power method. In [7, 136], matrix multiplicative weights algorithm is used to approximately solve (CVX), and these ideas are extended in [66] using sketching techniques [141]. However, these methods require constructing $X \in \mathrm{Sym}_n$ explicitly, which is prohibitive when n goes beyond a few thousands, whereas the Burer-Monteiro approach we consider here easily scales to very large instances as the low-rank factorization decreases the dimension of the problem from $\mathcal{O}(n^2)$ to $\mathcal{O}(nr)$ with $r \ll n$.

For time complexity comparison between these methods that are based on Lagrangian relaxation and the Burer-Monteiro approach in this thesis, we refer to Corollary 6.14.

Other CD-Based Methods

Coordinate descent methods have been successfully applied to non-convex differentiable optimization problems in several papers [143, 116, 125, 94]. In [143], the authors propose a coordinate gradient descent approach that may be viewed as a hybrid of gradient-projection and coordinate descent to minimize the sum of a smooth function and a convex separable function. They analyze the greedy coordinate selection rule and present local linear convergence, although no rate estimates are provided. [125] considers a similar composite but convex optimization problem and provides explicit rate estimates. These results are then generelized to non-convex problems by [116] and [94]. However, these approaches heavily rely on the Euclidean geometry and cannot handle non-convex constraints, which is the main focus of our thesis.

Computational Complexity Comparison

Per iteration computational cost of the CD method with uniform sampling is $\mathcal{O}(nr)$ as after i_k is chosen uniformly at random, $g_{i_k}^k$ can be computed in $2(n-1)r$ floating point operations. On the other hand, the CD method with importance sampling and greedy coordinate selection requires all $\{\|g_i^k\|\}_{i=1}^n$, which can be naively computed in $\mathcal{O}(n^2 r)$ floating point operations per iteration. Instead, a smarter implementation is to keep both $\{\sigma_i^k\}_{i=1}^n$ and $\{g_i^k\}_{i=1}^n$'s in the memory (only the current iterates, not all the past ones) and update them as presented in Algorithm 3, which can be done in $2(n-1)r$ floating point operations. Therefore, per iteration computational cost of the CD method with all three coordinate selection rules is $\mathcal{O}(nr)$ for dense A (i.e., when no structure is available on A). However, in many SDP applications (such as Max-Cut and graphical model inference), A is induced by a graph and letting d denote the maximum degree of the graph that induces A, the computational cost of the CD method becomes $\mathcal{O}(dr)$. In comparison, per iteration computational complexity of the Riemannian gradient ascent algorithm is $\mathcal{O}(n^2r)$, whereas the Riemannian trust-region algorithm runs a few iterations of a subroutine (e.g., power method) to solve the trust-region subproblem, whose per iteration cost is typically $\mathcal{O}(n^2r)$.

6.4 Numerical Results

In this section, we evaluate the empirical performance of the CD method. In what follows, n is the dimension of the cost matrix $A \in \mathbb{R}^{n\times n}$, and r refers to the rank of factorization. All algorithms are implemented on Matlab and the experiments are run on a computer with 2.9 GHz processor and 16 GB memory. RGD and RTR algorithms are implemented using the Manopt package [31] with the default options and the algorithms are terminated when the maximum allowed time is reached.

In all experiments, the cost matrix is generated as $A = (G+G^\top)/n$, where $G_{ij} \sim \mathsf{N}(0,1)$ for all $i \neq j$, and $G_{ii} = 0$ for all $i \in [n]$. All experiments are based on 50 Monte Carlo runs over the initialization. For each run, the initial iterate $\sigma^0 \in \mathbb{R}^{n\times r}$ is the same for all algorithms and each row of σ^0 is generated uniformly at random on $\mathcal{S}^{r-1}$.

First, we compare various coordinate selection schemes for CD (see Algorithm 3). We compare cyclic order $i = (1, 2, \ldots, n)$, uniform random selection, random permutation order (i follows a cyclic order of a uniformly random permutation of $[n]$), greedy coordinate selection, and selection by importance sampling. Figure 6-1 summarizes the results of our experiments on $n \in \{200, 1000\}$ with $r = \lceil\sqrt{2n}\rceil$. We observe that greedy coordinate selection achieves higher function value after running each algorithm the same number of iterations; yet, due to its high per-iteration cost, cyclic, uniformly random, and random permutation selection rules perform better in terms of overall runtime complexity. Furthermore, randomized rules that do not cycle through all coordinates achieve lower function values after running each algorithm the same number of iterations. This phenomenon is observed for a number of numerical examples in different papers and unfortunately we do not have a good theoretical understanding about this behavior except for a few preliminary

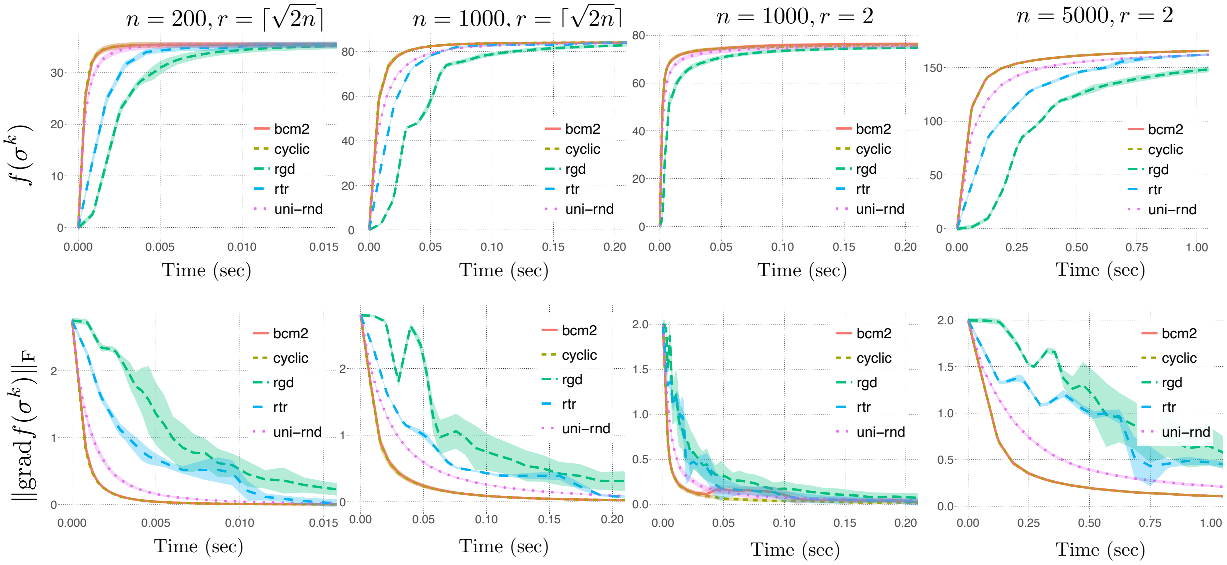

Figure 6-2: Performance of CD and CD2 (Algorithms 3 and 4) compared to other methods. Here, RTR and RGD refer to Riemannian Trust Region and Riemannian Gradient Descent, respectively.

results [72, 73, 88]. It would be an interesting future direction to theoretically understand the slower convergence of randomized coordinate selection rules in practice.

Next, we evaluate the performance of these algorithms for $n \in \{200, 1000, 5000\}$ with $r = 2$ and $r = \sqrt{2n}$. In Figure 6-2, empirical results illustrate the fast convergence of CD and CD2 (see Algorithm 4) compared to RGD and RTR, for both $r = \lceil\sqrt{2n}\rceil$ and $r = 2$. The numerical results indicate our algorithms return a high-quality solution much faster than RGD and RTR regardless of the rank of the factorization is larger or smaller than the Barvinok-Pataki bound.

We next compare the final performance of different methods after convergence. That is, we run all algorithms sufficiently enough until their function value stabilize, and compare the final value obtained. In Figure 6-3, we clearly observe that the final function values obtained through CD2 (Algorithm 4 with Lanczos method) and RTR are larger than those obtained by other algorithms. We also observe that even when the problem size is large (e.g., $n = 20,000$), CD returns a desirable solution within ∼20 seconds, whereas it takes approximately a minute for RTR to return such a solution.

Finally, we consider a random SDP with a planted solution. In particular, we consider a matrix $X \succeq 0$ such that $\mathrm{rank}(X) = r$ and $X \in \mathcal{S}^n$ where $n = \frac{r(r+1)}{2}$. We then generate a MaxCut SDP for which X is an optimal solution, i.e., we find a cost matrix A in the normal cone of X (this requires solving an auxiliary SDP). For each $r \in \{4, 7, 10\}$, we generate 100 random MaxCut SDPs as described above. We perform a Burer-Monteiro factorization to these SDPs for a range of ranks in $[r-4, r+4]$. We solve the resulting non-convex problem using our CD2 algorithm. Figure 6-4 shows the percentage of experiments solved correctly for each value of r. We consider a trial correct if the solution returned by CD2 is sufficiently close to the maximizer of the SDP. Figure 6-4 shows that there is a sharp phase transition at the Barvinok-Pataki bound. Above this bound, the solutions returned by our CD2 algorithm is approximately optimal to (CVX) with high probability.

6.5 Discussion

In this chapter, we studied the Burer-Monteiro approach to solve large-scale SDPs. We considered to solve this non-convex problem using the block-coordinate maximization algorithm that is extremely simple to implement. We proved that for various coordinate selection rules, CD attains a global sublinear convergence rate of $\mathcal{O}(1/\epsilon)$ to guarantee $\mathbb{E}\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 \leq \epsilon$. We also showed the linear convergence of CD around a local max-

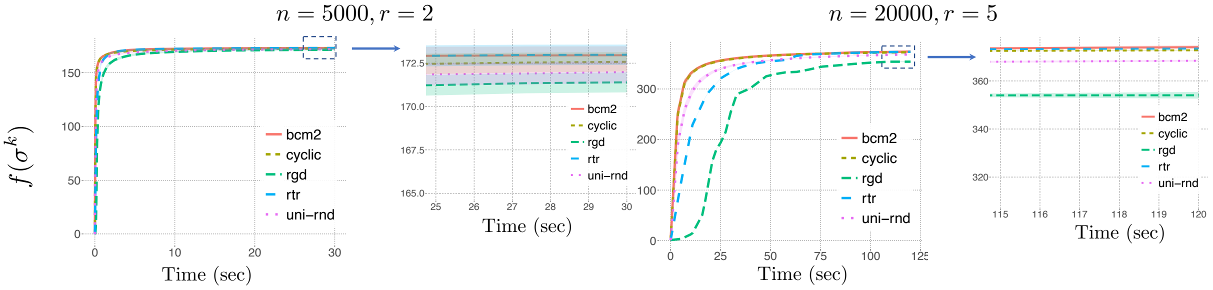

Figure 6-3: Comparing the final performance of different methods after (near) convergence.

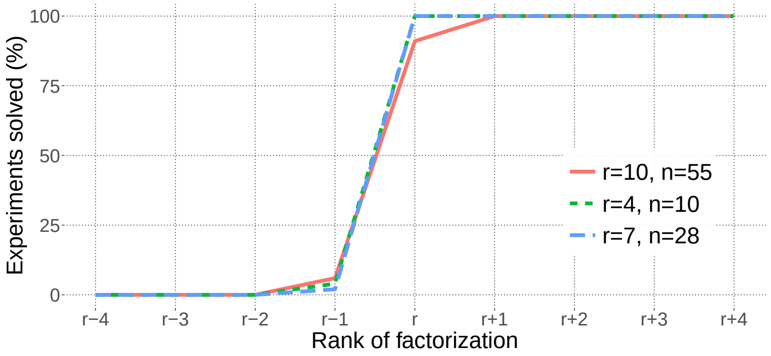

Figure 6-4: Phase transition in recovering the optimal solution of (CVX) by an approximately second-order stationary solution of (Non-CVX).

imum that satisfy the quadratic decay condition. We proved that the quadratic decay condition generically holds for all local maxima provided that $r \geq \sqrt{2n}$. These are the first precise rate estimates for the non-convex Burer-Monteiro approach in the literature to the best of our knowledge. We then introduced a new algorithm called CD2 based on CD and Lanczos methods. We showed that CD2 is guaranteed to return a solution that provides $1 - \mathcal{O}(1/r)$ approximation to the SDP without any assumptions on the cost matrix A, where the r-dependence of this approximation is optimal under the unique games conjecture. We also presented numerical results that verify our theoretical findings and show that CD is faster than the state-of-the-art methods. Even though in this thesis, we only considered SDPs with diagonal constraints, it would be of interest to study the block-coordinate maximization approach in more generic problems.

6.6 Additional Proofs

6.6.1 Proof of Corollary 6.3

Similar to the proof of Theorem 6.2, from Proposition 6.1, we have

$$\begin{aligned} f(\sigma^{k+1}) - f(\sigma^k) &= 2\big(\|g^k_{i_k}\| - \langle \sigma^k_{i_k}, g^k_{i_k}\rangle\big), \\ &= \frac{2\|g^k_{i_k}\|\big(\|g^k_{i_k}\| - \langle \sigma^k_{i_k}, g^k_{i_k}\rangle\big)}{\|g^k_{i_k}\|}, \\ &\geq \frac{\|g^k_{i_k}\|^2 - \langle \sigma^k_{i_k}, g^k_{i_k}\rangle^2}{\|g^k_{i_k}\|}, \end{aligned} \tag{6.6.1}$$

where the inequality follows since $\|g^k_{i_k}\| \geq \langle \sigma^k_{i_k}, g^k_{i_k}\rangle$, for all $\sigma^k_{i_k} \in \mathbb{R}^{n\times r}$. Letting $\mathbb{E}_k$ denote the expectation over i_k given σ^k, we have

$$\mathbb{E}_k f(\sigma^{k+1}) - f(\sigma^k) \geq \sum_{i=1}^n p_i \frac{\|g^k_i\|^2 - \langle \sigma^k_i, g^k_i\rangle^2}{\|g^k_i\|}.$$

In particular, when $p_i = \frac{1}{n}$, for all $i \in [n]$ (i.e., for uniform sampling case), we have

$$\mathbb{E}_k f(\sigma^{k+1}) - f(\sigma^k) \geq \frac{1}{n\|A\|_1} \sum_{i=1}^n \big(\|g^k_i\|^2 - \langle \sigma^k_i, g^k_i\rangle^2\big),$$

since $\|g^k_i\| \leq \|A\|_1$, for all $i \in [n]$ by (6.1.9). Therefore, we have

$$\mathbb{E}_k f(\sigma^{k+1}) - f(\sigma^k) \geq \frac{\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2}{2n\|A\|_1}. \tag{6.6.2}$$

On the other hand, when $p_i = \frac{\|g^k_i\|}{\sum_{j=1}^n \|g^k_j\|}$ (i.e., for importance sampling case), we have

$$\mathbb{E}_k f(\sigma^{k+1}) - f(\sigma^k) \geq \frac{\sum_{i=1}^n \|g^k_i\|^2 - \langle \sigma^k_i, g^k_i\rangle^2}{\sum_{j=1}^n \|g^k_j\|} = \frac{\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2}{2\sum_{j=1}^n \|g^k_j\|}.$$

Letting $\|A\|_{1,1} = \sum_{i,j=1}^{n} |A_{ij}|$ denote the $L_{1,1}$ norm of matrix A, we observe that $\sum_{j=1}^{n} \|g_j^k\| \leq \|A\|_{1,1}$, which in the above inequality yields

$$\mathbb{E}_k f(\sigma^{k+1}) - f(\sigma^k) \geq \frac{\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2}{2\|A\|_{1,1}}. \tag{6.6.3}$$

In order to prove (6.1.11), which corresponds to uniform sampling case, we assume the contrary that $\mathbb{E}\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 > \epsilon$, for all $k \in [K-1]$. Then, using the boundedness of f, we get

$$f^* - f(\sigma^0) \geq \mathbb{E} f(\sigma^K) - f(\sigma^0) = \sum_{k=0}^{K-1} \mathbb{E}\big[f(\sigma^{k+1}) - f(\sigma^k)\big] = \sum_{k=0}^{K-1} \mathbb{E}\big[\mathbb{E}_k f(\sigma^{k+1}) - f(\sigma^k)\big].$$

Using the expected functional ascent of CD in (6.6.2) above, we get

$$f^* - f(\sigma^0) \geq \sum_{k=0}^{K-1} \frac{\mathbb{E}\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2}{2n\|A\|_1} > \frac{K\epsilon}{2n\|A\|_1}, \tag{6.6.4}$$

where the last inequality follows by the assumption. Then, by contradiction, the algorithm returns a solution with $\mathbb{E}\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 \leq \epsilon$, for some $k \in [K-1]$, provided that

$$K \geq \frac{2n\|A\|_1(f^* - f(\sigma^0))}{\epsilon}.$$

The proof of (6.1.12), which corresponds to importance sampling case, can be obtained by using (6.6.3) (instead of (6.6.2)) in (6.6.4), and hence is omitted.

6.6.2 Rest of the Proof of Theorem 6.5

In order to quantify how close σ^0 and σ should be so that this convergence rate holds, we need to derive explicit bounds on the higher order terms in (6.1.19) and (6.1.21), which we do in the following. The Taylor expansion of σ^k around σ yields

$$\begin{aligned}\sigma_i^k &= \sigma_i \cos(\|u_i\|t) + \frac{u_i}{\|u_i\|}\sin(\|u_i\|t),\\ &= \sigma_i\left[\sum_{\ell=0}^{\infty}\frac{(-1)^\ell}{(2\ell)!}(\|u_i\|t)^{2\ell}\right] + \frac{u_i}{\|u_i\|}\left[\sum_{\ell=0}^{\infty}\frac{(-1)^\ell}{(2\ell+1)!}(\|u_i\|t)^{2\ell+1}\right].\end{aligned}$$

Using this expansion, we can compute $f(\sigma^k) = \sum_{i,j=1}^n A_{ij}\langle\sigma_i^k, \sigma_j^k\rangle$. The first three terms in the expansion are already given in (6.1.20) as follows

$$f(\sigma^k) = f(\sigma) + t^2\sum_{i=1}^n\big(\langle u_i, v_i\rangle - \|u_i\|^2\|g_i\|\big) + \beta_f, \tag{6.6.5}$$

where β_f represents the higher order terms. In order to find an upper bound on $|\beta_f|$, we use the Cauchy-Schwarz inequality in the higher order terms in the expansion of $f(\sigma^k)$, which yields the following bound

$$|\beta_f| \le \sum_{i,j=1}^n |A_{ij}|\left(\sum_{\ell=3}^{\infty}\frac{t^\ell}{\ell!}(\|u_i\|+\|u_j\|)^\ell\right).$$

As $\|u\|_{\mathrm{F}} = 1$, we have $\|u_i\| \le 1$ for all $i \in [n]$, which implies

$$|\beta_f| \le \sum_{i,j=1}^n |A_{ij}|\left(\sum_{\ell=3}^{\infty}\frac{t^\ell}{\ell!}2^\ell\right),$$

where we note that t denotes the geodesic distance between σ^k and $[\bar{\sigma}]$ as highlighted in (6.1.17). Assuming that $t \le 1$, we obtain the following upper bound

$$|\beta_f| \le t^3 n\|A\|_1\left(\sum_{\ell=3}^{\infty}\frac{2^\ell}{\ell!}\right).$$

Using the inequality $\sum_{\ell=3}^{\infty}\frac{2^\ell}{\ell!} = e^2 - 5 \le 5/2$ above, we get

$$|\beta_f| \le \frac{5n\|A\|_1 t^3}{2}.$$

Plugging this value back in (6.6.5), we obtain

$$f(\sigma^k) \leq f(\sigma) + t^2 \sum_{i=1}^{n} \big(\langle u_i, v_i \rangle - \|u_i\|^2 \|g_i\|\big) + \frac{5n\|A\|_1 t^3}{2}. \tag{6.6.6}$$

Considering the same expansion for $\|\text{grad} f(\sigma^k)\|_{\mathrm{F}}^2 = 2\sum_{i=1}^{n}(\|g_i^k\|^2 - \langle \sigma_i^k, g_i^k \rangle^2)$, we get the following (see (6.1.19)):

$$\|\text{grad} f(\sigma^k)\|_{\mathrm{F}}^2 = 2t^2 \sum_{i=1}^{n} \left(\|u_i\|\|g_i\| - \langle \frac{u_i}{\|u_i\|}, v_i \rangle \right)^2 + \beta_g, \tag{6.6.7}$$

where β_g represents the higher order terms. Upper bounding each higher order terms using the Cauchy-Schwarz inequality as follows, we obtain

$$\begin{aligned} |\beta_g| \leq 2 \sum_{i=1}^{n} \Bigg[\sum_{j,m=1}^{n} |A_{ij}||A_{im}| \left(\sum_{\ell=3}^{\infty} \frac{t^\ell}{\ell!} (\|u_j\| + \|u_m\|)^\ell \right) \\ + \sum_{j,m=1}^{n} |A_{ij}||A_{im}| \left(\sum_{\substack{\ell,s=0 \\ \ell+s\geq 3}}^{\infty} \frac{t^{\ell+s}}{\ell! s!} (\|u_i\| + \|u_j\|)^{\ell+s} \right) \Bigg]. \end{aligned}$$

Using the fact that $\|u_i\| \leq 1$ for all $i \in [n]$, we get the following upper bound

$$|\beta_g| \leq 2 \sum_{i=1}^{n} \left[\sum_{j,m=1}^{n} |A_{ij}||A_{im}| \left(\sum_{\ell=3}^{\infty} \frac{t^\ell}{\ell!} 2^\ell \right) + \sum_{j,m=1}^{n} |A_{ij}||A_{im}| \left(\sum_{\substack{\ell,s=0 \\ \ell+s\geq 3}}^{\infty} \frac{t^{\ell+s}}{\ell! s!} 2^{\ell+s} \right) \right].$$

Using the upper bound $\sum_{j,m=1}^{n} |A_{ij}||A_{im}| \leq \|A\|_1^2$ above, we obtain

$$|\beta_g| \leq 2\|A\|_1^2 \sum_{i=1}^{n} \left[\sum_{\ell=3}^{\infty} \frac{t^\ell}{\ell!} 2^\ell + \sum_{\substack{\ell,s=0 \\ \ell+s\geq 3}}^{\infty} \frac{t^{\ell+s}}{\ell! s!} 2^{\ell+s} \right].$$

Introducing a change of variables in the last sum, we get

$$\begin{aligned}|\beta_g| &\leq 2\|A\|_1^2 \sum_{i=1}^{n}\left[\sum_{\ell=3}^{\infty}\frac{t^\ell}{\ell!}2^\ell + \sum_{\ell=3}^{\infty}\frac{t^\ell}{\ell!}2^\ell\left(\sum_{s=0}^{\ell}\frac{\ell!}{s!(\ell-s)!}\right)\right],\\ &= 2\|A\|_1^2\sum_{i=1}^{n}\left[\sum_{\ell=3}^{\infty}\frac{t^\ell}{\ell!}\big(2^\ell+4^\ell\big)\right].\end{aligned}$$

Assuming that $t \leq 1$, we obtain the following upper bound

$$|\beta_g| \leq 2\|A\|_1^2 t^3 \sum_{i=1}^{n}\left[\sum_{\ell=3}^{\infty}\frac{1}{\ell!}\big(2^\ell+4^\ell\big)\right].$$

Using the inequality $\sum_{\ell=3}^{\infty}\frac{2^\ell+4^\ell}{\ell!} = e^2+e^4-18 \leq 44$ above, we get

$$|\beta_g| \leq 88n\|A\|_1^2 t^3.$$

Plugging this value back in (6.6.7), we obtain

$$\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 \geq 2t^2\sum_{i=1}^{n}\left(\|u_i\|\|g_i\| - \langle\frac{u_i}{\|u_i\|}, v_i\rangle\right)^2 - 88n\|A\|_1^2 t^3. \tag{6.6.8}$$

Using the same bounding technique as in (6.1.22), we get

$$\begin{aligned}\|\mathrm{grad} f(\sigma^k)\|_{\mathrm{F}}^2 &\geq \frac{\mu}{n}\left(f(\bar{\sigma}) - f(\sigma^k) - \frac{5n\|A\|_1 t^3}{2}\right) - 88n\|A\|_1^2 t^3,\\ &= \frac{\mu}{n}\big(f(\bar{\sigma}) - f(\sigma^k)\big) - t^3\|A\|_1(3\mu + 88n\|A\|_1).\end{aligned}$$

Therefore, in order for (6.1.23) to hold, we need

$$t^3\|A\|_1(3\mu + 88n\|A\|_1) \leq \frac{\mu}{2n}\big(f(\bar{\sigma}) - f(\sigma^k)\big),$$

which can be equivalently rewritten as follows

$$t^3 \leq \frac{\mu(f(\bar{\sigma}) - f(\sigma^k))}{2n\|A\|_1(3\mu + 88n\|A\|_1)}.$$

As $f(\sigma^k)$ is a monotonically non-decreasing sequence, then as soon as σ^0 is sufficiently close to $[\bar{\sigma}]$ in the sense that

$$\operatorname{dist}(\sigma^0, [\bar{\sigma}]) \leq \left(\frac{\mu(f(\bar{\sigma}) - f(\sigma^k))}{2n\|A\|_1(3\mu + 88n\|A\|_1)}\right)^{1/3},$$

then the linear convergence rate presented in (6.1.24) holds.

6.6.3 Proof of Theorem 6.17

Before presenting the proof of Theorem 6.17, we first introduce the following theorem that characterizes the convergence rate of the Lanczos method with random initialization.

Theorem 6.18 ([84, Theorem 4.2])**.** *Let $A \in \mathbb{R}^{n\times n}$ be a positive semidefinite matrix, $b \in \mathbb{R}^n$ be an arbitrary vector and $\lambda_L^\ell(A, b)$ denote the output of the Lanczos algorithm after ℓ iterations when applied to find the leading eigenvalue of A (denoted by $\lambda_1(A)$) with initialization b. In particular,*

$$\lambda_L^\ell(A, b) = \max\left\{\frac{\langle x, Ax\rangle}{\langle x, x\rangle} : 0 \neq x \in \operatorname{span}(b, \dots, A^{\ell-1}b)\right\}.$$

Assume that b is uniformly distributed over the set $\{b \in \mathbb{R}^n : \|b\| = 1\}$ and let $\epsilon \in [0, 1)$. Then, the probability that the Lanczos algorithm does not return an ϵ-approximation to the leading eigenvalue of A exponentially decreases as follows

$$\mathbb{P}\big(\lambda_L^\ell(A, b) < (1-\epsilon)\lambda_1(A)\big) \begin{cases} \leq 1.648\sqrt{n}e^{-\sqrt{\epsilon}(2\ell-1)}, & \text{if } 0 < \ell < n(r-1), \\ = 0, & \text{if } \ell \geq n(r-1). \end{cases}$$

Using this result, Theorem 6.17 is proven as follows. Since the tangent space $T_\sigma\mathcal{M}_r$

has dimension $n(r-1)$, then we can define a symmetric matrix (where we drop the notational dependency on σ for simplicity) $H \in \mathbb{R}^{n(r-1)\times n(r-1)}$ that represents the linear operator $\mathrm{Hess}f(\sigma)$ in the basis $\{u^1, \ldots, u^{n(r-1)}\}$ such that $\mathrm{span}(u^1, \ldots, u^{n(r-1)}) = T_\sigma\mathcal{M}_r$. In particular, letting $H_{ij} = \langle u^i, \mathrm{Hess}f(\sigma)[u^j]\rangle$ yields the desired matrix H and the Lanczos algorithm is run to find the leading eigenvalue of this matrix. Here, it is important to note that H is not a psd matrix, so it is required to shift H with a large enough multiple of the identity matrix so that the resulting matrix is guaranteed to be positive semidefinite. In particular, by inspecting the definition of $\mathrm{Hess}f(\sigma)$ in (6.1.2), it is easy to observe that $\|\mathrm{Hess}f(\sigma)\|_{\mathrm{op}} \leq 4\|A\|_1$. Therefore, it is sufficient to run the Lanczos algorithm to find the leading eigenvalue of $\widetilde{H} = H + 4\|A\|_1 I$, where I denotes the appropriate sized identity matrix. On the other hand, we initialize the Lanczos algorithm with a random vector u of unit norm (i.e., $\|u\|_{\mathrm{F}} = 1$) in the tangent space $T_\sigma\mathcal{M}_r$. Notice that u can equivalently be represented as a vector $b \in \mathbb{R}^{n(r-1)}$ in the basis $\{u^1, \ldots, u^{n(r-1)}\}$ as $u = \sum_{i=1}^{n(r-1)} b_i u^i$ such that $\|b\| = 1$. Then, by Theorem 6.18, we have

$$\mathbb{P}\Big(\lambda_L^\ell(\widetilde{H}, b) < (1-\epsilon)\lambda_1(\widetilde{H})\Big) \leq 1.648\sqrt{n(r-1)}e^{-\sqrt{\epsilon}(2\ell-1)}.$$

Letting $\lambda_1(H)$ denote the leading eigenvalue of H, we run the Lanczos algorithm to obtain a vector b^* such that $\|b^*\| = 1$ and $\langle b^*, Hb^*\rangle \geq \lambda_1(H)/2$. Thus, we want the following to be small:

$$\mathbb{P}\Big(\lambda_L^\ell(\widetilde{H}, b) < 4\|A\|_1 + \lambda_1(H)/2\Big). \tag{6.6.9}$$

Setting $\epsilon^* = \frac{\lambda_1(H)}{16\|A\|_1}$, we can observe that

$$\begin{aligned}(1-\epsilon^*)\lambda_1(\widetilde{H}) &= \Big(1 - \frac{\lambda_1(H)}{16\|A\|_1}\Big)(4\|A\|_1 + \lambda_1(H)),\\ &= 4\|A\|_1 + \frac{3\lambda_1(H)}{4} - \frac{(\lambda_1(H))^2}{16\|A\|_1},\\ &\geq 4\|A\|_1 + \frac{\lambda_1(H)}{2},\end{aligned}$$

where the inequality follows since $\lambda_1(H) \leq 4\|A\|_1$. Consequently, we have

$$\mathbb{P}\Big(\lambda_L^\ell(\widetilde{H}, b) < 4\|A\|_1 + \lambda_1(H)/2\Big) \leq \mathbb{P}\Big(\lambda_L^\ell(\widetilde{H}, b) < (1-\epsilon^*)\lambda_1(\widetilde{H})\Big)$$
$$\leq 1.648\sqrt{n(r-1)}e^{-\sqrt{\epsilon^*}(2\ell-1)}.$$

By Theorem 6.13, we know that the Lanczos method is called at most $\lceil 675n\|A\|_1^2/\varepsilon^2 \rceil$ times to search for an ε-approximate concave point and for any non-desired solution we have $\lambda_1(H) \geq \varepsilon$ by the definition of ε-approximate concave point. Then, by using a union bound over all calls to the Lanczos method, we conclude that when the Lanczos method is run for ℓ iterations, we have the following guarantee

$$\mathbb{P}(\text{Algorithm 4+5 fails to return an } \varepsilon\text{-approximate concave point})$$
$$\leq \left\lceil \frac{675n\|A\|_1^2}{\varepsilon^2} \right\rceil 1.648\sqrt{n(r-1)}e^{-\sqrt{\frac{\varepsilon}{16\|A\|_1}}(2\ell-1)}.$$

In order to set this probability to some $\delta \in (0,1)$, we let

$$\ell^* = \left\lceil \left(\frac{1}{2} + 2\sqrt{\frac{\|A\|_1}{\varepsilon}}\right) \log\left(\frac{\left\lceil \frac{675n\|A\|_1^2}{\varepsilon^2} \right\rceil 1.648\sqrt{n(r-1)}}{\delta}\right) \right\rceil$$
$$= \widetilde{\mathcal{O}}\left(\sqrt{\frac{\|A\|_1}{\varepsilon}} \log\left(\frac{n\sqrt{n(r-1)}}{\delta}\right)\right),$$

where tilde is used to hide poly-logarithmic factors in $\|A\|_1/\varepsilon$. Since the Lanczos algorithm is guaranteed to return the leading eigenvalue with probability 1 in at most $n(r-1)$ iterations, then running each Lanczos subroutine for $\min(\ell^*, n(r-1))$ iterations, it is guaranteed that Algorithm 4+5 returns an ε-approximate concave point with probability at least $1-\delta$.

Part II

Mirror Descent Method

Chapter 7

An Overview

In this chapter, we present a comprehensive review of the mirror descent method. In Section 7.1, we describe the mirror descent dynamics proposed by Nemirovski and Yudin [106]. In Section 7.2, we discuss sufficient conditions under which mirror descent dynamics are well-defined. In Section 7.3, we present a generalization of Bregman divergence that will be used to analyze the mirror descent method throughout this part. We then study the convergence of mirror descent in Section 7.4. The contents of this chapter mostly follow from the existing studies in the literature. Yet, our analyses do not rely on certain assumptions that have been extensively used in the literature, and hence provide a more general notion of convergence for the mirror descent method. We rigorously state the distinctions in the sequel. We conclude the chapter in Section 7.5 by motivating Chapters 8 & 9, and providing an outline.

7.1 Mirror Descent Dynamics

Consider a constrained convex optimization problem of the form

$$\min_{x \in \mathcal{X}} f(x), \tag{7.1.1}$$

where $\mathcal{X}$ is a nonempty closed convex subset of $\mathcal{E}$ and f is a continuously differentiable convex function. Suppose $(\mathcal{E}, \|\cdot\|)$ is a Banach space, where the norm we use to measure distance does not necessarily derive from an inner product. In this case, many optimization methods are not even applicable since $x \in \mathcal{E}$, whereas $\nabla f(x) \in \mathcal{E}^*$. We do not face this problem for optimization problems in Hilbert spaces since $\mathcal{E}^*$ is isometric to $\mathcal{E}$ by Riesz representation theorem. Nemirovski and Yudin [106] proposed to handle this issue by defining a gradient flow in the dual space $\mathcal{E}^*$ and then mapping the corresponding trajectory to the primal space $\mathcal{E}$. The resulting algorithm is called *mirror descent*.

The original approach of Nemirovski and Yudin [106] constructs the mirror descent method by functional analytic arguments between primal and dual spaces. In particular suppose on $\mathcal{E}^*$, there exists a continuously differentiable function h. Then for any $y \in \mathcal{E}^*$, $\nabla h(y)$ is an element of $\mathcal{E}$. Using this observation, the authors define the following differential equation in the dual space:

$$\dot{y}(t) = -\nabla f(\nabla h(y(t))), \tag{7.1.2a}$$

$$y(0) = y^0 \in \operatorname{dom}(\nabla \Phi^*), \tag{7.1.2b}$$

where ∇h is called *mirror map* from the dual space to the primal space. Suppose $y(t)$ is a solution to (7.1.2). Then, the image of this trajectory in the primal space is given by

$$x(t) = \nabla h(y(t)).$$

Indeed, h needs to satisfy certain conditions for $f(x(t))$ to converge to the minimum of (7.1.1) and in the following section, we present sufficient conditions on h, under which this holds true. However, before discussing these conditions, we first introduce a change of notation and let $h = \Phi^*$, where Φ^* is the convex conjugate of Φ, where Φ is a function defined on $\mathcal{E}$. The main motivation behind this change of notation is that in practice, we are given the problem (7.1.1) in the primal space and we are usually interested in constructing the mirror map according to the geometry of this problem. Therefore, we usually construct

h through a function Φ defined on the primal space, called the *distance generating function*. With this change of variables, we equivalently write the continuous dynamics in (7.1.2) as follows:

$$\dot{y}(t) = -\nabla f(\nabla \Phi^*(y(t))), \tag{7.1.3a}$$

$$y(0) = y^0 \in \operatorname{dom}(\nabla \Phi^*), \tag{7.1.3b}$$

where the corresponding primal trajectory is given by $x(t) = \nabla \Phi^*(y(t))$. In the following section, we present conditions on the distance generating function Φ (and also on Φ^*) such that the continuous dynamics (7.1.3) is well-defined.

7.2 Sufficient Conditions for Well-Defined Dynamics

We first recall that f is assumed to be continuously differentiable, so ∇f is well-defined. We require Φ^* to be continuously differentiable on its domain for the dynamics (7.1.3) to be well-defined. Next, we observe that without an additional assumption on f, we cannot make any conclusions on $\operatorname{rge} \nabla f$. Therefore, unless we have additional knowledge on f, the differential equation (7.1.3) should be well-defined on the entire dual space $\mathcal{E}^*$. As y is an input of $\nabla \Phi^*$ in (7.1.3), we also need $\operatorname{dom} \nabla \Phi^* = \mathcal{E}^*$. Finally, we need to have $\operatorname{rge} \nabla \Phi^* \subseteq \mathcal{X}$ so that ∇f can be evaluated at $\nabla \Phi^*(y)$ for any $y \in \mathcal{E}^*$. To recap, in order to have a well-defined dynamics (7.1.3), Φ^* needs to satisfy the following conditions:

1. Φ^* is differentiable on its domain.

2. $\operatorname{dom} \nabla \Phi^* = \mathcal{E}^*$.

3. $\operatorname{rge} \nabla \Phi^* \subseteq \mathcal{X}$.

Below we relate these conditions on Φ^* to conditions on a closed proper convex distance generating function Φ:

- *Condition 3:* Let us begin our discussion from the third condition and disregard the differentiability of Φ^* for the time being. As Φ is closed proper convex, subdifferential $\partial\Phi^*$ is well-defined, and the range of $\partial\Phi^*$ satisfies [128, p. 227]: $\operatorname{rge}\partial\Phi^* \subseteq \operatorname{dom}\Phi$. Therefore, as long as $\operatorname{dom}\Phi = \mathcal{X}$, subdifferential of Φ^* maps to $\mathcal{X}$.

- *Condition 2:* Turning our attention back to the second condition and again disregarding the differentiability of Φ^*, we can observe (see e.g., [128, p. 227]) that $\operatorname{ri}(\operatorname{dom}\Phi^*) \subseteq \operatorname{dom}\partial\Phi^* \subseteq \operatorname{dom}\Phi^*$. Therefore, $\operatorname{dom}\partial\Phi^* = \mathcal{E}^*$ if and only if $\operatorname{dom}\Phi^* = \mathcal{E}^*$. Furthermore, $\operatorname{dom}\Phi^* = \mathcal{E}^*$ if and only if Φ is co-finite (see e.g. [128, Corollary 13.3.1]), i.e., $\operatorname{epi}\Phi$ contains no non-vertical half-lines. Since $\mathcal{E}^*$ is finite-dimensional, the latter condition holds if and only if [16, Theorems 3.3 & 3.4] Φ is supercoercive, i.e.,
$$\lim_{\|x\|\to+\infty} \frac{\Phi(x)}{\|x\|} = +\infty.$$

- *Condition 1:* Finally, we consider the first condition: $\partial\Phi^*$ is single-valued, i.e., Φ^* is differentiable, on its domain if and only if Φ^* is essentially smooth [128, Theorem 26.1], or equivalently if and only if Φ is essentially strictly convex [128, Theorem 26.3].

More compactly, equivalent conditions on Φ to have well-defined dynamics (7.1.3) are given below:

Proposition 7.1. *Let $f : \mathcal{X} \to \mathbb{R}$ be a continuously differentiable convex function. Then, the mirror descent dynamics* (7.1.3) *is well-defined for any closed proper convex function Φ that satisfies the following conditions:*

1. Φ is essentially strictly convex.

2. Φ is supercoercive.

3. $\operatorname{dom}\Phi = \mathcal{X}$.

These three conditions are satisfied when Φ is strongly convex with $\operatorname{dom}\Phi = \mathcal{X}$ or when $\mathcal{X}$ is bounded and Φ is strictly convex with $\operatorname{dom}\Phi = \mathcal{X}$. As strong/strict convexity is easier to interpret, it is extensively used in the literature to analyze the convergence of the mirror descent methods. It is important to note that the original description of the mirror descent method by Nemirovski and Yudin [106] also relies on supercoercivity and differentiability assumptions, and the reasoning behind these assumptions are also discussed in [83]. Before discussing the convergence of the mirror descent dynamics, we first observe that Φ need not be differentiable to have a well-defined dynamics according to our discussion above, whereas in the standard literature, the analysis of mirror descent is carried out using Bregman divergences which requires Φ to be continuously differentiable. Therefore, to analyze the convergence of mirror descent dynamics, we next describe a generalized Bregman divergence that can be associated with a non-differentiable distance generating function.

7.3 Generalized Bregman Divergence

Let Φ be a closed proper convex function, and recall the definition of the Bregman divergence associated with Φ when Φ is continuously differentiable:

$$D_{\Phi}(x,u) = \Phi(x) - \Phi(u) - \langle \nabla\Phi(u), x-u\rangle. \tag{7.3.1}$$

An important consequence of the convexity of Φ is that $D_{\Phi}(x,u) \geq 0$ for all $x, u \in \mathcal{E}$, which enables us to use Bregman divergence as a Lyapunov function in the analysis of mirror descent methods. Unfortunately, this definition is not valid when Φ is not continuously differentiable as $\nabla\Phi(u)$ is not well-defined. We can address this issue by replacing the gradient $\nabla\Phi(u)$ with a subgradient $y \in \partial\Phi(u)$, which still yields a nonnegative function since

$$\Phi(x) - \Phi(u) - \langle y, x-u\rangle \geq 0, \quad \forall x, u \in \mathcal{E} \text{ and } \forall y \in \partial\Phi(u). \tag{7.3.2}$$

We cannot however define (7.3.2) as the divergence between two primal points x and u since the choice of the subgradient $y \in \partial\Phi(u)$ would change the value of this divergence.

As an example, consider the function $\Phi(x) = |x| + x^2$ in Figure 7-1. When we try to define the Bregman divergence between points x and u according to (7.3.2), the choice of the subgradient causes an ambiguity in the value of the divergence. We can get rid of this ambiguity by noticing that while a point u does not define a unique tangent plane to Φ, a slope y does. There is always at most one plane with slope y tangent to Φ, and when it exists it is given by the equation (as a function of x):

$$\Phi(u) + \langle y, x - u \rangle, \quad \text{where } u \in \partial\Phi^*(y).$$

The same ambiguity seems to appear here again: $\partial\Phi^*(y)$ can be a set-valued map, and consequently there is not a unique choice of primal variable $u \in \partial\Phi^*(y)$. However, this ambiguity does not matter since when $\partial\Phi^*(y)$ is a set-valued map, each primal variable $u \in \partial\Phi^*(y)$ lies on a linear segment of Φ due to the convexity of Φ. Thus, both the tangent plane and the value of the divergence is the same for any such primal variables.

The above discussion motives to define the divergence as a function of a primal point and a slope of a tangent (a dual point). In particular, we define the generalized Bregman divergence between a primal point $x \in \mathcal{E}$ and a dual point $y \in \mathcal{E}^*$ as follows:

$$B_\Phi(x, y) = \Phi(x) + \Phi^*(y) - \langle x, y \rangle. \tag{7.3.3}$$

It is easy to observe that (7.3.3) generalizes the original Bregman divergence because when Φ is continuously differentiable, we have $y = \nabla\Phi(u)$ and $\Phi^*(y)$ is given by its convex conjugate as $\Phi^*(y) = \langle y, u \rangle - \Phi(u)$, which yields $B_\Phi(x, y) = D_\Phi(x, u)$. An advantage of the definition (7.3.3) is that the generalized Bregman divergence is symmetric in the sense that

$$B_\Phi(x, y) = B_{\Phi^*}(y, x). \tag{7.3.4}$$

We can also observe that $B_\Phi(x, y)$ is convex in both x and y separately (although is not necessarily convex jointly in x and y). We discuss further properties of the generalized

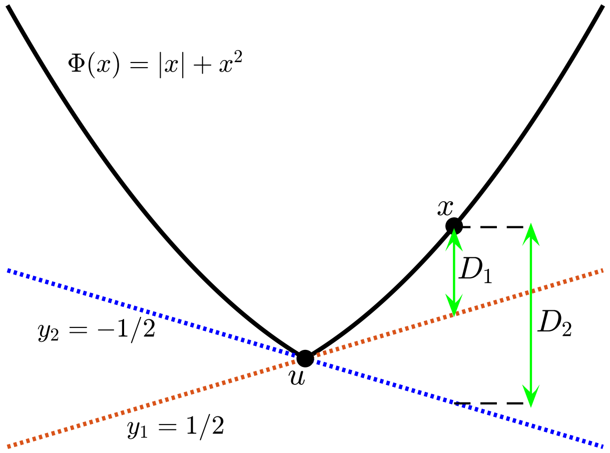

Figure 7-1: Generalized Bregman divergences when Φ is non-differentiable. The value of the divergence depends on the choice of subgradient.

Bregman divergence (7.3.3) in Section 7.3.2, and in the following section we present related works in the literature that discuss different generalizations of Bregman divergence.

7.3.1 Related Literature

In the original work of Bregman [34], Bregman divergence is not defined with respect to a distance generating function, but is assumed to have certain distance-like properties. In [34, Equation 1.4], Bregman showed that a strictly convex continuously differentiable distance generating function defines a divergence that satisfies the required conditions. In more recent literature, Bregman divergence is often associated with a Legendre distance generating function, see e.g., [53, Definition 2.1], which relaxes certain boundary conditions, such as closedness and differentiability on the boundary. This allows us to cover important classes of Bregman divergences, e.g., negative entropy and Burg's entropy, see [15, Remark

4.2]. An extension of these ideas to non-differentiable functions is presented by [80], where the author considers any closed proper convex function and define two different Bregman divergences according to minimal and maximal subgradients, which we explain next. For a given set $\mathcal{X}$, let $\iota_{\mathcal{X}}$ denote the indicator function of $\mathcal{X}$, i.e., $\iota_{\mathcal{X}}(x) = 0$ if $x \in \mathcal{X}$, and $+\infty$ otherwise. The support function of $\iota_{\mathcal{X}}$ is given by $\iota^*_{\mathcal{X}}(\cdot) = \sup_{x \in \mathcal{X}} \langle \cdot, x \rangle$. By [128, Theorem 23.2], any directional derivative of Φ at x is lower bounded by the support function of its subgradient, i.e., $\Phi'(x; d) \geq \iota^*_{\partial\Phi(x)}(d)$, where $\Phi'(x; d)$ is the derivative of Φ at x in direction d. Using this description, Kiwiel [80] proposed the following generalizations of Bregman divergence:

$$\begin{aligned} D^\flat_\Phi(x, u) &= \Phi(x) - \Phi(u) - \iota^*_{\partial\Phi(u)}(x - u), \\ D^\sharp_\Phi(x, u) &= \Phi(x) - \Phi(u) - \iota^*_{\partial\Phi(u)}(u - x). \end{aligned}$$

Similar generalized Bregman divergence definitions using directional derivatives are also used in [140] to analyze agglomerative clustering with Bregman divergences. In [81], Kiwiel proposed to specify the Bregman divergence with two points and a subgradient at the second point as follows

$$D^y_\Phi(x, u) = \Phi(x) - \Phi(u) - \langle y, x - u \rangle, \quad \text{where } y \in \partial\Phi(u). \tag{7.3.5}$$

Using this definition, the author then described a proximal point method that is closely related to mirror descent as we discuss in Chapter 8. This algorithm is reinvented in [112], called Bregman iterative algorithm, using the same generalized Bregman divergence definition. It is easy to observe that (7.3.5) and our definition (7.3.3) are equivalent as $B_\Phi(x, y) = D^y_\Phi(x, u)$ for any $u \in \partial\Phi^*(y)$. Our description above closely follows [68], where the author uses generalized Bregman divergence to derive regret bounds for maximum a posteriori estimation methods under different priors.

7.3.2 Properties of Generalized Bregman Divergence

We next present a few fundamental properties of generalized Bregman divergences that will be useful in the sequel. First, we establish three-points identity, which is a natural generalization of a quadratic identity valid for the Euclidean norm. Generalization of this identity to Bregman divergences (7.3.1) has been presented in [49, Lemma 3.1]. Its extension to the generalized Bregman divergence defined in (7.3.5) is presented in [81, Lemma 4.1], which relaxes the strict convexity and differentiability assumptions on the distance generating function. Below, we establish it for the generalized Bregman divergence in (7.3.3).

Lemma 7.2 (three-points identity)**.** *Let $\Phi : \mathcal{X} \to \mathcal{E}^*$ be a closed proper convex distance generating function. Then for any three points $x_1, x_3 \in \mathcal{X}$ and $y_2 \in \operatorname{dom} \partial\Phi^*$, the following identity holds:*

$$B_\Phi(x_3, y_1) + B_\Phi(x_1, y_2) - B_\Phi(x_3, y_2) = \langle x_3 - x_1, y_2 - y_1 \rangle, \quad \forall y_1 \in \partial\Phi(x_1).$$

Proof By the definition of B_Φ, we have

$$B_\Phi(x_3, y_1) = \Phi(x_3) + \Phi^*(y_1) - \langle x_3, y_1 \rangle, \tag{7.3.6}$$

$$B_\Phi(x_1, y_2) = \Phi(x_1) + \Phi^*(y_2) - \langle x_1, y_2 \rangle, \tag{7.3.7}$$

$$B_\Phi(x_3, y_2) = \Phi(x_3) + \Phi^*(y_2) - \langle x_3, y_2 \rangle. \tag{7.3.8}$$

Subtracting (7.3.8) from the sum of (7.3.6) and (7.3.7), we get

$$B_\Phi(x_3, y_1) + B_\Phi(x_1, y_2) - B_\Phi(x_3, y_2) = \Phi(x_1) + \Phi^*(y_1) + \langle x_3, y_2 \rangle - \langle x_3, y_1 \rangle - \langle x_1, y_2 \rangle. \tag{7.3.9}$$

By the conjugate subgradient theorem, we have $\Phi(x_1) + \Phi^*(y_1) = \langle x_1, y_1 \rangle$. Plugging in this equation above and reorganizing terms, we obtain the desired result. □

We next show a standard result in convex optimization: If a distance generating function is strongly convex, then so is the corresponding generalized Bregman divergence.

Lemma 7.3 (strong convexity)**.** *Let $\Phi : \mathcal{X} \to \mathcal{E}^*$ be a μ-strongly convex distance generating function with respect to the norm $\|\cdot\|$. Then, B_Φ is μ-strongly convex as well, i.e.,*

$$B_\Phi(x_1, y_2) \geq \frac{\mu}{2}\|x_1 - x_2\|^2, \quad \forall\, x_1, x_2 \in \mathcal{X} \text{ and } \forall\, y_2 \in \partial\Phi(x_2). \tag{7.3.10}$$

Proof By the definition of B_Φ, we have

$$B_\Phi(x_1, y_2) = \Phi(x_1) + \Phi^*(y_2) - \langle y_2, x_1 \rangle.$$

By the conjugate subgradient theorem, we have $\Phi^*(y_2) = \langle y_2, x_2 \rangle - \Phi(x_2)$, which implies

$$B_\Phi(x_1, y_2) = \Phi(x_1) - \Phi(x_2) - \langle y_2, x_1 - x_2 \rangle.$$

As Φ is μ-strongly convex, the right-hand side in the above equality is lower bounded by $\mu\|x_1 - x_2\|^2/2$, which concludes the proof. □

7.4 Convergence of Mirror Descent Dynamics

In this section, we study the convergence of mirror descent dynamics. We assume $f : \mathcal{E} \to \mathbb{R}$ is a continuously differentiable convex function and $\Phi : \mathcal{X} \to \mathcal{E}^*$ satisfies the conditions in Proposition 7.1 so that dynamics in (7.1.3) are well-defined. Let $y : [0, +\infty) \to \mathcal{E}^*$ be a solution of the mirror descent dynamics (7.1.3) and define the corresponding primal trajectory $x(t) = \nabla\Phi^*(y(t))$. Let $\mathcal{X}^*$ be the set of minimizers of (7.1.1), i.e.,

$$\mathcal{X}^* = \arg\min_{x \in \mathcal{X}} f(x),$$

and let $\mathcal{Y}^*$ be the preimage of $\mathcal{X}^*$ under the map $\nabla\Phi^*$:

$$\mathcal{Y}^* = \{y \in \mathcal{E}^* : \nabla\Phi^*(y^*) \in \mathcal{X}^*\}.$$

If the function values $f(x(t))$ converge to $\min_{x\in\mathcal{X}} f(x)$ as $t \to \infty$, any accumulation point of $x(t)$ is contained in $\mathcal{X}^*$, i.e., x converges *weakly* to $\mathcal{X}^*$, which implies y converges *weakly* to $\mathcal{Y}^*$. Although the limit points of $y(t)$ are contained in $\mathcal{Y}^*$, $y(t)$ does not necessarily have a limit as $t \to \infty$. If $\lim_{t\to\infty} y(t)$ exists and is an element of $\mathcal{Y}^*$, we say that $y(t)$ converges *strongly* to a point in $\mathcal{Y}^*$. In Section 7.4.1, we study weak convergence of y, i.e., function value convergence of mirror descent dynamics. In Section 7.4.2, we study strong convergence of x and y, i.e., pointwise convergence of mirror descent dynamics.

7.4.1 Weak Convergence

In this section, we study weak convergence of mirror descent dynamics. We first establish asymptotic convergence, then present an ergodic convergence rate and finally show that a non-ergodic convergence rate can be obtained when Φ^* is twice-differentiable. We note that these results are standard in the literature, although they are often given under the assumption that the distance generating function is continuously differentiable, see e.g., [83, 149]. Below, we present a similar analysis for non-differentiable distance generating functions using generalized Bregman divergence (7.3.3). We also show that this generalized Bregman divergence is equivalent to the potential function Nemirovski and Yudin [106] used to analyze the asymptotic convergence of mirror descent dynamics.

First, we will observe that

$$V(y(t)) = B_{\Phi^*}(y(t), x^*) + \int_0^t (f(\nabla\Phi^*(y(s))) - f(x^*))\, ds$$

is a Lyapunov function for the dynamics in (7.1.3), where $x^* \in \mathcal{X}^*$ is arbitrary. Let

$x(t) = \nabla\Phi^*(y(t))$ and consider the derivative of $V(y(t))$ with respect to time:

$$\begin{aligned}\dot{V}(y(t)) &= \langle \dot{y}(t),\, \nabla\Phi^*(y(t)) - x^* \rangle + f(\nabla\Phi^*(y(t))) - f(x^*)\\ &= -\langle \nabla f(x(t)),\, x(t) - x^* \rangle + f(x(t)) - f(x^*),\end{aligned}$$

where the last equation follows by the definition of the dynamics $\dot{y}(t) = -\nabla f(\nabla\Phi^*(y(t)))$. Since f is continuously differentiable, we have

$$\dot{V}(y(t)) = -D_f(x^*, x(t)), \tag{7.4.1}$$

which is upper bounded by zero for any $y(t)$ since f is convex. Moreover, $\dot{V}(y(t)) = 0$ if and only if $y(t) \in \mathcal{Y}^*$. Therefore, LaSalle's invariance principle [86, Theorem 2] implies the asymptotic convergence of this differential equation, i.e., every solution $y(t)$ originating in some compact set tends to $\mathcal{Y}^*$ as $t \to \infty$. In [106], Nemirovski and Yudin prove the asymptotic convergence of the mirror descent dynamics by showing that the following function decreases along the trajectory of y:

$$W(y(t)) = \Phi^*(y(t)) - \langle y(t), x^* \rangle.$$

Comparing this function with the Lyapunov function above, we can observe that $V(y(t)) = W(y(t)) + \Phi(x^*) + \int_0^t (f(\nabla\Phi^*(y(s))) - f(x^*))\, ds$ and the corresponding arguments are closely related. We next derive convergence rate estimates for the function values.

Theorem 7.4. *Let $f : \mathcal{E} \to \mathbb{R}$ be a continuously differentiable convex function and $\Phi : \mathcal{X} \to \mathcal{E}^*$ satisfy the conditions in Proposition 7.1. Suppose $y : [0, +\infty) \to \mathcal{E}^*$ is a solution of (7.1.3) and let $x(t) = \nabla\Phi^*(y(t))$ be the corresponding primal trajectory. Then,*

$$f\left(\frac{1}{t}\int_0^t x(s)\, ds\right) - f(x^*) \le \frac{B_{\Phi^*}(y^0, x^*)}{t}, \quad \textit{where } x^* \in \mathcal{X}^*.$$

Proof Applying Jensen's inequality to the definition of $V(y(t))$ and lower bounding

$B_{\Phi^*}(y(t), x^*) \geq 0$, we get

$$f\left(\frac{1}{t}\int_0^t x(s)\,ds\right) - f(x^*) \leq \frac{V(y(t))}{t}.$$

Since $V(y(t))$ is non-increasing, right-hand side is upper bounded by $V(y(0))$, which yields the desired result. □

When Φ^* is twice differentiable, we show that function values generated by mirror descent method enjoys a non-ergodic convergence rate.

Theorem 7.5. *Let $f : \mathcal{E} \to \mathbb{R}$ be a continuously differentiable convex function and $\Phi : \mathcal{X} \to \mathcal{E}^*$ be a twice-differentiable function that satisfies the conditions in Proposition 7.1. Suppose $y : [0, +\infty) \to \mathcal{E}^*$ is a solution of (7.1.3) and let $x(t) = \nabla\Phi^*(y(t))$ be the corresponding primal trajectory. Then,*

$$f(x(t)) - f(x^*) \leq \frac{B_{\Phi^*}(y^0, x^*)}{t}, \quad \textit{where } x^* \in \mathcal{X}^*. \tag{7.4.2}$$

Proof Let us consider the following Lyapunov function

$$V(y(t)) = t\,(f(\nabla\Phi^*(y(t))) - f(x^*)) + B_{\Phi^*}(y(t), x^*). \tag{7.4.3}$$

The derivative of V with respect to time is given by

$$\dot{V}(y(t)) = f(\nabla\Phi^*(y(t))) - f(x^*) + t\,\langle \nabla f(\nabla\Phi^*(y(t))), \nabla^2\Phi^*(y(t)) \times \dot{y}\rangle + \langle \dot{y}(t),\, \nabla\Phi^*(y(t)) - x^*\rangle,$$

where the equality follows since Φ^* is twice differentiable. Using the definition of the dynamics $\dot{y}(t) = -\nabla f(x(t))$, where $x(t) = \nabla\Phi^*(y(t))$, we obtain

$$\dot{V}(y(t)) = f(x(t)) - f(x^*) + t\,\langle \nabla f(x(t)), \nabla^2\Phi^*(y(t)) \times \nabla f(x(t))\rangle - \langle \nabla f(x(t)),\, x(t) - x^*\rangle,$$

As Φ^* is convex, $\nabla^2\Phi^*(y)$ is positive semidefinite, and consequently

$$\dot{V}(y(t)) \leq f(x(t)) - f(x^*) - \langle \nabla f(x(t)), x(t) - x^* \rangle,$$

with equality if $y(t) \in \mathcal{Y}^*$. Similar to (7.4.1), the right-hand side above equals $-D_f(x^*, x(t))$, which yields $\dot{V}(y(t)) \leq 0$ with equality if and only if $y(t) \in \mathcal{Y}^*$, i.e., V is a non-increasing function of time. A simple manipulation of (7.4.3) yields $f(x(t)) - f(x^*) \leq V(y(t))/t$ since $B_{\Phi^*}(y(t), x^*) \geq 0$. Since $V(y(t))$ is a non-increasing function of time, we obtain the desired result. □

7.4.2 Strong Convergence

Our discussions so far illustrate the function values $f(x(t))$ converge to $\min_{x\in\mathcal{X}} f(x)$ as $t \to \infty$, which does not say anything about the pointwise convergence (i.e., strong convergence) of the trajectories x or y. Strong convergence often requires additional assumptions on the objective function other than convexity even for gradient systems, see e.g., the counterexample of Baillon [12]. For mirror descent methods, we refer to the recent paper [25] that constructs an example, where f is linear and Φ is Legendre, and mirror descent converges weakly but not strongly. Although strong convergence need not hold for gradient-like systems in general, it can be shown to hold under certain assumptions on f. In particular, strong convergence of gradient systems has been shown when f is even or strongly convex [38], or when $\operatorname{int}\mathcal{X}^* \neq \emptyset$ [35]. These results have been extended to heavy ball [4], Newton's method [5], and Nesterov's accelerated gradient descent [8]. We establish in the following theorem that under a similar assumption, mirror descent trajectories converge strongly.

Theorem 7.6. *Let $f : \mathcal{E} \to \mathbb{R}$ be a continuously differentiable convex function such that* $\operatorname{int}\mathcal{X}^* \neq \emptyset$ *and let $\Phi : \mathcal{X} \to \mathcal{E}^*$ be a twice-differentiable function that satisfies the conditions in Proposition 7.1. Suppose $y : [0, +\infty) \to \mathcal{E}^*$ is a solution of (7.1.3) and let $x(t) = \nabla\Phi^*(y(t))$ be the corresponding primal trajectory. Then, $y(t)$ converges strongly as $t \to \infty$*

to a point in $\mathcal{Y}^*$, and consequently $x(t)$ converges strongly to a point in $\mathcal{X}^*$.

Proof Consider the following potential function

$$V_\beta(t) = t(f(x(t)) - f(x^*)) + \beta B_{\Phi^*}(y(t), x^*).$$

Following similar lines to the proof of Theorem 7.5, we can obtain the derivative of $V_\beta(t)$ with respect to time as follows

$$\begin{aligned} \frac{d}{dt}V_\beta(t) &= f(x(t)) - f(x^*) + t\left\langle \nabla f(x(t)), \dot{x}(t) \right\rangle - \beta\left\langle \nabla f(x(t)), x(t) - x^* \right\rangle \\ &\leq f(x(t)) - f(x^*) - \beta\left\langle \nabla f(x(t)), x(t) - x^* \right\rangle, \qquad (7.4.4) \end{aligned}$$

where the inequality follows as Φ^* is convex and twice differentiable. The length of the trajectory $y(t)$ is given by $\int_0^\infty \|\dot{y}(t)\|_2\, dt$, and our aim is to show that this length is finite. Since $\operatorname{int}\mathcal{X}^* \neq \emptyset$, there exists a $x^* \in \mathcal{X}^*$ and an $\epsilon > 0$ such that $\nabla f(x') = 0$ for all $x' \in \mathcal{N}_\epsilon(x^*) = \{x : \|x - x'\|_2 \leq \epsilon\}$. As f is a continuously differentiable convex function, we have

$$f(x') \geq f(x) + \langle \nabla f(x), x' - x \rangle, \quad \forall x' \in \mathcal{N}_\epsilon(x^*).$$

This implies $\langle \nabla f(x), x' - x \rangle \leq 0$ since $x' \in \mathcal{X}^*$. We then obtain

$$\langle \nabla f(x), x' - x^* \rangle \leq \langle \nabla f(x), x - x^* \rangle.$$

Taking the supremum over $x' \in \mathcal{N}_\epsilon(x^*)$, we get

$$\epsilon \|\nabla f(x)\|_2 \leq \langle \nabla f(x), x - x^* \rangle.$$

Using this inequality in (7.4.4), we obtain

$$\frac{d}{dt}V_\beta(t) + \epsilon\beta \|\nabla f(x(t))\|_2 \leq f(x(t)) - f(x^*).$$

Integrating this inequality from 0 to t, we get

$$V_\beta(t) - V_\beta(0) + \epsilon\beta \int_0^t \|\nabla f(x(s))\|_2 \, ds \leq \int_0^t (f(x(s)) - f(x^*)) \, ds. \tag{7.4.5}$$

As $\dot{y}(t) = -\nabla f(x(t))$, $V_\beta(t) \geq 0$, and $V_\beta(0) < +\infty$, we can conclude that $y(t)$ has finite length if the right-hand side of (7.4.5) is finite. To establish this, we upper bound the right-hand side of (7.4.4) using convexity of f, similar to (7.4.1), as follows:

$$\frac{d}{dt} V_\beta(t) \leq -(\beta - 1)(f(x(t)) - f(x^*)).$$

For any $\beta > 1$, integrating the above inequality from 0 to t yields

$$\int_0^t (f(x(s)) - f(x^*)) \, ds \leq \frac{V_\beta(0) - V_\beta(t)}{\beta - 1} < +\infty.$$

This concludes the proof of strong convergence of $y(t)$ and consequently of $x(t) = \nabla\Phi^*(y(t))$ as $t \to \infty$. By LaSalle's invariance principle, we can conclude that $y(t)$ converges strongly to a point in $\mathcal{Y}^*$, and consequently $x(t)$ converges strongly to a point in $\mathcal{X}^*$ as $t \to \infty$. □

We next present a similar strong convergence result when the set of optimal solutions is polyhedral and characterize the point that mirror descent trajectory converges to. An earlier result of this kind is also presented in the original work of Bregman [34] for a relaxation method based on Bregman projections, see [34, Lemma 3 & Theorem 3]. More recently, a similar result has been shown in [70, Theorem 1] for mirror descent updates in discrete-time with continuously differentiable distance generating functions. However, this result relies on the assumption that the discrete-time sequence $\{x^k\}$ generated by the mirror descent method converges to a global minimizer, i.e., $x^\infty = \lim_{k\to\infty} x^k$ exists and $x^\infty \in \mathcal{X}^*$. This assumption is quite restrictive and is relaxed in the theorem below. Furthermore unlike [70], our results hold for non-differentiable distance generating functions and hence covers a broader class of problems including constrained optimization problems and problems with mixed norm penalties such as elastic norm.

Theorem 7.7. *Let $\Phi : \mathcal{X} \to \mathcal{E}^*$ satisfy the conditions in Proposition 7.1. Suppose $f(x) = g(Ax - b)$, where $b \in \operatorname{rge} A$, g is a strictly convex continuously differentiable function with $0 = \arg\min_z g(z)$, and $\mathcal{X}^* = \mathcal{X} \cap \{x \in \mathcal{E} : Ax = b\} \neq \emptyset$. Let $y : [0, +\infty) \to \mathcal{E}^*$ be a solution of (7.1.3). Then, the corresponding primal trajectory $x(t) = \nabla\Phi^*(y(t))$ satisfies*

$$\lim_{t\to\infty} x(t) = \arg\min_{\substack{x\in\mathcal{E}\\ Ax=b}} B_\Phi(x, y^0). \tag{7.4.6}$$

Proof Since Φ is strictly convex, there exists a unique solution to (7.4.6) denoted by x^* and the following KKT conditions are satisfied at x^*:

$$\begin{aligned} &\text{(feasibility)} && Ax^* = b, \\ &\text{(stationarity)} && \exists z^* \,:\, y^0 + A^\top z^* \in \partial\Phi(x^*). \end{aligned}$$

LaSalle's invariance principle [86, Theorem 2] implies that if $\bar{y}$ is an accumulation point of $y(\cdot)$, say with the corresponding subsequence $\{t_k\}_{k\geq 0}$, then $\nabla\Phi^*(\bar{y})$ is an element of the set of optimal solutions $\mathcal{X}^*$. Consequently, the corresponding accumulation point $\bar{x} = \nabla\Phi^*(\bar{y})$ is feasible for (7.4.6). Stationarity condition can be verified by integrating the mirror descent dynamics in the dual space:

$$\int_0^t \dot{y}(\tau)\, d\tau = -\int_0^t \nabla f(\nabla\Phi^*(y(\tau)))\, d\tau,$$

which is well-defined by Peano existence theorem. This yields

$$y(t) - y^0 = -A^\top \int_0^t \nabla g(A\nabla\Phi^*(y(\tau)) - b)\, d\tau.$$

Plugging in $t = t_k$ and taking the limit as $k \to \infty$ in the above equation, we obtain

$$\bar{y} - y^0 = -A^\top \lim_{t_k\to\infty} \int_0^{t_k} \nabla g(A\nabla\Phi^*(y(\tau)) - b)\, d\tau.$$

Since $\bar{y} \in \partial\Phi(\bar{x})$ by conjugate subgradient theorem, the above equation implies the existence of a z such that the stationary condition holds. Consequently $\bar{x}$ is an optimal solution to the problem (7.4.6). As the optimal solution of (7.4.6) is unique and given by x^*, then any accumulation point of $x(\cdot)$ is x^*, which concludes the proof. □

An immediate consequence of this theorem is presented below, which shows that when applied to quadratic problems, mirror descent returns the solution with smallest Φ in the least squares sense.

Corollary 7.8. *Let $\Phi : \mathcal{X} \to \mathcal{E}^*$ satisfy the conditions in Proposition 7.1 and suppose $f(x) = \frac{1}{2}\|Ax - b\|_2^2$. Let $y : [0, +\infty) \to \mathcal{E}^*$ be a solution of (7.1.3) and suppose $\mathcal{X}^* = \mathcal{X} \cap \{x \in \mathcal{E} : Ax = Pb\} \neq \emptyset$, where $P = A(A^\top A)^{-1}A^\top$ is the projection matrix onto* rge A. *Then, the corresponding primal trajectory $x(t) = \nabla\Phi^*(y(t))$ satisfies*

$$\lim_{t\to\infty} x(t) = \arg \min_{\substack{x\in\mathcal{E} \\ Ax=Pb}} B_\Phi(x, y^0).$$

Theorem 7.7 and Corollary 7.8 have strong implications on the mirror descent methods. In particular, it illustrates that mirror descent dynamics does not converge to an arbitrary minimizer of f, but converges to the one that minimizes the generalized Bregman divergence with respect to the initialization. We will discuss the ramifications of this theorem in more detail in Section 7.5. Before doing so, we next present another strong convergence result under a certain growth condition that is used to show the length of the gradient curves are bounded. In particular, Lojasiewicz [91] showed that for any real analytic function f, there exist $\rho \in [0, 1)$ and $C > 0$ such that the following inequality

$$\|\nabla f(x)\| \geq C|f(x) - f(x^*)|^\rho \tag{7.4.7}$$

holds for all x around a neighborhood of a critical point x^* of f. This inequality is known as *Lojasiewicz gradient inequality* and is sufficient show that the length of the gradient flow trajectory is finite [91, p. 1592]. Below, we consider an application of Lojasiewicz gradient inequality to mirror descent dynamics and prove the strong convergence of its trajectory.

Theorem 7.9. *Suppose f is a real analytic function and $\Phi : \mathcal{X} \to \mathcal{E}^*$ is an L-smooth function that satisfies the conditions in Proposition 7.1, where Φ^* is twice-differentiable. Let $y : [0, +\infty) \to \mathcal{E}^*$ be a solution of (7.1.3). Then, either $\lim_{t\to\infty} \|y(t)\| = \infty$ or there exists $y^* \in \mathcal{E}^*$ such that $\lim_{t\to\infty} y(t) = y^*$.*

Proof Assume that $\|y(t)\|$ does not go to infinity as $t \to \infty$ (or else the proof is complete), then $y(t)$ has an accumulation point $y^* \in \mathcal{E}^*$. It remains to show that $\lim_{t\to\infty} y(t) = y^*$. To this end, we consider

$$\frac{d}{dt} f(\nabla\Phi^*(y(t))) = \langle \nabla f(\nabla\Phi^*(y(t))), \nabla^2\Phi^*(y(t)) \times \dot{y}(t)\rangle,$$

similar to our derivations in the proof of Theorem 7.5. Since $\dot{y}(t) = -\nabla f(\nabla\Phi^*(y(t)))$, we then have

$$\frac{d}{dt} f(\nabla\Phi^*(y(t))) = -\langle \nabla f(\nabla\Phi^*(y(t))), \nabla^2\Phi^*(y(t)) \times \nabla f(\nabla\Phi^*(y(t)))\rangle,$$

Since Φ is assumed to be L-smooth, convex conjugacy implies Φ^* is $1/L$-strongly convex. Therefore, we obtain the following upper bound on $\dot{V}(t)$:

$$\frac{d}{dt} f(\nabla\Phi^*(y(t))) \leq -\frac{1}{L}\|\nabla f(\nabla\Phi^*(y(t)))\|^2, \tag{7.4.8}$$

which implies $f(\nabla\Phi^*(y(t)))$ is non-increasing. As y^* is an accumulation point of $y(t)$, we consequently have $\lim_{t\to\infty} f(\nabla\Phi^*(y(t))) = f(\nabla\Phi^*(y^*))$. Observe that if there exists $T > 0$ such that $f(\nabla\Phi^*(y(T))) = f(\nabla\Phi^*(y^*))$, then it is immediate that $\lim_{t\to\infty} y(t) = y^*$ since $\nabla f(\nabla\Phi^*(y(t))) = 0$ for any $t \geq T$.

We next consider the complementary case: $f(\nabla\Phi^*(y(t))) > f(\nabla\Phi^*(y^*))$ for all $t \geq 0$. As f is assumed to be real analytic, (7.4.7) holds around some neighborhood of y^*, denoted by $\mathcal{N}^*$, and using this inequality in (7.4.8), we obtain

$$\frac{d}{dt} f(\nabla\Phi^*(y(t))) \leq -\frac{C}{L}\|\nabla f(\nabla\Phi^*(y(t)))\| \times |f(\nabla\Phi^*(y(t)))|^{\rho}, \tag{7.4.9}$$

provided that $y(t) \in \mathcal{N}^*$, where we assumed $f(\nabla\Phi^*(y^*)) = 0$ for simplicity. Since we assumed that $f(\nabla\Phi^*(y(t))) > f(\nabla\Phi^*(y^*))$ for all $t \geq 0$, then (7.4.9) can be equivalently written as follows

$$M\frac{d}{dt}[f(\nabla\Phi^*(y(t)))]^{1-\rho} \leq -\|\nabla f(\nabla\Phi^*(y(t)))\|, \quad \text{where } M = \frac{L}{C(1-\rho)}. \tag{7.4.10}$$

For any $t_2 > t_1 \geq 0$, the length of the curve y between t_1 and t_2 is given by

$$\text{Length}_y(t_1, t_2) = \int_{t_1}^{t_2} \|\dot{y}(t)\|\, dt.$$

Since $\dot{y}(t) = -\nabla f(\nabla\Phi^*(y(t)))$, we obtain the following upper bound on $\text{Length}_y(t_1, t_2)$ using inequality (7.4.10):

$$\text{Length}_y(t_1, t_2) \leq -M\int_{t_1}^{t_2}\left(\frac{d}{dt}[f(\nabla\Phi^*(y(t)))]^{1-\rho}\right)dt, \tag{7.4.11}$$

provided that $y(t) \in \mathcal{N}^*$ for all $t \in (t_1, t_2)$. A straightforward manipulation of (7.4.11) yields

$$\begin{aligned}\text{Length}_y(t_1, t_2) &\leq M[f(\nabla\Phi^*(y(t_1))) - f(\nabla\Phi^*(y(t_2)))]^{1-\rho} \\ &\leq M[f(\nabla\Phi^*(y(t_1)))]^{1-\rho}, \end{aligned} \tag{7.4.12}$$

where the last inequality follows since $f(\nabla\Phi^*(y(t_2))) \geq 0$. This proves that the length of curve y between any (t_1, t_2) is finite provided that the curve lies in the neighborhood $\mathcal{N}^*$. We next show that indeed there exists $t_1 \geq 0$ such that $y(t) \in \mathcal{N}^*$ for all $t \geq t_1$, which concludes the proof.

As $\mathcal{N}^*$ is non-empty by the definition of the Lojasiewicz gradient inequality, there exists

$r > 0$ such that the radius-r ball around y^*, denoted by $\mathcal{B}_r = \{y \in \mathcal{E}^* : \|y - y^*\| < r\}$, is contained in $\mathcal{N}^*$. Since y^* is an accumulation point of $y(t)$, there exists $t_1 \geq 0$ such that $\|y(t_1) - y^*\| < r/2$ and $M[f(\nabla\Phi^*(y(t_1)))]^{1-\rho} < r/2$. Assume the contrary that $y(t)$ is not contained in $\mathcal{B}_r$ for all $t \geq t_1$. Then necessarily for some $t > t_1$, $\|y(t) - y^*\| = r$ holds. Denoting the smallest such time instance by t_2, we observe that $y(t) \in \mathcal{N}^*$ for all $t \in [t_1, t_2)$. Then, (7.4.12) implies $\text{Length}_y(t_1, t_2) < r/2$. Consequently, $\|y(t_2) - y^*\| \leq \|y(t_2) - y(t_1)\| + \|y(t_1) - y^*\| < r$, which is a contradiction. Therefore, $y(t) \in \mathcal{N}^*$ for all $t \geq t_1$. □

Theorem 7.9 is a generalization of the original work of Lojasiewicz [91] for gradient flows. These ideas are generalized to gradient-like systems in [85], where pointwise convergence is proven under an angle condition. Our proof above follows similar lines to [85], where we use the smoothness of Φ and twice-differentiability of Φ^* to obtain an equivalent condition. We also refer to [1], where the authors show pointwise convergence for discrete-time systems under the same assumptions as in [85]. The ideas of [1] can be used to prove the pointwise convergence of mirror descent in discrete-time, which we do not discuss further in this thesis.

7.5 Motivation

As we discussed in the previous sections, mirror descent methods are attractive for large-scale optimization problems since they can enjoy almost dimension-free convergence rates by considering a pertinent geometry to the problem at hand. Theorem 7.7 and Corollary 7.8 suggest another advantage of mirror descent methods: When the optimization problem has many global minima, which is often the case for large-scale problems, mirror descent recovers the solution that minimizes the corresponding Bregman divergence with respect to the initial solution provided to the algorithm. More specifically, let us consider the initialization $y^0 = \arg\min_y \Phi^*(y)$. Then, Theorem 7.7 implies that mirror descent dynamics return the solution to the problem:

$$\begin{aligned} \min \quad & \Phi(x) && \text{(7.5.1a)} \\ \text{s.t.} \quad & Ax = b. && \text{(7.5.1b)} \end{aligned}$$

Such convex optimization problems with linear constraints are central in the optimization literature [21, 22]. Our discussions in the earlier sections imply that mirror descent can be used to solve these problems as long as the conditions in Proposition 7.1 are satisfied (recall that these conditions are satisfied when Φ is strongly convex). An important example that satisfies these conditions is when f is the least squares objective $\frac{1}{2}\|Ax - b\|_2^2$ and Φ is the elastic net regularizer. Our results imply that mirror descent can be used to recover the minimum $\ell_1 + \ell_2$ norm solution over the linear system $Ax = b$. Although the problems where Φ is strongly convex can be handled with the theory presented in this chapter, there are several celebrated large-scale optimization problems that unfortunately do not satisfy the strong convexity condition. Among these we highlight the following two that have attracted significant attention from optimization, signal processing and machine learning communities: ℓ_1-norm minimization and nuclear norm minimization subject to linear constraints. These two problems are shown to provide sparsest and minimum rank (respectively) solutions to linear system of equations, and recovering such sparse and minimum rank solutions are fundamental problems in signal processing and machine learning as we discuss below.

Consider the problem of finding the sparsest solution of the linear system:

$$\begin{aligned} \min \quad & \|x\|_0 && \text{(7.5.2a)} \\ \text{s.t.} \quad & Ax = b, && \text{(7.5.2b)} \end{aligned}$$

where $\|x\|_0 = |\{i : x_i \neq 0\}|$ counts the number of nonzero entries of x. Recovering such sparse solutions is a central problem in signal processing, where the aim is to recover the sparse signal $x^* \in \mathbb{R}^n$ from the measurements $b \in \mathbb{R}^m$ that is obtained through the sensing matrix $A \in \mathbb{R}^{m\times n}$. It is well-known that (7.5.2) is NP-hard [104], which prompted researchers and practitioners to use its convex relaxation: the ℓ_1-norm minimization problem subject to linear constraints, which is also known as the basis pursuit problem and is given by

$$\begin{aligned} \min \quad & \|x\|_1 && (7.5.3a) \\ \text{s.t.} \quad & Ax = b. && (7.5.3b) \end{aligned}$$

Note that this problem is the convex relaxation of (7.5.2) as the objective function $\|\cdot\|_0$ is replaced by $\|\cdot\|_1$, i.e., its convex envelope on the unit ball. Compressed sensing theory guarantees that under certain conditions [44, 54], the optimal solution of the NP-hard problem (7.5.2) is given by the optimal solution of its convex relaxation (7.5.3). Although this convex problem can be solved by off-the-shelf linear programming solvers such as interior point methods or the simplex algorithm, such methods do not exploit the special structure of (7.5.3) and usually do not scale well as the problem dimension increases [56]. This urged many researchers to develop efficient solvers tailored to the basis pursuit problem, see e.g., [20, 56, 145].

Another problem of great interest is the low-rank matrix recovery:

$$\begin{aligned} \min \quad & \operatorname{rank}(X) && (7.5.4a) \\ \text{s.t.} \quad & \mathcal{A}(X) = b. && (7.5.4b) \end{aligned}$$

This problem has many applications such as matrix completion [123], collaborative filtering [135], minimum order linear system realization [63], and output feedback stabilization [58]. Similar to the previous example, the problem (7.5.4) is NP-hard in general. That is why, $\operatorname{rank}(X)$ is replaced by its convex envelope on the unit ball of matrices with spectral norm less than one. It is shown that [62] the nuclear norm of X, i.e., $\|X\|_* = \operatorname{tr}(\sqrt{X^\top X})$ the sum of the singular values of X, is the desired convex envelope for $\operatorname{rank}(X)$, which yields the following convex relaxation:

$$\min \quad \|X\|_* \tag{7.5.5a}$$

$$\text{s.t.} \quad \mathcal{A}(X) = b. \tag{7.5.5b}$$

It is proven that under certain conditions, the optimal solution of the NP-hard problem (7.5.4) is given by the optimal solution of the convex problem (7.5.5) [45, 46, 123]. Although this problem can be cast as a semidefinite programming problem, the standard semidefinite programming solvers is observed to be computationally expensive for large-scale problems, and consequently many specialized algorithms have been developed to solve (7.5.5), see e.g., [43, 97].

7.6 Summary of Contributions

As we discussed in the previous section, there is a prominent set of large-scale optimization problems that can be addressed by mirror descent methods provided that mirror descent can be implemented with non-strictly convex and non-differentiable distance generating functions. This corresponds to generalizing the mirror descent method to non-smooth geometries, which to our knowledge is not studied before in the literature.

In Chapter 8, we discuss how to discretize the mirror descent dynamics under the conditions of Proposition 7.1. The mirror descent differential equation (7.1.3) we are discretizing is defined in the dual space (similar to the original work of Nemirovski and Yudin [106]), and hence we call the resulting discrete-time methods as dual methods. This is in contrast to some existing works in the literature (see e.g., [39, 77, 149]), where mirror descent dynamics is defined as a differential equation in the primal space, see (8.2.1), and consequently the discrete-time methods that arise from this dynamics are to be viewed as primal methods. We first highlight the differences between these two approaches and characterize that strict convexity of Φ is necessary for the dual methods, whereas essential smoothness of Φ is necessary for the primal methods. When both conditions hold, primal and dual methods are equivalent to each other. This characterization enables us to study the corresponding methods under weaker assumptions. We next show that even though the primal and dual methods arise from a different geometric intuition, their convergence can be analyzed under a unified framework. This is thanks to our generalized Bregman divergence definition that does not require strict convexity or essential smoothness of Φ, and provides a measure of slackness for a primal and dual variable pair. We establish that convergence of all these methods can be reduced down to three components: An approximation error that arises from using explicit discretization which can be controlled by Lipschitz-like conditions or bounded subgradients, a subproblem optimality term that arises from the update rule and is controlled by the normal cone condition, and a function value improvement term that arise from the fact that negative gradient is a descent direction and is controlled by the well-known geometric property called the three-points identity. We apply our techniques to show the convergence of the aforementioned methods for non-smooth and relatively smooth problems. For non-smooth problems, we recover the existing convergence guarantees [18, 49, 81, 107] on the primal and dual methods through a simple unified approach. For relatively smooth problems, we recover the existing convergence guarantees for the primal methods presented in [15, 93] and we extend the existing analysis to dual methods that enables to use non-differentiable distance generating functions. Finally, we show that the forward Euler discretization of the mirror descent dynamics yields some of the celebrated methods in the literature. In particular, we illustrate that linearized Bregman iterations [152] that is used to solve compressed sensing problems is equivalent to the mirror descent method. Similarly, we show that the celebrated singular value thresholding algorithm [43] that is used to solve low-rank recovery problems is an instance of the mirror descent method. Our results provide a strong link between the existing methods in the literature and their analyses.

In Chapter 9, we discuss a systematic approach to generalize the mirror descent method to handle non-strictly convex and non-differentiable distance generating functions. To the best of our knowledge, there has been no work in the literature that studied this problem.

When the strict convexity condition on Φ is relaxed, the continuous-time mirror descent dynamics is given by a differential inclusion. We show that this differential inclusion is well-defined and has solutions that satisfy the convergence guarantees enjoyed by the mirror descent differential equation presented Section 7.4 provided that $\operatorname{dom}\Phi$ is bounded. Keep in mind that this procedure is not a simple change from a smooth optimization problem to a non-smooth one since the composite differential inclusion structure interferes with the monotonicity of the right-hand side of the differential inclusion, which we handle by using other properties. When $\operatorname{dom}\Phi$ is not bounded, we show that mirror descent can still be applied to quadratic optimization problems using the monotonicity preserving property of the linear maps. We show that in this case the mirror descent differential inclusion yields a unique solution that is given in the form of a differential equation almost everywhere. We then discuss how to discretize the mirror descent differential inclusion either through using ϵ-subgradients or by regularizing the differential inclusion and converting it to a differential equation that can be solved using the methods presented in Chapter 8. We also claim that when the surfaces of discontinuity in the right-hand side of the differential inclusion admit a stratified structure and when an efficient method of sliding on the corresponding hypersurfaces is known, then the trajectory of the mirror descent differential inclusion can be recovered by a combinatorial algorithm. We illustrate this approach on a sparse recovery problem, which yields a structure given by a sequence of nonnegative least squares problems. This formulation is considered before in the compressed sensing literature, where the method is called the adaptive inverse scale space method [42]. Unfortunately, this method is not efficient unless the solution that we are looking for is super sparse. More specifically, the original sparse recovery problem is of size n with linear objective and linear equality constraints, whereas the resulting formulation requires solving a sequence of quadratic optimization problems with linear inequality constraints (where the length of the sequence and the size of the problem can be as large as n). Unlike [42], we handle this issue by constructing an active-set method that uses the solution of the previous subproblem as a warm start to the next problem. We discuss how to efficiently implement the resulting algorithm.

Chapter 8

A Unified View of Mirror Descent Method in Discrete-Time and Related Methods

In this chapter, we study the mirror descent method in discrete-time. In Section 8.1, we discuss forward and backward Euler discretization applied to mirror descent dynamics. We then present a detailed discussion on the resulting discrete-time mirror descent methods and other existing methods in the literature such as the Bregman proximal gradient and dual averaging methods. We then present in Section 8.2 a universal recipe to prove convergence of all these methods. In Section 8.3, we apply the presented methodology to non-smooth and relatively smooth optimization problems and recover the existing rate estimates in the literature as well as certain novel rate estimates in a simple comprehensible way. For a more detailed summary of our contributions and comparisons to existing works in the literature, we refer to Section 8.3.1. Finally, in Section 8.4, we show the celebrated sparse optimization methods, the linearized Bregman iterative method and the singular value thresholding method, are instances of the mirror descent method. We conclude the chapter in Section 8.5 with a few remarks.

8.1 Discretization of Mirror Descent Dynamics

In this chapter, we discuss discretization of mirror descent dynamics and the resulting algorithms. Recall the mirror descent dynamics to minimize a convex continuously differentiable function f over a closed convex set $\mathcal{X}$:

$$\dot{y}(t) = -\nabla f(\nabla\Phi^*(y(t))), \tag{8.1.1a}$$

$$y(0) = y^0 \in \mathrm{dom}(\nabla\Phi^*). \tag{8.1.1b}$$

Below, we discuss forward and backward Euler methods applied to (8.1.1).

8.1.1 Forward Euler Discretization

We first consider the forward Euler method, which yields the following discrete-time update:

$$y^{k+1} = y^k - \eta_k \nabla f(\nabla\Phi^*(y^k)), \tag{8.1.2}$$

with the initialization $y^0 \in \mathrm{dom}(\nabla\Phi^*)$. This yields a sequence of primal solutions by a simple computation $x^k = \nabla\Phi^*(y^k)$, and hence we can equivalently write the update rule (8.1.2) as follows:

$$x^k = \nabla\Phi^*(y^k), \tag{8.1.3a}$$

$$y^{k+1} = y^k - \eta_k \nabla f(x^k). \tag{8.1.3b}$$

We also observe that another equivalent formulation to (8.1.2) can be obtained using generalized Bregman divergences as follows:

$$x^{k+1} = \arg\min_x \left\{ \langle x, \nabla f(x^k)\rangle + \frac{1}{\eta_k} B_\Phi(x, y^k) \right\}, \tag{8.1.4a}$$

$$y^{k+1} = y^k - \eta_k \nabla f(x^k). \tag{8.1.4b}$$

Indeed, the optimality condition of (8.1.4a) is given by

$$y^k - \eta_k \nabla f(x^k) \in \partial \Phi(x^{k+1}),$$

and when y^{k+1} is updated as in (8.1.4b), we have $y^{k+1} \in \partial\Phi(x^{k+1})$, which implies $x^{k+1} = \nabla\Phi^*(y^{k+1})$ as Φ^* is differentiable. In the sequel, we will use both (8.1.3) and (8.1.4) to relate the mirror descent method to existing methods in the literature. Throughout the thesis, we refer to (8.1.3) as the explicit mirror descent method in discrete-time.

Follow-the-regularized-leader

In the online convex optimization literature, any algorithm minimizing a linearized sum of past losses plus a regularization term is considered a follow-the-regularized-leader variant. Therefore, the online version of (8.1.3) is also referred as follow-the-regularized-leader algorithm with linear losses or online mirror descent [133]. Its extension to online composite optimization problems is called regularized dual averaging method [151].

Dual Averaging

Dual averaging is proposed by Nesterov in [107] for non-smooth optimization problems, where the main idea is to prevent the subgradients enter the optimization model with decreasing weights. The author addresses this issue by using two control sequences: one is the step size that aggregates subgradients in the dual space (denoted with η below) and the other is a dynamically updated scale between the primal and dual spaces (denoted with β below). The update rule of dual averaging is given by

$$x^k = \arg\min_{x\in\mathcal{E}} \{\beta_k \Phi(x) - \langle x, y^k\rangle\}, \tag{8.1.5a}$$

$$y^{k+1} = y^k - \eta_k \nabla f(x^k). \tag{8.1.5b}$$

The only difference between (8.1.5) and the description in [107, Equations 2.4 & 2.14] is the definition of Φ: Here, Φ is defined on $\mathcal{E}$ with $\operatorname{dom}\Phi = \mathcal{X}$, i.e., it is an extended real-valued function, whereas in [107] Φ is defined on $\mathcal{X}$ and thus the minimization in (8.1.5a) taken over $\mathcal{X}$. Recalling our generalized Bregman divergence definition, (8.1.5a) can be equivalently written as follows:

$$x^k = \arg\min_{x \in \mathcal{E}} B_\Phi(x, y^k/\beta_k). \tag{8.1.6}$$

Observe that the optimality condition of (8.1.6) implies that $y^k/\beta_k \in \partial\Phi(x^k)$ and hence we have $x^k \in \nabla\Phi^*(y^k/\beta_k)$ since Φ^* is continuously differentiable. Therefore, the dual averaging updates are given by

$$x^k = \nabla\Phi^*(y^k/\beta_k), \tag{8.1.7a}$$

$$y^{k+1} = y^k - \eta_k \nabla f(x^k). \tag{8.1.7b}$$

Comparing (8.1.7) and (8.1.3), we can see how dual averaging generalizes mirror descent with an additional control sequence $\{\beta_k\}$, a scale between the primal and dual spaces.

Bregman Proximal Gradient

When Φ is continuously differentiable on its domain, the computation in (8.1.4b) is redundant since $y^{k+1} = \nabla\Phi(x^{k+1})$ and x^{k+1} is already computed in (8.1.4a). That is, we can equivalently write (8.1.4) as follows:

$$x^{k+1} = \arg\min_{x \in \mathcal{E}} \left\{ \langle x, \nabla f(x^k) \rangle + \frac{1}{\eta_k} B_\Phi(x, \nabla\Phi(x^k)) \right\}. \tag{8.1.8}$$

Furthermore, since Φ is assumed to be differentiable, we can replace the generalized Bregman divergence in (8.1.8) with the classical Bregman divergence (7.3.1), i.e., we obtain the update rule of the Bregman proximal gradient method:

$$x^{k+1} = \arg\min_{x\in\mathcal{X}}\left\{\langle x, \nabla f(x^k)\rangle + \frac{1}{\eta_k}D_\Phi(x, x^k)\right\}. \tag{8.1.9}$$

Note that we changed the optimization space from $\mathcal{E}$ in (8.1.8) to $\mathcal{X}$ in (8.1.9). That is because, the classical Bregman divergence definition is valid only on the space, where Φ is differentiable. Since $\mathcal{X}$ is assumed to be a closed set, Φ is differentiable on $\mathcal{X}$ when one of the following two conditions hold:

1. $\mathcal{X}$ is both open and closed, that is $\mathcal{X} = \mathcal{E}$.

2. $\nabla\Phi$ diverges as its argument goes to $\partial\mathcal{X}$, that is $\lim_{k\to\infty}\|\nabla\Phi(x^k)\| = +\infty$ for any sequence $\{x^k\}$ in $\mathcal{X}$ converging to a boundary point x of $\mathcal{X}$. Note that this is equivalent to saying that Φ is essentially smooth.

These two conditions ensure that the iterates generated by the mirror descent and Bregman proximal gradient methods remain in the *interior* of the feasible set $\mathcal{X}$. Furthermore, since Φ and Φ^* are differentiable, there is a bijective mapping between the primal and dual spaces, which imply the equivalence of these two methods.

These two conditions above (first discussed in [18]) are essential to be able to conclude the equivalence of the mirror descent and Bregman proximal gradient methods. In particular, if $\mathcal{X}$ is a bounded closed convex set and $\nabla\Phi$ does not diverge as its argument goes to $\partial\mathcal{X}$, then Φ cannot be differentiable on $\partial\mathcal{X}$. As an example, consider $\mathcal{X} = \{x : \|x\|_2 \leq 1\}$ and suppose we want to choose $\frac{1}{2}\|x\|_2^2$ as the distance generating function. If we define $\Phi(x) = \frac{1}{2}\|x\|_2^2$ when $x \in \mathcal{X}$ and $+\infty$ otherwise, then Φ is not differentiable on $\partial\mathcal{X}$. This is not an issue for the mirror descent method: it does not require Φ to be differentiable since it uses subgradients of Φ, see (8.1.3b). On the other hand, when Φ is not differentiable on $\mathcal{X}$, the Bregman proximal gradient method becomes ill-defined. However, this issue can be handled by setting $\Phi(x) = \frac{1}{2}\|x\|_2^2$ for every $x \in \mathcal{E}$, which ensures differentiability of Φ on $\mathcal{X}$. Then, the subproblem in (8.1.9) is solved over the set $\mathcal{X}$, which ensures the primal iterates generated by the Bregman proximal gradient method remains feasible. This procedure is often called forward-backward splitting [52].

8.1.2 Backward Euler Discretization

We next consider the backward Euler method, which yields the following discrete-time update:

$$y^{k+1} = y^k - \eta_k \nabla f(\nabla \Phi^*(y^{k+1})), \tag{8.1.10}$$

with the initialization $y^0 \in \mathrm{dom}(\nabla\Phi^*)$. Similar to the previous section, primal solutions are given by $x^k = \nabla\Phi^*(y^k)$. Using generalized Bregman divergences, it is straightforward to observe that (8.1.10) can be equivalently written as follows:

$$x^{k+1} = \arg\min_x \left\{ f(x) + \frac{1}{\eta_k} B_\Phi(x, y^k) \right\}, \tag{8.1.11a}$$

$$y^{k+1} = y^k - \eta_k \nabla f(x^{k+1}). \tag{8.1.11b}$$

Indeed, the optimality condition of (8.1.11a) is given by

$$y^k - \eta_k \nabla f(x^{k+1}) \in \partial\Phi(x^{k+1}),$$

and when y^{k+1} is updated as in (8.1.11b), we have $y^{k+1} \in \partial\Phi(x^{k+1})$, which implies $x^{k+1} = \nabla\Phi^*(y^{k+1})$ as Φ is strictly convex. The equivalence of (8.1.10) and (8.1.11) hence follows.

Throughout the thesis, we refer to (8.1.11) as the implicit mirror descent method in discrete-time. In [81], the algorithm in the exact form (8.1.11), but with a different generalized Bregman divergence definition (see (7.3.5)), is proposed and its convergence is proven. Later in [112], the same algorithm is reinvented and is called Bregman iterative regularization algorithm. Its convergence is proven for the case f is a quadratic and Φ is the total variation seminorm. There is a huge literature on this method and for a detailed analysis and applications we refer the reader to [81, 152].

Bregman Proximal Point Method

Similar to our discussions for the Bregman proximal gradient method in the previous section, when Φ is continuously differentiable on its domain, the computation in (8.1.11b) is redundant since $y^{k+1} = \nabla\Phi(x^{k+1})$ and x^{k+1} is already computed in (8.1.11a). That is, we can equivalently write (8.1.11) as follows:

$$x^{k+1} = \arg\min_{x\in\mathcal{E}} \left\{ f(x) + \frac{1}{\eta_k} B_\Phi(x, \nabla\Phi(x^k)) \right\}. \tag{8.1.12}$$

Furthermore, since Φ is assumed to be differentiable, we can replace the generalized Bregman divergence in (8.1.12) with the classical Bregman divergence (7.3.1), i.e., we obtain the update rule of the Bregman proximal point method:

$$x^{k+1} = \arg\min_{x\in\mathcal{X}} \left\{ f(x) + \frac{1}{\eta_k} D_\Phi(x, x^k) \right\}. \tag{8.1.13}$$

This method in (8.1.13) is first proposed in [47], where the authors call it the Bregman proximal minimization method. The convergence of this method is extensively analyzed in [47, 49, 142].

8.2 A Unified View of Existing Methods

According to our discussion above, the explicit and implicit mirror descent methods as well as the dual averaging method can be viewed as dual algorithms. That is, these methods arise from the discretization of the mirror descent dynamics in continuous-time (8.1.1), which describes a flow in the dual space. For these methods to be well-defined, Φ needs to be strictly/strongly convex, so that the mapping from the dual space to the primal space is single-valued, i.e., $\partial\Phi^* = \nabla\Phi^*$. Notice that these methods do not require Φ to be continuously differentiable.

On the other hand, the Bregman proximal gradient and Bregman proximal point methods are primal algorithms. This can be observed more clearly by considering the following continuous-time dynamics in the primal space:

$$\frac{d}{dt}\nabla\Phi(x(t)) = -\nabla f(x(t)) - \mathcal{N}_{\mathcal{X}}(x(t)), \tag{8.2.1}$$

Indeed, forward-backward splitting applied to (8.2.1) yields the Bregman proximal gradient method in (8.1.9), whereas backward Euler discretization applied to (8.2.1) yields the Bregman proximal point method in (8.1.13). For these methods to be well-defined, we need the additional assumption that Φ is continuously differentiable, i.e., the mapping from the primal space to the dual space is single-valued $\partial\Phi = \nabla\Phi$. For an overview of all these algorithms and their differences, see Table 8.1.

As the remarks above make it clear, all aforementioned methods have different update rules, yet their convergence follows by a unified theory. To observe this, we first unify the notation and for dual methods we let $\{y^k\}$ denote the dual sequence generated by the method. On the other hand, for primal methods we let $\{y^k = \nabla\Phi(x^k)\}$ denote the image of the primal sequence generated by the method, which is well-defined as Φ is assumed to be differentiable for primal methods. We also let $\{d^k\}$ denote the update sequence, i.e., $d^k = -\eta_k\nabla f(x^{k+1})$ for the implicit mirror descent and Bregman proximal point methods, whereas $d^k = -\eta_k\nabla f(x^k)$ for the rest of the methods. We first observe that all methods satisfy the following subproblem optimality condition.

Lemma 8.1. *Consider the methods in Table 8.1 applied to solve the problem (7.1.1). Every iteration of these methods satisfies*

$$(y^k + d^k) - y^{k+1} \in \mathcal{N}_X(x^{k+1}), \quad \forall k \geq 0. \tag{8.2.2}$$

Proof For explicit and implicit mirror descent as well as dual averaging, we have $y^{k+1} = y^k + d^k$ by their definition (recall these are dual methods). Then, (8.2.2) trivially holds since $0 \in \mathcal{N}_{\mathcal{X}}(x)$ for every $x \in \mathcal{X}$. On the other hand, for the Bregman proximal gradient

Table 8.1: A summary of existing methods.

	Primal Method	Dual Method
Continuous-Time Dynamics	$\frac{d}{dt}\nabla\Phi(x(t)) = -\nabla f(x(t)) - \mathcal{N}_{\mathcal{X}}(x(t))$	$\frac{d}{dt}y(t) = -\nabla f(\nabla\Phi^*(y(t)))$
Backward Euler Discretization	*Bregman proximal point* $x^{k+1} = \arg\min_{x\in\mathcal{X}}\Big\{f(x) + \frac{1}{\eta_k}D_\Phi(x, x^k)\Big\}$	*Implicit mirror descent* $x^{k+1} = \arg\min_{x\in\mathcal{E}}\Big\{f(x) + \frac{1}{\eta_k}B_\Phi(x, y^k)\Big\}$ $y^{k+1} = y^k - \eta_k\nabla f(x^{k+1})$
Forward Euler Discretization	*Bregman proximal gradient* $x^{k+1} = \arg\min_{x\in\mathcal{X}}\Big\{\langle x, \nabla f(x^k)\rangle + \frac{1}{\eta_k}D_\Phi(x, x^k)\Big\}$	*Explicit mirror descent* $x^k = \nabla\Phi^*(y^k)$ $y^{k+1} = y^k - \eta_k\nabla f(x^k)$
Forward Euler + Dynamic Scaling	×	*Dual averaging* $x^k = \nabla\Phi^*(y^k/\beta_k)$ $y^{k+1} = y^k - \eta_k\nabla f(x^k)$

method, the optimality of (8.1.9) implies

$$(\nabla\Phi(x^k) - \eta_k \nabla f(x^k)) - \nabla\Phi(x^{k+1}) \in \mathcal{N}_X(x^{k+1}),$$

which is equivalent to (8.2.2). Similarly, for the Bregman proximal point method, the optimality of (8.1.13) implies

$$(\nabla\Phi(x^k) - \eta_k \nabla f(x^{k+1})) - \nabla\Phi(x^{k+1}) \in \mathcal{N}_X(x^{k+1}),$$

which is equivalent to (8.2.2). □

The normal cone condition in Lemma 8.1 together with the three-points identity presented in Lemma 7.2 and the fact that $-\nabla f$ is a descent direction imply the convergence of all aforementioned methods as we discuss below. Throughout the section for generality, we consider the case where f is non-differentiable and let $x^* \in \arg\min f$ be an optimal solution of (7.1.1). In the remainder of this section, we present a universal recipe of convergence for the explicit mirror descent, Bregman Proximal gradient, implicit mirror descent, Bregman proximal point and dual averaging methods. The main idea is to represent the function suboptimality by the following three terms:

1. Approximation error that arises in explicit methods due to approximating $f(x^{k+1})$ around x^k, which is controlled by a Lipschitz-like condition or arithmetic-geometric mean inequality.

2. Optimality condition of the subproblem defined by the update rule of the method, which is controlled by Lemma 8.1.

3. Function value improvement measure that can be characterized by three-points identity Lemma 7.2 or four-points inequality Lemma 8.2.

Explicit Mirror Descent & Bregman Proximal Gradient Methods

Recall that for the explicit mirror descent and Bregman proximal gradient methods, we have

$$d^k = -\eta_k g^k, \quad \text{where } g^k \in \partial f(x^k).$$

Then, convexity of f implies

$$\eta_k (f(x^k) - f(x^*)) \leq \langle d^k, x^* - x^k \rangle,$$

After a slight massaging, we obtain

$$\begin{aligned} \eta_k (f(x^k) - f(x^*)) &\leq \langle d^k, x^{k+1} - x^k \rangle + \langle d^k, x^* - x^{k+1} \rangle \\ &\leq T_1^k + T_2^k + T_3^k, \end{aligned} \tag{8.2.3}$$

where

$$\begin{aligned} T_1^k &= \langle d^k, x^{k+1} - x^k \rangle, \\ T_2^k &= \langle y^k + d^k - y^{k+1}, x^* - x^{k+1} \rangle, \\ T_3^k &= \langle y^{k+1} - y^k, x^* - x^{k+1} \rangle. \end{aligned}$$

Observe that for a convergent algorithm $\|x^{k+1} - x^k\|$ goes to zero as $k \to \infty$, i.e., T_1^k is a decreasing sequence of approximation errors (that arise from approximating $f(x^{k+1})$ around x^k). Furthermore, $T_2^k \leq 0$ by Lemma 8.1 (optimality of subproblems) and using Lemma 7.2 (three-points identity) with $x_1 = x^{k+1}$, $x_3 = x^*$, $y_1 = y^{k+1}$ and $y_2 = y^k$, we obtain

$$T_3^k = B_\Phi(x^*, y^k) - B_\Phi(x^*, y^{k+1}) - B_\Phi(x^{k+1}, y^k).$$

Implicit Mirror Descent & Bregman Proximal Point Methods

Recall that for the implicit mirror descent and Bregman proximal point methods, we have

$$d^k = -\eta_k g^{k+1}, \quad \text{where } g^{k+1} \in \partial f(x^{k+1}).$$

Then, convexity of f implies $\eta_k(f(x^{k+1}) - f(x^*)) \leq \langle d^k, x^* - x^{k+1}\rangle$. Using the definitions in the previous section, it is straightforward to observe that

$$\eta_k(f(x^{k+1}) - f(x^*)) \leq T_2^k + T_3^k, \tag{8.2.4}$$

i.e., the upper bounds in (8.2.3) and (8.2.4) are identical up to an approximation error T_1^k.

Dual Averaging Method

Similar to the explicit mirror descent method, the dual averaging update directions are given by

$$d^k = -\eta_k g^k, \quad \text{where } g^k \in \partial f(x^k),$$

and similar to (8.2.3) we have

$$\eta_k(f(x^k) - f(x^*)) \leq T_1^k + T_2^k + T_3^k.$$

T_1^k and T_2^k can be upper bounded similar to the explicit mirror descent and Bregman proximal gradient methods. On the other hand for T_3^k, we introduce the following four-points inequality that follows from the three-points identity (Lemma 7.2) and Fenchel's inequality.

Lemma 8.2 (four-points inequality)**.** *Let $\Phi : \mathcal{X} \to \mathcal{E}^*$ be a closed proper convex distance generating function. Then for any four points $x_1, x_3 \in \mathcal{X}$ and $y_1, y_2 \in \operatorname{dom} \partial\Phi^*$, the* *following inequality holds:*

$$B_\Phi(x_3, y_1) + B_\Phi(x_1, y_2) - B_\Phi(x_3, y_2) \geq \langle x_3 - x_1, y_2 - y_1 \rangle.$$

Proof Similar to the proof of Lemma 7.2, (7.3.9) yields

$$B_\Phi(x_3, y_1) + B_\Phi(x_1, y_2) - B_\Phi(x_3, y_2) = \Phi(x_1) + \Phi^*(y_1) + \langle x_3, y_2 \rangle - \langle x_3, y_1 \rangle - \langle x_1, y_2 \rangle.$$

By Fenchel's inequality, we have $\Phi(x_1) + \Phi^*(y_1) \geq \langle x_1, y_1 \rangle$. Plugging in this equation above and reorganizing terms, we obtain the desired result. □

To upper bound T_3^k, we apply Lemma 8.2 with $x_1 = x^{k+1}$, $x_3 = x^*$, $y_1 = y^{k+1}/\beta_k$ and $y_2 = y^k/\beta_k$, which yields

$$T_3^k \leq \beta_k \big[B_\Phi(x^*, y^k/\beta_k) - B_\Phi(x^*, y^{k+1}/\beta_k) - B_\Phi(x^{k+1}, y^k/\beta_k) \big].$$

Using the recipe described in this section, we next establish the convergence rate of the aforementioned methods for non-smooth and relatively smooth problems.

8.3 A Unified Convergence Analysis

We next apply the universal convergence recipe presented in Section 8.2 to non-smooth and relatively smooth problems in Sections 8.3.2 and 8.3.3, respectively. Before doing so, we first discuss the related work in the literature and our contributions in the following section.

8.3.1 Related Work & Contributions

Non-Smooth Problems

A unified analysis of the explicit mirror descent and Bregman proximal gradient methods is first presented in the seminal work of Beck and Teboulle [18]. In particular, the authors showed that when Φ is Legendre, the iterates generated by the explicit mirror descent and Bregman proximal gradient methods are equivalent. Thus, the convergence of explicit mirror descent can be analyzed via Bregman divergences. Unfortunately, this equivalence does not hold unless Φ is essentially smooth on $\mathcal{X}$ as discussed in the previous section. Our analysis here only requires assumptions to have well-defined discrete-time methods, i.e., Φ is not assumed to be continuously differentiable for the dual methods.

In [100], online versions of the explicit mirror descent and Bregman proximal gradient methods (as well as the follow-the-regularized-leader method) are considered. The authors consider unconstrained composite optimization problems with a time-varying non-smooth component. In this case, the difference between the explicit mirror descent and Bregman proximal gradient methods appear as a result of having a time-varying regularizer. The authors show that unification of these methods can still be shown if the mirror map is varied over time as well with a suitable scaling. This idea is quite different from what we present in this chapter and see (8.3.1) for a more rigorous description.

The most closely related work to this chapter is by Juditsky et al. [77], where the authors consider the explicit mirror descent and Bregman proximal gradient methods (in their paper these methods are called dual averaging and mirror descent, respectively). The main contribution of [77] is the definition of a unified mirror descent method that reduces to either the Bregman proximal gradient method (i.e., when Φ is continuously differentiable on $\mathcal{E}$, see [77, Proposition 2.2]) or the explicit mirror descent method (i.e., Φ is strongly convex or Φ is strictly convex and $\mathcal{X}$ is bounded, see [77, Proposition 3.2]). Indeed, it is obvious that under the former assumption the Bregman proximal gradient method is well-defined, whereas under the latter assumptions Proposition 7.1 holds (see the discussion following Proposition 7.1). The authors provide a unified framework for explicit mirror descent and Bregman proximal gradient methods, and recover their rate estimates for non-smooth problems presented in [18]. Our framework in this chapter additionally covers the dual averaging, implicit mirror descent and Bregman proximal point methods for non-smooth problems, and recovers the rate estimates presented in [107, 49, 81], respectively. Furthermore, we extend our analysis to relatively smooth problems and obtain the rate estimates presented in [15] and [93], while relaxing certain differentiability conditions as we discuss below in more detail. A similar unified framework for non-smooth and relatively smooth problems has been presented for the explicit mirror descent and Bregman proximal gradient methods in [139]. However, it relies on the assumption Φ is Legendre and thus does not have the generality of the results of this chapter.

Relatively Smooth Problems

Relative smoothness is first introduced in [15] for composite optimization problems, where one of the components is continuously differentiable while the other is not. For the sake of clarity, we discuss the non-composite case (7.1.1), where the methods we consider can be applied to composite problems with an additional proximal step as presented in [15]. In [15], the authors consider solving (7.1.1) with a Bregman proximal gradient algorithm, where f is continuously differentiable on $\operatorname{int}\mathcal{X}$ and Φ is Legendre such that f is L-smooth with respect to Φ:

$$\exists L > 0 \quad \text{such that} \quad L\Phi - f \text{ is convex on } \operatorname{int}\mathcal{X}.$$

As we discussed in the previous section, when Φ is Legendre, the iterates generated by the Bregman proximal gradient algorithm remain in $\operatorname{int}\mathcal{X}$ and are identical to the iterates of the explicit mirror descent method (for the composite case, see [15, Section 3] for additional assumptions on the non-differentiable component function). Under the relative smoothness assumption, the authors prove the weak convergence of the Bregman proximal gradient method (called NoLips algorithm in [15]) and with additional assumptions they also prove strong convergence of the iterates. Unlike [15], we do not only focus on the explicit mirror descent and Bregman proximal gradient methods when they are equivalent (i.e., when Φ is of Legendre type). More specifically, for our analysis to hold strict/strong convexity is sufficient for the dual methods and essential smoothness is sufficient for the primal methods.

In [93], the authors define relative smoothness in the same way and establish the convergence rate of the Bregman proximal gradient method. Their analysis does not require Φ to be strictly/strongly convex, but uses a slightly weaker assumption that Φ is strongly convex relative to f, and they provide a linear convergence rate estimate when f is relatively strongly convex with respect to Φ (i.e., $\exists \mu > 0$ such that $f - \mu\Phi$ is convex on $\operatorname{int}\mathcal{X}$). They also consider the explicit mirror descent method (which the authors call dual averaging, not to be confused with dual averaging we define in Table 8.1) and present its sublinear and linear rate estimates. Yet, their analysis requires Φ to be continuously differentiable on $\mathcal{E}$ while constraints are handled with a projection step similar to the Bregman proximal gradient method. Hence, this method is not well-defined for non-differentiable distance generating functions unlike the mirror descent and dual averaging methods we consider here. Nevertheless, the assumptions in [93] are milder compared to [15] such that the trajectory of the Bregman proximal gradient method is not necessarily identical to the trajectory of the mirror descent method. The authors also extend their analysis to composite optimization problems using a slight generalization of Bregman divergence, but that analysis is much different from the one we consider here in the sense that the subgradient of the non-smooth component function cancels out in the update rule and the algorithm effectively becomes a Bregman proximal gradient method similar to [15].

In [92], the above ideas are extended to non-smooth problems using subgradient methods, and an extension to stochastic composite optimization problems is presented in [64]. The algorithm in [64] reduces to regularized dual averaging method for deterministic problems of the type $\min_{\mathcal{X}} f + g$, where f is convex continuously differentiable and g is closed proper convex. Given a distance generating function Φ of Legendre type, the update rule of the regularized dual averaging method is given by (see [64, Section 2.2]):

$$y^k = y^{k-1} - \eta \nabla f(x^{k-1}), \tag{8.3.1a}$$

$$x^k = \nabla \Phi_k^*(y^k), \tag{8.3.1b}$$

where $\Phi_k = \Phi + k\eta g$ and $\eta > 0$ is a fixed step size. An important difference of this algorithm from [15] is that it does not necessarily operate on $\operatorname{int} \mathcal{X}$. Indeed when $g = \iota_{\mathcal{X}}$, it can be observed that this method reduces to the explicit mirror descent method since $\iota_{\mathcal{X}}$ is not affected by scaling in (8.3.1b). However, for an arbitrary regularizer g, this method is different from the explicit mirror descent method due to the time-varying distance generating function Φ_k. A generalization of this method is studied in [48], where the authors consider $\Phi_k = \Phi + h(k,\eta)g$ and the function h is a design parameter.

8.3.2 Convergence Analysis for Non-smooth Problems

In this section, we consider a constrained convex optimization problem of the form

$$\min_{x \in \mathcal{X}} f(x), \tag{8.3.2}$$

where $\mathcal{X}$ is a nonempty closed convex subset of $\mathcal{E}$ and f is a closed proper convex function. Let Φ be a μ-strongly convex function, which is assumed to be continuously differentiable on $\mathcal{X}$ whenever we are talking about primal methods, whereas it may be non-differentiable for dual methods.

Explicit Mirror Descent & Bregman Proximal Gradient Methods

By our discussions in Section 8.2, the following inequality holds for the explicit mirror descent and Bregman proximal gradient methods:

$$\eta_k(f(x^k) - f(x^*)) \leq \langle d^k, x^{k+1} - x^k\rangle + B_\Phi(x^*, y^k) - B_\Phi(x^*, y^{k+1}) - B_\Phi(x^{k+1}, y^k),$$

where $d^k = -\eta_k g^k$ and $g^k \in \partial f(x^k)$. Using arithmetic-geometric mean inequality on the first term, we obtain

$$\eta_k(f(x^k) - f(x^*)) \leq \frac{\eta_k^2}{2\mu}\|g^k\|_*^2 + \frac{\mu}{2}\|x^{k+1} - x^k\|^2 + B_\Phi(x^*, y^k) - B_\Phi(x^*, y^{k+1}) - B_\Phi(x^{k+1}, y^k).$$

Using Lemma 7.3 with $x_1 = x^{k+1}$, $x^2 = x^k$ and $y^2 = y^k$, we can observe that $\frac{\mu}{2}\|x^{k+1} - x^k\|^2 - B_\Phi(x^{k+1}, y^k) \leq 0$, which substituting into the above inequality yields

$$\eta_k(f(x^k) - f(x^*)) \leq \frac{\eta_k^2}{2\mu}\|g^k\|_*^2 + B_\Phi(x^*, y^k) - B_\Phi(x^*, y^{k+1}).$$

Summing this inequality for $k = 1, \ldots, \ell$ (where we start the initial index from 1 for notational convenience), we get

$$\sum_{k=1}^{\ell} \eta_k(f(x^k) - f(x^*)) \leq B_\Phi(x^*, y^1) - B_\Phi(x^*, y^{\ell+1}) + \frac{1}{2\mu}\sum_{k=1}^{\ell} \eta_k^2 \|g^k\|_*^2.$$

Lower bounding each $f(x^k)$ by $\min_k f(x^k)$ in the left-hand side and noticing that $B_\Phi(x^*, y^{\ell+1})$ is non-negative, we obtain

$$\min_{1\leq k\leq \ell} f(x^k) - \min_{x\in\mathcal{X}} f(x) \leq \frac{B_\Phi(x^*, y^1) + \frac{1}{2\mu}\sum_{k=1}^{\ell} \eta_k^2 \|g^k\|_*^2}{\sum_{k=1}^{\ell} \eta_k},$$

which is identical to the rate estimate for Bregman proximal gradient method presented in [18, Theorem 4.1]. Note that our analysis shows that the same estimate holds also for the explicit mirror descent method without the assumption Φ is continuously differentiable, whereas the analysis in [18] holds for the mirror descent method under differentiability assumption. Choosing a suitable stepsize as described in [18, Proposition 4.1]:

$$\eta_k = \frac{\sqrt{2\mu B_\Phi(x^*, y^1)}}{Lk}, \quad \forall k \geq 1,$$

we obtain the following rate estimate (cp. [18, Theorem 4.2]):

$$\min_{1\leq \ell \leq k} f(x^\ell) - \min_{x\in\mathcal{X}} f(x) \leq \frac{L\sqrt{2\mu B_\Phi(x^*, y^1)}}{\sqrt{\mu k}}.$$

Implicit Mirror Descent & Bregman Proximal Point Methods

By our discussions in Section 8.2, the following inequality holds for the implicit mirror descent and Bregman proximal point methods:

$$\eta_k(f(x^{k+1}) - f(x^*)) \leq B_\Phi(x^*, y^k) - B_\Phi(x^*, y^{k+1}) - B_\Phi(x^{k+1}, y^k),$$

Summing this inequality for $k = 0, \dots, \ell - 1$, we get

$$\sum_{k=0}^{\ell-1} \eta_k(f(x^{k+1}) - f(x^*)) \leq B_\Phi(x^*, y^0) - B_\Phi(x^*, y^\ell) - \sum_{k=0}^{\ell-1} B_\Phi(x^{k+1}, y^k).$$

Lower bounding each $f(x^k)$ by $\min_k f(x^k)$ in the left-hand side and noticing that $B_\Phi(x^*, y^{\ell+1})$ and $B_\Phi(x^{k+1}, y^k)$ are non-negative, we obtain

$$\min_{1\leq k\leq \ell} f(x^k) - \min_{x\in\mathcal{X}} f(x) \leq \frac{B_\Phi(x^*, y^0)}{\sum_{k=0}^{\ell-1}\eta_k},$$

which is precisely the rate estimated obtained in [49, Theorem 3.4] for the Bregman proximal point method and in [81, Lemma 4.1] for the implicit mirror descent method.

Dual Averaging Method

By our discussions in Section 8.2, the following inequality holds for the dual averaging method:

$$\eta_k(f(x^k)-f(x^*)) \leq \langle d^k, x^{k+1} - x^k\rangle + \beta_k\big[B_\Phi(x^*, y^k/\beta_k) - B_\Phi(x^*, y^{k+1}/\beta_k) - B_\Phi(x^{k+1}, y^k/\beta_k)\big],$$

where $d^k = -\eta_k g^k$ and $g^k \in \partial f(x^k)$. Using arithmetic-geometric mean inequality on the first term, we obtain

$$\eta_k(f(x^k) - f(x^*)) \le \frac{\eta_k^2}{2\mu\beta_k}\|g^k\|_*^2 + \frac{\mu\beta_k}{2}\|x^{k+1} - x^k\|^2 \\ + \beta_k\big[B_\Phi(x^*, y^k/\beta_k) - B_\Phi(x^*, y^{k+1}/\beta_k) - B_\Phi(x^{k+1}, y^k/\beta_k)\big].$$

Using Lemma 7.3 with $x_1 = x^{k+1}$, $x^2 = x^k$ and $y^2 = y^k/\beta_k$, we can observe that $\frac{\mu}{2}\|x^{k+1} - x^k\|^2 - B_\Phi(x^{k+1}, y^k/\beta_k) \le 0$, which substituting into the above inequality yields

$$\begin{aligned}\eta_k(f(x^k) - f(x^*)) &\le \frac{\eta_k^2}{2\mu\beta_k}\|g^k\|_*^2 + \beta_k\big[B_\Phi(x^*, y^k/\beta_k) - B_\Phi(x^*, y^{k+1}/\beta_k)\big] \\ &= \frac{\eta_k^2}{2\mu\beta_k}\|g^k\|_*^2 + \beta_k\Phi^*(y^k/\beta_k) - \beta_k\Phi^*(y^{k+1}/\beta_k) - \langle x^*, y^k\rangle + \langle x^*, y^{k+1}\rangle.\end{aligned}$$

Summing this inequality for $k = 0, \ldots, \ell$, we get

$$\sum_{k=0}^{\ell}\eta_k(f(x^k) - f(x^*)) \le \frac{1}{2\mu}\sum_{k=0}^{\ell}\frac{\eta_k^2}{\beta_k}\|g^k\|_*^2 - \langle x^*, y^0\rangle + \langle x^*, y^{\ell+1}\rangle \\ + \beta_0\Phi^*(y^0/\beta_0) - \beta_\ell\Phi^*(y^{\ell+1}/\beta_\ell) - \sum_{k=0}^{\ell-1}(\beta_{k+1} - \beta_k)\,\Phi(x^{k+1}).$$

where the inequality follows by

$$\begin{aligned}\beta_{k+1}\Phi^*(y^{k+1}/\beta_{k+1}) - \beta_k\Phi^*(y^{k+1}/\beta_k) &= \langle x^{k+1}, y^{k+1}\rangle - \beta_{k+1}\Phi(x^{k+1}) - \beta_k\Phi^*(y^{k+1}/\beta_k) \\ &\le -(\beta_{k+1} - \beta_k)\,\Phi(x^{k+1}),\end{aligned}$$

which follows by the Fenchel's inequality. Assuming that $\{\beta_k\}$ is a non-decreasing sequence, we obtain

$$\sum_{k=0}^{\ell}\eta_k(f(x^k) - f(x^*)) \le \beta_0\,B_\Phi(x^*, y^0/\beta_0) + \frac{1}{2\mu}\sum_{k=0}^{\ell}\frac{\eta_k^2}{\beta_k}\|g^k\|_*^2,$$

which is identical to [107, Theorem 1].

8.3.3 Convergence Analysis for Relatively Smooth Optimization Problems

In this section, we consider a constrained convex optimization problem of the form

$$\min_{x \in \mathcal{X}} f(x), \tag{8.3.3}$$

where $\mathcal{X}$ is a nonempty closed convex subset of $\mathcal{E}$ and $f : \mathcal{E} \to \mathbb{R}$ is a continuously differentiable convex function. We assume Φ is continuously differentiable on $\mathcal{X}$ for the primal methods and Φ is assumed to satisfy the conditions in Proposition 7.1 for the dual methods. We also assume that f is L-smooth relative to Φ, i.e., $\exists L > 0$ such that

$$f(x) \leq f(u) + \langle \nabla f(u), x - u \rangle + L B_\Phi(x, y), \quad \forall x, u \in \mathcal{X} \text{ and } \forall y \in \partial \Phi(u). \tag{8.3.4}$$

In the remainder of this section, we apply the universal convergence recipe presented in Section 8.2 to relatively smooth problems.

Explicit Mirror Descent & Bregman Proximal Gradient Methods

By our discussions in Section 8.2, the following inequality holds for the explicit mirror descent and Bregman proximal gradient methods:

$$\eta_k(f(x^k) - f(x^*)) \leq \langle d^k, x^{k+1} - x^k \rangle + B_\Phi(x^*, y^k) - B_\Phi(x^*, y^{k+1}) - B_\Phi(x^{k+1}, y^k),$$

where $d^k = -\eta_k \nabla f(x^k)$. Using the relative smoothness assumption (8.3.4) with $x = x^{k+1}$, $u = x^k$ and $y = y^k$, the above inequality can be equivalently written as follows (cp. [15, Lemma 5]):

$$\eta_k(f(x^{k+1}) - f(x^*)) \leq B_\Phi(x^*, y^k) - B_\Phi(x^*, y^{k+1}) - (1 - \eta_k L) B_\Phi(x^{k+1}, y^k).$$

Let $\eta_k = 1/(2L)$ for all k such that the last term above guarantees $\{f(x^k)\}$ is a non-increasing sequence converging to $\min_{\mathcal{X}} f$. Then, we have

$$f(x^{k+1}) - f(x^*) \leq 2L(B_\Phi(x^*, y^k) - B_\Phi(x^*, y^{k+1})).$$

Summing this inequality for $k = 0, \ldots, \ell - 1$, we get

$$\sum_{k=0}^{\ell-1}(f(x^{k+1}) - f(x^*)) \leq 2L(B_\Phi(x^*, y^0) - B_\Phi(x^*, y^\ell)).$$

Observing $B_\Phi(x^*, y^\ell) \geq 0$ and applying Jensen's inequality, we obtain

$$f(x^k) - f(x^*) \leq \frac{2L}{k} B_\Phi(x^*, y^0),$$

which is identical to [15, Theorem 1] for the Bregman proximal gradient method and [93, Theorem 3.2] for the explicit mirror descent method (note that the method we call explicit mirror descent is called dual averaging in [93]). It is important to highlight that [15] and [93] assume Φ is continuously differentiable on $\mathcal{X}$, whereas Φ is non-differentiable in our analysis.

Implicit Mirror Descent & Bregman Proximal Point Methods

By our discussions in Section 8.2, the following inequality holds for the implicit mirror descent and Bregman proximal point methods:

$$\eta_k(f(x^{k+1}) - f(x^*)) \leq B_\Phi(x^*, y^k) - B_\Phi(x^*, y^{k+1}) - B_\Phi(x^{k+1}, y^k),$$

Choosing a constant stepsize $\eta_k = \eta$ and summing this inequality for $k = 0, \ldots, \ell - 1$, we get

$$\eta \sum_{k=0}^{\ell-1}(f(x^\ell) - f(x^*)) \leq B_\Phi(x^*, y^0) - B_\Phi(x^*, y^\ell) - \sum_{k=0}^{\ell-1} B_\Phi(x^{k+1}, y^k).$$

Observing $\{f(x^k)\}$ is a non-increasing sequence and applying Jensen's inequality, we obtain

$$f(x^k) - f(x^*) \leq \frac{1}{\eta k} B_\Phi(x^*, y^0),$$

Dual Averaging Method

By our discussions in Section 8.2, the following inequality holds for the dual averaging method:

$$\eta_k(f(x^k) - f(x^*)) \leq \langle d^k, x^{k+1} - x^k \rangle + \beta_k \big[B_\Phi(x^*, y^k/\beta_k) - B_\Phi(x^*, y^{k+1}/\beta_k) - B_\Phi(x^{k+1}, y^k/\beta_k)\big],$$

where $d^k = -\eta_k \nabla f(x^k)$. Using the relative smoothness assumption (8.3.4) with $x = x^{k+1}$, $u = x^k$ and $y = y^k$, the above inequality can be equivalently written as follows:

$$\eta_k(f(x^{k+1}) - f(x^*)) \leq \beta_k[B_\Phi(x^*, y^k/\beta_k) - B_\Phi(x^*, y^{k+1}/\beta_k)] - (\beta_k - \eta_k L) B_\Phi(x^{k+1}, y^k/\beta_k).$$

Let $\eta_k = 1/(2L)$ for all k and $\{\beta_k\}$ be a non-decreasing sequence with $\beta_0 > 1/2$ such that the last term above guarantees $\{f(x^k)\}$ is a non-increasing sequence converging to $\min_{\mathcal{X}} f$. Then, we have

$$f(x^{k+1}) - f(x^*) \leq 2L\beta_k[B_\Phi(x^*, y^k/\beta_k) - B_\Phi(x^*, y^{k+1}/\beta_k)] - L(2\beta_k - 1) B_\Phi(x^{k+1}, y^k/\beta_k).$$

Summing this inequality for $k = 0, \ldots, \ell - 1$ and using similar tricks to our analysis in the previous section, we get

$$\sum_{k=0}^{\ell-1} (f(x^{k+1}) - f(x^*)) \leq 2L\beta_0\, B_\Phi(x^*, y^0/\beta_0),$$

which by applying Jensen's inequality yields

$$f(x^k) - f(x^*) \leq \frac{2L\beta_0}{k} B_\Phi(x^*, y^0/\beta_0).$$

8.4 Applications

In this section, we consider a few celebrated problems in the literature and specify the mirror descent method applied to these problems. We establish that the linearized Bregman iterative method [152] and the singular value thresholding method [43] are instances of the explicit mirror descent method, in Sections 8.4.1 and 8.4.2, respectively.

8.4.1 Sparse Recovery Problem

Here we consider the setting:

$$f(x) = \frac{1}{2}\|Ax - b\|_2^2, \quad \mathcal{X} = \mathbb{R}^n \quad \text{and} \quad \Phi(x) = \|x\|_1 + \frac{\lambda}{2}\|x\|_2^2.$$

As we have discussed in Theorem 7.7, when applied to solve $\min_{\mathcal{X}} f$, the mirror descent method returns the solution to the following problem

$$\begin{aligned} \min \quad & \|x\|_1 + \frac{\lambda}{2}\|x\|_2^2, \\ \text{s.t.} \quad & Ax = b, \end{aligned}$$

provided that $y^0 = 0$. This problem is used as a surrogate to the basis pursuit problem ($\lambda = 0$) and one of the most celebrated solvers is the linearized Bregman iterative method [152]. The linearized Bregman iterative method is designed to minimize $\mu\|u\|_1 + \frac{1}{2\delta}\|u\|_2^2$ subject to $Au = b$ and consists of the following iterations:

$$v^{k+1} = v^k + A^\top(b - Au^k) \tag{8.4.2a}$$

$$u^{k+1} = \delta\, T_\mu(v^{k+1}), \tag{8.4.2b}$$

where

$$T_\mu(v) = [t_\mu(v_1), \ldots, t_\mu(v_n)]^\top$$

is the soft thresholding operator given by

$$t_\mu(\xi) = \begin{cases} 0 & \text{if } |\xi| \le \mu, \\ \operatorname{sgn}(\xi)(|\xi| - \mu) & \text{if } |\xi| > \mu. \end{cases}$$

Compare the update rule of linearized Bregman iterations (8.4.2) with the update rule of the explicit mirror descent method:

$$x^k = \nabla\Phi^*(y^k), \tag{8.4.3a}$$

$$y^{k+1} = y^k - \eta A^\top(Ax^k - b). \tag{8.4.3b}$$

Indeed, it can be observed that these two methods are equivalent as we highlight below.

Proposition 8.3. *Let (x^k, y^k) denote the primal and dual sequences generated by the explicit mirror descent method (8.4.3) and (u^k, v^k) denote the primal and dual sequences generated by the linearized Bregman iterative method (8.4.2). When $\eta = 1/\mu$ and $\lambda = 1/(\mu\delta)$, the iterates generated by these two methods are equivalent, i.e., $x^k = u^k$ and $y^k = v^k/\mu$ for all k.*

Proof We begin the proof by computing $\nabla\Phi^*$. The subdifferential of Φ at x is given by $\partial\Phi(x) = \lambda x + \operatorname{Sgn}(x)$. Therefore, for any $y \in \lambda x + \operatorname{Sgn}(x)$, we have $x = \nabla\Phi^*(y)$, i.e.,

$$\nabla\Phi^*(y) = \frac{1}{\lambda} T_1(y).$$

Observe that $y^k = v^k/\mu$ for all k provided that $x^k = u^k$. We prove this by induction: $y^0 = v^0 = 0$ by the initialization and suppose these relations hold for some k. Then, we have

$$T_1(y^k) = T_1(v^k/\mu) = \frac{1}{\mu} T_\mu(v^k).$$

Consequently, we obtain

$$x^k = \nabla\Phi^*(y^k) = \frac{1}{\lambda\mu} T_\mu(v^k).$$

As $\lambda = 1/(\mu\delta)$, we get $x^k = u^k$. Since we have $y^k = v^k/\mu$ by the induction step, we obtain

$$y^{k+1} = \frac{v^k}{\mu} - \frac{1}{\mu} A^\top (Au^k - b) = \frac{v^{k+1}}{\mu},$$

which conclude the proof. □

8.4.2 Low-Rank Recovery Problem

Here we consider the setting:

$$f(x) = \frac{1}{2}\|\mathcal{A}X - b\|_2^2, \quad \mathcal{X} = \mathbb{R}^{n\times m} \quad \text{and} \quad \Phi(x) = \|X\|_* + \frac{\lambda}{2}\|X\|_{\mathrm{F}}^2,$$

where $\mathcal{A}$ is a linear operator acting on the space of $n \times m$ matrices and $b \in \mathbb{R}^p$. As we have discussed in Theorem 7.7, when applied to solve $\min_{\mathcal{X}} f$, the mirror descent method returns the solution to the following problem

$$\begin{aligned} \min \quad & \|X\|_* + \frac{\lambda}{2}\|X\|_{\mathrm{F}}^2 \\ \text{s.t.} \quad & \mathcal{A}X = b, \end{aligned}$$

provided that $y^0 = 0$. This problem is used as a surrogate to the low-rank matrix recovery problem ($\lambda = 0$) and one of the most celebrated solvers is the singular value thresholding method [43], which is an extension of the linearized Bregman iterative method [152]. The linearized Bregman iterative method is designed to minimize $\tau\|W\|_1 + \frac{1}{2}\|W\|_2^2$ subject to $\mathcal{A}W = b$ and consists of the following iterations:

$$W^k = D_\tau(\mathcal{A}^\top z^k), \tag{8.4.5a}$$

$$z^{k+1} = z^k + \delta(b - \mathcal{A}W^k), \tag{8.4.5b}$$

where similar to the previous section

$$D_\tau(W) = U D_\tau(\Sigma) V^\top, \quad \text{where} \quad D_\tau(\Sigma) = \operatorname{diag}(\{\sigma_i - \tau\}_+),$$

and $t_+ = \max(0, t)$ is the positive part of t. Compare the update rule of the singular value thresholding method (8.4.5) with the update rule of the explicit mirror descent method:

$$X^k = \nabla \Phi^*(Y^k), \tag{8.4.6a}$$

$$Y^{k+1} = Y^k - \eta \, \mathcal{A}^\top (\mathcal{A} X^k - b). \tag{8.4.6b}$$

Indeed, it can be observed that these two methods are equivalent as we highlight below.

Proposition 8.4. *Let (X^k, Y^k) denote the primal and dual sequences generated by the explicit mirror descent method (8.4.6) and (W^k, z^k) denote the primal and dual (in the Lagrange sense) sequences generated by the linearized Bregman iterative method (8.4.5). When $\eta = \delta/\tau$ and $\lambda = 1/\tau$, the iterates generated by these two methods are equivalent, i.e., $X^k = W^k$ and $Y^k = \mathcal{A}^\top z^k/\tau$ for all k.*

Proof We begin the proof by computing $\nabla\Phi^*$. The subdifferential of Φ at $X = U\Sigma V^\top$ is given by

$$\partial \Phi(X) = \lambda X + \{UV^\top + S : X \text{ and } S \text{ have orthogonal row/column spaces and } \|S\| \le 1\},$$

where $\|\cdot\|$ denotes the spectral norm. Therefore, similar to the proof of Proposition 8.3, we have

$$\nabla \Phi^*(Y) = \frac{1}{\lambda} D_1(Y).$$

Observe that $Y^k = \mathcal{A}^\top z^k/\tau$ for all k provided that $X^k = W^k$. We prove this by induction: $Y^0 = \mathcal{A}^\top z^0 = 0$ by the initialization and suppose these relations hold for some k. Then, we have

$$D_1(Y^k) = D_1(\mathcal{A}^\top z^k/\tau) = \frac{1}{\tau} D_\tau(\mathcal{A}^\top z^k).$$

Consequently, we obtain

$$X^k = \nabla\Phi^*(Y^k) = D_\tau(\mathcal{A}^\top z^k) = W^k.$$

Since we have $Y^k = \mathcal{A}^\top z^k/\tau$ by the induction step, we obtain

$$Y^{k+1} = \frac{\mathcal{A}^\top z^k}{\tau} - \frac{\delta}{\tau}\mathcal{A}^\top(\mathcal{A}W^k - b) = \mathcal{A}^\top z^{k+1}/\tau,$$

which concludes the proof. □

8.5 Discussion

In this chapter, we presented a unified framework for explicit and implicit mirror descent, dual averaging, Bregman proximal gradient and Bregman proximal point methods. Our main aim in this chapter was to clarify the assumptions needed for each method to be well-defined, so that our discussion in the following chapter would be easier to comprehend. In doing so, we presented a universal convergence recipe for the aforementioned methods by characterizing the function value suboptimality as a composition of approximation error, update rule optimality and function value improvement. We also established the equivalence between certain celebrated optimization methods and mirror descent. Our results are more comprehensive than the existing studies in the literature under milder assumptions.

Chapter 9

Generalized Mirror Descent Methods

In this chapter, we study mirror descent methods with non-strictly convex and non-differentiable distance generating functions, which we call *generalized mirror descent methods.* Recall that the mirror descent dynamics is given by:

$$\dot{y}(t) = -\nabla f(\nabla \Phi^*(y(t))), \tag{9.0.1a}$$

$$y(0) = y^0 \in \mathcal{E}^*. \tag{9.0.1b}$$

As we discussed in Section 7.2, we require the following conditions on Φ to have a well-defined dynamics (9.0.1):

1. Φ is essentially strictly convex, so that Φ^* is differentiable on its domain.

2. Φ is supercoercive, so that $\operatorname{dom} \nabla \Phi^* = \mathcal{E}^*$.

3. $\operatorname{dom} \Phi = \mathcal{X}$, so that $\operatorname{rge} \nabla \Phi^* \subseteq \mathcal{X}$.

Our main purpose in this section is to relax the first two conditions and study the resulting dynamics. This corresponds to extending the mirror descent method to non-smooth geometries, which to our knowledge is not studied in the literature before. Note that the third condition ensures that the iterates generated by the mirror descent method remains feasible and hence cannot be relaxed unless a separate projection step is incorporated into (9.0.1a), which we do not pursue here.

Outline

In Section 9.1, we consider relaxing only the first condition and in order to guarantee that the second condition still holds, we consider the case $\operatorname{dom}\Phi = \mathcal{X}$ is bounded. We illustrate that the resulting mirror descent dynamics are still well-defined and has solutions that satisfy the convergence guarantees enjoyed by the mirror descent differential equation (cp. Section 7.4). In Section 9.2, we relax the second condition as well, but in doing so we face the problem of having a differential inclusion with non-compact values, which may prohibit convergence. Thus, we focus on quadratic problems for which we handle this problem by the monotonicity of the mirror descent differential inclusion. We show that the resulting dynamics enjoy similar convergence rates to the mirror descent differential equation. In Section 9.3, we discuss a few methods to obtain discrete-time solutions to the mirror descent differential inclusion. In Section 9.4, we illustrate that in certain examples the trajectory of the mirror descent method can be efficiently recovered by a discrete-time method in finitely many iterations. We provide numerical experiments in Section 9.5 and conclude the chapter with certain remarks in Section 9.6

9.1 Relaxing Strict Convexity Condition

When the distance generating function Φ is non-strictly convex, the mirror descent dynamics have the following differential inclusion form:

$$\dot{y}(t) = -\nabla f(\partial\Phi^*(y(t))), \tag{9.1.1a}$$

$$y(0) = y^0 \in \mathcal{E}^*, \tag{9.1.1b}$$

where $\partial\Phi^* : \mathcal{E}^* \rightrightarrows \mathcal{X}$ is a set-valued map. As we discussed in the previous section, here we keep the supercoercivity condition by assuming $\operatorname{dom}\Phi = \mathcal{X}$ is bounded. For the mirror descent differential equation (9.0.1), we have only dealt with the convergence of the solution in the previous sections. However for the mirror descent differential inclusion (9.1.1), even the existence of a solution is not immediate. The simplest approach to tackle this issue is to reduce the corresponding differential inclusion to a differential equation. More specifically, we investigate if there exists a differential equation $\dot{y}(t) = -g(y(t))$ concealed in the differential inclusion (9.1.1) in the sense that $g(y) \in \nabla f(\partial\Phi^*(y))$ for every y. If it is so, we can conclude that a solution of the differential equation is a solution of the differential inclusion (9.1.1), which resolves the existence issue. This approach reduces to the so called *selection problem*, where one tries to select a single-valued map in the set-valued map $\partial\Phi^*$ such that the selection satisfies some regularity conditions such as continuity and measurability. We do not discuss the details of such selection rules, but for an interested reader, we refer to [10, Chapter 9] for a detailed treatment of this topic. Our approach is based on showing that $\nabla f \circ \partial\Phi^*$ is an upper semi-continuous map, which in turn implies the existence of a selection rule by the approximate selection theorem [10, Theorem 9.2.1]. Before discussing this method in detail, we first define what we mean by a solution. We require that a solution $y(\cdot)$ has to be *absolutely continuous*, i.e., $y(\cdot)$ should be the primitive of its derivative:

$$y(t) = y^0 + \int_0^t \dot{y}(s)\, ds.$$

In the following section, we show the existence of a solution to mirror descent differential inclusion (9.1.1). We then present the convergence of a solution and discuss how to discretize the mirror descent differential inclusion.

9.1.1 Existence of a Solution

We begin our discussion by showing that the set-valued map $\nabla f \circ \partial\Phi^*$ is upper semi-continuous. To this end, we first introduce the following lemma, which is a standard result on upper semi-continuous maps.

Lemma 9.1 ([9, Proposition 1.1.1])**.** *Suppose $S : \mathcal{A} \rightrightarrows \mathcal{B}$ and $T : \mathcal{B} \rightrightarrows \mathcal{C}$ are upper semi-continuous maps, then so is $T \circ S : \mathcal{A} \rightrightarrows \mathcal{C}$.*

Below, we conclude the upper semi-continuity of $\nabla f \circ \partial\Phi^*$ using Lemma 9.1.

Proposition 9.2. *Let Φ be a closed proper convex function with $\operatorname{dom}\Phi = \mathcal{X}$, where $\mathcal{X}$ is a non-empty bounded closed convex set. Then, $\nabla f \circ \partial\Phi^*$ is upper semi-continuous.*

Proof We begin the proof by showing that $\partial\Phi^*$ is bounded and closed. Since Φ is a closed proper convex function, we have $\operatorname{rge}\partial\Phi^* \subseteq \operatorname{dom}\Phi$, see e.g., [128, p. 227]. As $\operatorname{dom}\Phi = \mathcal{X}$ and $\mathcal{X}$ is bounded, then $\operatorname{rge}\partial\Phi^*$ is bounded. Next, we consider closedness of $\partial\Phi^*$. $\partial\Phi^*$ is closed (i.e., has closed graph) when Φ^* is closed proper convex. Φ^* is closed proper convex when Φ is proper convex, which is satisfied by the assumption. Consequently, $\partial\Phi^*$ is closed. Since $\partial\Phi^*$ is bounded and closed on $\mathcal{E}^*$, then it is upper semi-continuous by [11, Theorem 1.4.1]. It is easy to observe that for a single-valued map ∇f, the definitions of upper inverse and lower inverse reduce to the inverse of the function ∇f. Therefore, the continuity of ∇f trivially implies that ∇f is upper semi-continuous. The result then follows by Lemma 9.1 with $T = \nabla f$ and $S = \partial\Phi^*$. □

The upper semi-continuity of $\nabla f \circ \partial\Phi^*$ is promising to show the local existence of a solution to the differential inclusion (9.1.1). In particular, since $\nabla f \circ \partial\Phi^*$ is upper semi-continuous around a neighborhood of $y^0 \in \operatorname{int}(\operatorname{dom}\partial\Phi^*)$, then the approximate selection theorem [10, Theorem 9.2.1] implies the existence of a solution $y(\cdot)$ in a neighborhood of y^0 defined on $[0, T]$ for some $T > 0$. Extending this result to entire $\mathcal{E}^*$, we obtain the desired result. This results follows by [9, Theorems 2.1.3 & 2.1.4] and we present a proof sketch in Section 9.7.1.

Theorem 9.3. *Suppose $f : \mathcal{E} \to \mathbb{R}$ is a continuously differentiable convex function with bounded gradients on a non-empty bounded closed convex set $\mathcal{X}$. Let Φ be a closed proper* *convex function with* $\operatorname{dom}\Phi = \mathcal{X}$*. Then, there exists a solution* $y(\cdot)$ *defined on* $[0, +\infty)$ *to the differential inclusion* (9.1.1)*.*

9.1.2 Convergence of Solutions

Let y be an absolutely continuous solution defined on $[0, +\infty)$ to the differential inclusion (9.1.1). We next show that the corresponding primal trajectory enjoys the same sublinear rate of convergence as the mirror descent differential equation.

Theorem 9.4. *Suppose the conditions in Theorem 9.3 hold. Let* y *be a solution to* (9.1.1)*. If* $\dot{y}$ *is continuous almost everywhere, then there exists a selection* $x(t) \in \partial\Phi^*(y(t))$ *such that every accumulation point of* x *is contained in* $\mathcal{X}^* = \arg\min f$ *and*

$$f\left(\frac{1}{t}\int_0^t x(s)\, ds\right) - f(x^*) \leq \frac{B_{\Phi^*}(y^0, x^*)}{t}, \quad \textit{where } x^* \in \mathcal{X}^*.$$

Proof Let $\mathcal{N}$ be the set of times t such that y is not differentiable at t, y is not right continuous at t or $\dot{y}(t) \notin -\nabla f(\partial\Phi^*(y(t)))$. For every $t \notin \mathcal{N}$, let $x(t) \in \partial\Phi^*(y(t))$ such that $\dot{y}(t) = -\nabla f(x(t))$. Then, similar to our discussions in Chapter 7, we consider the following function for every $t \notin \mathcal{N}$:

$$V(y(t)) = B_{\Phi^*}(y(t), x^*) + \int_0^t (f(x(s)) - f(x^*))\, ds,$$

which is a Lyapunov function candidate for the mirror descent differential inclusion (9.1.1). Recall the definition of generalized Bregman divergence:

$$V(y(t)) = \Phi^*(y(t)) + \Phi(x^*) - \langle y(t), x^* \rangle + \int_0^t (f(x(s)) - f(x^*))\, ds.$$

In order to conclude $V(y(t))$ is a Lyapunov function, we need to show that it is non-increasing. To this end, we consider

$$
\begin{aligned}
\lim_{s\to 0^+} \frac{1}{s}(\Phi^*(y(t+s)) - \Phi^*(y(t))) &\le \lim_{s\to 0^+} \frac{1}{s}\langle x(t+s), y(t+s) - y(t)\rangle, \\
\lim_{s\to 0^+} \frac{1}{s}(\Phi^*(y(t)) - \Phi^*(y(t+s))) &\le \lim_{s\to 0^+} \frac{1}{s}\langle x(t), y(t) - y(t+s)\rangle,
\end{aligned}
$$

where the inequality follows since $x(t) \in \partial\Phi^*(t)$ and $x(t+s) \in \partial\Phi^*(t+s)$. Taking the limit as $s \to 0^+$ and noting that x is right continuous, we get

$$
\lim_{s\to 0^+} \frac{1}{s}\Phi^*(y(t+s)) - \Phi^*(y(t)) = -\langle x(t), \nabla f(x(t))\rangle.
$$

Therefore, the right derivative of V with respect to t is given by

$$
\begin{aligned}
\dot{V}(y(t)) &= f(x(t)) - f(x^*) - \langle x(t), \nabla f(x(t))\rangle \\
&= -D_f(x^*, x(t)),
\end{aligned}
$$

which is upper bounded by zero. Thus, applying Jensen's inequality to V and following similar steps to Theorem 7.4, we obtain the desired result. □

9.1.3 Discrete-Time Solutions

We next discuss how to discretize the mirror descent differential inclusion using explicit Euler method. In particular, we consider the stochastic recursive inclusions:

$$
y^{k+1} = y^k - \eta_k \nabla f(x^k) + \eta_k \xi^k, \tag{9.1.2}
$$

where $x^k \in \partial\Phi^*(y^k)$ are the selections from inclusions, η_k is the stepsize and $\{\xi^k\}_{k\ge 0}$ is an arbitrary noise sequence. In order to compare the discrete-time system with the continuous-time dynamics, we define $\bar{y}$ as the linear interpolation of the iterates y^k given by

$$\bar{y}(t) = y^k + \frac{t - t_k}{t_{k+1} - t_k}(y^{k+1} - y^k), \quad \text{for } t \in [t_k, t_{k+1}),$$

where $t_k = \sum_{j=0}^{k} \eta_j$ is the cumulative stepsizes that correspond to discretization times. In order to characterize the time derivative of this solution, we also define the following constant interpolation:

$$\bar{x}(t) = x^k, \quad \text{for } t \in [t_k, t_{k+1}).$$

According to these definitions, we have $\dot{\bar{y}}(t) = -\nabla f(\bar{x}(t))$ almost everywhere. We then have the following convergence theorem on the interpolated sequence, cp. [27, Theorem 5.2] and [57, Theorem 2], whose proof is omitted.

Theorem 9.5. *Let conditions of Theorem 9.3 and the following hold.*

1. *Iterates are bounded:* $\sup_k \|y^k\| < \infty$ *and* $\sup_k \|x^k\| < \infty$.

2. *Stepsize sequence satisfies:* $\sum_{k=0}^{\infty} \eta_k = \infty$ *and* $\sum_{k=0}^{\infty} \eta_k^2 < \infty$.

3. *Weighted noise sequence converges:* $\lim_{k\to\infty} \sum_{j=0}^{k} \eta_j \xi^j = \omega$ *for some* $\omega \in \mathcal{E}^*$.

Then, every limit point $y(\cdot)$ *of* $\{\bar{y}(t+\cdot),\, t \geq 0\}$ *in* $C([0,\infty), \mathcal{E}^*)$ *as* $t \to \infty$ *satisfies* (9.1.1) *almost everywhere, i.e.,*

$$y(t) = y^0 - \int_0^t \nabla f(x(s))\, ds,\ t \geq 0, \quad \textit{where } x(t) \in \partial\Phi^*(y(t)),\ \forall t.$$

The theorem above establishes the convergence of the dual variable and we refer to Section 9.3 for a further discussion on the convergence of the primal trajectory.

9.2 Relaxing Bounded Domain Condition

We next investigate relaxing the bounded domain condition in addition to the differentiability condition. This corresponds to mirror descent dynamics with non-strictly convex distance generating functions with domain $\mathcal{E}$, which implies Φ^* need not be differentiable on its domain and $\operatorname{dom} \partial\Phi^*$ is not necessarily entire $\mathcal{E}^*$. Keep in mind that we still impose the third condition: $\operatorname{dom} \Phi = \mathcal{X}$, which is fundamental to ensure that the solutions generated by the mirror descent dynamics are feasible. When $\operatorname{dom} \partial\Phi^* \neq \mathcal{E}^*$, we are not only concerned with the existence of a solution $y(\cdot)$ but also its *viability* as well, i.e., $y(t)$ should be contained in $\operatorname{dom} \partial\Phi^*$ for all $t \geq 0$.

Consider the set-valued map $\partial\Phi^*$ described in the previous section and suppose we fix a closed convex set $\mathcal{Y} = \operatorname{dom} \partial\Phi^*$ by setting $\partial\Phi^*(y) = \emptyset$ for every $y \notin \mathcal{Y}$. Then, it can be shown that the trajectories generated by (9.1.1) remain feasible under mild conditions [9, Theorem 5.2.7]. In particular, when $-\nabla f(\partial\Phi^*(y)) \subset T_{\mathcal{Y}}(y)$, where $T_{\mathcal{Y}}(y)$ denotes the tangent cone of $\mathcal{Y}$ at y, there exists a feasible solution to (9.1.1). However, we are principally interested with the primal trajectory and consequently Φ^* is often constructed via its convex conjugate Φ. Therefore, in general it does not make much sense to impose a restriction on $\operatorname{dom} \partial\Phi^*$ unless $\operatorname{dom} \partial\Phi^*$ is bounded as a consequence of the definition of Φ. Recall that $\operatorname{dom} \partial\Phi^*$ is bounded if and only if Φ contains a non-vertical half line. Let y denote the slope of this half line. Then, $\partial\Phi^*(y)$ is not compact and the existence theorems in the previous section does not apply. Therefore, without additional conditions on the objective function f, mirror descent differential inclusion does not necessarily have a solution. Therefore in the sequel, we consider the particular case, where f is a convex quadratic function, and study the corresponding mirror descent differential inclusion.

Our main focus in this section is on unconstrained quadratic problems:

$$f(x) = \frac{1}{2}\|Ax - b\|_2^2 \quad \text{and} \quad \mathcal{X} = \mathcal{E},$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. We consider mirror descent dynamics to solve $\min_{\mathcal{X}} f$ with a non-strictly convex non-differentiable distance generating function $\Phi : \mathcal{E} \to \mathbb{R}$. The corresponding mirror descent differential inclusion is then given by

$$\dot{y}(t) = -A^\top (A\,\partial\Phi^*(y(t)) - b), \tag{9.2.1a}$$

$$y(0) = y^0 \in \operatorname{dom}\partial\Phi^*. \tag{9.2.1b}$$

Here, we do not rule out the possibility that $\operatorname{dom}\partial\Phi^*$ is a bounded subset of $\mathcal{E}^*$. Indeed, this is not just a theoretical curiosity, but it is the case for many interesting problems. As an example, consider $\Phi(x) = \|x\|_1$ whose subdifferential is given by $\partial\Phi(x) = \{y \in \mathcal{E}^* : \|y\|_\infty \le 1, \langle y, x\rangle = \|x\|_1\}$. Since the range of $\partial\Phi$ is bounded by the unit ℓ_∞-ball, it follows that the domain of $\partial\Phi^*$ is bounded by the same ball (as $x \in \partial\Phi^*(y)$ is equivalent to $y \in \partial\Phi(x)$, for any $x \in \mathcal{E}$ and $y \in \mathcal{E}^*$).

9.2.1 Existence and Uniqueness of the Solution

It can be observed from (9.2.1a) that $\dot{y} \in \operatorname{rge} A^\top$. Therefore, if $y^0 \in \operatorname{dom}\partial\Phi^* \cap \operatorname{rge} A^\top$, we can consider a change of variables $y = A^\top z$, where $z \in \mathbb{R}^m$, and consider the equivalent differential inclusion:

$$\dot{z}(t) = -(A\,\partial\Phi^*(A^\top z(t)) - b), \tag{9.2.2a}$$

$$z(0) = z^0 \in \mathcal{E}_A^*, \tag{9.2.2b}$$

where $\mathcal{E}_A^* = \{z \in \mathbb{R}^m : A^\top z \in \operatorname{dom}\partial\Phi^*\}$ is the pre-image of $\operatorname{dom}\partial\Phi^*$ under the linear map $A^\top$ and $z^0 \in \mathcal{E}_A^*$ such that $y^0 = A^\top z^0$. We start studying this differential inclusion by showing that the right-hand side of (9.2.2a) is a maximal monotone operator.

Lemma 9.6. *Let $\Phi : \mathcal{E} \to \mathbb{R}$ be a closed proper convex function and suppose $0 \in \operatorname{ri}(\operatorname{dom}\partial\Phi^*)$. Then, $T : z \mapsto A(\partial\Phi^*(A^\top z(t))) - b$ is a maximal monotone map.*

Proof It is well-known (see e.g., [130, Theorem 12.17]) that subdifferential of a closed proper convex function is maximal monotone. Since Φ is assumed to be closed proper convex, then so is Φ^*, which implies $\partial\Phi^*$ is a maximal monotone map. Furthermore, maximality is preserved under linear transformations [130, Theorem 12.43]. In particular, $S(z) = A\,\partial\Phi^*(A^\top z)$ is maximal monotone provided that $\operatorname{rge} A^\top \cap \operatorname{ri}(\operatorname{dom}\partial\Phi^*) \neq \emptyset$. Since $0 \in \operatorname{rge} A^\top$ and $0 \in \operatorname{ri}(\operatorname{dom}\partial\Phi^*)$ by the assumption of the lemma, it follows that S is maximal monotone. As any constant map is trivially maximal monotone with domain $\mathbb{R}^m$, then $T(z) = S(z) - b$ is maximal monotone, see e.g., [130, Corollary 12.44]. □

As an immediate consequence of the maximal monotonicity of the right-hand side of (9.2.2a), the differential inclusion (9.2.2) has a unique solution by [9, Theorem 3.2.1]. Furthermore, this solution is called *slow solution* and is characterized by the smallest norm element of the set-valued map (9.2.2a). Throughout this section, we will assume that $\mathcal{E}$ is equipped with ℓ_2-norm, and for a given closed convex set $\mathcal{K}$, we let

$$m(\mathcal{K}) = \Pi_{\mathcal{K}}(0)$$

denote the smallest ℓ_2-norm element of $\mathcal{K}$. Then, a solution z to the differential inclusion $\dot{z} = F(z)$ is called *slow solution* if $\dot{z} = m(F(z))$ almost everywhere. By [9, Theorem 3.2.1], the unique solution to (9.2.2) is characterized by the corresponding slow solution as we highlight below, whose proof is omitted.

Theorem 9.7. *Suppose the conditions in Lemma 9.6 hold. Then, there exists a unique solution $z(\cdot)$ defined on $[0,\infty)$ to the differential inclusion (9.2.2), which is the slow solution, i.e.,*

$$\dot{z}(t) = m(-(A\,\partial\Phi^*(A^\top z(t)) - b)) \quad \textit{almost everywhere.} \tag{9.2.3}$$

Furthermore, this solution enjoys the following properties:

1. *$t \mapsto \|\dot{z}(t)\|_2$ is non-increasing.*

2. *Let z_1 and z_2 be the solutions issued from z_1^0 and z_2^0, respectively. Then, $\|z_1(t) - z_2(t)\|_2 \leq \|z_1^0 - z_2^0\|_2$ for all $t \geq 0$.*

3. For all $t \geq 0$, $\dot{z}(t)$ is continuous from the right and

$$\dot{z}(t) = \lim_{s \to 0^+} \frac{z(t+s) - z(t)}{s}.$$

9.2.2 Convergence of Solutions

Observe that (9.2.3) establishes a subgradient selection rule for $\partial \Phi^*$, which is called *minimal selection*. We denote this selection by $x(t)$ for all $t \geq 0$ since this corresponds to the primal trajectory we are interested in. More specifically, we let

$$x(t) \in \arg \min_{x \in \partial \Phi^*(A^\top z(t))} \|Ax - b\|_2, \quad \forall t \geq 0. \tag{9.2.4}$$

We then observe that

$$V(z(t)) = B_{\Phi^*}(A^\top z(t), x^*) + \int_0^t (f(x(s)) - f(x^*))\, ds$$

is a Lyapunov function for mirror descent dynamics. Recall the definition of generalized Bregman divergence:

$$V(z(t)) = \Phi^*(A^\top z(t)) + \Phi(x^*) - \langle A^\top z(t), x^* \rangle + \int_0^t (f(x(s)) - f(x^*))\, ds.$$

Consider the right-derivative of V with respect to time, which is well-defined since Φ^* is continuous, $\dot{z}(t)$ is right-continuous and consequently $f(x(t))$ is right-continuous (since f is a quadratic function, also compare with (9.2.4)):

$$\begin{aligned} \dot{V}(z(t)) &= \lim_{s \to 0^+} \frac{1}{s}(V(z(t+s)) - V(z(t))) \\ &= f(x(t)) - f(x^*) + \lim_{s \to 0^+} \frac{1}{s}(\Phi^*(A^\top z(t+s)) - \Phi^*(A^\top z(t)) - \langle Ax^*, z(t+s) - z(t) \rangle). \end{aligned} \tag{9.2.5}$$

Since Φ^* is convex, the following inequalities hold:

$$\begin{aligned}
\Phi^*(A^\top z(t+s)) - \Phi^*(A^\top z(t)) &\leq -\langle x(t+s), A^\top(z(t) - z(t+s))\rangle, \\
\Phi^*(A^\top z(t)) - \Phi^*(A^\top z(t+s)) &\leq -\langle x(t), A^\top(z(t+s) - z(t))\rangle,
\end{aligned}$$

as $x(t+s) \in \partial\Phi^*(A^\top z(t+s))$ and $x(t) \in \partial\Phi^*(A^\top z(t))$. After some elementary operations, these inequalities can be equivalently written as follows

$$\begin{aligned}
\Phi^*(A^\top z(t+s)) - \Phi^*(A^\top z(t)) &\leq -\langle Ax(t+s) - b, z(t) - z(t+s)\rangle - \langle b, z(t) - z(t+s)\rangle, \\
\Phi^*(A^\top z(t)) - \Phi^*(A^\top z(t+s)) &\leq -\langle Ax(t) - b, z(t+s) - z(t)\rangle - \langle b, z(t+s) - z(t)\rangle.
\end{aligned}$$

Taking the limit as $s \to 0^+$, we observe that

$$\begin{aligned}
\lim_{s\to 0^+} \frac{1}{s}\Phi^*(A^\top z(t+s)) - \Phi^*(A^\top z(t)) &\leq -\langle \dot{z}(t), \dot{z}(t)\rangle + \langle b, \dot{z}(t)\rangle, \\
\lim_{s\to 0^+} \frac{1}{s}\Phi^*(A^\top z(t)) - \Phi^*(A^\top z(t+s)) &\leq \langle \dot{z}(t), \dot{z}(t)\rangle - \langle b, \dot{z}(t)\rangle.
\end{aligned}$$

Since these inequalities provide tight lower and upper bounds, we have

$$\lim_{s\to 0^+} \frac{1}{s}\Phi^*(A^\top z(t+s)) - \Phi^*(A^\top z(t)) = -\|\dot{z}(t)\|_2^2 + \langle b, \dot{z}(t)\rangle.$$

Using this equation in (9.2.5), we obtain

$$\dot{V}(z(t)) = f(x(t)) - f(x^*) - \|\dot{z}(t)\|_2^2 - \langle Ax^* - b, \dot{z}(t)\rangle).$$

Since $\dot{z}(t) = -(Ax(t) - b)$, we then get

$$\dot{V}(z(t)) = f(x(t)) - f(x^*) - \langle (Ax(t) - b) - (Ax^* - b), Ax(t) - b\rangle).$$

Rearranging terms, we can observe that

$$\dot{V}(z(t)) = f(x(t)) - f(x^*) - \langle x(t) - x^*, A^\top(Ax(t) - b)\rangle).$$

Since $\nabla f(x(t)) = A^\top(Ax(t) - b)$, we conclude that $\dot{V}(z(t)) = -D_f(x^*, x(t))$. Thus, we arrive at the following result.

Theorem 9.8. *Suppose the conditions in Lemma 9.6 hold and let*

$$x(t) \in \arg \min_{x \in \partial\Phi^*(A^\top z(t))} \|Ax - b\|_2, \quad \forall t \geq 0. \tag{9.2.6}$$

denote a primal trajectory corresponding to the minimal selection rule presented in Theorem 9.7. Then, every accumulation point of $x(t)$ is contained in $\mathcal{X}^* = \arg\min f$. Furthermore, the following convergence rate holds:

$$f\left(\frac{1}{t}\int_0^t x(s)\, ds\right) - f(x^*) \leq \frac{B_{\Phi^*}(A^\top z^0, x^*)}{t}, \quad \textit{where } x^* \in \mathcal{X}^*.$$

An immediate consequence of this theorem is that the mirror descent differential inclusion is implicitly biased towards minimum divergence solutions, similar to Theorem 7.7.

Corollary 9.9. *Suppose the conditions in Lemma 9.6 hold and let $x(t)$ denote a primal trajectory given by (9.2.6) corresponding to the minimal selection rule presented in Theorem 9.7. Then, every accumulation point $\bar{x}$ of $x(t)$ satisfies*

$$\bar{x} \in \arg \min_{\substack{x \in \mathcal{E} \\ Ax = Pb}} B_\Phi(x, A^\top z^0), \tag{9.2.7}$$

where $P = A(A^\top A)^{-1}A^\top$ is the projection matrix onto $\operatorname{rge} A$.

Proof The following KKT conditions are satisfied at any optimal solution x^* to (9.2.7):

$$\begin{aligned} &\text{(feasibility)} && Ax^* = Pb, \\ &\text{(stationarity)} && \exists \lambda^* : A^\top z^0 + A^\top \lambda^* \in \partial\Phi(x^*). \end{aligned}$$

As we have shown in Theorem 9.8 (and by LaSalle's invariance principle), any accumulation point $\bar{x}$ of $x(t)$ is contained in $\mathcal{X}^*$, i.e., satisfies $\nabla f(\bar{x}) = A^\top(A\bar{x} - b) = 0$, which implies that $A\bar{x} = Pb$. In order to verify the stationarity condition, we integrate the slow solution (9.2.3):

$$z(t) - z^0 = -\int_0^t (Ax(s) - b)\, ds,$$

where $x(t)$ is given by (9.2.6) (note that $x(t)$ need not be unique, whereas $\dot{z}(t) = -(Ax(s) - b)$ is). Let $\{x(t_k)\}$ denote the subsequence that converges to $\bar{x}$. Then, multiplying both sides by $A^\top$ from the left and taking the limit as $k \to \infty$, we get

$$A^\top z^0 - \lim_{k\to\infty} \int_0^{t_k} A^\top(Ax(s) - b)\, ds \in \partial\Phi(\bar{x}).$$

Thus, $\bar{x}$ is an optimal solution to the problem (9.2.7). □

Remark 9.10. *Our discussions in this section trivially extend to the case $\mathcal{X}$ is bounded. In particular, observe that in Lemma 9.6, the condition $0 \in \mathrm{ri}(\mathrm{dom}\,\partial\Phi^*)$ holds since bounded $\mathcal{X}$ implies $\mathrm{dom}\,\partial\Phi^* = \mathcal{E}^*$. Furthermore, the convergence results (Theorems 9.7 and 9.8) readily extend to the case $\mathrm{dom}\,\partial\Phi^* = \mathcal{E}^*$.*

9.2.3 Discrete-Time Solutions

Finally, we study discretization methods applied to mirror descent differential inclusion (9.2.2) for quadratic problems. We consider the backward Euler method (since $\mathcal{E}_A^*$ may be bounded and a projection step is required), which yields the following discrete-time update:

$$z^{k+1} = z^k - \eta_k\,(Ax^{k+1} - b), \quad \text{where } x^{k+1} = \partial\Phi^*(A^\top z^{k+1}). \tag{9.2.8}$$

Similar to the previous section, our focus will be on approximate update rules. Therefore, we first introduce the following notation. Let $T : z \mapsto A\partial\Phi^*(A^\top z) - b$ denote the maximal monotone mapping we have described in Lemma 9.6. The proximal point algorithm in

(9.2.8) is based on the Minty characterization [102] that for each z and $\eta_k > 0$, there is a unique u such that $u - z \in -\eta_k T(u)$, or equivalently

$$z \in (1 + \eta_k T)(u).$$

Consequently, $P_k = (1 + \eta_k T)^{-1}$ is a single-valued mapping called *proximal mapping*. Using this definition, we can observe that (9.2.8) can be equivalently written as $z^{k+1} = P_k(z^k)$. We consider the case, where the proximal map is approximately calculated yielding the update rule

$$z^{k+1} = P_k(z^k) + \xi^k. \tag{9.2.9}$$

We then have the following convergence theorem on the discrete-time sequence, as an immediate consequence of [129, Theorem 1].

Theorem 9.11. *Let $\{z^k\}_{k\geq 0}$ be any sequence generated by the discrete-time update rule in (9.2.9). Assume $z^k \in \mathcal{E}_A^*$ for all $k \geq 0$, the error sequence satisfies $\sum_{k=0}^{\infty} \|\xi^k\|_2 < \infty$ and $b \in \operatorname{rge} A^\top$. Then, z^k converges weakly to a point z^* such that $0 \in T(z^*)$ as $k \to \infty$.*

9.3 Discussion on Discrete-Time Solutions

In this section, we present a detailed discussion on the discrete-time solutions to the mirror descent differential inclusion. Theorem 9.5 and Theorem 9.11 show that the dual trajectory y generated by particular discretizations discussed in the corresponding theorems converges to a fixed point of the differential inclusion. This is sufficient to show the convergence of the dual variables to a solution of the initial value problem (9.1.1), whereas the convergence of primal variables generated by some $x \in \partial\Phi^*(y)$ need not converge to a minimizer of f in certain applications. This behavior can be observed by inspecting the graph of $\partial\Phi$ and $\partial\Phi^*$ as we discuss below with an example.

Consider the setting $\mathcal{E} = \mathbb{R}$, $\mathcal{X} = [2, 2]$ and $\Phi(x) = |x|$. Graphs of $\partial\Phi$ and $\partial\Phi^*$ are depicted in Figure 9-1. From the figure on the right, we observe that we have $x = 0$ when

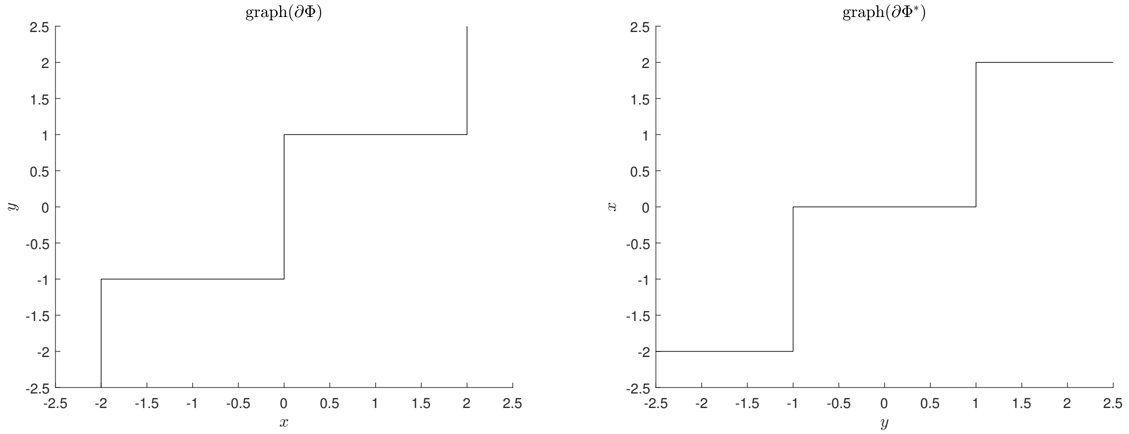

Figure 9-1: Graphs of $\partial\Phi$ and $\partial\Phi^*$ for $\Phi = |\cdot|$ and $\mathcal{X} = [-2, 2]$.

$y \in (-1, 1)$, $x = 2$ when $y > 1$ and $x = -2$ when $y < -1$. Therefore, the primal variable x takes only three different values almost everywhere for $y \in \mathbb{R}$. Suppose the minimizer of f is at $x^* = 1$, e.g., say $f(x) = (x-1)^2/2$. Then, unless $y = 1$ exactly, we cannot recover x^* without an error that is strictly bounded away from zero. The observation in this example holds for many interesting problems. In particular, the region of interest in the dual space (to generate a primal variable that minimizes f) corresponds to a set of measure zero. This issue can be handled in several ways as we discuss below.

The first approach is to solve a subproblem upon the termination of the discrete-time method. In particular, Theorems 9.5 and 9.11 guarantee that the dual trajectory converges weakly to a pre-image of $\mathcal{X}^*$. Let y^∞ denote the dual variable upon the termination of the discrete-time method. Then, we can solve the following subproblem:

$$x^\infty \in \arg \min_{\substack{x \in \partial\Phi^*(y) \\ \|y - y^\infty\|_2 \le \epsilon}} \|\nabla f(x)\|_2, \tag{9.3.1}$$

for some $\epsilon > 0$ to recover a minimizer of f. This approach makes sense if computing subdifferentials $\partial\Phi^*$ is relatively cheap and the problem (9.3.1) is not more expansive than solving the original problem. Note that the problem in (9.3.1) is a robust optimization problem variant and efficient methods for solving this problem exists in the literature [24].

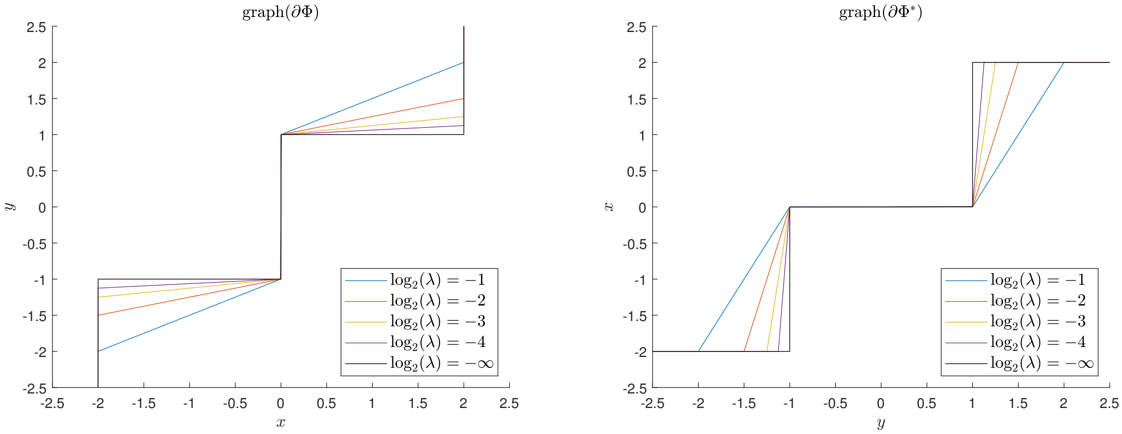

Figure 9-2: Graphs of $\partial\Phi$ and $\partial\Phi^*$ for $\Phi = |\cdot| + \frac{\lambda}{2}(\cdot)^2$ and $\mathcal{X} = [-2, 2]$.

The second approach is to regularize the distance generating function Φ and revert the problem to a differential equation. This resolves the issue as can be observed in Figure 9-2 for the example discussed above. For this example, the region of interest in the dual space $\mathbb{R}$ has measure λD, where D is the diameter of $\mathcal{X}$. This enables to use stepsizes that are proportional to λ by sacrificing from the sparsity induced by Φ, where the sacrifice also increases proportional to λ. As discussed in Section 8.4, this approach is taken in many state-of-the-art solvers since the resulting differential equation is easier to interpret and rate estimates can be obtained for sufficiently small stepsizes.

Another approach to handle the aforementioned issue is to consider ϵ-subdifferentials:

$$\partial_\epsilon \Phi(x) = \{y \in \mathcal{E}^* : \Phi(u) \geq \Phi(x) + \langle y, u - x \rangle - \epsilon, \ \forall u \in \mathcal{E}\}, \tag{9.3.2}$$

instead of regular subdifferentials. Figure 9-3 illustrates how using ϵ-subgradients result in obtaining a region of interest with strictly positive measure for the particular example discussed above. For this approach, the convergence of the resulting method can be analyzed using similar techniques to [127], which we do not discuss here. Suppose the discrete-time method terminates with the dual variable y^∞, then a primal solution can be found by

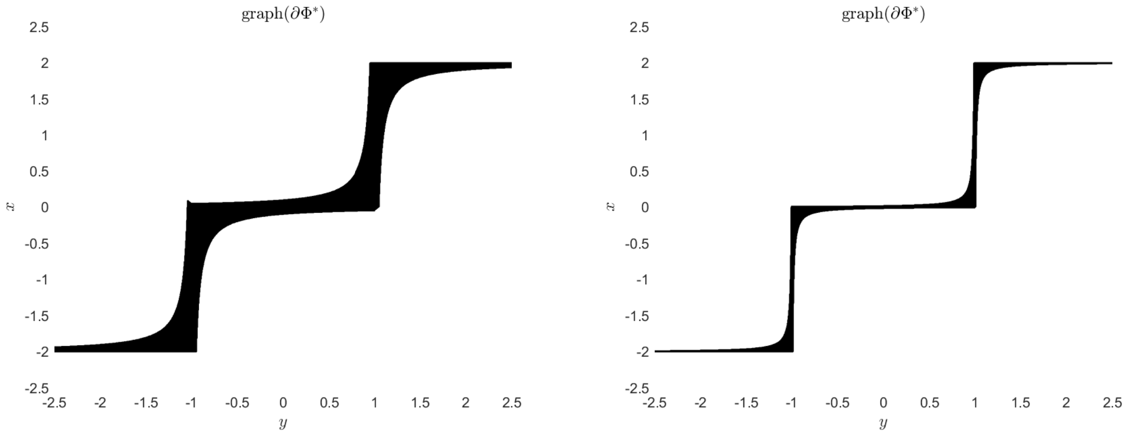

Figure 9-3: Graphs of $\partial_\epsilon \Phi^*$ for $\epsilon = 0.1$ (left) and $\epsilon = 0.02$ (right), where $\Phi = |\cdot|$ and $\mathcal{X} = [-2, 2]$.

solving a problem of the following type:

$$x^\infty \in \arg \min_{x \in \partial_\epsilon \Phi^*(y^\infty)} \|\nabla f(x)\|_2. \tag{9.3.3}$$

An advantage of this method over the one described in (9.3.1) is that the optimization space in (9.3.3) is explicitly defined by the choice of ϵ and using a sequence of non-increasing $\{\epsilon_k\}$'s naturally yield easier problems to solve.

A far more efficient approach is to track the trajectory of the differential inclusion explicitly. In particular, Theorem 9.11 shows that the solution to the mirror descent differential inclusion is uniquely determined by the minimum selection rule. Therefore, if we can trace the solution along the submanifolds where $\dot{z}$ (or $\dot{y}$) is continuous, then upon termination the final subgradient chosen according to the minimum selection rule becomes an optimal solution. In the following section, we illustrate this idea for the basis pursuit problem.

9.4 Application: Minimum ℓ_1-norm Solution to Linear Systems

The problem of finding the minimum ℓ_1-norm solution to an undetermined system of linear equations has attracted a lot of attention in optimization and signal processing communities. One of the main reasons for this interest is the fact that the minimum ℓ_1-norm solution is often the sparsest possible solution under certain conditions. More specifically, consider recovering an unknown signal $x_0 \in \mathbb{R}^n$, given a measurement vector $b \in \mathbb{R}^m$ and a sensing matrix $A \in \mathbb{R}^{m\times n}$ such that $b = Ax_0$. When x_0 is sufficiently sparse and the sensing matrix A is incoherent with the basis under which x_0 is sparse, then x_0 can be exactly recovered by computing the minimum ℓ_1-norm solution to this linear system, i.e., by the following problem:

$$\begin{aligned} \min \quad & \|x\|_1 && \text{(9.4.1a)} \\ \text{s.t.} \quad & Ax = b. && \text{(9.4.1b)} \end{aligned}$$

We find an optimal solution to (9.4.1) by solving the problem

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|Ax - b\|_2^2,$$

using the mirror descent method with $\Phi(x) = \|x\|_1$. As shown in Theorem 9.7, the unique solution to the mirror descent dynamics for this problem is given by

$$\dot{z}(t) = m(-(A\,\partial\Phi^*(A^\top z(t)) - b)), \tag{9.4.2}$$

and Corollary 9.9 guarantees that the limit points of the primal trajectory generated by the mirror descent method:

$$x(t) \in \arg\min_{x\in\partial\Phi^*(A^\top z(t))} \frac{1}{2}\|Ax - b\|_2^2 \tag{9.4.3}$$

are minimum ℓ_1-norm solutions among the least squares solutions.

We begin our discussion by observing that $\Phi^*(A^\top z) = \iota_{\mathcal{Z}}(z)$, where $\mathcal{Z} = \{z \in \mathbb{R}^m : \|A^\top z\|_\infty \le 1\}$. Then, letting a_i denote the i-th column of A, i.e., $A = [a_1, \ldots, a_n]$, the i-th coordinate of the subdifferential of Φ^* is given as follows

$$\partial_i \Phi^*(A^\top z) = \begin{cases} 0 & \text{if } |a_i^\top z| < 1, \\ \mathbb{R}_\ge & \text{if } a_i^\top z = 1, \\ \mathbb{R}_\le & \text{if } a_i^\top z = -1, \\ \emptyset & \text{if } |a_i^\top z| > 1. \end{cases}$$

For ease of presentation, we define the following functions:

$$\begin{aligned} \psi_i(z) &= a_i^\top z - 1, \\ \psi_{-i}(z) &= a_i^\top z + 1, \end{aligned}$$

which represent the boundary of $\mathcal{Z}$ as highlighted below:

$$\mathcal{Z} = \{z \in \mathbb{R}^m : \psi_i(z) \le 0 \text{ and } \psi_{-i}(z) \ge 0,\ \forall i \in [n]\}.$$

We finally let $I(z)$ denote the set of active constraints at solution z:

$$\mathcal{I}(z) = \mathcal{I}_+(z) \cup \mathcal{I}_-(z),$$

where

$$\begin{aligned} \mathcal{I}_+(z) &= \{i \in [n] \,:\, \psi_i(z) = 0\}, \\ \mathcal{I}_-(z) &= \{i \in [n] \,:\, \psi_{-i}(z) = 0\}. \end{aligned}$$

Throughout the paper, we let $\mathcal{I}(z) = \mathcal{I}_+(z) \cup \mathcal{I}_-(z)$ denote an active-set for simplicity, whereas in actual implementation we need to define both $\mathcal{I}_+(z)$ and $\mathcal{I}_-(z)$ separately. Every time it is stated that there exists an active set $\mathcal{I}(z)$, it should be understood that there exist two disjoint sets $\mathcal{I}_+(z)$ and $\mathcal{I}_-(z)$ that uniquely define the active-set $\mathcal{I}(z)$.

As we hinted in the previous section, our aim here is to trace the trajectory of the mirror descent differential inclusion. This can be done by tracking the set of active constraints at z. In particular, for two dual solutions z_1 and z_2, if their active sets are the same, then the corresponding primal solutions given by the minimum selection rule is the same as well. A closer look into this argument together with the right continuity of $\dot{z}$ as shown in Theorem 9.7 reveal that the mirror descent method applied to solve the quadratic problem $\frac{1}{2}\|Ax-b\|_2^2$ with distance generating function $\Phi(x) = \|x\|_1$ is equivalent to the gradient flow applied to the dual problem of (9.4.1). More specifically, the dual of the basis pursuit problem (9.4.1) is given by

$$\max \quad \langle b, z\rangle \tag{9.4.4a}$$

$$\text{s.t.} \quad \|A^\top z\|_\infty \leq 1, \tag{9.4.4b}$$

or equivalently $\min_{z\in\mathbb{R}^m} \langle b, z\rangle + \iota_Z(z)$. Since $\iota_Z(z) = \partial\Phi^*(A^\top z)$ as discussed above, a gradient descent dynamics applied to the dual problem yields the gradient flow presented in (9.4.2). For a more detailed description of this equivalence, we refer to Section 9.7.2. Since the boundary of ι_Z is polyhedral and $\dot{z}$ is right continuous as shown in Theorem 9.7, then addition operator allows us to slide on the subspace (submanifold in the general case) defined by the active constraints. Therefore, we can represent the trajectory of the mirror descent differential inclusion by the following discrete-time process:

$$z^{k+1} = z^k + \eta_k d^k, \tag{9.4.5}$$

where $\eta_k > 0$ is a stepsize and d^k is the update direction given by

$$d^k = b - \sum_{i \in \mathcal{I}(z^k)} x_i^k a_i, \tag{9.4.6}$$

and x_i^k, $i \in \mathcal{I}(z^k)$, is a solution to the following problem (cp. (9.4.3)):

$$\min \ \frac{1}{2}\left\| b - \sum_{i \in \mathcal{I}(z^k)} x_i a_i \right\|_2^2 \tag{9.4.7a}$$

$$\text{s.t.} \ \ x_i \geq 0, \quad \forall i \in \mathcal{I}_+(z^k), \tag{9.4.7b}$$

$$x_i \leq 0, \quad \forall i \in \mathcal{I}_-(z^k). \tag{9.4.7c}$$

It is important to highlight that there may be many optimal solutions of (9.4.7). However, they all yield to the same update direction given in (9.4.6), since (9.4.7) is essentially a Euclidean projection onto a closed convex set.

According to the update rule in (9.4.5), we can pick the stepsize η_k as large as possible until a new constraint becomes active (without violating the right continuity of $\dot{z}$). Before discussing how to pick this stepsize, we first make the following observation. For any $z^k \in \mathcal{Z}$ with the corresponding update direction d^k is defined according to (9.4.6), let $i \in \mathcal{I}(z^k)$ be an index such that $a_i^\top d^k \neq 0$. Then for any sufficiently small $\eta_k > 0$, we have $i \notin \mathcal{I}(z^k + \eta_k d^k)$, i.e., i leaves the active-set. We denote the set of variables that leave the active-set at iteration z^k by $\mathcal{B}(z^k) = \mathcal{B}_+(z^k) \cup \mathcal{B}_-(z^k)$, where

$$\mathcal{B}_+(z^k) = \left\{ i \in \mathcal{I}_+(z^k) \ : \ a_i^\top d^k \neq 0 \right\} \quad \text{and} \quad \mathcal{B}_-(z^k) = \left\{ i \in \mathcal{I}_-(z^k) \ : \ a_i^\top d^k \neq 0 \right\}. \tag{9.4.8}$$

Using these definitions, we next characterize the largest allowable stepsize in the following proposition. Note that the existence of a strictly positive stepsize trivially follows by the right continuity of $\dot{z}$.

Proposition 9.12. *Let $z^k \in \mathcal{Z}$ be a dual solution with the corresponding active-set $\mathcal{I}(z^k)$*

and the update direction d^k. Then, d^k is the update direction for all $z^k + \eta d^k$, where $0 \le \eta \le \bar{\eta}$ and

$$\bar{\eta} = \min_{i \notin \mathcal{I}(z^k)\setminus\mathcal{B}(z^k)} \frac{\operatorname{sgn}(a_i^\top d^k) - a_i^\top z}{a_i^\top d^k}. \tag{9.4.9}$$

Proposition 9.12 (whose proof is deferred to Section 9.7.3) establishes a method to track the continuous-time mirror descent solution given by (9.4.2). In particular, given a dual solution $z^k \in \mathcal{Z}$ with the corresponding set of active constraints $\mathcal{I}(z^k)$, we let x_i^k, $i \in \mathcal{I}(z^k)$, be a solution to the non-negative least squares problem presented in (9.4.7), construct the update direction as in (9.4.6) and perform the update (9.4.5) using the stepsize (9.4.9). At the next iterate z^{k+1}, we can recompute the active-set $\mathcal{I}(z^{k+1})$ from the scratch or update the active-set as follows

$$\mathcal{I}_+(z^{k+1}) = \big(\mathcal{I}_+(z^k) \setminus \mathcal{B}_+(z^k)\big) \cup \mathcal{F}_+(z^k) \quad \text{and} \quad \mathcal{I}_-(z^{k+1}) = \big(\mathcal{I}_-(z^k) \setminus \mathcal{B}_-(z^k)\big) \cup \mathcal{F}_-(z^k),$$

where $\mathcal{F}(z^k) = \mathcal{F}_+(z^k) \cup \mathcal{F}_-(z^k)$ denotes the set of constraints that will become active, i.e.,

$$\mathcal{F}_+(z^k) = \left\{ i \notin \mathcal{I}(z^k) \setminus \mathcal{B}(z^k) \,:\, \frac{1 - a_i^\top z^k}{a_i^\top d^k} = \eta^k \right\}, \tag{9.4.10a}$$

$$\mathcal{F}_-(z^k) = \left\{ i \notin \mathcal{I}(z^k) \setminus \mathcal{B}(z^k) \,:\, \frac{-1 - a_i^\top z^k}{a_i^\top d^k} = \eta^k \right\}. \tag{9.4.10b}$$

The resulting procedure is presented in Algorithm 6 and its finite-time convergence is proven in Theorem 9.13, whose proof is deferred to Section 9.7.4 in order not to distract the reader.

Theorem 9.13. *Algorithm 6 terminates with a pair of primal and dual optimal solutions in finitely many iterations provided that the system $Ax = b$ is realizable.*

Algorithm 6 presents that the basis pursuit problem can be solved by a sequence of nonnegative least squares problems (modulo a sign change for indices $I_-(z)$). However, a significant drawback of this approach is that the resulting algorithm is far more expensive than solving the basis pursuit problem directly: We started with an optimization problem

Algorithm 6: Discrete-Time Realization of Mirror Descent Dynamics

```
Initialize $z^0 = 0$ and $\mathcal{I}(z^0) = \emptyset$.
for $k \geq 0$ do
    Solve (9.4.7) to find $x_i^k$, $i \in \mathcal{I}(z^k)$.
    Compute the direction of update $d^k = b - \sum_{i \in \mathcal{I}(z^k)} x_i^k a_i$.
    Compute the stepsize $\eta_k$ by (9.4.9).
    Find the constraints that turn non-active $\mathcal{B}(z^k)$ by (9.4.8).
    Find the constraints that will become active $\mathcal{F}(z^k)$ by (9.4.10).
    Update the dual variable $z^{k+1} = z^k + \eta_k d^k$.
    Update the active-set $\mathcal{I}(z^{k+1}) = (\mathcal{I}(z^k) \setminus \mathcal{B}(z^k)) \cup \mathcal{F}(z^k)$.
end for
```

of size n with linear objective and linear equality constraints, and now we have a sequence of quadratic optimization problems with linear inequality constraints (where the length of the sequence and the size of the problem can be as large as n). In the following section, we discuss how to handle this issue iteratively, where the main idea is to use the solution of the previous subproblem as a warm start to the next problem. This corresponds to embedding another active-set algorithm into Algorithm 6 as detailed below.

9.4.1 Iteratively Solving the Nonnegative Least Squares Subproblems

In this section, we present an efficient method to solve the nonnegative least squares subproblems in Algorithm 6 by adding/removing variables to the active-set one at a time. Our approach is based on the active-set method of Lawson and Hanson [87] (given in Algorithm 7). This method can be viewed as a greedy active-set method, where at each outer iteration a variable is introduced into the active-set (denoted by $\mathcal{P}$, where $x_i > 0$, $i \in \mathcal{P}$ and $x_i = 0$, $i \in \mathcal{R}$) and at each inner iteration the first variable that turn negative (while introducing the new variable into the active-set) leaves the active-set. This algorithm converges in finitely many iterations as each inner iteration consists of at most $|\mathcal{P}|$ iterations and each outer iteration strictly decreases the objective $\frac{1}{2}\|Ax - b\|_2^2$, which implies an active-set cannot be visited twice.

Throughout this section, given any index set $\mathcal{I} \subset [n]$, for a vector $x \in \mathbb{R}^n$ we let $x_{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|}$ denote the subvector of x with indices $\mathcal{I}$, and for a matrix $A \in \mathbb{R}^{m \times n}$ we let $A_{\mathcal{I}} \in \mathbb{R}^{m \times |\mathcal{I}|}$ denote the submatrix of A with columns $\mathcal{I}$.

Algorithm 7: Lawson & Hanson Algorithm [87] for Nonnegative Least Squares Problem

```
Input: $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$.
Output: $x^* \in \arg\min_{x \geq 0} \frac{1}{2}\|Ax - b\|_2^2$.
Initialize $\mathcal{P} = \emptyset$, $\mathcal{R} = [n]$, $x = 0$, $d = A^\top(b - Ax)$.
while $\mathcal{R} \neq \emptyset$ and $d \not\leq 0$ do
  $j \in \arg\max_{i \in \mathcal{R}} d_i$.
  $\mathcal{P} \leftarrow \mathcal{P} \cup \{j\}$ and $\mathcal{R} \leftarrow \mathcal{R} \setminus \{j\}$.
  $s_{\mathcal{P}} = (A_{\mathcal{P}}^\top A_{\mathcal{P}})^{-1}(A_{\mathcal{P}})^\top b$.
  while $s_{\mathcal{P}} \not> 0$ do
    $\ell = \arg\min_{i \in \mathcal{P} : s_i \leq 0}\{x_i/(x_i - s_i)\}$.
    $\alpha = x_\ell/(x_\ell - s_\ell)$.
    $x \leftarrow x + \alpha(s - x)$.
    $\mathcal{P} \leftarrow \mathcal{P} \setminus \mathcal{S}$ and $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{S}$, where $\mathcal{S} = \{i \in \mathcal{P} : x_i = 0\}$.
  end while
  $x = s$ and $d = A^\top(b - Ax)$.
end while
```

Before describing how to incorporate Algorithm 7 into Algorithm 6, we first fix certain definitions to avoid any confusion as both algorithms are active-set algorithms. At iteration k, the active-set corresponding to Algorithm 6 is given by $\mathcal{I}^k = \mathcal{I}(z^k)$ and we let $\mathcal{Q}^k = [n]\backslash\mathcal{I}^k$ denote the *complementary-set*. When we apply Algorithm 7 to solve the nonnegative least squares problem over indices $\mathcal{I}^k$, it partitions $\mathcal{I}^k$ into two subsets: $\mathcal{P}^k$ and $\mathcal{R}^k$. We refer $\mathcal{P}^k$ as *active-set* since it denotes the strictly positive indices of x, and $\mathcal{R}^k$ as *nonstrict-set* since for any $i \in \mathcal{R}^k$ both $x_i = 0$ and $|a_t^\top x| = 1$ holds, i.e., it denotes the indices for which strict complementarity does not hold. To be able to keep track of the sign of each variable, we need to consider $\mathcal{P}^k = \mathcal{P}_+^k \cup \mathcal{P}_-^k$ and $\mathcal{R}^k = \mathcal{R}_+^k \cup \mathcal{R}_-^k$, whereas $\mathcal{Q}^k$ is sufficient to represent the remaining indices.

Suppose we apply Algorithm 7 to solve the subproblem in Algorithm 6 at iteration k. Then, Algorithm 7 returns a primal solution x^k and a direction of update d^k as well as an active-set $\mathcal{P}^k = \mathcal{P}^k_+ \cup \mathcal{P}^k_-$ and a nonstrict-set $\mathcal{R}^k = \mathcal{R}^k_+ \cup \mathcal{R}^k_-$, which we denote as the tuple $(x^k, d^k, \mathcal{P}^k, \mathcal{R}^k)$. As soon as a positive step in the direction d^k is taken in Algorithm 6, the constraints that become inactive is given by

$$\mathcal{B}^k_+ = \left\{ i \in \mathcal{R}^k_+ \ : \ a_i^\top d^k \neq 0 \right\} \quad \text{and} \quad \mathcal{B}^k_- = \left\{ i \in \mathcal{R}^k_- \ : \ a_i^\top d^k \neq 0 \right\}. \tag{9.4.11}$$

since for any $i \in \mathcal{P}^k$, $x_i^k \neq 0$ holds, which implies $a_i^\top d^k = 0$ by complementary slackness. Therefore, removing indices $\mathcal{B}^k$ from the complementary set $\mathcal{R}^k$ does not affect optimality. That is, the tuple $(x^k, d^k, \mathcal{P}^k, \mathcal{R}^k \setminus \mathcal{B}^k)$ is optimal for the corresponding nonnegative least squares problem over indices $\mathcal{P}^k \cup (\mathcal{R}^k \setminus \mathcal{B}^k)$. Thus, let us update the nonstrict-set and the complementary-set as follows: $\mathcal{R}^k \leftarrow \mathcal{R}^k \setminus \mathcal{B}^k$ and $\mathcal{Q}^k \leftarrow \mathcal{Q}^k \cup \mathcal{B}^k$. Then, we can pick the step size as follows

$$\eta^k = \min_{i \in \mathcal{Q}^k} \frac{\operatorname{sgn}(a_i^\top d^k) - a_i^\top z^k}{a_i^\top d^k}. \tag{9.4.12}$$

After performing the dual update $z^{k+1} = z^k + \eta^k d^k$, a certain set of indices become active at z^{k+1}, which is then given by

$$\mathcal{F}^k_+ = \left\{ i \in \mathcal{Q}^k \ : \ \frac{1 - a_i^\top z^k}{a_i^\top d^k} = \eta^k \right\} \quad \text{and} \quad \mathcal{F}^k_- = \left\{ i \in \mathcal{Q}^k \ : \ \frac{-1 - a_i^\top z^k}{a_i^\top d^k} = \eta^k \right\}. \tag{9.4.13}$$

Now, we arrive at the next dual solution z^{k+1} and we need to solve the nonnegative least squares problem over indices $\mathcal{P}^k \cup \mathcal{R}^k \cup \mathcal{F}^k$. We can solve this problem from the scratch by assigning $x = 0$, $d = b$, $\mathcal{P} = \emptyset$, and $\mathcal{R} = \mathcal{P}^k \cup \mathcal{R}^k \cup \mathcal{F}^k$ in Algorithm 7. However, a more efficient way is to initialize with $x = x^k$, $d = d^k$, $\mathcal{P} = \mathcal{P}^k$, and $\mathcal{R} = \mathcal{R}^k \cup \mathcal{F}^k$ as this is a valid active-set iterate for Algorithm 7. Since at each iteration of Algorithm 6 we expect that only a few indices become active, this initialization is expected to be very close to the solution of the corresponding nonnegative least squares problem, and only a few iterations (often just one!) of Algorithm 7 would yield the optimal solution. After we obtain the solution to the subproblem at iteration $k+1$, we end up at where we started in the previous iteration. The resulting algorithm is given in Algorithm 8.

Algorithm 8: An Efficient Discrete-Time Realization of Mirror Descent Dynamics

```
Initialize $z^0 = 0$, $x^0 = 0$, $d^0 = b$, $\mathcal{P}^0 = \emptyset$, $\mathcal{R}^0 = \emptyset$, $\mathcal{Q}^0 = [n]$, $k = 0$.
while $d^k \neq 0$ do
    Compute the step size $\eta^k$ (9.4.12) and find the constraints that become active $\mathcal{F}^k$ (9.4.13).
    Update the dual variable $z^{k+1} = z^k + \eta^k d^k$.
    Run Algorithm 7 initialized at $(x = x^k, d = d^k, \mathcal{P} = \mathcal{P}^k, \mathcal{R} = \mathcal{R}^k \cup \mathcal{F}^k)$, call the corresponding values upon termination as $(x^{k+1}, d^{k+1}, \mathcal{P}^{k+1}, \mathcal{R}^{k+1})$.
    Find the constraints that become non-active $\mathcal{B}^{k+1}$ (9.4.11) and update the sets $\mathcal{R}^{k+1} \leftarrow \mathcal{R}^{k+1} \setminus \mathcal{B}^{k+1}$ and $\mathcal{Q}^{k+1} \leftarrow \mathcal{Q}^{k+1} \cup \mathcal{B}^{k+1}$.
    Compute the direction of update $d^{k+1} = b - \sum_{i \in \mathcal{P}^{k+1}} x_i^{k+1} a_i$.
end while
```

9.4.2 A Numerically Efficient Implementation of Algorithm 8 via QR Decomposition

Even though the implementation in Algorithm 8 resolves the problem of solving the full nonnegative least squares problem from the scratch, it is not an efficient algorithm as each iteration of Algorithm 7 requires inverting the matrix $A_{\mathcal{P}}^\top A_{\mathcal{P}}$ for every active-set. Instead of performing this costly operation at each iteration, we can modify the matrix $R = [A,\, b]$ with elementary row operations to maintain a QR decomposition over the active-set of indices. More specifically, let $\mathcal{P}$ be an active set, then we want $R_{\mathcal{P}}$ to be the R-factor corresponding to the QR decomposition of $A_{\mathcal{P}}$. If such a relationship holds, then we can solve $A_{\mathcal{P}} x_{\mathcal{P}} = b$ for $x_{\mathcal{P}}$ via back substitution starting from the last index and working towards the first. Updating R when an index is added to $\mathcal{P}$ or removed from $\mathcal{P}$ can be efficiently performed via Givens rotation as follows. Suppose $\mathcal{P} = \{i_1, \ldots, i_p\}$, where the indices are written in increasing order, and assume that R is given such that $R_{\mathcal{P}}$ is the R-factor of the QR decomposition of $A_{\mathcal{P}}$.

- Suppose we want to remove i_ℓ from $\mathcal{P}$. To do so, we need to zero the unwanted subdiagonal elements R_{j,i_j}, $j = \ell+1, \dots, p$. This can be done by a sequence of Givens rotations: $R' = G_p \dots G_{\ell+1} R$, where G_j denote the rotation in planes (rows) $j-1$ and j such that when it multiplies a matrix R from the left, it sets the entry of R at the j-th row and i_j-th column to zero. Therefore, letting $\mathcal{P}' = (i_1, \dots, i_{\ell-1}, i_{\ell+1}, \dots, i_p)$, $R'_{\mathcal{P}'}$ is the R-factor of the QR-decomposition of $A_{\mathcal{P}'}$.

- Suppose we want to add i_ℓ to $\mathcal{P} = (i_1, \dots, i_{\ell-1}, i_{\ell+1}, \dots, i_p)$. To do so, we need to zero the unwanted subdiagonal elements R_{j,i_ℓ}, $j = \ell+1, \dots, m$. This can be done by a sequence of Givens rotations: $R' = J_{\ell+1} \dots J_m R$, where J_j denote the rotation in planes (rows) $j-1$ and j such that when it multiplies a matrix R from the left, it sets the entry of R at the j-th row and i_ℓ-th column to zero. Therefore, letting $\mathcal{P}' = (i_1, \dots, i_{\ell-1}, i_\ell, i_{\ell+1}, \dots, i_p)$, $R'_{\mathcal{P}'}$ is the R-factor of the QR-decomposition of $A_{\mathcal{P}'}$.

As the size of the active-set never exceeds m, both of these operations can be performed in $\mathcal{O}(mn)$ flops. Solving for the primal variable $x_\mathcal{P}$ can be done via back substitution, which takes additional $\mathcal{O}(m^2)$ flops. Hence, every step in Algorithm 8 takes at most $\mathcal{O}(mn)$ flops, which significantly reduces the computational cost as matrix inversion may require $\mathcal{O}(m^3)$ flops.

9.4.3 Related Work

Algorithm 6 is essentially a sequence of non-negative least squares problems with growing size. In [42], the authors define an algorithm called the adaptive inverse scale space method, which considers the same differential inclusion as (9.1.1). The authors establish that the sequence of nonnegative least squares problems that are defined above yields a solution to the ℓ_1-norm minimization problem, while our results here show that this solution is unique. A significant drawback of the method in [42] (and Algorithm 6 here) is that it requires to solve a non-negative least squares problems of evergrowing size. This is extremely discouraging because when $n \approx m$ and the minimum ℓ_1-norm solution is not super sparse, solving the sequence of subproblems becomes far more costly than solving the basis pursuit problem directly, say using the simplex method. However, in Sections 9.4.1 and 9.4.2, we present an efficient implementation whose iteration cost does not grow as the size of the subproblems grows. This yields an algorithm that can be efficiently used for solving the basis pursuit problem.

A closely related method to Algorithm 8 is the polytope faces pursuit (PFP) method proposed in [118]. The PFP method is a greedy active-set method to solve the dual of the basis pursuit problem. The variables that enter the active-set are determined according to the same principle as (9.4.13). However, the authors does not use the sequential nonnegative least squares formulation to determine which variables should leave the active-set, but instead randomly pick an index and remove it from the active-set when a strictly positive stepsize according to (9.4.12) cannot be chosen. Consequently, the algorithm is not necessarily convergent. In fact, the greedy update rule in [118] for linear programming with inequality constraints is first presented by the seminal work of Rosen [131]. This method is called the gradient projection method as it projects the gradient direction (i.e., negative gradient) onto the subspace spanned by the active constraints. In comparison, Algorithm 8 projects the gradient direction onto the dual feasible space, which is far more efficient because when the gradient direction is in the interior of the dual feasible space, Algorithm 8 allows us to cut through the dual polyhedron instead of walking on the boundary.

For a more detailed understanding on the differences of this method with respect to the Homotopy method [111, 98] and the orthogonal matching pursuit (OMP) method [115], we refer to [56, Section 8] and [42, Section 5.3], respectively. Comparisons between these methods are also discussed in the following numerical experiments.

9.5 Numerical Experiments

All algorithms are implemented on Matlab and the experiments are run on a computer with 2.9 GHz processor and 16 GB memory. Homotopy and PFP algorithms are implemented

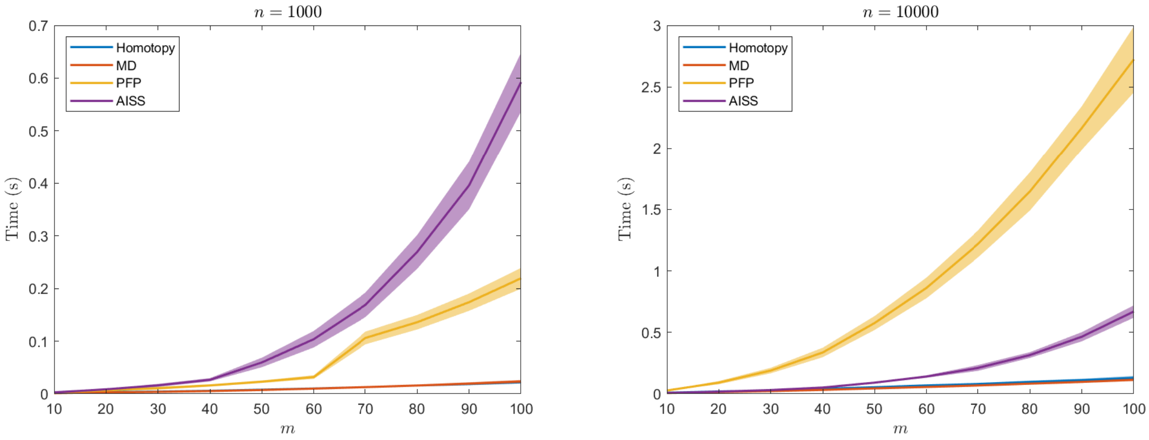

Figure 9-4: Running times of the Homotopy, MD, PFP and AISS methods averaged over 100 randomly generated problems.

using the SparseLab package [55] with the default options.

We first consider a random sparse problem, where the entries of the matrix A is sampled from $A_{ij} \sim \mathrm{N}(0,1)$. Then, the columns of A are normalized to have unit norm. We then plant a random solution sampled from $x_i \sim \mathrm{Unif}[-1,1]$. All experiments are based on 100 Monte Carlo runs over the initialization $x^0 = 0$. In Figure 9-4, we compare the running time of the MD method described in Algorithm 8 with respect to Homotopy, PFP and AISS methods. It can be seen from the left panel that the running time of the AISS method that does not exploit the efficient implementation of the nonnegative least squares formulation deteriorates as m increases. This results from the resolving the sequential nonnegative least squares problems from the stratch every time instead of using the previous solution as a warm-start to an active-set method as done in Algorithm 8. Similarly, as n increases, the running time of the PFP method quickly deteriorates due to the random variable removal, which results redundant basis changes in the evolution of the PFP method. On the other hand the MD method presented in Algorithm 8 finds the optimal solution as fast as the Homotopy method.

We next compare the MD and Homotopy methods on the diabetes dataset in Figure 9-5. As we can observe from the figures in the left column, the MD method monotonically increases the dual objective at each iteration, whereas the Homotopy method decreases the ℓ_∞-norm of the gradient of the primal objective at each iteration. Hence, the Homotopy method is a primal method, whereas the MD method is a dual method. In the right column of Figure 9-5, we observe that the primal trajectories generated by the Homotopy and MD methods are significantly different. Indeed, the Homotopy method returns the trajectory of $\min_x f(x) + \lambda\|x\|_1$ as λ goes from ∞ to 0. On the other hand, primal trajectory of the MD method is generated by a row generation method, i.e., by sequentially adding the rows of the linear system $Ax = b$ and solving the corresponding subproblem.

9.6 Discussion

In this chapter, we extended the mirror descent method to non-smooth geometries by relaxing the strict convexity condition on the distance generating function Φ. We showed that when $\operatorname{dom}\Phi$ is bounded, every limit point of a solution to the mirror descent differential inclusion is a pre-image of a primal variable that minimizes the objective function. On the other hand when $\operatorname{dom}\Phi$ is unbounded, we showed that the mirror descent method can still be applied to solve quadratic optimization problems, for which the differential inclusion admits a unique solution. We discussed how to obtain discrete-time solutions to the differential inclusion either through using ϵ-subgradients or by regularizing the differential inclusion and converting it to a differential equation that can be solved using the methods presented in Chapter 8. Finally, we discussed an application for which the trajectory of the solution to the differential inclusion can be exactly and efficiently recovered via an active-set method.

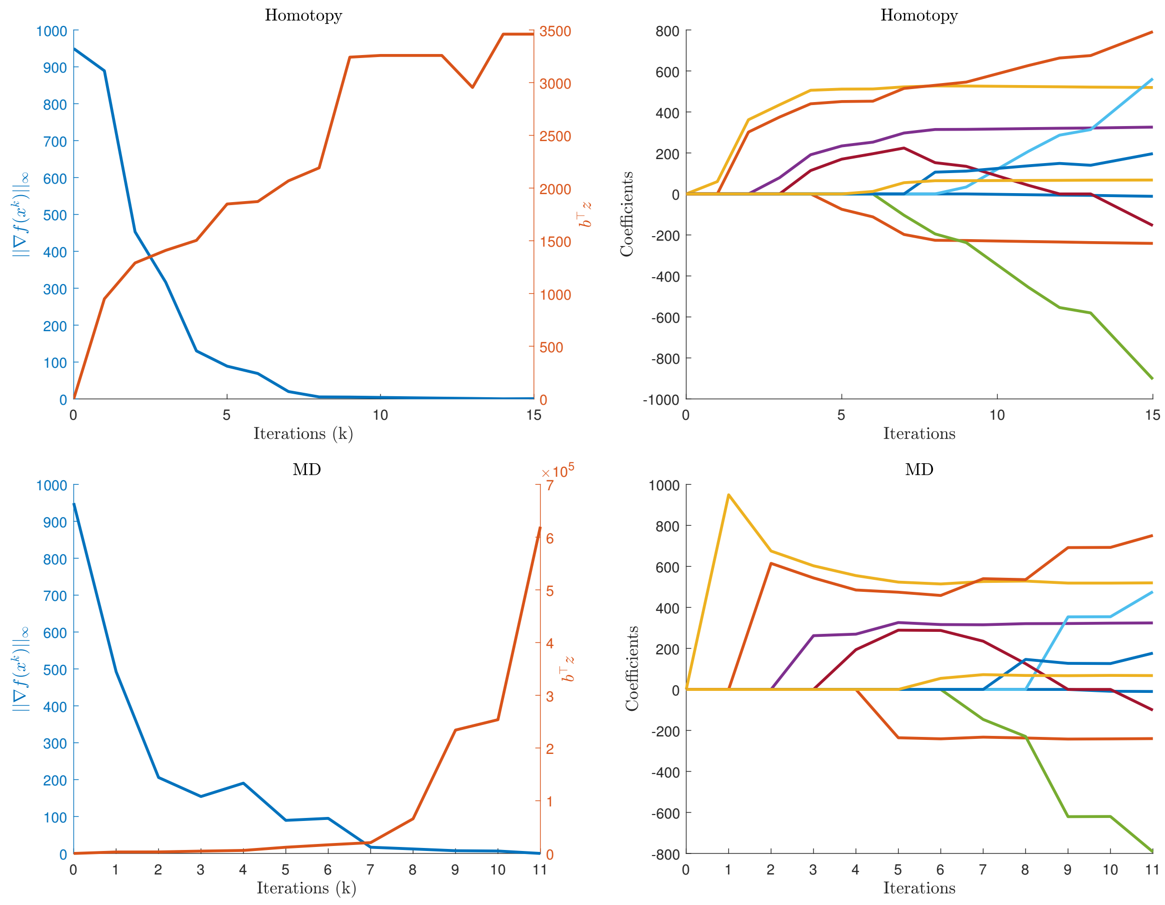

Figure 9-5: Trajectory of the iterations of the Homotopy and MD methods on the diabetes dataset. Top row corresponds to the Homotopy method, whereas the bottom row corresponds to the MD method. Left column shows the evolution of the dual objective and ℓ_∞-norm of the gradients, whereas the right column shows the evolution of the primal variables.

9.7 Additional Proofs

9.7.1 Proof of Theorem 9.3

We begin the proof by showing that $\nabla f \circ \partial\Phi^*$ has non-empty closed convex values for every $y \in \mathcal{E}^*$, and it maps to a bounded set. Since $\operatorname{dom}\Phi = \mathcal{X}$ is bounded, Φ is supercoercive, which implies $\operatorname{dom}\Phi^* = \mathcal{E}^*$ as we discussed in Section 7.2. This also implies $\partial\Phi^*(y)$ is non-empty at every $y \in \mathcal{E}^*$ by [128, Theorem 23.4]. Furthermore, it is straightforward to observe that $\partial\Phi^*$ is closed-valued and convex-valued since the subdifferential $\partial\Phi^*(y)$ is given by the intersection of infinite set of halfspaces:

$$\partial\Phi^*(y) = \bigcap_{v \in \operatorname{dom}\Phi^*} \{x : \Phi^*(v) \geq \Phi^*(y) + \langle x, v - y\rangle\}.$$

Since ∇f is continuous and $\operatorname{dom}\nabla f = \mathcal{E}$, it maps bounded closed convex sets to bounded closed convex sets. Thus, $\nabla f \circ \partial\Phi^*$ is an upper semi-continuous map (by Lemma 9.1) from $\mathcal{E}^*$ to some $\mathcal{Y} \subset \mathcal{E}^*$ with non-empty closed convex values, where $\mathcal{Y}$ is bounded since $\operatorname{rge}\partial\Phi^*$ is bounded and ∇f is Lipschitz continuous. Consequently, we can apply [9, Theorem 2.1.4] to show that for any $y^0 \in \operatorname{dom}\partial\Phi^*$, there exists an absolutely continuous solution $y(\cdot)$ defined on $[0, +\infty)$ to the differential inclusion (9.1.1). Below, we provide a short proof of this claim.

Let B denote the unit ball in $\mathcal{E}^* \times \mathcal{E}^*$ and let $\{g_n\}$ be a sequence of continuous single-valued maps such that $\operatorname{graph}(g_n) \subset \operatorname{graph}(\nabla f \circ \partial\Phi^*) + \varepsilon_n B$, where the existence of such maps for every $\varepsilon_n > 0$ follows by the approximate selection theorem [9, Theorem 1.12.1]. Suppose ε_n approaches to zero as $n \to \infty$ and let $y_n : [0, T] \to \mathcal{E}^*$ be solutions to the differential equation $\dot{y}_n(t) = -g_n(y_n(t))$ with $y_n(0) = y^0$. Since $\operatorname{rge}(\nabla f \circ \partial\Phi^*) = \mathcal{Y}$ is bounded and ε_n is finite, $\operatorname{rge} g_n$ is also bounded, i.e., each $\dot{y}_n$ takes values in a bounded set. Consequently, each y_n takes values in a compact set as T and y^0 are finite. Therefore, Ascoli-Arzela and Banach-Alaoglu theorems imply (see [9, Theorem 0.3.4]) that there exists a subsequence $\{y_{n_k}\}$ such that y_{n_k} converges uniformly to y on $[0, T]$ and $\dot{y}_{n_k}$ converges weakly to $\dot{y}$ in $L^1([0, T])$ (i.e., functions that are absolutely integrable on $[0, T]$). Then, by the convergence theorem for upper semi-continuous maps [9, Theorem 1.4.1], we conclude that for almost all $t \in [0, T]$, $(y(t), \dot{y}(t)) \in \operatorname{graph}(-\nabla f \circ \partial\Phi^*)$, i.e., $\dot{y}(t) \in -(\nabla f \circ \partial\Phi^*)(y(t))$. Taking $T \to \infty$ concludes the proof.

9.7.2 A Dual Viewpoint

Consider the dual of the basis pursuit problem given in (9.4.4). Letting $g(z) = \langle b, z \rangle$, the gradient flow direction at $z \in \mathcal{Z}$ is given by

$$d = \lim_{\eta \to 0^+} \frac{\mathrm{P}_{\mathcal{Z}}(z + \eta \nabla g(z)) - z}{\eta}, \tag{9.7.1}$$

where $\mathrm{P}_{\mathcal{Z}}(z)$ is the Euclidean projection of z onto $\mathcal{Z}$:

$$\mathrm{P}_{\mathcal{Z}}(z) = \arg\min_{z' \in \mathcal{Z}} \frac{1}{2} \|z - z'\|_2^2.$$

Then, as shown in the following proposition, the update direction d is given by a solution of the nonnegative least squares problem.

Proposition 9.14. *Let $z \in \mathcal{Z}$ be a dual solution and $\mathcal{I}(z) = \mathcal{I}_+(z) \cup \mathcal{I}_-(z)$ denote the set of active constraints at z. Then, the gradient flow direction d defined in (9.7.1) is given by*

$$d = b - \sum_{i \in \mathcal{I}(z)} x_i^* a_i,$$

where x_i^*, $i \in \mathcal{I}(z)$, is a solution to the following problem:

$$\begin{aligned} \min \quad & \frac{1}{2} \left\| b - \sum_{i \in \mathcal{I}(z)} x_i a_i \right\|_2^2 \\ \text{s.t.} \quad & x_i \geq 0, \quad \forall i \in \mathcal{I}_+(z), \\ & x_i \leq 0, \quad \forall i \in \mathcal{I}_-(z). \end{aligned}$$

Proof We begin the proof by rewriting the projection onto the dual feasible set as $\mathrm{P}_{\mathcal{Z}}(z + tb) = z + td^*$. Then, d^* can be found as the argument of the following minimization problem

$$\begin{aligned}\min \;& \frac{1}{2}\|(z+td)-(z+tb)\|_2^2\\ \text{s.t.}\;& \|A^\top(z+td)\|_\infty \le 1.\end{aligned}$$

For t sufficiently small, non-active constraints cannot be violated by $z+td$. Therefore, we can rewrite the above problem as follows:

$$\begin{aligned}\min \;& \frac{1}{2}\|(z+td)-(z+tb)\|_2^2\\ \text{s.t.}\;& \psi_i(z+td) \le 0, \quad \forall i \in \mathcal{I}_+(z),\\ & \psi_{-i}(z+td) \ge 0, \quad \forall i \in \mathcal{I}_-(z).\end{aligned}$$

As $\psi_i(z)=0$ for every $i \in \mathcal{I}_+(z)$ and $\psi_{-i}(z)=0$ for every $i \in \mathcal{I}_-(z)$, we equivalently have

$$\begin{aligned}\min \;& \frac{1}{2}\|d-b\|_2^2\\ \text{s.t.}\;& a_i^\top d \le 0, \quad \forall i \in \mathcal{I}_+(z),\\ & a_i^\top d \ge 0, \quad \forall i \in \mathcal{I}_-(z).\end{aligned}$$

The Lagrange dual corresponding to this problem is given by

$$\begin{aligned}\max \;& \min_d \left\{ \frac{1}{2}\|d-b\|_2^2 + \sum_{i\in\mathcal{I}(z)} x_i a_i^\top d \right\}\\ \text{s.t.}\;& x_i \ge 0, \quad \forall i \in \mathcal{I}_+(z),\\ & x_i \le 0, \quad \forall i \in \mathcal{I}_-(z).\end{aligned}$$

Massaging this problem, we obtain

$$\begin{aligned}\max \quad & \min_d \left\{ \frac{1}{2} d^\top d - d^\top \left(b - \sum_{i \in \mathcal{I}(z)} x_i a_i \right) \right\} \\ \text{s.t.} \quad & x_i \geq 0, \quad \forall i \in \mathcal{I}_+(z), \\ & x_i \leq 0, \quad \forall i \in \mathcal{I}_-(z).\end{aligned}$$

The minimizer of the inner problem is given by

$$d^* = b - \sum_{i \in \mathcal{I}(z)} x_i a_i,$$

and plugging in this value to the above problem, we get

$$\begin{aligned}\max \quad & -\frac{1}{2} \left\| b - \sum_{i \in \mathcal{I}(z)} x_i a_i \right\|_2^2 \\ \text{s.t.} \quad & x_i \geq 0, \quad \forall i \in \mathcal{I}_+(z), \\ & x_i \leq 0, \quad \forall i \in \mathcal{I}_-(z).\end{aligned}$$

Taking the minus sign outside of the maximization, we get the desired result. □

9.7.3 Proof of Theorem 9.12

As we discussed in Section 9.4, the update direction does not change until a new variable enters into the active-set. In order to find which constraint becomes active first, we perform the following ratio test among all non-active constraints. To derive the ratio test, we first observe that for every $i \notin \mathcal{I}(z^k) \setminus \mathcal{B}(z^k)$, we have

$$\begin{aligned} \psi_i(z^k + \eta d^k) &= \psi_i(z^k) + \eta a_i^\top d^k, \quad \text{where} \quad \psi_i(z^k) < 0, \\ \psi_{-i}(z^k + \eta d^k) &= \psi_{-i}(z^k) + \eta a_i^\top d^k, \quad \text{where} \quad \psi_{-i}(z^k) > 0. \end{aligned}$$

Solving $\psi_i(z^k + \eta d^k) = 0$ and $\psi_{-i}(z^k + \eta d^k) = 0$ for η, we find the values of η for which the corresponding inequality becomes active. Among these values the minimum positive one is the largest possible step size that we are looking for. Furthermore, for each $i \notin \mathcal{I}(z^k) \backslash \mathcal{B}(z^k)$, exactly one of ψ_i and ψ_{-i} becomes active with some $\eta > 0$ and the other becomes active with some $\eta < 0$. Therefore, $\bar{\eta}$ is given by

$$\bar{\eta} = \min_{i \notin \mathcal{I}(z^k) \backslash \mathcal{B}(z^k)} \left\{ \max\left(-\frac{\psi_i(z^k)}{a_i^\top d^k}, -\frac{\psi_{-i}(z^k)}{a_i^\top d^k} \right) \right\}.$$

Using the definitions $\psi_i(z^k) = a_i^\top z^k - 1$ and $\psi_{-i}(z^k) = a_i^\top z^k + 1$, and observing that ψ_i becomes active with some positive η whenever $a_i^\top d^k > 0$ and ψ_{-i} becomes active otherwise, we can rewrite $\bar{\eta}$ as follows:

$$\bar{\eta} = \min_{i \notin \mathcal{I}(z^k) \backslash \mathcal{B}(z^k)} -\frac{a_i^\top z^k - \operatorname{sgn}(a_i^\top d^k)}{a_i^\top d^k},$$

which is what we wanted to prove.

9.7.4 Proof of Theorem 9.13

We prove this theorem in three steps. First, we show that $\{\|d^k\|_2^2\}_{k \geq 0}$ is a strictly decreasing sequence (this trivially follows by Theorem 9.7, yet below we include a proof for the discrete-time method as well for completeness). We then claim that there are finitely many values $\|d^k\|_2^2$ can take due to finitely many active-set combinations. We conclude the proof by showing that at every iteration k with $d^k \neq 0$, objective value increases by a positive amount, ruling out termination before the optimal solution is obtained.

Let $\{z^k\}_{k \geq 0}$ be a sequence of dual variables generated by Algorithm 6 with the corresponding sequence of active-sets $\{\mathcal{I}(z^k)\}_{k\geq 0}$. Let f^k denote the optimum value of the subproblem (9.4.7) solved at iteration k, i.e.,

$$f^k = \min \ \frac{1}{2}\left|\left| b - \sum_{i\in\mathcal{I}(z^k)} x_i a_i \right|\right|_2^2 \tag{9.7.2a}$$

$$\text{s.t.} \ \ x_i \geq 0, \quad \forall i \in \mathcal{I}_+(z^k), \tag{9.7.2b}$$

$$x_i \leq 0, \quad \forall i \in \mathcal{I}_-(z^k), \tag{9.7.2c}$$

and let x_i^k, $i \in \mathcal{I}(z^k)$, be a corresponding minimizer. Then, we claim that $\{f^k\}_{k\geq 0}$ is a nonincreasing sequence. This can be concluded by first observing that the objective value remains the same when we change the constraints as follows

$$f^k = \min \ \frac{1}{2}\left|\left| b - \sum_{i\in\mathcal{I}(z^k)} x_i a_i \right|\right|_2^2 \tag{9.7.3a}$$

$$\text{s.t.} \ \ x_i \geq 0, \quad \forall i \in \mathcal{I}_+(z^k) \setminus \mathcal{B}_+(z^k), \tag{9.7.3b}$$

$$x_i \leq 0, \quad \forall i \in \mathcal{I}_-(z^k) \setminus \mathcal{B}_-(z^k), \tag{9.7.3c}$$

$$x_i = 0, \quad \forall i \in \mathcal{F}_+(z^k) \cup \mathcal{F}_-(z^k), \tag{9.7.3d}$$

which follows since $\mathcal{I}_+(z^k) \cap \mathcal{F}_+(z^k) = \emptyset$ and $\mathcal{I}_-(z^k) \cap \mathcal{F}_-(z^k) = \emptyset$, and for any $i \in \mathcal{B}(z^k)$, $a_i^\top d^k = 0$ holds, which implies $x_i^k = 0$ by complementary slackness. From equation 9.7.3, we can observe that $f^{k+1} \leq f^k$, as $x_i = x_i^k$, $i \in \mathcal{I}(z^k) \setminus \mathcal{B}(z^k)$, and $x_i = 0$, $i \in \mathcal{F}(z^k)$, is a feasible solution for the subproblem at iteration $k+1$. Furthermore, the equality $f^{k+1} = f^k$ holds if and only if $d^{k+1} = d^k$. This follows due to the strong convexity of ℓ_2-norm, but below we include a proof for completeness. It is easy to observe the *if* part of the proof, and the *only if* part is can be shown by contraposition. To that end, with an abuse of notation, we let x^k denote a solution to the subproblem at iteration $k+1$, with $x_i^k = 0$, $i \in \mathcal{F}(z^k)$. Suppose $d^k \neq d^{k+1}$, which implies there exists x^k and x^{k+1} such that $x_j^k \neq x_j^{k+1}$ for some $j \in \mathcal{I}(z^{k+1})$. Consider a new solution to this problem defined as $\bar{x}_i = (x_i^k + x_i^{k+1})/2$,

$i \in \mathcal{I}(z^{k+1})$. Then, $\bar{x}$ is dual feasible as the feasible set is convex and

$$\left\| b - \sum_{i \in \mathcal{I}(z^{k+1})} \bar{x}_i a_i \right\|_2^2 < \left\| b - \sum_{i \in \mathcal{I}(z^{k+1})} x_i^k a_i \right\|_2^2 + \left\| b - \sum_{i \in \mathcal{I}(z^{k+1})} x_i^{k+1} a_i \right\|_2^2, \tag{9.7.4}$$

where the strict inequality holds due the Cauchy-Schwarz inequality since $x_j^k \neq x_j^{k+1}$. Therefore, x^k cannot be optimal for the subproblem at iteration $k+1$, i.e., $f^k \neq f^{k+1}$. This concludes that $\{f^k\}_{k \geq 0}$ is a strictly decreasing sequence.

We observe that if active-sets at two distinct iterations $k \neq j$ are the same, i.e., $\mathcal{I}_+(z^k) = \mathcal{I}_+(z^j)$ and $\mathcal{I}_-(z^k) = \mathcal{I}_-(z^j)$, then $f^k = f^j$ since an active-set uniquely characterizes the optimal value of the problem (9.7.2). As $\{f^k\}_{k \geq 0}$ is a strictly decreasing sequence, an active-set cannot be visited twice. Finally, whenever $d^k \neq 0$, we have $0 < \eta^k < \infty$. The upper bound follows since $\eta^k = \infty$ holds when $a_i^\top d^k = 0$ for all $i \notin \mathcal{I}(z^k) \setminus \mathcal{B}(z^k)$. However, when the latter condition holds the ray emanating from z^k in the direction d^k is an extreme ray, i.e., $z(\eta) = z^k + \eta d^k$ is feasible for every $\eta \geq 0$ and the objective value at $z(\eta)$ is an increasing function of η. Therefore, dual problem is unbounded, which implies primal problem is infeasible, and this conflicts with the assumption that $Ax = b$ is realizable; consequently $\eta^k < \infty$ holds. Therefore, there exists some $i \notin \mathcal{I}(z^k) \setminus \mathcal{B}(z^k)$ such that $a_i^\top d^k \neq 0$. Since $|a_i^\top z^k| \neq 1$ for every $i \notin \mathcal{I}(z^k)$ and $\operatorname{sgn}(a_i^\top d^k) = -a_i^\top z^k$ for every $i \in \mathcal{B}(z^k)$, it follows that the numerator of (9.4.9) is strictly positive, i.e., $\eta^k > 0$. As the algorithm does not terminate until $d^k = 0$, $\{\|d^k\|_2^2\}_{k \geq 0}$ is a strictly decreasing sequence, and there are finitely many values $\|d^k\|_2^2$ can take, it follows that the algorithm returns an optimal solution in finitely many iterations.

Bibliography

[1] P. A. Absil, R. Mahony, and B. Andrews. Convergence of the iterates of descent methods for analytic cost functions. *SIAM Journal on Optimization*, 16(2):531–547, 2005.

[2] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, USA, 2007.

[3] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton. Complementarity and nondegeneracy in semidefinite programming. *Mathematical Programming*, 77(1):111–128, 1997.

[4] F. Alvarez. On the minimizing property of a second order dissipative system in Hilbert spaces. *SIAM Journal on Control and Optimization*, 38(4):1102–1119, 2000.

[5] F. Alvarez, H. Attouch, J. Bolte, and P. Redont. A second-order gradient-like dissipative dynamical system with hessian-driven damping. application to optimization and mechanics. *Journal de MathÃľmatiques Pures et AppliquÃľes*, 81(8):747–779, 2002.

[6] M. Anitescu. Degenerate nonlinear programming with a quadratic growth condition. *SIAM Journal on Optimization*, 10(4):1116–1135, 2000.

[7] S. Arora, E. Hazan, and S. Kale. Fast algorithms for approximate semidefiniite programming using the multiplicative weights update method. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '05, pages 339–348, 2005.

[8] H. Attouch, Z. Chbani, J. Peypouquet, and P. Redont. Fast convergence of inertial dynamics and algorithms with asymptotic vanishing viscosity. *Mathematical Programming*, 168:123–175, 2018.

[9] J.-P. Aubin and A. Cellina. *Differential Inclusions: Set-Valued Maps and Viability Theory*. Springer-Verlag, 1984.

[10] J.-P. Aubin and H. Frankowska. *Set-Valued Analysis*. Birkhauser, 1990.

[11] A. Auslender and M. Teboulle. *Asymptotic cones and functions in optimization and variational inequalities*. Springer Monographs in Mathematics, 2006.

[12] J. B. Baillon. Un exemple concernant le comportement asymptotique de la solution du problÃĺme $du/dt + \partial\phi(\mu) \ni 0$. *Journal of Functional Analysis*, 28(3):369–376, 1978.

[13] A. S. Bandeira, N. Boumal, and V. Voroninski. On the low-rank approach for semidefinite programs arising in synchronization and community detection. *arXiv:1602.04426*, 2016.

[14] A. I. Barvinok. Problems of distance geometry and convex properties of quadratic maps. *Discrete & Computational Geometry*, 13(2):189–202, 1995.

[15] 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(2):330–348, 2017.

[16] H. H. Bauschke, J. M. Borwein, and P. L. Combettes. Essential smoothness, essential strict convexity, and Legendre functions in Banach spaces. *Communications in Contemporary Mathematics*, 3(4):615–647, 2001.

[17] A. Beck. On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes. *SIAM Journal on Optimization*, 25(1):185–209, 2015.

[18] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[19] A. Beck and L. Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060, 2013.

[20] S. Becker, J. Bobin, and E. J. CandÃĺs. NESTA: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39, 2011.

[21] D. P. Bertsekas. *Nonlinear programming.* Athena Scientific, 1999.

[22] D. P. Bertsekas. *Convex Optimization Algorithms.* Athena Scientific, 2015.

[23] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods.* Prentice-Hall, Inc., 1989.

[24] D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

[25] J. Bolte and E. Pauwels. Curiosities and counterexamples in smooth convex optimization. *arXiv:2001.07999*, 2020.

[26] J. F. Bonnans and A. Ioffe. Second-order sufficiency and quadratic growth for nonisolated minima. *Mathematics of Operations Research*, 20(4):801–817, 1995.

[27] V. S. Borkar. *Stochastic Approximation*. Hindustan Book Agency, 2008.

[28] L. Bottou. Curiously fast convergence of some stochastic gradient descent algorithms. In *Proceedings of the symposium on learning and data science*, 2009.

[29] L. Bottou. Stochastic gradient descent on toy problems. http://leon.bottou.org/projects/sgd, 2012.

[30] N. Boumal, P.-A. Absil, and C. Cartis. Global rates of convergence for nonconvex optimization on manifolds. *arXiv:1605.08101*, 2016.

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

[32] N. Boumal, V. Voroninski, and A. S. Bandeira. The non-convex Burer-Monteiro approach works on smooth semidefinite programs. In *Advances in Neural Information Processing Systems*, pages 2757–2765, 2016.

[33] N. Boumal, V. Voroninski, and A. S. Bandeira. Deterministic guarantees for BurerâĂŞMonteiro factorizations of smooth semidefinite programs. *arXiv:1804.02008*, 2018.

[34] 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(3):200–217, 1967.

[35] H. Brezis. *OpÃľrateurs Maximaux Monotones et semigroups de Contractions dans les Espaces de Hilbert*. North-Holland Publishing Co., 1973.

[36] C. Briat. *Linear parameter-varying and time-delay systems*. Springer, 2014.

[37] J. Briët, F. M. de Oliveira Filho, and F. Vallentin. The positive semidefinite grothendieck problem with rank constraint. In *Automata, Languages and Programming*, pages 31–42, 2010.

[38] R. E. Bruck. Asymptotic convergence of nonlinear contraction semigroups in Hilbert space. *Journal of Functional Analysis*, 18(1):15–26, 1975.

[39] S. Bubeck. Convex optimization: Algorithms and complexity. *arXiv:1405.4980*, 2014.

[40] S. Burer and R. D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

[41] S. Burer and R. D. C. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444, Jul 2005.

[42] M. Burger, M. Moller, M. Benning, and S. Osher. An adaptive inverse scale space method for compressed sensing. *Mathematics of Computation*, 82(281):269–299, 2013.

[43] J.-F. Cai, E. J. CandÃÍs, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[44] E. J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[45] E. J. CandÃÍs and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(717), 2009.

[46] E. J. CandÃÍs and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[47] Y. Censor and S. A. Zenios. Proximal minimization algorithm with D-functions. *Journal of Optimization Theory and Applications*, 73:451–464, 1992.

[48] S.-K. Chao and G. Cheng. A generalization of regularized dual averaging and its dynamics. *arXiv:1909.10072*, 2019.

[49] G. Chen and M. Teboulle. Convergence analysis of a proximal-like minimization algorithm using Bregman functions. *SIAM Journal on Optimization*, 3(3):538–543, 1993.

[50] D. Cifuentes and A. Moitra. Polynomial time guarantees for the Burer-Monteiro method. *arXiv:1912.01745*, 2019.

[51] E. S. Coakley and V. Rokhlin. A fast divide-and-conquer algorithm for computing the spectra of real symmetric tridiagonal matrices. *Applied and Computational Harmonic Analysis*, 34(3):379 – 414, 2013.

[52] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

[53] A. R. De Pierro and A. N. Iusem. A relaxed version of Bregman's method for convex programming. *Journal of Optimization Theory and Applications*, 51:421âĂŞ440, 1986.

[54] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[55] D. L. Donoho and Others. Sparselab: Seeking sparse solutions to linear systems of equations. https://sparselab.stanford.edu/, 2009.

[56] D. L. Donoho and Y. Tsaig. Fast solution of ℓ_1-norm minimization problems when the solution may be sparse. *IEEE Transactions on Information Theory*, 54(11):4789–4812, 2008.

[57] J. C. Duchi and F. Ruan. Stochastic methods for composite and weakly convex optimization problems. *SIAM Journal on Optimization*, 28(4):3229–3259, 2018.

[58] L. El Ghaoui and P. Gahinet. Rank minimization under LMI constraints: A framework for output feedback problems. In *Proceedings of the European Control Conference*, 1993.

[59] G. M. Engel and H. Schneider. Cyclic and diagonal products on a matrix. *Linear Algebra and its Applications*, 7(4):301 – 335, 1973.

[60] M. A. Erdogdu, Y. Deshpande, and A. Montanari. Inference in graphical models via semidefinite programming hierarchies. In *Advances in Neural Information Processing Systems*, pages 416–424, 2017.

[61] M. A. Erdogdu, A. Ozdaglar, P. A. Parrilo, and N. D. Vanli. Convergence rate of block-coordinate maximization burer-monteiro method for solving large SDPs. *Mathematical Programming*, 2021.

[62] M. Fazel. *Matrix rank minimization with applications*. PhD thesis, Stanford University, 2002.

[63] M. Fazel, H. Hindi, and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the 2001 American Control Conference*, volume 6, pages 4734–4739, 2001.

[64] N. Flammarion and F. Bach. Stochastic composite least-squares regression with convergence rate $O(1/n)$. In *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 831–875. PMLR, 2017.

[65] D. Gamarnik and Q. Li. On the max-cut of sparse random graphs. *arXiv:1411.1698*, 2014.

[66] D. Garber and E. Hazan. Approximating semidefinite programs in sublinear time. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, pages 1080–1088, 2011.

[67] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[68] G. J. Gordon. Regret bounds for prediction problems. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, COLT '99, page 29âĂŞ40, 1999.

[69] L. Grippo and M. Sciandrone. On the convergence of the block nonlinear GaussâĂŞ-Seidel method under convex constraints. *Operations Research Letters*, 26(3):127–136, 2000.

[70] S. Gunasekar, J. Lee, D. Soudry, and N. Srebro. Characterizing implicit bias in terms of optimization geometry. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1832–1841. PMLR, 2018.

[71] M. Gurbuzbalaban, A. Ozdaglar, and P. Parrilo. Why random reshuffling beats stochastic gradient descent. *Mathematical Programming*, 186:49 – 84, 2021.

[72] M. Gurbuzbalaban, A. Ozdaglar, P. A. Parrilo, and N. D. Vanli. When cyclic coordinate descent outperforms randomized coordinate descent. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 6999–7007. Curran Associates, Inc., 2017.

[73] M. Gurbuzbalaban, A. Ozdaglar, N. D. Vanli, and S. J. Wright. Randomness and permutations in coordinate descent methods. *Mathematical Programming*, 181, 03 2018.

[74] A. Israel, F. Krahmer, and R. Ward. An arithmeticâĂŞgeometric mean inequality for products of three matrices. *Linear Algebra and its Applications*, 488:1–12, 2016.

[75] A. Javanmard, A. Montanari, and F. Ricci-Tersenghi. Phase transitions in semidefinite relaxations. *Proceedings of the National Academy of Sciences*, 113(16):E2218–E2223, 2016.

[76] M. Journee, F. Bach, P.-A. Absil, and R. Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010.

[77] A. Juditsky, J. Kwon, and E. Moulines. Unifying mirror descent and dual averaging. *arXiv:1910.13742*, 2020.

[78] J. F. C. Kingman. A convexity property of positive matrices. *The Quarterly Journal of Mathematics*, 12(1):283–284, 1961.

[79] S. J. Kirkland and M. Neumann. *Group inverses of M-matrices and their applications*. CRC Press, 2012.

[80] K. C. Kiwiel. Free-steering relaxation methods for problems with strictly convex costs and linear constraints. *Mathematics of Operations Research*, 22(2):326–349, 1997.

[81] K. C. Kiwiel. Proximal minimization methods with generalized Bregman functions. *SIAM Journal on Control and Optimization*, 35(4):1142–1168, 1997.

[82] P. Klein and H.-I Lu. Efficient approximation algorithms for semidefinite programs arising from MAX CUT and COLORING. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 338–347, New York, NY, USA, 1996. ACM.

[83] W. Krichene. *Continuous and discrete dynamics for online learning and convex optimization.* PhD thesis, University of California at Berkeley, 2016.

[84] J. Kuczyński and H. Woźniakowski. Estimating the largest eigenvalues by the power and lanczos algorithms with a random start. *SIAM J. Matrix Anal. Appl.*, 13(4):1094–1122, 1992.

[85] C. Lageman. Pointwise convergence of gradient-like systems. *Mathematische Nachrichten*, 280(13-14):1543–1558, 2007.

[86] J. LaSalle. Some extensions of Liapunov's second method. *IRE Transactions on Circuit Theory*, 7(4):520–527, 1960.

[87] C. L. Lawson and R. J. Hanson. *Solving least squares problems.* SIAM, 1995.

[88] C.-P. Lee and S. J. Wright. Random permutations fix a worst case for cyclic coordinate descent. *IMA Journal of Numerical Analysis*, 39(3):1246–1275, 2019.

[89] C.-P. Lee and S. J. Wright. Analyzing random permutations for cyclic coordinate descent. *Mathematics of Computation*, 89:2217–2248, 2020.

[90] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht. Gradient descent only converges to minimizers. In *29th Annual Conference on Learning Theory*, volume 49, pages 1246–1257. PMLR, 23–26 Jun 2016.

[91] S. Lojasiewicz. Sur la géométrie semi- et sous- analytique. *Annales de l'Institut Fourier*, 43(5):1575–1595, 1993.

[92] H. Lu. âĂIJrelative continuityâĂİ for non-lipschitz nonsmooth convex optimization using stochastic (or deterministic) mirror descent. *INFORMS Journal on Optimization*, 1(4):288–303, 2019.

[93] H. Lu, R. M. Freund, and Y. Nesterov. Relatively smooth convex optimization by first-order methods, and applications. *SIAM Journal on Optimization*, 28(1):333–354, 2018.

[94] Z. Lu and L. Xiao. Randomized block coordinate non-monotone gradient method for a class of nonlinear programming. Technical Report MSR-TR-2013-66, June 2013.

[95] Z.-Q. Luo and P. Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72(1):7–35, 1992.

[96] Z.-Q. Luo and P. Tseng. Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, 46(1):157–178, 1993.

[97] S. Ma, D. Goldfarb, and L. Chen. Fixed point and Bregman iterative methods for matrix rank minimization. *Mathematical Programming*, 128:321–353, 2011.

[98] D. M. Malioutov, M. Cetin, and A. S. Willsky. Homotopy continuation for sparse signal representation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing Proceedings. (ICASSP '05).*, volume 5, pages 733–736, 2005.

[99] D. M. Malioutov, J. K. Johnson, and A. S. Willsky. Walk-sums and belief propagation in gaussian graphical models. *Journal of Machine Learning Research*, 7:2031–2064, 2006.

[100] B. McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 525–533, 2011.

[101] S. Mei, T. Misiakiewicz, A. Montanari, and R. I. Oliveira. Solving SDPs for synchronization and MaxCut problems via the Grothendieck inequality. *arXiv:1703.08729*, 2017.

[102] G. J. Minty. Monotone (nonlinear) operators in hilbert space. *Duke Mathematical Journal*, 29(3):341–346, 1962.

[103] A. Montanari. A Grothendieck-type inequality for local maxima. *arXiv:1603.04064*, 2016.

[104] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995.

[105] D. Needell and J. A. Tropp. Paved with good intentions: Analysis of a randomized block Kaczmarz method. *Linear Algebra and its Applications*, 441:199 – 221, 2014.

[106] A. S. Nemirovski and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization.* John Wiley & Sons, 1983.

[107] Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120:221–259, 2009.

[108] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[109] R. D. Nussbaum. Convexity and log convexity for the spectral radius. *Linear Algebra and its Applications*, 73:59–122, 1986.

[110] J. Ortega and W. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables.* Society for Industrial and Applied Mathematics, 2000.

[111] M. R. Osborne, B. Presnell, and B. A. Turlach. A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20(3):389–403, 2000.

[112] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005.

[113] P. Oswald and W. Zhou. Random reordering in SOR-type methods. *Numerische Mathematik*, 135(4):1207–1220, 2017.

[114] G. Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Mathematics of Operations Research*, 23(2):339–358, 1998.

[115] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, pages 40–44, 1993.

[116] A. Patrascu and I. Necoara. Efficient random coordinate descent algorithms for large-scale structured nonconvex optimization. *Journal of Global Optimization*, 61, 2013.

[117] R. J. Plemmons. M-matrix characterizations.IâĂŤnonsingular m-matrices. *Linear Algebra and its Applications*, 18(2):175 – 188, 1977.

[118] M. D. Plumbley. Recovery of sparse representations by polytope faces pursuit. In *Independent Component Analysis and Blind Signal Separation*, pages 206–213, 2006.

[119] M. Powell. On search directions for minimization algorithms. *Mathematical Programming*, 4:193–201, 1973.

[120] T. Pumir, S. Jelassi, and N. Boumal. Smoothed analysis of the low-rank approach for smooth semidefinite programs. In *Advances in Neural Information Processing Systems*, pages 2281–2290, 2018.

[121] A. Rantzer. Distributed control of positive systems. In *50th IEEE Conference on Decision and Control and European Control Conference*, pages 6608–6611, 2011.

[122] M. Razaviyayn, M. Hong, and Z.-Q. Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013.

[123] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[124] B. Recht and C. Ré. Toward a noncommutative arithmetic-geometric mean inequality: Conjectures, case-studies, and consequences. *JMLR Workshop and Conference Proceedings*, 23:11.1–11.24, 2012.

[125] P. Richtarik and M. Takac. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144, 07 2011.

[126] P. Richtarik and M. Takac. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156:433–484, 2016.

[127] S. M. Robinson. Linear convergence of epsilon-subgradient descent methods for a class of convex functions. *Mathematical Programming*, 86:41–50, 1999.

[128] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[129] R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[130] R. T. Rockafellar and R. J.-B. Wets. *Variational analysis*. Springer, 2009.

[131] J. B. Rosen. The gradient projection method for nonlinear programming. part i. linear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 8(1):181–217, 1960.

[132] A. Saha and A. Tewari. On the nonasymptotic convergence of cyclic coordinate descent methods. *SIAM Journal on Optimization*, 23(1):576–601, 2013.

[133] S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107âĂŞ194, 2012.

[134] O. Shamir. Without-replacement sampling for stochastic gradient methods. In *Advances in Neural Information Processing Systems*, pages 46–54, 2016.

[135] N. Srebro. *Learning with Matrix Factorizations*. PhD thesis, Massachusetts Institute of Technology, 2004.

[136] D. Steurer. Fast SDP algorithms for constraint satisfaction problems. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 684–697, 2010.

[137] R. Sun and M. Hong. Improved iteration complexity bounds of cyclic block coordinate descent for convex problems. In *Advances in Neural Information Processing Systems 28*, pages 1306–1314. 2015.

[138] R. Sun and Y. Ye. Worst-case complexity of cyclic coordinate descent: $O(n^2)$ gap with randomized version. *Mathematical Programming*, 185:485–520, 2021.

[139] M. Teboulle. A simplified view of first order methods for optimization. *Mathematical Programming*, 170:67âĂŞ96, 2018.

[140] M. Telgarsky and S. Dasgupta. Agglomerative Bregman clustering. In *Proceedings of the 29th International Conference on Machine Learning*, ICML '12, pages 1527–1534, 2012.

[141] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher. Practical sketching algorithms for low-rank matrix approximation. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1454–1485, 2017.

[142] P. Tseng and D. P. Bertsekas. On the convergence of the exponential multiplier method for convex programming. *Mathematical Programming*, 60:1–19, 1993.

[143] P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117:387–423, 03 2009.

[144] R. Tutunov, H. Bou-Ammar, and A. Jadbabaie. Distributed SDDM solvers: Theory & applications. *arXiv:1508.04096*, 2015.

[145] E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890âĂŞ912, 2008.

[146] N. D. Vanli, M. Gurbuzbalaban, and A. Ozdaglar. Global convergence rate of proximal incremental aggregated gradient methods. *SIAM Journal on Optimization*, 28(2):1282–1300, 2018.

[147] R. S. Varga. *Matrix iterative analysis.* Springer Science & Business Media, 2009.

[148] P.-W. Wang, W.-C. Chang, and J. Z. Kolter. The mixing method: Coordinate descent for low-rank semidefinite programming. *arXiv:1706.00476*, 2017.

[149] A. Wilson. *Lyapunov Arguments in Optimization.* PhD thesis, University of California at Berkeley, 2018.

[150] S. J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

[151] L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(88):2543–2596, 2010.

[152] W. Yin, S. Osher, D. Goldfarb, and J. Darbon. Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing. *SIAM Journal on Imaging Sciences*, 1(1):143–168, 2008.

[153] D. M. Young. *Iterative solution of large linear systems.* Academic Press, 1971.

[154] T. Zhang. A note on the non-commutative arithmetic-geometric mean inequality. *arXiv:1411.5058*, 2014.