TEL-AVIV UNIVERSITY
THE RAYMOND AND BEVERLY SACKLER
FACULTY OF EXACT SCIENCES
SCHOOL OF MATHEMATICAL SCIENCES
DEPARTMENT OF STATISTICS AND OPERATIONS RESEARCH

Contributions to the Complexity Analysis of Optimization Algorithms

Dissertation submitted to the Tel-Aviv University Senate for the degree of "Doctor of Philosophy"

by

Yoel Drori

Under the supervision of Prof. Marc Teboulle

December 2014

Acknowledgments

The last few years were an extremely interesting and satisfying period of my life, this is first and foremost thanks to my advisor, Professor Marc Teboulle, whose guidance and patience were crucial for the success of this work. I would also like to take this opportunity to extend my love to my family and thank them for their unconditional support throughout the years.

Contents

List of Figures

List of Tables

Abstract

The main focus of this work is the introduction of a novel approach for the worst-case analysis of first-order methods. The approach is based on the observation that the worst-case performance of a given algorithm can be expressed as an optimization problem (which attempts to find the "worst-case input" to the algorithm), and that problem can be tackled using tools from the field of optimization. In Chapter 2, we focus on smooth and convex optimization problems, and show how to apply this approach on the gradient method, thereby achieving a new and tight complexity result for this algorithm. In addition, we show how to apply the approach on a wide family of algorithms, which includes the fast gradient method and the heavy ball method, and show that when an analytical solution to the resulting optimization problem is not available, it is possible to efficiently approximate its solution using numerical tools. Furthermore, we show how to numerically find the best algorithm in this class, and that it has an efficiency estimate that it two times better than the known bounds on fast gradient method.

In Chapter 3, we further extend the aforementioned approach, and show how it can be used to find a new optimization method in the non-smooth case. We detail the construction of the algorithm and prove that it attains the optimal efficiency estimate on the class of Lipschitz-continuous functions. Surprisingly, the resulting algorithm turns out to be very similar to Kelley's cutting-plane method.

In Chapter 4, we suggest a new method for solving structured saddle-point problems. The method is simple and possesses some technical advantages over existing methods, such as Nesterov's smoothing technique. We present the method, establish its efficiency estimate and demonstrate its effectiveness on some practical problems. In addition, we demonstrate how to apply the technique introduced in the previous chapters in order to derive a numerical bound on the efficiency of this method.

Finally, in Chapter 5, we consider a class non-convex quadratically constrained quadratic problems. By a refined examination on the problem structure, we derive an improved characterization on the situation where the convex semidefinite relaxation provides an exact solution. We demonstrate the usefulness of the results both in practice and in theory by several examples.

Chapter 1

Introduction

Consider the following (unconstrained) minimization problem

$$(P) \quad f^* = \inf_{x \in \mathbb{R}^d} f(x).$$

A *first-order* method is an iterative algorithm that approximates the solution of (P) by generating a sequence of points $\{x_i \in \mathbb{R}^d : i = 0, \ldots, N\}$, where the algorithm can only gain information on the objective f by evaluating it and its gradient at the selected points. The performance or *efficiency estimate* of a first-order method on a given family of functions is often measured as the worst-case absolute inaccuracy, i.e., $f(x_N) - f^*$, over all possible functions f in the given family, where the distance from the starting point of the algorithm to an optimal solution is assumed to be bounded. For the formal definitions we refer the reader to [23, Chapter 5].

The earliest and arguably the most fundamental first-order method is the gradient method, defined by the rule $x_{i+1} = x_i - h_i \nabla f(x_i)$ for adequately chosen step-sizes h_i. This scheme is applicable for many classes of problems, for example, when f is known to have a Lipschitz-continuous gradient with constant L, then by taking $h_i \equiv \frac{1}{L}$ the efficiency estimate of the method can be shown to be in the order of $O(1/N)$.

There have been several attempts to find a first-order method with an improved efficiency estimate, most notably with the introduction of the heavy ball method [83], the conjugate gradient methods [79], and quasi-Newton methods [39]. Although these methods perform well in practice, it was only in 1983, with the introduction of the fast gradient method by Nesterov [73], were a method with worst-case efficiency estimate if $O(1/N^2)$ was introduced. The efficiency estimate of Nesterov's algorithm is also optimal, as it is possible to show that a first-order method acting on convex functions with Lipschitz-continuous gradient cannot have an efficiency estimate with a better rate of convergence [72, 74].

As a result of these and other advances in the field, first-order methods methods have gained popularity both in theoretical optimization and in many scientific applications, such as signal and image processing, communications, machine learning, and many more. These problems are very large scale, and first-order methods, which in general involve very cheap and simple computational iterations, are often the best option to tackle such problems in a reasonable time when moderate accuracy solutions are sufficient.

As the number of applications rise and their scope widens, the importance of an accurate analysis of the optimization methods increases. On the theoretical front, an accurate analysis

might provide a deeper understanding of the operation of the optimization method and as a result can help us devise more efficient methods. On the practical front, an accurate analysis can help improve the performance of existing methods, in particular for non-convex optimization problems, where many problems are solved indirectly through approximations and therefore rely on the quality of these approximations.

In this dissertation, we address the increasing need for an accurate analysis of optimization algorithms by suggesting a new approach for performing this analysis. The approach is based on the observation that the worst-case performance of a given first-order method (and in fact, the worst-case performance of any type of method) can be posed as an optimization problem and, as a result, this problem can be solved using tools from the field of optimization. In addition, we discuss two classes of problems where the structure of the optimization problem can be used to obtain an efficient solution method.

Following is a brief overview of the main results presented in this dissertation.

1.1 Analysis of Smooth First-Order Methods

We start, in Chapter 2, by focusing on problems where the objective is convex and has a Lipschitz-continuous gradient. We discuss in detail a new approach for analyzing optimization methods and show how this approach can be used to convert the problem of finding the efficiency estimate for a first-order method into a optimization problem, which we call a Performance Estimation Problem (PEP).

We demonstrate this approach on the gradient method and find an analytical solution for the resulting PEP, thereby obtaining a new and tight efficiency estimate on the method, which is two times better than the previously known efficiency estimate.

We then broaden our attention to a wide family of first-order methods, which includes the fast gradient method and the heavy ball method, and show that when an analytical solution to the corresponding PEP is not known, it is possible to efficiently approximate its value using numerical tools.

Finally, since we have formulated the efficiency estimate of a method as an optimization problem, we can naturally express the problem of finding steps sizes which results with the best possible efficiency estimate as a minimax problem. We analyze this minimax problem and show that, after some transformations, it can be efficiently solved using standard numerical tools for any fixed N (the total number of steps the method makes). We demonstrate this result for various values of N and show that the computed method has an efficiency estimate which is approximately two times better than the efficiency estimate for Nesterov's fast gradient method.

1.2 An Optimal Method for of Non-Smooth Optimization

We turn our attention to the class of non-smooth minimization problems, where the objective is convex and Lipschitz-continuous. This class is inherently difficult, as problems in this class require, in general, at least $O(1/\varepsilon^2)$ steps in order to reach a given accuracy ε [72, 74]. As a result of this difficulty, it is important to exploit the properties of the specific function given to

the optimization procedure to achieve the fastest possible converge, especially when an accurate solution is required.

A natural approach for exploiting the structure of the objective is adopted by Kelley's cutting-plane method [53], which maintains a lower-bound model for the objective by considering the supporting hyperplanes to the objective at the past iterates and chooses the next iterate in way that minimizes the value predicted by the model. Despite its intuitive nature, this method was proved to be inefficient both in practice and in theory [72], where the source of the poor performance seems to be the instability of the solution. This observation inspired the introduction of several methods, including the successful bundle method [61], which introduces a form of regularization in order to "motivate" the next iterate to remain close to a previous iterates, where the model is more accurate.

In Chapter 3, we develop a new method that exploits the structure of the objective by choosing the next iterate according to a model that, in addition to the supporting hyperplanes to the objective, includes a novel type of regularization. In order to derive the method, we further extend the approach proposed in the previous chapter and show how it can be used to construct an efficient and practical method. The resulting method turns out to be surprisingly similar to Kelley's cutting-plane method, yet it attains the best possible efficiency estimate on this class of problems.

1.3 Saddle-Point Problems

Another important class of problems of the form (P) is the class of convex-concave saddle-point problems

$$\text{(M)} \qquad \min_{u\in U}\max_{v\in V}\{K(u,v) := f(u) + \langle Au, v\rangle - g(v)\},$$

where f and g are convex functions, A is a linear operator, and U, V are convex sets. Historically, the first approaches for solving saddle-point problems considered the problem via the the more general framework of variational inequalities, and the problem was then solved using methods designed for solving such problems, such as the extragradient method [59], which was shown in [71] to require $O(1/\varepsilon)$ iterations to achieve a given accuracy $\varepsilon > 0$. The main difficulty with this approach is that when the objective K is not differentiable, known methods either require $O(1/\varepsilon^2)$ iterations to achieve a given accuracy ε or must make some additional assumptions on the problem structure (see, for example, [70, 77] and references therein).

A novel approach in solving non-differentiable saddle-point problems was suggested by Nesterov in [76], where he developed a smoothing technique that specifically exploits the structure of problem (M) thereby achieving a method with an $O(1/\varepsilon)$ efficiency estimate. The smoothing approach assumes that the function f is differentiable and that g is possibly non-differentiable, but is relatively "simple". The disadvantage in Nesterov's smoothing approach is that it requires the user to choose the desired target accuracy before starting the optimization process thereby not allowing the method to take advantage of favourable problem instances. As an attempt to rectify this problem, Nesterov proposed an excessive gap technique, described in [75]. This approach requires both function f and g to have a "simple" structure; however, in practice it is hard to implement and was fully described only for the case where the functions f and g are the indicators functions of some convex sets.

More recently, Chambolle and Pock presented in [33] a method that successfully overcomes the latter issue with Nesterov's excessive gap technique. The method is highly successful in solving a wide variety of problems, and is easy to implement, however, it still requires both functions f and g to have a "simple" structure.

In Chapter 4, we revisit the model assumed by Nesterov's smoothing approach and propose a method that generates an approximation sequence for the problem that converges to the solution at the rate of $O(1/\varepsilon)$ without the need to choose the desired accuracy of the result before beginning the computation and without any significant impact on the computational effort. Numerical experiments on the image deblurring and fused lasso problems confirm the theory and demonstrate that our algorithm is competitive when compared to related state of art schemes.

1.4 Nonconvex Quadratic Optimization

The class of nonconvex quadratically constrained quadratic programming (QCQP) problems plays a key role in both subproblems arising in optimization algorithms such as trust region methods (see for example [31, 45]) and is also a bridge to the analysis of many combinatorial optimization problems that can be formulated as such. In principal, nonconvex QCQP problems are hard to solve, and as result many approximation techniques were devised in order to tackle it. Many of these techniques rely on the so-called semidefinite relaxation (SDR), which is a related convex problem over the matrix space that can be solved efficiently, see e.g., [51, 99].

A key issue in the analysis of QCQPs is to determine under which conditions the semidefinite relaxation is tight, meaning that it has the same optimal value as the original QCQP problem. In these cases, one can construct the global optimal solution of the QCQP problem from the optimal solution of the SDR via a rank reduction procedure. There are several classes of QCQP problems which posses this "tight semidefinite relaxation" result; among them are the class of generalized trust region subproblems [45, 67] which are QCQPs with a single quadratic constraint, problems with two constraints over the complex number field [17] as well as problems arising in the context of quadratic assignment problem [1, 2].

Another class of QCQP problems is the class of *Quadratic Matrix Programming* (QMP) problems whose general form is given by

$$\begin{aligned} \text{(QMP)} \quad \min_{X\in\mathbb{R}^{n\times r}} \ & \operatorname{tr}(X^T A_0 X) + 2\operatorname{tr}(\tilde{B}_0^T X) + c_0 \\ \text{s.t. } & \operatorname{tr}(X^T A_i X) + 2\operatorname{tr}(\tilde{B}_i^T X) + c_i \le \alpha_i, \quad i \in \mathscr{I}, \\ & \operatorname{tr}(X^T A_j X) + 2\operatorname{tr}(\tilde{B}_j^T X) + c_j = \alpha_j, \quad j \in \mathscr{E}, \end{aligned}$$

where n, r are positive integers, $\mathscr{I}$ and $\mathscr{E}$ are sets of indices such that $\mathscr{I} \cap \mathscr{E} = \emptyset$, $A_i \in \mathbb{S}^n$, $\tilde{B}_i \in \mathbb{R}^{n\times r}$ and $c_i, \alpha_i \in \mathbb{R}$. This class of problems was introduced and studied in [15] where it was also shown that it encompasses a broad class of important problems both in theory and in applications. The main result in [15] is that problem (QMP) with at most r constraints has a tight SDR property. In the homogenous case (i.e., when $\tilde{B}_i = 0$ for all i) this question was already studied by Barvinok [13, 14] for the problem of determining the feasibility of this problem; Barvinok's results were then extended by Pataki [82] to include any homogeneous quadratic objective function. In both cases it was shown that it is possible to use the SDP relaxation to solve the original nonconvex problem when the number of constraints is at most $\binom{r+2}{2} - 1$.

In Chapter 5, we concentrate on a special type of QMP problems defined by

$$\begin{array}{lll} & \min_{X\in\mathbb{R}^{n\times r}} & \mathrm{tr}(X^TA_0X)+2\,\mathrm{tr}(V^TB_0^TX)+c_0 \\ \text{(sQMP)} & \text{s.t.} & \mathrm{tr}(X^TA_iX)+2\,\mathrm{tr}(V^TB_i^TX)+c_i\le\alpha_i, \quad i\in\mathscr{I}, \\ & & \mathrm{tr}(X^TA_jX)+2\,\mathrm{tr}(V^TB_j^TX)+c_j=\alpha_j, \quad j\in\mathscr{E}, \end{array} \tag{1.4.1}$$

with $A_i\in\mathbb{S}^n$, $B_i\in\mathbb{R}^{n\times s}$ $(i\in\{0\}\cup\mathscr{I}\cup\mathscr{E})$ and $0\neq V\in\mathbb{R}^{s\times r}$, $s\le r$. Essentially, this type of QMP problems is characterized by the property that the matrices $\tilde{B}_i$ are of the special form $\tilde{B}_i=B_iV$; for the case $n>r>s$, this means that the range spaces of the $n\times r$ matrices $\tilde{B}_i$, $(i\in\{0\}\cup\mathscr{I}\cup\mathscr{E})$ are all contained in the same s-dimensional subspace, which is the range space of V. Note that when $s=r$ and $V=I_r$ we are back to the original QMP setting. At a first glance it seems that this property of the matrices $\tilde{B}_i$ is quite restrictive, however, it naturally appears in applications, as described in §5.4.

Chapter 2

A new approach for analyzing optimization algorithms

We introduce a novel approach for analyzing the worst-case performance of first-order black-box optimization methods. We focus on smooth unconstrained convex minimization over the Euclidean space. Our approach relies on the observation that by definition, the worst-case behavior of a black-box optimization method is by itself an optimization problem, which we call the Performance Estimation Problem (PEP). We formulate and analyze the PEP for two classes of first-order algorithms. We first apply this approach on the classical gradient method and derive a new and tight analytical bound on its performance. We then consider a broader class of first-order black-box methods, which among others, include the so-called heavy-ball method and the fast gradient schemes. We show that for this broader class, it is possible to derive new bounds on the performance of these methods by solving an adequately relaxed convex semidefinite PEP. Finally, we show an efficient procedure for finding optimal step sizes which results in a first-order black-box method that achieves best worst-case performance.

This chapter is based on the published paper [40].

2.1 Introduction

First-order convex optimization methods have recently gained in popularity both in theoretical optimization and in many scientific applications, such as signal and image processing, communications, machine learning, and many more. These problems are very large scale, and first-order methods, which in general involve very cheap and simple computational iterations, are often the best option to tackle such problems in a reasonable time, when moderate accuracy solutions are sufficient. For convex optimization problems, there exists an extensive literature on the development and analysis of first-order methods, and in recent years, this has been revitalized at a quick pace due to the emergence of many fundamental new applications alluded above. On the theoretical front see e.g., the recent works [47, 60, 85] and for applications see the collections [80, 92] and references therein.

This work is not on the development of new algorithms, rather it focuses on the theoretical performance analysis of first-order methods for unconstrained minimization with an objective function which is known to belong to a given family $\mathscr{F}$ of smooth convex functions over the

Euclidean space $\mathbb{R}^d$.

Following the seminal work of Nemirovsky and Yudin [72] in the complexity analysis of convex optimization methods, we measure the computational cost based on the oracle model of optimization. According to this model, a *first-order black-box* optimization method is an algorithm $\mathscr{A}$ which has knowledge of the underlying space $\mathbb{R}^d$ and the family $\mathscr{F}$, where the function itself is not known. To gain information on the objective function f to be minimized, the algorithm queries a first-order oracle, that is, a subroutine which given as input a point in $\mathbb{R}^d$, returns the value of the objective function and its gradient at that point. The algorithm starts with a given point $x_0 \in \mathbb{R}^d$ and generates a finite sequence of points $\{x_i \in \mathbb{R}^d : i = 1,\dots,N\}$, where at each step the algorithm can depend only on the previous steps, their function values and gradients via some rule

$$x_{i+1} = \mathscr{A}(x_0,\dots,x_i; f(x_0),\dots,f(x_i); f'(x_0),\dots,f'(x_i)),\ i = 0,\dots,N-1,$$

where $f'(\cdot)$ stands for the gradient of $f(\cdot)$. Note that the algorithm has another implicit knowledge, i.e., that the distance from its initial point x_0 to a minimizer $x_* \in X_*(f)$ of f is bounded by some constant $R > 0$, see more precise definitions in the next section.

Given a desired accuracy $\varepsilon > 0$, applying the given algorithm on the function f in the class $\mathscr{F}$, the algorithm stops when it produces an approximate solution x_ε which is ε-optimal, that is such that

$$f(x_\varepsilon) - f(x_*) \leq \varepsilon.$$

The worst-case performance (or complexity) of a first-order black-box optimization algorithm is then measured by the number of oracle calls the algorithm needs to find such an approximate solution. Equivalently, we can measure the worst-case performance of an algorithm by looking at the absolute inaccuracy

$$\delta(f, x_N) = f(x_N) - f(x_*),$$

where x_N is the result of the algorithm after making N calls to the oracle. Throughout this chapter we will use the latter form to measure the performance of a given algorithm.

Building on this model, in this work we introduce a novel approach for analyzing the performance of a given first-order scheme. Our approach relies on the observation that by definition, the worst-case behavior of a first-order black-box optimization algorithm is by itself an optimization problem which consists of finding the maximal absolute inaccuracy over all possible inputs to the algorithm. Thus, with x_N being the output of the algorithm after making N calls to the oracle, we look at the solution of the following *Performance Estimation Problem* (PEP):

$$\begin{aligned}
&\max\ f(x_N) - f(x_*)\\
&\text{s.t. } f \in \mathscr{F},\\
&\quad x_{i+1} = \mathscr{A}(x_0,\dots,x_i; f(x_0),\dots,f(x_i); f'(x_0),\dots,f'(x_i)),\ i = 0,\dots,N-1,\\
&\quad x_* \in X_*(f),\ \|x_* - x_0\| \leq R,\\
&\quad x_0,\dots,x_N,x_* \in \mathbb{R}^d.
\end{aligned} \tag{P}$$

At first glance this problem seems very hard or impossible to solve. We overcome this difficulty through an analysis that relies on various types of relaxations, including duality and semi-definite relaxation techniques. The problem setting and an outline of the underlying idea

of the proposed approach for analyzing (P) are described in Section 2.2. In order to develop the basic idea and tools underlying our proposed approach, we first focus on the fundamental gradient method (GM) for smooth convex minimization, and then extend it to a broader class of first-order black-box minimization methods. Obviously, the gradient method is a particular case of this broader class that will be analyzed below. However, it is quite important to start with the gradient method for two reasons. First, it allows to acquaint the reader in a more transparent way with the techniques and methodology we need to develop in order to analyze (P), thus paving the way to tackle more general schemes. Secondly, for the gradient method, we are able to prove a new and tight bound on its performance which is given *analytically*, see Section 2.3. Capitalizing on the methodology and tools developed in the past section, in Section 2.4, we consider a broader class of first-order black-box methods, which among others, is shown to include the so-called heavy-ball [83] and fast gradient schemes [74]. Although an analytical solution is not available for this general case, we show that for this broader class of methods, it is possible to compute *numerical estimates* for an adequate relaxation of the corresponding PEP, allowing to derive new bounds on the performance of these methods. We then derive in Section 2.5 an efficient procedure for finding optimal step sizes which results in a first-order method that achieves best worst-case performance. Our approach and analysis give rise to some interesting problems leading us to suggest some conjectures. We conclude with three appendices: the first includes the proof of a technical result, the second demonstrates our approach on the projected gradient method, and the third appendix presents some preliminary results on strongly convex functions.

Notation. For a differentiable function f, its gradient at x is denoted by $f'(x)$. The Euclidean norm of a vector $x \in \mathbb{R}^d$ is denoted as $\|x\|$. The set of symmetric matrices in $\mathbb{R}^{n\times n}$ is denoted by $\mathbb{S}^n$. For two symmetric matrices A and B, $A \succeq B, (A \succ B)$ means $A - B \succeq 0\,(A - B \succ 0)$ is positive semidefinite (positive definite). We use e_i for the i-th canonical basis vector in $\mathbb{R}^N$, which consists of all zero components, except for its i-th entry which is equal to one, and use v to denote a unit vector in $\mathbb{R}^d$. For an optimization problem (P), val(P) stands for its optimal value.

2.2 The Problem and the Main Approach

2.2.1 The Problem and Basic Assumptions

Let $\mathscr{A}$ be a first-order algorithm for solving the optimization problem

$$(M) \quad \min\{f(x) : x \in \mathbb{R}^d\}.$$

Throughout the chapter we make the following assumptions:

- $f : \mathbb{R}^d \to \mathbb{R}$ is a convex function of the type $C_L^{1,1}(\mathbb{R}^d)$, i.e., continuously differentiable with Lipschitz continuous gradient:

  $$\|f'(x) - f'(y)\| \leq L\|x - y\|,\ \forall x, y \in \mathbb{R}^d,$$

  where $L > 0$ is the Lipschitz constant.

- We assume that (M) is solvable, i.e., the optimal set $X_*(f) := \arg\min f$ is nonempty.
- There exists $R > 0$, such that the distance from the given starting point of the algorithm x_0 to an optimal solution $x_* \in X_*(f)$ is bounded by R.[1]

Given a convex function f in the class $C_L^{1,1}(\mathbb{R}^d)$ and any starting point $x_0 \in \mathbb{R}^d$, the algorithm $\mathscr{A}$ is a first-order black-box scheme, i.e., it is allowed to access f only through the sequential calls to the first-order oracle that returns the value and the gradient of f at any input point x. The algorithm $\mathscr{A}$ then generates a sequence of points $x_i \in \mathbb{R}^d$, $i = 0, \ldots, N$.

2.2.2 Basic Idea and Main Approach

We are interested in measuring the worst-case behavior of a given algorithm $\mathscr{A}$ in terms of the absolute inaccuracy $f(x_N) - f(x_*)$, by solving problem (P) defined in the introduction, namely

$$\begin{aligned} &\max\ f(x_N) - f(x_*) \\ &\text{s.t. } f \in C_L^{1,1}(\mathbb{R}^d),\ f \text{ is convex}, \\ &\quad x_{i+1} = \mathscr{A}(x_0, \ldots, x_i; f(x_0), \ldots, f(x_i); f'(x_0), \ldots, f'(x_i)),\ i = 0, \ldots, N-1, \\ &\quad x_* \in X_*(f), \|x_* - x_0\| \le R, \\ &\quad x_0, \ldots, x_N, x_* \in \mathbb{R}^d. \end{aligned} \tag{P}$$

To tackle this problem, we suggest to perform a series of relaxations thereby reaching a tractable optimization problem.

A main difficulty in problem (P) lies in the functional constraint (the variable f is a convex function in $C_L^{1,1}(\mathbb{R}^d)$), i.e., we are facing an abstract hard optimization problem in infinite dimensions. To overcome this difficulty, the approach taken in this chapter is to *relax* this constraint so that the problem can be reduced and formulated as an explicit finite dimensional problem that can eventually be adequately analyzed.

An informal description of the underlying idea consists of two main steps as follows:

- Given an algorithm $\mathscr{A}$ that generates a finite sequence of points, to build a problem in finite dimensions we replace the functional constraint $f \in C_L^{1,1}$ in (P) by new variables in $\mathbb{R}^d$. These variables are the points $\{x_0, x_1, \ldots x_N, x_*\}$ themselves, the function values and their gradients at these points. Roughly speaking, this can be seen as a sort of discretization of f at a selected set of points.
- To define constraints that relate the new variables, we use relevant/useful properties characterizing the family of convex functions in $C_L^{1,1}$, as well as the rule(s) describing the given algorithm $\mathscr{A}$.

This approach can, in principle, be applied to any optimization algorithm. Note that any relaxation performed on the maximization problem (P) may increase its optimal value, however,

[1] In general, the terms L and R are unknown or difficult to compute, in which case some upper bound estimates can be used in place. Note that all currently known complexity results for first-order methods depend on L and R.

the optimal value of the relaxed problem still remains a valid upper bound on $f(x_N) - f(x_*)$. Also note that once a bound on the absolute inaccuracy has been established, it is possible to find a bound that does not depend on the unknown term $f(x_*)$, e.g., from the well-known property $\|f'(x_N)\|^2 \leq f(x_N) - f(x_*)$.

A formal description on how this approach can be applied to the gradient method is described in the next section, which as we shall see, allows us to derive a new tight bound on the performance of the gradient method.

2.3 An Analytical Bound for the Gradient Method

To develop the basic idea and tools underlying the proposed approach for analyzing the performance of iterative optimization algorithms, in this section we focus on the simplest fundamental method for smooth convex minimization, the *Gradient Method* (GM). It will also pave the way to tackle more general first-order schemes as developed in the forthcoming sections.

2.3.1 A Performance Estimation Problem for the Gradient Method

Consider the gradient algorithm with constant step size, as applied to problem (M), which generates a sequence of points as follows:

Algorithm GM

0. Input: $f \in C_L^{1,1}(\mathbb{R}^d)$ convex, $x_0 \in \mathbb{R}^d$.
1. For $i = 0, \ldots, N-1$, compute $x_{i+1} = x_i - \frac{h}{L} f'(x_i)$.

Here $h > 0$ is fixed. Note that while simple, this algorithm is restricted to problems where the Lipschitz constant L is known or can be efficiently estimated.

At this point, recall that for $h = 1$, the convergence rate of the Algorithm GM can be shown to be (see for example [19, 74]):

$$f(x_N) - f(x_*) \leq \frac{L\|x_0 - x_*\|^2}{2N}, \quad \forall x_* \in X_*(f). \tag{2.3.1}$$

We begin our analysis with a well-known fundamental property for the class of convex $C_L^{1,1}$ functions, see e.g., [74, Theorem 2.1.5].

Proposition 2.3.1. Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and $C_L^{1,1}$. Then,

$$\frac{1}{2L}\|f'(x) - f'(y)\|^2 \leq f(x) - f(y) - \langle f'(y), x - y\rangle, \text{ for all } x, y \in \mathbb{R}^d. \tag{2.3.2}$$

Let $x_0 \in \mathbb{R}^d$ be any starting point, let $\{x_1, \ldots, x_N\}$ be the points generated by Algorithm GM and let x_* be a minimizer of f. Applying (2.3.2) on the points $\{x_0, \ldots, x_N, x_*\}$, we get:

$$\frac{1}{2L}\|f'(x_i) - f'(x_j)\|^2 \leq f(x_i) - f(x_j) - \langle f'(x_j), x_i - x_j\rangle, \quad i,j = 0,\ldots,N,*. \tag{2.3.3}$$

Now define

$$\delta_i := \frac{1}{L\|x_* - x_0\|^2}(f(x_i) - f(x_*)), \quad i = 0,\ldots,N,*$$
$$g_i := \frac{1}{L\|x_* - x_0\|} f'(x_i), \quad i = 0,\ldots,N,*$$

and note that we always have $\delta_* = 0$ and $g_* = 0$.

In terms of δ_i, g_i, condition (2.3.3) becomes

$$\frac{1}{2}\|g_i - g_j\|^2 \leq \delta_i - \delta_j - \frac{\langle g_j, x_i - x_j\rangle}{\|x_* - x_0\|}, \quad i,j = 0,\ldots,N,*, \tag{2.3.4}$$

and the recurrence defining Algorithm GM reads:

$$x_{i+1} = x_i - h\|x_* - x_0\| g_i, \quad i = 0,\ldots,N-1.$$

Problem (P) can now be relaxed by *discarding* the underlying function $f \in C_L^{1,1}$ in (P). That is, the constraint in the function space $f \in C_L^{1,1}$ with f convex, is replaced by the inequalities (2.3.4) characterizing this family of functions and expressed in terms of the variables $x_0,\ldots,x_N,x_* \in \mathbb{R}^d$, $g_0,\ldots,g_N \in \mathbb{R}^d$ and $\delta_0,\ldots,\delta_N \in \mathbb{R}$ generated by Algorithm GM. Thus, an upper bound on the worst-case behavior of $f(x_N) - f(x_*) = L\|x_* - x_0\|^2\delta_N$ can be obtained by solving the following relaxed PEP:

$$\begin{aligned}
\max_{\substack{x_0,\ldots,x_N,x_* \in \mathbb{R}^d,\\ g_0,\ldots,g_N \in \mathbb{R}^d,\\ \delta_0,\ldots,\delta_N \in \mathbb{R}}} \quad & L\|x_* - x_0\|^2\delta_N \\
\text{s.t.} \quad & x_{i+1} = x_i - h\|x_* - x_0\| g_i, \quad i = 0,\ldots,N-1, \\
& \frac{1}{2}\|g_i - g_j\|^2 \leq \delta_i - \delta_j - \frac{\langle g_j, x_i - x_j\rangle}{\|x_* - x_0\|}, \quad i,j = 0,\ldots,N,*, \\
& \|x_* - x_0\| \leq R.
\end{aligned}$$

Simplifying the PEP The obtained problem remains nontrivial to tackle. We will now perform some simplifications on this problem that will be useful for the forthcoming analysis.

First, we observe that the problem is invariant under the transformation $g_i' \leftarrow Qg_i$, $x_i' \leftarrow Qx_i$ for any orthogonal transformation Q. We can therefore assume without loss of generality that $x_* - x_0 = \|x_* - x_0\|\nu$, where ν is any given unit vector in $\mathbb{R}^d$. Therefore, for $i = *$ the inequality constraints reads

$$\frac{1}{2}\|g_* - g_j\|^2 \leq \delta_* - \delta_j - \frac{\langle g_j, \|x_* - x_0\|\nu + x_0 - x_j\rangle}{\|x_* - x_0\|}, \quad j = 0,\ldots,N.$$

Secondly, we consider (2.3.4) for the four cases $i = *$, $j = *$, $i < j$ and $j < i$, and use the equality constraints

$$x_{i+1} = x_i - h\|x_* - x_0\|g_i, \quad i = 0,\dots,N-1$$

to eliminate the variables $x_1,\dots,x_N$. After some algebra, we reach the following form for the PEP:

$$\begin{aligned}
\max_{x_0,x_*,g_i\in\mathbb{R}^d,\delta_i\in\mathbb{R}} \; & L\|x_* - x_0\|^2\delta_N \\
\text{s.t. } & \frac{1}{2}\|g_i - g_j\|^2 \le \delta_i - \delta_j - \langle g_j, \sum_{t=i+1}^{j} hg_{t-1}\rangle, \quad i < j = 0,\dots,N, \\
& \frac{1}{2}\|g_i - g_j\|^2 \le \delta_i - \delta_j + \langle g_j, \sum_{t=j+1}^{i} hg_{t-1}\rangle, \quad j < i = 0,\dots,N, \\
& \frac{1}{2}\|g_i\|^2 \le \delta_i, \quad i = 0,\dots,N, \\
& \frac{1}{2}\|g_i\|^2 \le -\delta_i - \langle g_i, v + \sum_{t=1}^{i} hg_{t-1}\rangle, \quad i = 0,\dots,N, \\
& \|x_* - x_0\| \le R,
\end{aligned}$$

where $i < j = 0,\dots,N$ is a shorthand notation for $i = 0,\dots,N-1, j = i+1,\dots,N$.

Finally, we note that the optimal solution for this problem is attained when $\|x_* - x_0\| = R$, and hence we can also eliminate the variables x_0 and x_*. This produces the following PEP for the gradient method, a *nonconvex quadratic* minimization problem:

$$\begin{aligned}
\max_{g_i\in\mathbb{R}^d,\delta_i\in\mathbb{R}} \; & LR^2\delta_N \\
\text{s.t. } & \frac{1}{2}\|g_i - g_j\|^2 \le \delta_i - \delta_j - \langle g_j, \sum_{t=i+1}^{j} hg_{t-1}\rangle, \quad i < j = 0,\dots,N, \\
& \frac{1}{2}\|g_i - g_j\|^2 \le \delta_i - \delta_j + \langle g_j, \sum_{t=j+1}^{i} hg_{t-1}\rangle, \quad j < i = 0,\dots,N, \\
& \frac{1}{2}\|g_i\|^2 \le \delta_i, \quad i = 0,\dots,N, \\
& \frac{1}{2}\|g_i\|^2 \le -\delta_i - \langle g_i, v + \sum_{t=1}^{i} hg_{t-1}\rangle, \quad i = 0,\dots,N.
\end{aligned}$$

This problem can be written in a more compact and useful form. Let G denote the $(N+1)\times d$ matrix whose rows are $g_0^T,\dots g_N^T$, and for notational convenience let $u_i \in \mathbb{R}^{N+1}$ denote the canonical unit vector

$$u_i = e_{i+1}, \quad i = 0,\dots,N.$$

Then for any i, j, we have

$$g_i = G^T u_i,\ \mathrm{tr}(G^T u_i u_j^T G) = \langle g_i, g_j\rangle,\ \text{and } \langle G^T u_i, v\rangle = \langle g_i, v\rangle.$$

Therefore, by defining the following $(N+1)\times(N+1)$ symmetric matrices

$$\begin{aligned}
A_{i,j} &= \frac{1}{2}(u_i-u_j)(u_i-u_j)^T + \frac{1}{2}\sum_{t=i+1}^{j} h(u_j u_{t-1}^T + u_{t-1}u_j^T),\\
B_{i,j} &= \frac{1}{2}(u_i-u_j)(u_i-u_j)^T - \frac{1}{2}\sum_{t=j+1}^{i} h(u_j u_{t-1}^T + u_{t-1}u_j^T),\\
C_i &= \frac{1}{2}u_i u_i^T,\\
D_i &= \frac{1}{2}u_i u_i^T + \frac{1}{2}\sum_{t=1}^{i} h(u_i u_{t-1}^T + u_{t-1}u_i^T),
\end{aligned} \tag{2.3.5}$$

we can express our nonconvex quadratic minimization problem in terms of $\delta := (\delta_0,\ldots,\delta_N) \in \mathbb{R}^{N+1}$ and the new matrix variable $G \in \mathbb{R}^{(N+1)\times d}$ as follows

$$\begin{aligned}
\max_{G\in\mathbb{R}^{(N+1)\times d},\,\delta\in\mathbb{R}^{N+1}} \quad & LR^2\delta_N\\
\text{s.t.} \quad & \mathrm{tr}(G^T A_{i,j}G) \le \delta_i - \delta_j, \quad i<j=0,\ldots,N,\\
& \mathrm{tr}(G^T B_{i,j}G) \le \delta_i - \delta_j, \quad j<i=0,\ldots,N,\\
& \mathrm{tr}(G^T C_i G) \le \delta_i, \quad i=0,\ldots,N,\\
& \mathrm{tr}(G^T D_i G + \nu u_i^T G) \le -\delta_i, \quad i=0,\ldots,N.
\end{aligned} \tag{G}$$

Problem (G) is a nonhomogeneous *Quadratic Matrix Program*, a class of problems introduced and studied by Beck [15] and will be further studied in Chapter 5.

2.3.2 A Tight Performance Estimate for the Gradient Method

We now proceed to establish the two main results of this section. First, we derive an upper bound on the performance of the gradient method; this is accomplished using duality arguments. Then, we show that this bound can actually be attained by applying the gradient method to a specific convex function in the class $C_L^{1,1}$.

In order to simplify the following analysis, we will remove some constraints from (G) and consider the bound produced by the following relaxed problem:

$$\begin{aligned}
\max_{G\in\mathbb{R}^{(N+1)\times d},\,\delta\in\mathbb{R}^{N+1}} \quad & LR^2\delta_N\\
\text{s.t.} \quad & \mathrm{tr}(G^T A_{i-1,i}G) \le \delta_{i-1} - \delta_i, \quad i=1,\ldots,N,\\
& \mathrm{tr}(G^T D_i G + \nu u_i^T G) \le -\delta_i, \quad i=0,\ldots,N.
\end{aligned} \tag{G$'$}$$

As we shall show below, it turns out that this additional relaxation has no damaging effects and produces the desired performance bound when $0 < h \le 1$.

We are interested in deriving a dual problem for (G$'$) which is as simple as possible, especially with respect to its dimension. As noted earlier, problem (G$'$) is a nonhomogeneous quadratic matrix program, and a dual problem for (G$'$) could be directly obtained by applying

the results developed by Beck [15]. However, the resulting obtained dual will involve an additional matrix variable $\Phi \in \mathbb{S}^d$, where d can be very large. Instead, by exploiting the special structure of the second set of nonhomogeneous inequalities given in (G′), we derive here an alternative dual problem, but with only one additional variable $t \in \mathbb{R}$.

To establish our dual result, the next lemma shows that a dimension reduction is possible when minimizing a quadratic matrix function sharing the special form as the one that appears in problem (G′).

Lemma 2.1. *Let $f(X) = \mathrm{tr}(X^T QX + 2ba^T X)$ be a quadratic function, where $X \in \mathbb{R}^{n\times m}$, $Q \in \mathbb{S}^n$, $a \in \mathbb{R}^n$ and $0 \neq b \in \mathbb{R}^m$. Then*

$$\inf_{X\in\mathbb{R}^{n\times m}} f(X) = \inf_{\xi\in\mathbb{R}^n} f(\xi b^T).$$

Proof. First, we recall (this can be easily verified) that $\inf\{f(X) : X \in \mathbb{R}^{n\times m}\} > -\infty$ if and only if $Q \succeq 0$, and there exists at least one solution $\bar{X}$ such that

$$Q\bar{X} + ab^T = 0 \;\Leftrightarrow\; \bar{X}^T Q + ba^T = 0, \tag{2.3.6}$$

i.e., the above is just $\nabla f(X) = 0$ and characterizes the minimizers of the convex function $f(X)$. Using (2.3.6) it follows that $\inf_X f(X) = f(\bar{X}) = \mathrm{tr}(ba^T\bar{X})$. Now, for any $\xi \in \mathbb{R}^n$, we have $f(\xi b^T) = \|b\|^2(\xi^T Q\xi + 2a^T\xi)$. Thus, likewise, $\inf\{f(\xi b^T) : \xi \in \mathbb{R}^n\} > -\infty$ if and only if $Q \succeq 0$ and there exists $\bar{\xi} \in \mathbb{R}^n$ such that

$$Q\bar{\xi} + a = 0, \tag{2.3.7}$$

and using (2.3.7) it follows $\inf_\xi f(\xi b^T) = f(\bar{\xi} b^T) = \|b\|^2 a^T\bar{\xi} = \mathrm{tr}(ba^T\bar{\xi}b^T)$. Now, using (2.3.6)-(2.3.7), one obtains $\bar{X}^T Q = -ba^T$ and $Q(\bar{X} - \bar{\xi}b^T) = 0$, and hence it follows that

$$\begin{aligned} f(\bar{X}) - f(\bar{\xi}b^T) &= \mathrm{tr}(ba^T(\bar{X} - \bar{\xi}b^T)) \\ &= \mathrm{tr}(-\bar{X}^T Q(\bar{X} - \bar{\xi}b^T)) = 0. \end{aligned}$$

□

Equipped with Lemma 2.1, we now derive a Lagrangian dual for problem (G′).

Lemma 2.2. *Consider problem* (G′) *for any fixed $h \in \mathbb{R}$ and $L, R > 0$. A Lagrangian dual of* (G′) *is given by the following convex program:*

$$\min_{\lambda\in\mathbb{R}^N, t\in\mathbb{R}} \{\frac{1}{2}LR^2 t : \lambda \in \Lambda,\, S(\lambda, t) \succeq 0\}, \tag{DG′}$$

where $\Lambda := \{\lambda \in \mathbb{R}^N : \lambda_{i+1} - \lambda_i \geq 0, \quad i = 1,\ldots,N-1,\ 1 - \lambda_N \geq 0,\ \lambda_i \geq 0, \quad i = 1,\ldots,N\}$, the matrix $S(\cdot,\cdot) \in \mathbb{S}^{N+2}$ is given by

$$S(\lambda, t) = \begin{pmatrix} (1-h)S_0 + hS_1 & q \\ q^T & t \end{pmatrix},$$

with $q = (\lambda_1, \lambda_2 - \lambda_1, \ldots, \lambda_N - \lambda_{N-1}, 1 - \lambda_N)^T$ *and where the matrices* $S_0, S_1 \in \mathbb{S}^{N+1}$ *are defined by:*

$$S_0 = \begin{pmatrix} 2\lambda_1 & -\lambda_1 & & & & \\ -\lambda_1 & 2\lambda_2 & -\lambda_2 & & & \\ & -\lambda_2 & 2\lambda_3 & -\lambda_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -\lambda_{N-1} & 2\lambda_N & -\lambda_N \\ & & & & -\lambda_N & 1 \end{pmatrix} \tag{2.3.8}$$

and

$$S_1 = \begin{pmatrix} 2\lambda_1 & \lambda_2 - \lambda_1 & \ldots & \lambda_N - \lambda_{N-1} & 1 - \lambda_N \\ \lambda_2 - \lambda_1 & 2\lambda_2 & & \lambda_N - \lambda_{N-1} & 1 - \lambda_N \\ \vdots & & \ddots & & \vdots \\ \lambda_N - \lambda_{N-1} & \lambda_N - \lambda_{N-1} & & 2\lambda_N & 1 - \lambda_N \\ 1 - \lambda_N & 1 - \lambda_N & \ldots & 1 - \lambda_N & 1 \end{pmatrix}. \tag{2.3.9}$$

Proof. For convenience, we recast (G′) as a minimization problem, and we also omit the fixed term LR^2 from the objective. That is, we consider the equivalent problem (G″) defined by

$$\begin{aligned} \min_{G \in \mathbb{R}^{(N+1)\times d}, \delta \in \mathbb{R}^{N+1}} \quad & -\delta_N \\ \text{s.t. } & \operatorname{tr}(G^T A_{i-1,i} G) \le \delta_{i-1} - \delta_i, \quad i = 1, \ldots, N, \\ & \operatorname{tr}(G^T D_i G + \nu u_i^T G) \le -\delta_i, \quad i = 0, \ldots, N. \end{aligned} \tag{G″}$$

Attaching the dual multipliers $\lambda = (\lambda_1, \ldots, \lambda_N) \in \mathbb{R}^N_+$ and $\tau := (\tau_0, \ldots, \tau_N)^T \in \mathbb{R}^{N+1}_+$ to the first and second set of inequalities respectively, and using the notation $\delta = (\delta_0, \ldots, \delta_N)$, we get that the Lagrangian of this problem is given as a sum of two separable functions in the variables (δ, G):

$$\begin{aligned} L(G, \delta, \lambda, \tau) &= -\delta_N + \sum_{i=1}^N \lambda_i (\delta_i - \delta_{i-1}) + \sum_{i=0}^N \tau_i \delta_i \\ &\quad + \sum_{i=1}^N \lambda_i \operatorname{tr}(G^T A_{i-1,i} G) + \sum_{i=0}^N \tau_i \operatorname{tr}(G^T D_i G + \nu u_i^T G) \\ &\equiv L_1(\delta, \lambda, \tau) + L_2(G, \lambda, \tau). \end{aligned}$$

The dual objective function is then defined by

$$H(\lambda, \tau) = \min_{G, \delta} L(G, \delta, \lambda\, \tau) = \min_{\delta} L_1(\delta, \lambda, \tau) + \min_{G} L_2(G, \lambda, \tau),$$

and the dual problem of (G″) is then given by

$$\max\{H(\lambda, \tau) : \lambda \in \mathbb{R}^N_+, \tau \in \mathbb{R}^{N+1}_+\}. \tag{DG″}$$

Since $L_1(\cdot, \lambda, \tau)$ is linear in δ, we have $\min_\delta L_1(\delta, \lambda, \tau) = 0$ whenever

$$\begin{aligned} -\lambda_1 + \tau_0 &= 0, \\ \lambda_i - \lambda_{i+1} + \tau_i &= 0 \quad (i = 1, \ldots, N-1), \\ -1 + \lambda_N + \tau_N &= 0, \end{aligned} \tag{2.3.10}$$

and $-\infty$ otherwise. Invoking Lemma 2.1, we get

$$\min_{G\in\mathbb{R}^{(N+1)\times d}} L_2(G,\lambda,\tau) = \min_{w\in\mathbb{R}^{N+1}} L_2(wv^T,\lambda,\tau).$$

Therefore for any (λ,τ) satisfying (2.3.10), we have obtained that the dual objective reduces to

$$\begin{aligned} H(\lambda,\tau) &= \min_{w\in\mathbb{R}^{N+1}} \{w^T\left(\sum_{i=1}^N \lambda_i A_{i-1,i} + \sum_{i=0}^N \tau_i D_i\right) w + \tau^T w\} \\ &= \max_{t\in\mathbb{R}} \{-\frac{1}{2}t : w^T\left(\sum_{i=1}^N \lambda_i A_{i-1,i} + \sum_{i=0}^N \tau_i D_i\right) w + \tau^T w \le -\frac{1}{2}t,\ \forall w\in\mathbb{R}^{N+1}\} \\ &= \max_{t\in\mathbb{R}} \left\{-\frac{1}{2}t : \begin{pmatrix} \sum_{i=1}^N \lambda_i A_{i-1,i} + \sum_{i=0}^N \tau_i D_i & \frac{1}{2}\tau \\ \frac{1}{2}\tau^T & \frac{1}{2}t \end{pmatrix} \succeq 0\right\}. \end{aligned}$$

where the last equality follows from the well known lemma [23, Page 163][2].

Now, recalling the definition of the matrices $A_{i-1,i}, D_i$ (see (2.3.5)), we obtain

$$\sum_{i=1}^N \lambda_i A_{i-1,i} = \frac{1}{2}\begin{pmatrix} \lambda_1 & (h-1)\lambda_1 & & & & \\ (h-1)\lambda_1 & \lambda_1+\lambda_2 & (h-1)\lambda_2 & & & \\ & (h-1)\lambda_2 & \lambda_2+\lambda_3 & (h-1)\lambda_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & (h-1)\lambda_{N-1} & \lambda_{N-1}+\lambda_N & (h-1)\lambda_N \\ & & & & (h-1)\lambda_N & \lambda_N \end{pmatrix}$$

and

$$\sum_{i=0}^N \tau_i D_i = \frac{1}{2}\begin{pmatrix} \tau_0 & h\tau_1 & \dots & h\tau_{N-1} & h\tau_N \\ h\tau_1 & \tau_1 & & h\tau_{N-1} & h\tau_N \\ \vdots & & \ddots & & \vdots \\ h\tau_{N-1} & h\tau_{N-1} & & \tau_{N-1} & h\tau_N \\ h\tau_N & h\tau_N & \dots & h\tau_N & \tau_N \end{pmatrix}.$$

Finally, using the relations (2.3.10) to eliminate τ_i, and recalling that val(G″) was defined as $-LR^2$ val(G′), the desired form of the stated dual problem follows. □

The next lemma will be crucial in invoking duality in the forthcoming theorem. The proof for this lemma is quite technical and appears in the appendix.

Lemma 2.3. *Let*

$$\lambda_i = \frac{i}{2N+1-i}, \qquad i = 1,\dots,N,$$

then the matrices $S_0, S_1 \in \mathbb{S}^{N+1}$ defined in (2.3.8)–(2.3.9) are positive definite for every $N\in\mathbb{N}$.

[2]Let M be a symmetric matrix. Then, $x^TMx + 2b^Tx + c \ge 0, \forall x\in\mathbb{R}^d$ if and only if the matrix $\begin{pmatrix} M & b \\ b^T & c \end{pmatrix}$ is positive semidefinite.

We are now ready to establish a new upper bound on the complexity of the gradient method for values of h between 0 and 1. To the best of our knowledge, the tightest bound thus far is given by (2.3.1).

Theorem 2.4. *Let $f \in C_L^{1,1}(\mathbb{R}^d)$ and let $x_0, \ldots, x_N \in \mathbb{R}^d$ be generated by Algorithm GM with $0 < h \leq 1$. Then*

$$f(x_N) - f(x_*) \leq \frac{LR^2}{4Nh+2}. \tag{2.3.11}$$

Proof. First note that both (G) and (G′) are clearly feasible and $\mathrm{val}(G) \leq \mathrm{val}(\mathrm{G}')$. Invoking Lemma 2.2, by weak duality for the pair of primal-dual problems (G′) and (DG′), we thus obtain that $\mathrm{val}(\mathrm{G}') \leq \mathrm{val}(\mathrm{DG}')$ and hence:

$$f(x_N) - f(x_*) \leq \mathrm{val}(G) \leq \mathrm{val}(\mathrm{G}') \leq \mathrm{val}(\mathrm{DG}'). \tag{2.3.12}$$

Now consider the following point (λ, t) for the dual problem (DG′):

$$\lambda_i = \frac{i}{2N+1-i}, \quad i = 1, \ldots, N,$$
$$t = \frac{1}{2Nh+1}.$$

Assuming that this point is (DG′)-feasible, it follows from (2.3.12) that

$$f(x_N) - f(x_*) \leq \mathrm{val}(\mathrm{DG}') \leq \frac{LR^2}{4Nh+2},$$

which proves the desired result. Thus, it remains to show that the above given choice (λ, t) is feasible for (DG′). First, it is easy to see that all the linear constraints of (DG′) on the variables $\lambda_i, i = 1, \ldots, N$ described through the set Λ hold true. Now we prove that the matrix $S \equiv S(\lambda, t)$ is positive semidefinite. From Lemma 2.3, with $h \in [0,1]$, we get that $(1-h)S_0 + hS_1$ is positive definite, as a convex combination of positive definite matrices. Next, we argue that the determinant of S is zero. Indeed, take $u := (1, \ldots, 1, -(2Nh+1))^T$, then from the definition of S and the choice of λ_i and t it follows by elementary algebra that $Su = 0$. To complete the argument, we note that the determinant of S can also be found via the identity (see, e.g., [29, Section A.5.5]):

$$\det(S) = (t - q^T((1-h)S_0 + hS_1)^{-1}q)\det((1-h)S_0 + hS_1).$$

Since we have just shown that $(1-h)S_0 + hS_1 \succ 0$, then $\det((1-h)S_0 + hS_1) > 0$ and we get from the above identity that the value of $t - q^T((1-h)S_0 + hS_1)^{-1}q$, which is the Schur complement of the matrix S, is equal to 0. By a well known lemma on the Schur complement [23, Lemma 4.2.1], we conclude that S is positive semidefinite. □

The next theorem gives a lower bound on the worst-case complexity of Algorithm GM. In particular, it shows that the bound (2.3.11) is tight and that it is attained by a specific convex function in $C_L^{1,1}$.

Theorem 2.5. *Let $L > 0$, $N \in \mathbb{N}$ and $d \in \mathbb{N}$. Then for every $h > 0$ there exists a convex function $\varphi \in C_L^{1,1}(\mathbb{R}^d)$ and a point $x_0 \in \mathbb{R}^d$ such that after N iterations, Algorithm GM reaches an approximate solution x_N with the following absolute inaccuracy*

$$\varphi(x_N) - \varphi^* = \frac{LR^2}{4Nh+2}.$$

Proof. For the sake of simplicity we will assume that $L = 1$ and $R = \|x_* - x_0\| = 1$. Generalizing this proof to general values of L and R can be done by an appropriate scaling.

Consider the function

$$\varphi(x) = \begin{cases} \frac{1}{2Nh+1}\|x\| - \frac{1}{2(2Nh+1)^2}, & \text{if } \|x\| \geq \frac{1}{2Nh+1}, \\ \frac{1}{2}\|x\|^2, & \text{if } \|x\| < \frac{1}{2Nh+1}. \end{cases}$$

Note that this function is nothing else but the Moreau proximal envelope [68] of the function $x \mapsto \|x\|/(2Nh+1)$. It is well known that this function is convex, continuously differentiable with Lipschitz constant $L = 1$, and that its minimal value $\varphi(x_*) = 0$, see e.g., [68, 88]. Applying the gradient method on $\varphi(x)$ with $x_0 = v$ where, as before, v is a unit vector in $\mathbb{R}^d$ (note that only the first part the φ is relevant), we obtain that for $i = 0, \ldots, N$:

$$\begin{aligned} x_i &= \left(1 - \frac{ih}{2Nh+1}\right) v, \\ \varphi'(x_i) &= \frac{1}{2Nh+1} v, \\ \text{and } \varphi(x_i) &= \frac{1}{2Nh+1}\left(1 - \frac{ih}{2Nh+1}\right) - \frac{1}{2(2Nh+1)^2} \\ &= \frac{1}{4Nh+2}\left(\frac{4Nh+1-2ih}{2Nh+1}\right). \end{aligned}$$

Therefore,

$$\varphi(x_N) - \varphi(x_*) = \varphi(x_N) = \frac{1}{4Nh+2}$$

and the desired claim is proven. □

We conclude this section by raising a conjecture on the worst-case performance of the gradient method with a constant step size $0 < h < 2$. Note that when $0 < h \leq 1$ the bound below coincides with (2.3.11).

Conjecture 2.3.1. Suppose the sequence $x_0, \ldots, x_N$ is generated by Algorithm GM with $0 < h < 2$, then

$$f(x_N) - f(x_*) \leq \frac{LR^2}{2} \max\left(\frac{1}{2Nh+1}, (1-h)^{2N}\right).$$

2.4 New Bounds on a Class of First-Order Methods

The framework developed in the previous sections will now serve as a basis to extend the worst-case performance analysis for a broader class of first-order methods for minimizing a smooth convex function over $\mathbb{R}^d$. First, we define a general class of first-order algorithms and we show that it encompasses some interesting first-order methods. Then, following our approach, we define the corresponding PEP associated with this class. Although for this more general case, an analytical solution is not available for determining the bound, we establish that given a fixed number of steps N, a bound on the performance of algorithms in this class can be *estimated numerically*. We then illustrate how this result can be applied for deriving new complexity bounds on two first-order methods.

2.4.1 A General First-Order Algorithm: Definition and Examples

As before, our family $\mathscr{F}$ is the class of convex functions in $C_L^{1,1}(\mathbb{R}^d)$, and $\{d, N, L, R\}$ are fixed. Consider the following class of first-order methods:

Algorithm FO

0. Input: $f \in C_L^{1,1}(\mathbb{R}^d)$, $x_0 \in \mathbb{R}^d$.
1. For $i = 0, \ldots, N-1$, compute $x_{i+1} = x_i - \frac{1}{L}\sum_{k=0}^{i} h_k^{(i+1)} f'(x_k)$.

Here, $h_k^{(i)} \in \mathbb{R}$ play the role of step-sizes, which we assume to be determined by each specific algorithm in this class in a way that is independent of the problem data (i.e., f and x_0).

The interest in the analysis of first-order algorithms of this type is motivated by the fact that it covers some fundamental first-order schemes beyond the gradient method. In particular, to motivate Algorithm FO, let us consider the following two algorithms which are of particular interest, and as we shall see below can be seen as special cases of Algorithm FO.

We start with the so-called Heavy Ball Method (HBM). For earlier work on this method see Polyak [83], and for some interesting modern developments and applications, we refer the reader to Attouch et al. [4, 5] and references therein.

Example 2.4.1. The Heavy Ball Method (HBM)

Algorithm HBM

0. Input: $f \in C_L^{1,1}(\mathbb{R}^d)$, $x_0 \in \mathbb{R}^d$,
1. $x_1 = x_0 - \frac{\alpha}{L} f'(x_0)$
2. For $i = 1, \ldots, N-1$ compute: $x_{i+1} = x_i - \frac{\alpha}{L} f'(x_i) + \beta(x_i - x_{i-1})$

Here the step sizes α and β are chosen such that $0 \leq \beta < 1$ and $0 < \alpha < 2(1+\beta)$, see [83].

By recursively eliminating the term $x_i - x_{i-1}$ in step 2 of Algorithm HBM, we can rewrite this step as

$$x_{i+1} = x_i - \frac{1}{L}\sum_{k=0}^{i} \alpha\beta^{i-k} f'(x_k), \quad i = 1,\ldots,N-1.$$

Therefore, the heavy ball method is a special case of Algorithm FO with the choice

$$h_k^{(i+1)} = \alpha\beta^{i-k}, \quad k = 0,\ldots,i,\ i = 0,\ldots N-1.$$

The next algorithm is Nesterov's celebrated *Fast Gradient Method* [73].

Example 2.4.2. The fast gradient method (FGM)

Algorithm FGM

0. Input: $f \in C_L^{1,1}(\mathbb{R}^d)$, $x_0 \in \mathbb{R}^d$,
1. $y_1 = x_0$, $t_1 = 1$,
2. For $i = 1,\ldots,N$ compute:
   (a) $x_i = y_i - \frac{1}{L} f'(y_i)$,
   (b) $t_{i+1} = \frac{1+\sqrt{1+4t_i^2}}{2}$,
   (c) $y_{i+1} = x_i + \frac{t_i - 1}{t_{i+1}}(x_i - x_{i-1})$.

A major breakthrough was achieved by Nesterov in [73], where he proved that the FGM, which requires almost no increase in computational effort when compared to the basic gradient scheme, achieves the improved rate of convergence $O(1/N^2)$ for function values. More precisely, one has[3]

$$f(x_N) - f(x_*) \leq \frac{2L\|x_0 - x_*\|^2}{(N+1)^2}, \quad \forall x_* \in X_*(f). \tag{2.4.1}$$

The order of complexity of Nesterov's algorithm is also *optimal*, as it is possible to show that there exists a convex function $f \in C_L^{1,1}(\mathbb{R}^d)$ such that when $d \geq 2N+1$, and under some other mild assumptions, *any* first-order algorithm that generates a point x_N by performing N calls to a first-order oracle of f satisfies [74, Theorem 2.1.7]

$$f(x_N) - f(x_*) \geq \frac{3L\|x_0 - x_*\|^2}{32(N+1)^2}, \quad \forall x_* \in X_*(f).$$

This fundamental algorithm discovered about 30 years ago by Nesterov [73] has been recently revived and is currently subject of intensive research activities. For some of its extensions and

[3] See remark following the proof of Theorem 1 in [73].

many applications, see e.g., the recent survey paper Beck-Teboulle [20] and references therein.

At first glance, Algorithm FGM does not seem to fit the class of algorithms defined above (Algorithm FO). Here two sequences are defined: the main sequence $x_0,\ldots,x_N$ and an auxiliary sequence $y_1,\ldots,y_N$. Observing that the gradient of the function is only evaluated on the *auxiliary sequence* of points $\{y_i\}$, we show in the next proposition that Algorithm FGM fits in this class through the following algorithm:

Algorithm FGM′

0. Input: $f \in C_L^{1,1}(\mathbb{R}^d)$, $x_0 \in \mathbb{R}^d$,
1. $y_1 = x_0$, $t_1 = 1$,
2. For $i = 1,\ldots,N-1$ compute:
   (a) $t_{i+1} = \frac{1+\sqrt{1+4t_i^2}}{2}$,
   (b) $y_{i+1} = y_i - \frac{1}{L}\sum_{k=1}^{i} h_k^{(i+1)} f'(y_k)$,
3. $x_N = y_N - \frac{1}{L}f'(y_N)$,

with

$$h_k^{(i+1)} = \begin{cases} \frac{t_i-1}{t_{i+1}} h_k^{(i)}, & \text{if } k+2 \le i, \\ \frac{t_i-1}{t_{i+1}}(h_{i-1}^{(i)} - 1), & \text{if } k = i-1, \\ 1 + \frac{t_i-1}{t_{i+1}}, & \text{if } k = i. \end{cases} \quad (i = 1,\ldots,N-1,\ k = 1,\ldots,i) \tag{2.4.2}$$

Proposition 2.4.1. The points $y_1,\ldots,y_N,x_N$ generated by Algorithm FGM′ are identical to the respective points generated by Algorithm FGM.

Proof. We will show by induction that the sequence y_i generated by Algorithm FGM′ is identical to the sequence y_i generated by Algorithm FGM, and that the value of x_N generated by Algorithm FGM′ is equal to the value of x_N generated by Algorithm FGM.

First note that the sequence t_i is defined by the two algorithms in the same way. Now let $\{x_i,\ y_i\}$ be the sequences generated by Algorithm FGM and denote by $\{y_i'\}$, x_N' the sequence generated by Algorithm FGM′. Obviously, $y_1' = y_1$ and since $t_1 = 1$ we get using the relations 2.4.2:

$$y_2' = y_1' - \frac{1}{L}h_1^{(2)} f'(y_1') = y_1 - \frac{1}{L}\left(1 + \frac{t_1 - 1}{t_2}\right) f'(y_1) = y_1 - \frac{1}{L}f'(y_1) = x_1 = y_2.$$

Assuming $y'_i = y_i$ for $i = 1, \ldots, n$, we then have

$$\begin{aligned}
y'_{n+1} &= y'_n - \frac{1}{L} h_n^{(n+1)} f'(y'_n) - \frac{1}{L} h_{n-1}^{(n+1)} f'(y'_{n-1}) - \frac{1}{L} \sum_{k=1}^{n-2} h_k^{(n+1)} f'(y'_k) \\
&= y_n - \frac{1}{L} \left(1 + \frac{t_n - 1}{t_{n+1}}\right) f'(y_n) - \frac{1}{L} \frac{t_n - 1}{t_{n+1}} \left(h_{n-1}^{(n)} - 1\right) f'(y_{n-1}) - \frac{1}{L} \sum_{k=1}^{n-2} \frac{t_n - 1}{t_{n+1}} h_k^{(n)} f'(y'_k) \\
&= y_n - \frac{1}{L} f'(y_n) + \frac{t_n - 1}{t_{n+1}} \left(-\frac{1}{L} f'(y_n) + \frac{1}{L} f'(y_{n-1}) - \frac{1}{L} \sum_{k=1}^{n-1} h_k^{(n)} f'(y'_k)\right) \\
&= x_n + \frac{t_n - 1}{t_{n+1}} \left(-\frac{1}{L} f'(y_n) + \frac{1}{L} f'(y_{n-1}) + y'_n - y'_{n-1}\right) \\
&= x_n + \frac{t_n - 1}{t_{n+1}} (x_n - x_{n-1}) \\
&= y_{n+1}.
\end{aligned}$$

Finally,

$$x'_N = y'_N - \frac{1}{L} f'(y'_N) = y_N - \frac{1}{L} f'(y_N) = x_N.$$

□

2.4.2 Numerical Estimation of a Bound on Algorithm FO

To build the performance estimation problem for Algorithm FO, from which a complexity bound can be derived, we follow the approach used to derive problem (G) for the gradient method. The only difference being that here, of course, the relation between the variables x_i is derived from the main iteration of Algorithm FO. After some algebra, the resulting PEP reads

$$\begin{aligned}
\max_{G \in \mathbb{R}^{(N+1)\times d}, \delta_i \in \mathbb{R}} \quad & LR^2 \delta_N \\
\text{s.t.} \quad & \operatorname{tr}(G^T \tilde{A}_{i,j} G) \leq \delta_i - \delta_j, \quad i < j = 0, \ldots, N, \\
& \operatorname{tr}(G^T \tilde{B}_{i,j} G) \leq \delta_i - \delta_j, \quad j < i = 0, \ldots, N, \\
& \operatorname{tr}(G^T \tilde{C}_i G) \leq \delta_i, \quad i = 0, \ldots, N, \\
& \operatorname{tr}(G^T \tilde{D}_i G + v u_i^T G) \leq -\delta_i, \quad i = 0, \ldots, N,
\end{aligned} \tag{Q}$$

where $\tilde{A}_{i,j}$, $\tilde{B}_{i,j}$, $\tilde{C}_i$ and $\tilde{D}_i$ are defined, similarly to (2.3.5), by

$$\begin{aligned}
\tilde{A}_{i,j} &= \frac{1}{2}(u_i - u_j)(u_i - u_j)^T + \frac{1}{2} \sum_{t=i+1}^{j} \sum_{k=0}^{t-1} h_k^{(t)} (u_j u_k^T + u_k u_j^T), \\
\tilde{B}_{i,j} &= \frac{1}{2}(u_i - u_j)(u_i - u_j)^T - \frac{1}{2} \sum_{t=j+1}^{i} \sum_{k=0}^{t-1} h_k^{(t)} (u_j u_k^T + u_k u_j^T), \\
\tilde{C}_i &= \frac{1}{2} u_i u_i^T, \\
\tilde{D}_i &= \frac{1}{2} u_i u_i^T + \frac{1}{2} \sum_{t=1}^{i} \sum_{k=0}^{t-1} h_k^{(t)} (u_i u_k^T + u_k u_i^T)
\end{aligned} \tag{2.4.3}$$

and we recall that $v \in \mathbb{R}^d$ is a given unit vector, $u_i = e_{i+1} \in \mathbb{R}^{N+1}$ and the notation $i < j = 0,\ldots,N$ is a shorthand notation for $i = 0,\ldots,N-1, j = i+1,\ldots,N$.

In view of the difficulties in the analysis required to find the solution of (G), an analytical solution to this more general case seems unlikely. However, as we now proceed to show, we can find an upper bound on the optimal solution of this problem by solving a semidefinite program that can be computed numerically using state of the art SDP software.

Following the analysis of the gradient method, (cf. (G′) in §2.3.2) we consider the following simpler relaxed problem:

$$\begin{aligned} \max_{G\in\mathbb{R}^{(N+1)\times d},\delta_i\in\mathbb{R}} \quad & LR^2\delta_N \\ \text{s.t. } & \operatorname{tr}(G^T\tilde{A}_{i-1,i}G) \le \delta_{i-1}-\delta_i, \quad i=1,\ldots,N, \\ & \operatorname{tr}(G^T\tilde{D}_iG + v u_i^T G) \le -\delta_i, \quad i=0,\ldots,N. \end{aligned} \tag{Q′}$$

With the same proof as given in Lemma 2.2 for problem (Q′), we obtain that a dual problem for (Q′) is given by the following convex semidefinite optimization problem (as before, we omit the term LR^2):

$$\begin{aligned} \min_{\lambda,\tau,t} \quad & \frac{1}{2}t \\ \text{s.t. } & \begin{pmatrix} \sum_{i=1}^N \lambda_i\tilde{A}_{i-1,i} + \sum_{i=0}^N \tau_i\tilde{D}_i & \frac{1}{2}\tau \\ \frac{1}{2}\tau^T & \frac{1}{2}t \end{pmatrix} \succeq 0, \\ & (\lambda,\tau)\in\tilde{\Lambda}, \end{aligned} \tag{DQ′}$$

where

$$\tilde{\Lambda} = \{(\lambda,\tau)\in\mathbb{R}^N_+\times\mathbb{R}^{N+1}_+ : \tau_0=\lambda_1,\ \lambda_i-\lambda_{i+1}+\tau_i=0,\ i=1,\ldots,N-1,\ \lambda_N+\tau_N=1\}. \tag{2.4.4}$$

The structure of problem (DQ′) will be very helpful in the analysis of the next section which further addresses the role of the step-sizes. Note that the data matrices of both primal-dual problems (Q′) and (DQ′) depend on the step-sizes $h_k^{(i)}$.

To avoid a trivial bound on problem (Q′), here we need the following assumption on the dual problem (DQ′):

Assumption 1 Problem (DQ′) is solvable, i.e., the minimum is finite and attained for the given step-sizes $h_k^{(i)}$.

Actually, the attainment requirement can be avoided if we can exhibit a feasible point (λ,τ,t) for the problem (DQ′). As noted earlier, given the difficulties already encountered for the simpler gradient method, finding explicitly such a point for the general Algorithm FO is unlikely.

The promised complexity bound on Algorithm FO now easily follows and is determined by the optimal value of the dual problem (DQ'), which can be efficiently computed by any numerical solvers for SDP [23, 51, 99] for small to medium scale problems.

Proposition 2.4.2. Fix any $N,d\in\mathbb{N}$. Let $f\in C_L^{1,1}(\mathbb{R}^d)$ be convex and suppose that $x_0,\ldots,x_N\in\mathbb{R}^d$ are generated by Algorithm FO, and that Assumption 1 holds. Then,

$$f(x_N)-f(x_*) \le LR^2\,\mathrm{val}\,(\mathrm{DQ}').$$

Proof. Follows from weak duality for the pair of primal-dual problems (Q′)-(DQ′) □

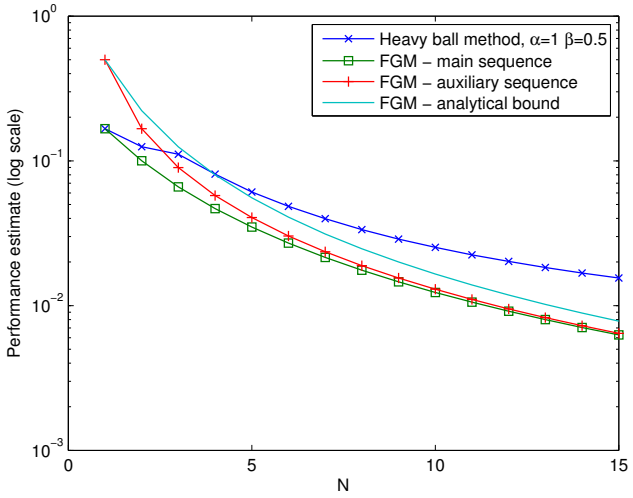

Figure 2.1: The computed worst-case bounds on the HBM and FGM versus the classical analytical bound on the FGM (2.4.1).

2.4.3 Numerical Illustrations

We apply Proposition 2.4.2 to find bounds on the complexity of the heavy ball method (HBM) with[4] $\alpha = 1$ and $\beta = \frac{1}{2}$ and on the fast gradient method (FGM) with $h_k^{(i)}$ as given in (2.4.2), which as shown earlier, can both be viewed as particular realizations of Algorithm FO.

The resulting SDP programs were solved for different values of N using `CVX` [48, 49]. These results, together with the classical bound on the convergence rate of the main sequence of the fast gradient method (2.4.1), are summarized in Figure 2.1 and Table 2.1.

Note that as far as the authors are aware, there is no known convergence rate result for the HBM on the class of convex functions in $\mathcal{C}_L^{1,1}$. As can be seen from the above results, the numerically estimated bound for the HBM behaves slightly better than the gradient method (compare with the explicit bound given in Theorem 2.4), but remains much slower than the fast gradient scheme (FGM).

Considering the results on the FGM, note that the numerically estimated bounds for the main sequence of point x_i and the corresponding values at the auxiliary sequence y_i of the fast gradient method are very similar and perform slightly better than predicted by the classical bound (2.4.1). To the best of our knowledge, the complexity of the auxiliary sequence is yet unknown, thus these results encourage us to raise the following conjecture.

Conjecture 2.4.1. Let $x_0, x_1, \ldots$ and $y_1, y_2, \ldots$ be the main and auxiliary sequences defined by FGM (respectively), then $\{f(x_i)\}$ and $\{f(y_i)\}$ converge to the optimal value of the problem with the same rate of convergence.

[4] According to our simulations, this choice for the values of α, β produces results that are typical of the behavior of the algorithm.

N	Heavy Ball	FGM, main	FGM, auxiliary	FGM, analytical bound
1	$LR^2/6.00$	$LR^2/6.00$	$LR^2/2.00$	$LR^2/2.0=2LR^2/(1+1)^2$
2	$LR^2/7.99$	$LR^2/10.00$	$LR^2/6.00$	$LR^2/4.5=2LR^2/(2+1)^2$
3	$LR^2/9.00$	$LR^2/15.13$	$LR^2/11.13$	$LR^2/8.0=2LR^2/(3+1)^2$
4	$LR^2/12.35$	$LR^2/21.35$	$LR^2/17.35$	$LR^2/12.5=2LR^2/(4+1)^2$
5	$LR^2/16.41$	$LR^2/28.66$	$LR^2/24.66$	$LR^2/18.0=2LR^2/(5+1)^2$
10	$LR^2/39.63$	$LR^2/81.07$	$LR^2/77.07$	$LR^2/60.5=2LR^2/(10+1)^2$
20	$LR^2/89.45$	$LR^2/263.65$	$LR^2/259.65$	$LR^2/220.5=2LR^2/(20+1)^2$
40	$LR^2/188.99$	$LR^2/934.89$	$LR^2/930.89$	$LR^2/840.5=2LR^2/(40+1)^2$
80	$LR^2/387.91$	$LR^2/3490.22$	$LR^2/3486.22$	$LR^2/3280.5=2LR^2/(80+1)^2$
160	$LR^2/785.68$	$LR^2/13427.43$	$LR^2/13423.43$	$LR^2/12960.5=2LR^2/(160+1)^2$
500	$LR^2/2476.11$	$LR^2/127224.44$	$LR^2/127220.32$	$LR^2/125500.5=2LR^2/(500+1)^2$
1000	$LR^2/4962.01$	$LR^2/504796.99$	$LR^2/504798.28$	$LR^2/501000.5=2LR^2/(1000+1)^2$

Table 2.1: The computed worst-case bounds on the HBM and FGM versus the classical analytical bound on the FGM (2.4.1).

2.5 A Best Performing Algorithm: Optimal Step Sizes for Algorithm FO

We now consider the problem of finding the "best" performing algorithm of the form FO with respect to the new bounds. Namely, we consider the problem of minimizing $\mathrm{val}\,(\mathrm{Q}')$, the optimal value of (Q'), with respect to the step sizes $h := (h_k^{(i)})_{0\le k<i\le N}$ defining the Algorithm FO, and which are now considered as unknown variables in FO.

We denote by $\tilde{A}_{i,j}(h)$ and $\tilde{D}_i(h)$, the matrices given in (2.4.3), which are functions of the algorithm step sizes h. The resulting bound derived in Proposition 2.4.2 is thus a function of h, and the problem of minimizing $\mathrm{val}\,(\mathrm{DQ}')$ with respect to the step sizes h thus consists of solving the following bilinear problem:

$$\min_{h,\lambda,\tau,t}\left\{\frac{1}{2}t : \begin{pmatrix}\sum_{i=1}^N \lambda_i\tilde{A}_{i-1,i}(h)+\sum_{i=0}^N \tau_i\tilde{D}_i(h) & \frac{1}{2}\tau\\ \frac{1}{2}\tau^T & \frac{1}{2}t\end{pmatrix}\succeq 0, (\lambda,\tau)\in\tilde{\Lambda}\right\}, \tag{BIL}$$

with $\tilde{\Lambda}$ defined as in (2.4.4).

Note that the feasibility of (BIL) follows from the proof of Theorem 2.5, where an explicit feasible point is given to (DG'), which is a special instance of (BIL) when the steps $(h_k^{(i)})$ are chosen as in the gradient method.

From the definition of the matrices $\tilde{A}_{i,j}(h)$ and $\tilde{D}_i(h)$, we get

$$\begin{aligned}\sum_{i=1}^N \lambda_i\tilde{A}_{i-1,i}(h)+\sum_{i=0}^N \tau_i\tilde{D}_i(h) &= \frac{1}{2}\sum_{i=1}^N \lambda_i(u_{i-1}-u_i)(u_{i-1}-u_i)^T+\frac{1}{2}\sum_{i=0}^N \tau_i u_i u_i^T\\ &\quad+\frac{1}{2}\sum_{i=1}^N\sum_{k=0}^{i-1}\left(\lambda_i h_k^{(i)}+\tau_i\sum_{t=k+1}^{i} h_k^{(t)}\right)(u_iu_k^T+u_ku_i^T).\end{aligned}$$

Introducing the new variables:

$$r_{i,k} = \lambda_i h_k^{(i)} + \tau_i \sum_{t=k+1}^{i} h_k^{(t)}, \quad i = 1,\dots,N,\ k = 0,\dots,i-1 \tag{2.5.1}$$

and denoting $r = (r_{i,k})_{0\le k<i\le N}$, we obtain the following *linear* SDP relaxation of (BIL):

$$\min_{r,\lambda,\tau,t} \left\{ \frac{1}{2}t : \begin{pmatrix} S(r,\lambda,\tau) & \frac{1}{2}\tau \\ \frac{1}{2}\tau^T & \frac{1}{2}t \end{pmatrix} \succeq 0,\ (\lambda,\tau) \in \tilde{\Lambda} \right\}, \tag{LIN}$$

where

$$S(r,\lambda,\tau) = \frac{1}{2}\sum_{i=1}^{N} \lambda_i (u_{i-1}-u_i)(u_{i-1}-u_i)^T + \frac{1}{2}\sum_{i=0}^{N} \tau_i u_i u_i^T + \frac{1}{2}\sum_{i=1}^{N}\sum_{k=0}^{i-1} r_{i,k}(u_i u_k^T + u_k u_i^T).$$

This convex SDP can now be efficiently solved by numerical methods. As the following theorem shows, the optimal solution of (LIN) can then be used to construct an optimal solution for (BIL) and hence recover optimal values for the step sizes h.

Theorem 2.6. *Suppose $(r^*,\lambda^*,\tau^*,t^*)$ is an optimal solution for (LIN), then (h,λ^*,τ^*,t^*) is an optimal solution for* (BIL)*, where $h = (h_k^{(i)})_{0\le k<i\le N}$ is defined by the following recursive rule*[5]

$$h_k^{(i)} = \begin{cases} \frac{r_{i,k}^* - \tau_i^* \sum_{t=k+1}^{i-1} h_k^{(t)}}{\lambda_i^* + \tau_i^*} & \text{if } \lambda_i^* + \tau_i^* \neq 0, \\ 0 & \text{otherwise,} \end{cases} \quad i = 1,\dots,N,\ k = 0,\dots,i-1. \tag{2.5.2}$$

Proof. As (LIN) is a relaxation of (BIL), it is enough to show that (BIL) can achieve the same objective value. Let $(r^*,\lambda^*,\tau^*,t^*)$ be an optimal solution for (LIN). If $\lambda_i^* \neq 0$ for all $1 \le i \le N$, then (2.5.2) satisfies all the equations in (2.5.1) and therefore (h,λ^*,τ^*,t^*) is feasible for (BIL).

Suppose $\lambda_m^* = 0$ for some $1 \le m \le N$ and that m is the maximal index with this property. Then by the equality and non-negativity constraints in (LIN), we get that $\lambda_1^* = \lambda_2^* = \cdots = \lambda_m^* = 0$ and $\tau_0^* = \tau_1^* = \cdots = \tau_{m-1}^* = 0$. Let $S := S(r,\lambda^*,\tau^*)$, then by the positive semidefinite constraint in (LIN), we have $S \succeq 0$. From the linear equalities connecting λ and τ it follows that

$$S_{i,i} = \begin{cases} \frac{1}{2}(\lambda_1^* + \tau_0^*) = \lambda_1^*, & \text{if } i = 1, \\ \frac{1}{2}(\lambda_i^* + \lambda_{i-1}^* + \tau_{i-1}^*) = \lambda_i^*, & \text{if } i = 2,\dots,N, \end{cases}$$

and we get that $S_{1,1} = \cdots = S_{m,m} = 0$. By the properties of positive semidefinite matrices we now get that $r_{i,k}^* = 0$ for $i = 1,\dots,m$ and $k = 0,\dots,i-1$, hence the set of equations (2.5.1) with the chosen values of $h_k^{(i)}$ is consistent. □

The optimal value of LR^2 val(LIN) for various values of N is summarized in Table 2.2. As can be seen from these results (compare with Table 2.1), the worst-case performance of the new algorithm is almost exactly two times better than the worst-case performance of the fast

[5] We thank Donghwan Kim and Jeffrey A. Fessler for spotting a typo in the journal version of this work [40].

N	LR^2val(LIN)	N	LR^2val(LIN)
1	$LR^2/8.00$	20	$LR^2/525.09$
2	$LR^2/16.16$	40	$LR^2/1869.22$
3	$LR^2/26.53$	80	$LR^2/6983.13$
4	$LR^2/39.09$	160	$LR^2/26864.04$
5	$LR^2/53.80$	500	$LR^2/254482.61$
10	$LR^2/159.07$	1000	$LR^2/1009628.17$

Table 2.2: An approximate solution of LR^2 val(LIN) for various values of N.

gradient method. Note that the bounds given here are worst-case bounds: the performance of the considered methods on a specific application can be very different.

The resulting first-order algorithm with the computed optimal step sizes $h_k^{(i)}$ for $N = 5$ is illustrated in the example below.

Example 2.5.1. Consider the following first-order method, which was constructed by solving (LIN) for $N = 5$.

0. Input: $f \in C_L^{1,1}(\mathbb{R}^d)$, $x_0 \in \mathbb{R}^d$,
1. $x_1 = x_0 - \frac{1.6180}{L} f'(x_0)$,
2. $x_2 = x_1 - \frac{0.1741}{L} f'(x_0) - \frac{2.0194}{L} f'(x_1)$,
3. $x_3 = x_2 - \frac{0.0756}{L} f'(x_0) - \frac{0.4425}{L} f'(x_1) - \frac{2.2317}{L} f'(x_2)$,
4. $x_4 = x_3 - \frac{0.0401}{L} f'(x_0) - \frac{0.2350}{L} f'(x_1) - \frac{0.6541}{L} f'(x_2) - \frac{2.3656}{L} f'(x_3)$,
5. $x_5 = x_4 - \frac{0.0178}{L} f'(x_0) - \frac{0.1040}{L} f'(x_1) - \frac{0.2894}{L} f'(x_2) - \frac{0.6043}{L} f'(x_3) - \frac{2.0778}{L} f'(x_4)$.

A bound on the worst-case performance the algorithm in this example is given by the following inequality (see $N = 5$ in Table 2.2):

$$f(x_5) - f(x_*) \leq \frac{L\|x_0 - x_*\|^2}{53.80}, \quad \forall x_* \in X_*(f).$$

2.6 Conclusions

We introduced a novel approach for estimating the worst-case complexity of first-order methods for convex optimization via the PEP problem, its relaxations, and exact or approximate solution using duality. Using this approach we derived a tight bound on the worst-case performance of the fixed-size gradient method and established new bounds that can be numerically estimated for

a general class of first-order algorithms, which includes the Heavy Ball method and Nesterov's fast gradient method. We then showed how to construct optimal stepsizes for this first-order class.

While the proposed approach and the PEP problem offer a novel way to measure the complexity of any algorithm, it should be stressed that this approach is of course not without limitations. Indeed, as shown in the chapter, finding a bound on the PEP problem is challenging. In the case of the gradient method with a fixed step size, the derivation of simple closed form expression for the bound required a dedicated analysis. Furthermore, for more general first-order algorithms, we are left with the problem of approximating the solution of the problem using SDP solvers which are often efficient only for small to medium scale problems. Nevertheless, the novelty of the proposed approach offers possible directions for extensions that could be considered in future research by formulating and analyzing the corresponding PEP problem for other first-order algorithms. This includes, for example, the analysis of gradient methods with different variable step-size strategies, which is briefly discussed in Appendix II, and the analysis of algorithms for different classes $\mathscr{F}$ of input functions, such as the class of strongly convex functions, which is briefly discussed in Appendix III.

Finally, we would like to mention the very recent work [54] by Donghwan Kim and Jeffrey A. Fessler who further analyzed the results presented in this chapter. Among other results, they confirmed Conjecture 2.4.1 and derived two efficient implementations to the method presented in Section 2.5.

2.7 Appendix I: Proof of Lemma 2.3

We now establish the positive definiteness of the matrices S_0 and S_1 given in (2.3.8) and (2.3.9), respectively.

Part I: $S_0 \succ 0$

We begin by showing that S_0 is positive definite. Recall that

$$S_0 = \begin{pmatrix} 2\lambda_1 & -\lambda_1 & & & & \\ -\lambda_1 & 2\lambda_2 & -\lambda_2 & & & \\ & -\lambda_2 & 2\lambda_3 & -\lambda_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -\lambda_{N-1} & 2\lambda_N & -\lambda_N \\ & & & & -\lambda_N & 1 \end{pmatrix}$$

for

$$\lambda_i = \frac{i}{2N+1-i}, \qquad i = 1,\ldots,N.$$

Let us look at $\xi^T S_0 \xi$ for any $\xi = (\xi_0, \ldots, \xi_N)^T$:

$$\begin{aligned}\xi^T S_0 \xi &= \sum_{i=0}^{N-1} 2\lambda_{i+1}\xi_i^2 - 2\sum_{i=0}^{N-1} \lambda_{i+1}\xi_i\xi_{i+1} + \xi_N^2 \\ &= \sum_{i=0}^{N-1} \lambda_{i+1}(\xi_{i+1} - \xi_i)^2 + \lambda_1\xi_0^2 + \sum_{i=1}^{N-1}(\lambda_{i+1} - \lambda_i)\xi_i^2 + (1-\lambda_N)\xi_N^2\end{aligned}$$

which is always positive for $\xi \neq 0$. We conclude that S_0 is positive definite.

Part II: $S_1 \succ 0$

We will show that S_1 is positive definite using Sylvester's criterion[6].

Recall that

$$S_1 = \begin{pmatrix} 2\lambda_1 & \lambda_2 - \lambda_1 & \ldots & \lambda_N - \lambda_{N-1} & 1 - \lambda_N \\ \lambda_2 - \lambda_1 & 2\lambda_2 & & \lambda_N - \lambda_{N-1} & 1 - \lambda_N \\ \vdots & & \ddots & & \vdots \\ \lambda_N - \lambda_{N-1} & \lambda_N - \lambda_{N-1} & & 2\lambda_N & 1 - \lambda_N \\ 1 - \lambda_N & 1 - \lambda_N & \ldots & 1 - \lambda_N & 1 \end{pmatrix}$$

for

$$\lambda_i = \frac{i}{2N + 1 - i}, \qquad i = 1, \ldots, N.$$

A recursive expression for the determinants We begin by deriving a recursion rule for the determinant of matrices of the following form:

$$M_k = \begin{pmatrix} d_0 & a_1 & a_2 & \ldots & a_{k-1} & a_k \\ a_1 & d_1 & a_2 & & a_{k-1} & a_k \\ a_2 & a_2 & d_2 & & a_{k-1} & a_k \\ \vdots & & & \ddots & & \vdots \\ a_{k-1} & a_{k-1} & a_{k-1} & & d_{k-1} & a_k \\ a_k & a_k & a_k & \ldots & a_k & d_k \end{pmatrix}.$$

To find the determinant of M_k, subtract the one before last row multiplied by $\frac{a_k}{a_{k-1}}$ from the last row: the last row becomes

$$(0, \ldots, 0, a_k - \frac{a_k}{a_{k-1}} d_{k-1}, d_k - \frac{a_k}{a_{k-1}} a_k).$$

Expanding the determinant along the last row we get

$$\det M_k = (d_k - \frac{a_k}{a_{k-1}} a_k) \det M_{k-1} - (a_k - \frac{a_k}{a_{k-1}} d_{k-1}) \det (M_k)_{k,k-1}$$

[6]Despite the interesting structure of the matrix S_1, this proof is quite involved. A simpler proof would be most welcome!

where $(M_k)_{k,k-1}$ denotes the $k,k-1$ minor:

$$(M_k)_{k,k-1} = \begin{pmatrix} d_0 & a_1 & a_2 & \dots & a_{k-2} & a_k \\ a_1 & d_1 & a_2 & & a_{k-2} & a_k \\ a_2 & a_2 & d_2 & & a_{k-2} & a_k \\ \vdots & & & \ddots & & \\ a_{k-2} & a_{k-2} & a_{k-2} & & d_{k-2} & a_k \\ a_{k-1} & a_{k-1} & a_{k-1} & & a_{k-1} & a_k \end{pmatrix}.$$

If we multiply the last column of $(M_k)_{k,k-1}$ by $\frac{a_{k-1}}{a_k}$ we get a matrix that is different from M_{k-1} by only the corner element. Thus by basic determinant properties we get that

$$\frac{a_{k-1}}{a_k}\det(M_k)_{k,k-1} = \det M_{k-1} + (a_{k-1} - d_{k-1})\det M_{k-2}.$$

Combining these two results, we have found the following recursion rule for $\det M_k$, $k \geq 2$:

$$\begin{aligned} \det M_k &= (d_k - \frac{a_k}{a_{k-1}}a_k)\det M_{k-1} \\ &\quad - (a_k - \frac{a_k}{a_{k-1}}d_{k-1})\left(\frac{a_k}{a_{k-1}}\det M_{k-1} + (a_k - \frac{a_k}{a_{k-1}}d_{k-1})\det M_{k-2}\right) \\ &= \left((d_k - \frac{a_k}{a_{k-1}}a_k) - (a_k - \frac{a_k}{a_{k-1}}d_{k-1})\frac{a_k}{a_{k-1}}\right)\det M_{k-1} - \left(a_k - \frac{a_k}{a_{k-1}}d_{k-1}\right)^2 \det M_{k-2} \end{aligned}$$

or

$$\det M_k = \left(d_k - \frac{2a_k^2}{a_{k-1}} + \frac{a_k^2 d_{k-1}}{a_{k-1}^2}\right)\det M_{k-1} - a_k^2\left(1 - \frac{d_{k-1}}{a_{k-1}}\right)^2 \det M_{k-2}. \tag{2.7.1}$$

Obviously, the recursion base cases are given by

$$\begin{aligned} \det M_0 &= d_0, \\ \det M_1 &= d_0 d_1 - a_1^2. \end{aligned}$$

Closed form expressions for the determinants Going back to our matrix, S_1, by choosing

$$\begin{aligned} d_i &= 2\frac{i+1}{2N-i}, \quad i = 0,\dots,N-1 \\ d_N &= 1 \\ a_i &= \frac{i+1}{2N-i} - \frac{i}{2N+1-i}, \quad i = 1,\dots,N-1 \\ a_N &= 1 - \frac{N}{N+1} = \frac{1}{N+1}, \end{aligned}$$

we get that M_k is the $k+1$'th leading principal minor of the matrix S_1. The recursion rule (2.7.1) can now be solved for this choice of a_i and d_i. The solution is given by:

$$\det M_k = \frac{(2N+1)^2}{(2N-k)^2}\left(1 + \sum_{i=0}^{k}\frac{2N-2k-1}{2N+4Ni-2i^2+1}\right)\prod_{i=0}^{k}\frac{2N+4Ni-2i^2+1}{(2N+1-i)^2}, \tag{2.7.2}$$

for $k = 0, \ldots, N-1$, and

$$\det M_N = \det L_1 = \frac{(2N+1)^2}{(N+1)^2} \prod_{i=0}^{N-1} \frac{2N+4Ni-2i^2+1}{(2N+1-i)^2}. \tag{2.7.3}$$

Verification We now proceed to verify the expressions (2.7.2) and (2.7.3) given above. We will show that these expressions satisfy the recursion rule (2.7.1) and the base cases of the problem. We begin by verifying the base cases:

$$\det M_0 = \frac{(2N+1)^2}{(2N)^2}\left(1+\frac{2N-1}{2N+1}\right)\frac{1}{2N+1} = \frac{1}{N} = d_0,$$

$$\begin{aligned}\det M_1 &= \frac{(2N+1)^2}{(2N-1)^2}\left(1+\frac{2N-3}{2N+1}+\frac{2N-3}{6N-1}\right)\frac{1}{2N+1}\frac{6N-1}{(2N)^2} \\ &= \frac{28N^2-20N-1}{4N^2(2N-1)^2} = \frac{4}{N(2N-1)} - \left(\frac{2}{2N-1}-\frac{1}{2N}\right)^2 = d_0 d_1 - a_1^2.\end{aligned}$$

Now suppose $2 \le k \le N$. Denote

$$\alpha_k = d_k - \frac{2a_k^2}{a_{k-1}} + \frac{a_k^2 d_{k-1}}{a_{k-1}^2} = \begin{cases} 4\frac{(2N+1)k-k^2-1}{(2N-k)^2}, & \text{if } k < N, \\ 3\frac{2N^2+2N-1}{(2N+1)^2}, & \text{if } k = N, \end{cases}$$

$$\beta_k = a_k^2\left(1-\frac{d_{k-1}}{a_{k-1}}\right)^2 = \begin{cases} \frac{(4kN-2N-2k^2+4k-1)^2}{(2N-k)^2(2N-k+1)^2}, & \text{if } k < N, \\ \frac{(2N^2+2N-1)^2}{(N+1)^2(2N+1)^2}, & \text{if } k = N, \end{cases}$$

then the recursion rule (2.7.1) can be written as

$$\det M_k = \alpha_k \det M_{k-1} - \beta_k \det M_{k-2}.$$

Further denote

$$\begin{aligned} r_i &= \frac{1}{2N+4Ni-2i^2+1}, \quad i = 0, \ldots, N-1, \\ s_i &= \frac{(2N+1)^2}{(2N-i)^2}, \quad i = 0, \ldots, N-1, \\ p_i &= 2N-2i-1, \quad i = 0, \ldots, N-1, \\ q_i &= \frac{2N+4Ni-2i^2+1}{(2N+1-i)^2}, \quad i = 0, \ldots, N-1, \end{aligned}$$

then the solution (2.7.2) becomes

$$\det M_k = s_k\left(1+p_k\sum_{i=0}^{k} r_i\right)\prod_{i=0}^{k} q_i,$$

and (2.7.3) becomes

$$\det M_N = \frac{(2N+1)^2}{(N+1)^2} \prod_{i=0}^{N-1} q_i.$$

Substituting (2.7.2) in the RHS of (2.7.1) we get that for $k = 2, \ldots, N$

$$\begin{aligned}
&\alpha_k \det M_{k-1} - \beta_k \det M_{k-2} \\
&= \alpha_k s_{k-1} \left(1 + p_{k-1} \sum_{i=0}^{k-1} r_i\right) \prod_{i=0}^{k-1} q_i - \beta_k s_{k-2} \left(1 + p_{k-2} \sum_{i=0}^{k-2} r_i\right) \prod_{i=0}^{k-2} q_i \\
&= \left(\alpha_k s_{k-1} \left(1 + p_{k-1} r_{k-1} + p_{k-1} \sum_{i=0}^{k-2} r_i\right) - \frac{\beta_k}{q_{k-1}} s_{k-2} - \frac{\beta_k}{q_{k-1}} s_{k-2} p_{k-2} \sum_{i=0}^{k-2} r_i\right) \prod_{i=0}^{k-1} q_i \\
&= \left(\alpha_k s_{k-1}(1 + p_{k-1} r_{k-1}) - \frac{\beta_k}{q_{k-1}} s_{k-2} + \left(\alpha_k s_{k-1} p_{k-1} - \frac{\beta_k}{q_{k-1}} s_{k-2} p_{k-2}\right) \sum_{i=0}^{k-2} r_i\right) \prod_{i=0}^{k-1} q_i.
\end{aligned}$$

It is straightforward (although somewhat involved) to verify that for $k < N$

$$\alpha_k s_{k-1}(1 + p_{k-1} r_{k-1}) - \frac{\beta_k}{q_{k-1}} s_{k-2} = s_k q_k (1 + p_k r_{k-1} + p_k r_k),$$

and

$$\alpha_k s_{k-1} p_{k-1} - \frac{\beta_k}{q_{k-1}} s_{k-2} p_{k-2} = s_k p_k q_k.$$

We therefore get

$$\begin{aligned}
&\alpha_k \det M_{k-1} - \beta_k \det M_{k-2} \\
&= \left(s_k q_k (1 + p_k r_{k-1} + p_k r_k) + s_k p_k q_k \sum_{i=0}^{k-2} r_i\right) \prod_{i=0}^{k-1} q_i \\
&= s_k \left(1 + p_k \sum_{i=0}^{k} r_i\right) \prod_{i=0}^{k} q_i \\
&= \det M_k,
\end{aligned}$$

and thus (2.7.2) satisfies (2.7.1). It is also possible to show that

$$\begin{aligned}
\alpha_N s_{N-1}(1 + p_{N-1} r_{N-1}) - \frac{\beta_N}{q_{N-1}} s_{N-2} &= \frac{(2N+1)^2}{(N+1)^2}, \\
\alpha_N s_{N-1} p_{N-1} - \frac{\beta_N}{q_{N-1}} s_{N-2} p_{N-2} &= 0,
\end{aligned}$$

thus, for $k = N$

$$\begin{aligned}
&\alpha_N \det M_{N-1} - \beta_N \det M_{N-2} \\
&= \frac{(2N+1)^2}{(N+1)^2} \prod_{i=0}^{N-1} q_i \\
&= \det M_N,
\end{aligned}$$

and the expression (2.7.3) is also verified.

To complete the proof, note that the closed form expressions for $\det M_k$ consist of sums and products of positive values, hence $\det M_k$ is positive, and thus by Sylvester's criterion S_1 is positive definite.

2.8 Appendix II: An Analytical Bound for the Projected Gradient Method

Let $C \subseteq \mathbb{R}^d$ be a convex set and suppose f is a convex differentiable function with Lipschitz continuous gradient with the constant L. In this section, we consider the projected gradient method:

Algorithm PG

1. Choose $x_0 \in C$
2. For $i = 1, \ldots, N$
   (a) Set $y_i \leftarrow x_{i-1} - \frac{h_{i-1}}{L} f'(x_{i-1})$
   (b) Set $x_i \leftarrow P_C(y_i)$.

We show that under some conditions on the step sizes, h_i, the efficiency estimate of the method is given by

$$f(x_N) - f(x_*) \leq \frac{L\|x_0 - x^*\|^2}{4\sum_{k=0}^{N-1} h_k}.$$

We start the derivation of the bound by writing the corresponding PEP:

$$\begin{aligned} \max_{\substack{\varphi \in C_L^{1,1} \text{ convex}, \\ x_0 \in C}} \quad & \varphi(x_N) - \varphi^* \\ \text{s.t. } & y_i = x_{i-1} - \frac{h_{i-1}}{L}\varphi'(x_{i-1}), \quad i = 1, \ldots, N, \\ & x_i = P_C(y_i), \quad i = 1, \ldots, N, \\ & \|x_0 - x_*\| \leq R. \end{aligned}$$

By the properties of convex sets and convex functions in $C_L^{1,1}$, the following inequalities hold:

$$\langle x - P_C(x), y - P_C(x) \rangle \leq 0, \quad \forall x \in \mathbb{R}^d, \ \forall y \in C, \tag{2.8.1}$$

$$\frac{1}{2L}\|\varphi'(x) - \varphi'(y)\|^2 \leq \varphi(x) - \varphi(y) - \langle \varphi'(y), x - y \rangle, \quad \forall \varphi \in F_L, \ \forall x, y. \tag{2.8.2}$$

Let

$$\begin{aligned}
&x_* \in \arg\min_{x\in C} \varphi(x),\\
&g_* = \frac{1}{L}\varphi'(x_*),\\
&\delta_i = \frac{1}{L}(\varphi(x_i)-\varphi(x_*)), \quad (i=0,\dots,N,*),\\
&g_i = \frac{1}{L}\varphi'(x_i), \quad (i=0,\dots,N,*),
\end{aligned}$$

then by applying (2.8.1) and (2.8.2) on $x_0,\dots,x_N,x_*$, $y_0,\dots,y_N$ and treating these variables as the new optimization variables instead of φ, we arrive to the following relaxation:

$$\begin{aligned}
\max_{x_i,y_i,g_i\in\mathbb{R}^d,\delta_i\in\mathbb{R}} \quad & L\delta_N\\
\text{s.t.} \quad & y_i = x_{i-1} - h_{i-1}g_{i-1}, \quad i=1,\dots,N,\\
& \langle y_i - x_i, x_j - x_i\rangle \le 0, \quad j=0,\dots,N,*,\ i=1,\dots,N,\\
& \frac{1}{2}\|g_i-g_j\|^2 \le \delta_j - \delta_i - \langle g_i, x_j - x_i\rangle, \quad i,j=0,\dots,N,*,\\
& \|x_0 - x_*\| \le R.
\end{aligned}$$

Eliminating y_i and removing some constraints (which were numerically found to be inactive), we reach the following relaxed problem:

$$\begin{aligned}
\max_{x_i,g_i\in\mathbb{R}^d,\delta_i\in\mathbb{R}} \quad & L\delta_N\\
\text{s.t.} \quad & \|x_i - x_{i+1}\|^2 - h_i\langle g_i, x_i - x_{i+1}\rangle \le 0, \quad i=1,\dots,N-1,\\
& \langle x_i - x_{i+1}, x_* - x_{i+1}\rangle - h_i\langle g_i, x_* - x_{i+1}\rangle \le 0, \quad i=0,\dots,N-1,\\
& \frac{1}{2}\|g_{i+1}-g_i\|^2 + \langle g_{i+1}, x_i - x_{i+1}\rangle \le \delta_i - \delta_{i+1}, \quad i=0,\dots,N-1,\\
& \frac{1}{2}\|g_i - g_*\|^2 + \langle g_i, x_* - x_i\rangle \le -\delta_i, \quad i=0,\dots,N,\\
& \|x_0 - x_*\| \le R.
\end{aligned}$$

The SDP relaxation is performed by defining the variable:

$$Z = \left(\begin{array}{ccc|ccc|cc}
\langle x_0,x_0\rangle & \dots & \langle x_0,x_N\rangle & \langle x_0,g_0\rangle & \dots & \langle x_0,g_N\rangle & \langle x_0,x_*\rangle & \langle x_0,g_*\rangle\\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots\\
\langle x_N,x_0\rangle & \dots & \langle x_N,x_N\rangle & \langle x_N,g_0\rangle & \dots & \langle x_N,g_N\rangle & \langle x_N,x_*\rangle & \langle x_N,g_*\rangle\\
\hline
\langle g_0,x_0\rangle & \dots & \langle g_0,x_N\rangle & \langle g_0,g_0\rangle & \dots & \langle g_0,g_N\rangle & \langle g_0,x_*\rangle & \langle g_0,g_*\rangle\\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots\\
\langle g_N,x_0\rangle & \dots & \langle g_N,x_N\rangle & \langle g_N,g_0\rangle & \dots & \langle g_N,g_N\rangle & \langle g_N,x_*\rangle & \langle g_N,g_*\rangle\\
\hline
\langle x_*,x_0\rangle & \dots & \langle x_*,x_N\rangle & \langle x_*,g_0\rangle & \dots & \langle x_*,g_N\rangle & \langle x_*,x_*\rangle & \langle x_*,g_*\rangle\\
\langle g_*,x_0\rangle & \dots & \langle g_*,x_N\rangle & \langle g_*,g_0\rangle & \dots & \langle g_*,g_N\rangle & \langle g_*,x_*\rangle & \langle g_*,g_*\rangle
\end{array}\right)$$

thereby reaching the problem

$$\begin{aligned}
\text{(PG-R)} \quad \max_{Z\in\mathbb{S}^{2N+2},\delta_i\in\mathbb{R}} \quad & L\delta_N \\
\text{s.t.} \quad & \operatorname{tr}(A_iZ) \le 0, \quad i=1,\dots,N-1, \\
& \operatorname{tr}(B_iZ) \le 0, \quad i=0,\dots,N-1, \\
& \operatorname{tr}(C_iZ) \le \delta_i-\delta_{i+1}, \quad i=0,\dots,N-1, \\
& \operatorname{tr}(D_iZ) \le -\delta_i, \quad i=0,\dots,N, \\
& \operatorname{tr}(EZ) \le R^2, \\
& Z \succeq 0,
\end{aligned}$$

where A_i, B_i, C_i, D_i and E are $(2N+2)\times(2N+2)$ symmetrical matrices appropriately defined according to the optimization problem above. I.e., if we denote

$$\begin{aligned}
u_i &= e_{i+1} \quad (i=0,\dots,N) \\
v_i &= e_{i+N+2} \quad (i=0,\dots,N) \\
u_* &= e_{2N+1} \\
v_* &= e_{2N+2}
\end{aligned}$$

($e_i\in\mathbb{R}^{2N+2}$ is the i'th canonical unit vector) and define

$$\begin{aligned}
A_i' &= (u_i-u_{i+1})(u_i-u_{i+1})^T - h_iv_i(u_i-u_{i+1})^T, \quad i=1,\dots,N-1 \\
B_i' &= (u_i-u_{i+1})(u_*-u_{i+1})^T - h_iv_i(u_*-u_{i+1})^T, \quad i=0,\dots,N-1 \\
C_i' &= \frac{1}{2}(v_{i+1}-v_i)(v_{i+1}-v_i)^T + v_{i+1}(u_i-u_{i+1})^T, \quad i=0,\dots,N-1 \\
D_i' &= \frac{1}{2}(v_i-v_*)(v_i-v_*)^T + v_i(u_*-u_i)^T, \quad i=0,\dots,N \\
E' &= (u_0-u_*)(u_0-u_*)^T
\end{aligned}$$

we get

$$\begin{aligned}
A_i &= \frac{1}{2}(A_i'+A_i'^T), \quad i=1,\dots,N-1, \\
B_i &= \frac{1}{2}(B_i'+B_i'^T), \quad i=0,\dots,N-1, \\
C_i &= \frac{1}{2}(C_i'+C_i'^T), \quad i=0,\dots,N-1, \\
D_i &= \frac{1}{2}(D_i'+D_i'^T), \quad i=0,\dots,N, \\
E &= E'.
\end{aligned}$$

The dual of (PG-R) is then given by

$$\begin{aligned}
\min_{a_i,b_i,c_i,d_t,e} \quad & LR^2 e \\
\text{s.t.} \quad & \sum_{i=1}^{N-1} a_i A_i + \sum_{i=0}^{N-1} b_i B_i + \sum_{i=0}^{N-1} c_i C_i + \sum_{i=0}^{N} d_i D_i + eE \succeq 0, \\
& c_0 - d_0 = 0, \\
& c_i - c_{i-1} - d_i = 0, \quad i = 1,\ldots,N-1, \\
& c_{N-1} + d_N = 1, \\
& a_i, b_i, c_i, d_i, e \geq 0.
\end{aligned}$$

To show an upper bound on the primal problem (and hence an upper bound on the efficiency estimate of the method), it is enough to find a dual feasible point. Let

$$\begin{aligned}
\hat{a}_i &= \frac{\sum_{k=0}^{i-1} h_k}{h_i\left(2\sum_{k=0}^{N-1} h_k - \sum_{k=0}^{i-1} h_k\right)}, \quad i = 1,\ldots,N-1, \\
\hat{b}_0 &= \frac{h_1}{h_0}\hat{a}_1, \\
\hat{b}_i &= \frac{h_{i+1}}{h_i}\hat{a}_{i+1} - \hat{a}_i, \\
\hat{b}_{N-1} &= \frac{1}{h_{N-1}} - \hat{a}_{N-1}, \\
\hat{c}_i &= \frac{\sum_{k=0}^{i} h_k}{2\sum_{k=0}^{N-1} h_k - \sum_{k=0}^{i-1} h_k}, \quad i = 0,\ldots,N-1, \\
\hat{d}_0 &= \hat{c}_0, \\
\hat{d}_i &= \hat{c}_i - \hat{c}_{i-1}, \\
\hat{d}_N &= 1 - \hat{c}_{N-1}, \\
\hat{e} &= \frac{1}{4\sum_{k=0}^{N-1} h_k},
\end{aligned}$$

then to complete the derivation of the bound, we need to verify that the constraints in the dual problem are satisfied. Since the linear constraints are trivially satisfied, we are left with verifying the matrix inequality:

$$\hat{S} = \sum_{i=0}^{N-1} \hat{a}_i A_i + \sum_{i=0}^{N-1} \hat{b}_i B_i + \sum_{i=0}^{N-1} \hat{c}_i C_i + \sum_{i=0}^{N} \hat{d}_i D_i + \hat{e}E \succeq 0.$$

Let

$$\xi = (\xi_0,\ldots,\xi_N,\psi_0,\ldots,\psi_N,\xi_*,\psi_*)^T \in \mathbb{R}^{2N+2}$$

then since

$$\begin{aligned}
\xi^T \sum_{i=1}^{N-1} a_i A_i \xi &= \sum_{i=1}^{N-1} a_i \left((\xi_i - \xi_{i+1})^2 - h_i \psi_i (\xi_i - \xi_{i+1}) \right), \\
\xi^T \sum_{i=0}^{N-1} b_i B_i \xi &= \sum_{i=0}^{N-1} b_i \left((\xi_i - \xi_{i+1})(\xi_* - \xi_{i+1}) - h_i \psi_i (\xi_* - \xi_{i+1}) \right), \\
\xi^T \sum_{i=0}^{N-1} c_i C_i \xi &= \sum_{i=0}^{N-1} c_i \left(\frac{1}{2}(\psi_{i+1} - \psi_i)^2 + \psi_{i+1}(\xi_i - \xi_{i+1}) \right), \\
\xi^T \sum_{i=0}^{N} d_i D_i \xi &= \sum_{i=0}^{N} d_i \left(\frac{1}{2}(\psi_i - \psi_*)^2 + \psi_i (\xi_* - \xi_i) \right), \\
\xi^T e E \xi &= e(\xi_0 - \xi_*)^2,
\end{aligned}$$

we have

$$\begin{aligned}
\xi^T \hat{S} \xi &= \sum_{i=1}^{N-1} \hat{a}_i \left((\xi_i - \xi_{i+1})^2 - h_i \psi_i (\xi_i - \xi_{i+1}) \right) + \sum_{i=0}^{N-1} \hat{b}_i \left((\xi_i - \xi_{i+1})(\xi_* - \xi_{i+1}) - h_i \psi_i (\xi_* - \xi_{i+1}) \right) \\
&+ \sum_{i=0}^{N-1} \hat{c}_i \left(\frac{1}{2}(\psi_{i+1} - \psi_i)^2 + \psi_{i+1}(\xi_i - \xi_{i+1}) \right) + \sum_{i=0}^{N} \hat{d}_i \left(\frac{1}{2}(\psi_i - \psi_*)^2 + \psi_i (\xi_* - \xi_i) \right) + \hat{e}(\xi_0 - \xi_*)^2.
\end{aligned}$$

Setting $\hat{a}_0 = 0$ and extending the definition of $\hat{a}_i$ to include $i = N$ (h_N can be set to any arbitrary value since it is not actually used), we have the following identity, which was derived by inspecting the Cholesky decomposition of $\hat{S}$:

$$\begin{aligned}
&\xi^T \hat{S} \xi = \\
&\sum_{i=0}^{N-1} \frac{\hat{c}_i (2\hat{a}_i + \hat{b}_i)}{2\hat{a}_{i+1} h_{i+1}} \left(\xi_i - \frac{\hat{a}_{i+1} h_{i+1}}{\hat{c}_i} \xi_{i+1} - \frac{\hat{a}_{i+1} h_{i+1}}{2\hat{a}_i + \hat{b}_i} \psi_i + \frac{\hat{a}_{i+1} h_{i+1}}{2\hat{a}_i + \hat{b}_i} \psi_{i+1} + \frac{\hat{a}_{i+1} h_{i+1} (\hat{c}_i - \hat{a}_i h_i)}{\hat{c}_i} \xi_* \right)^2 \\
&\quad + \sum_{i=0}^{N-1} \frac{\hat{c}_i}{2} \left(1 - \frac{\hat{a}_{i+1} h_{i+1}}{2\hat{a}_i + \hat{b}_i} \right) (\psi_i - \psi_{i+1})^2 + \sum_{i=0}^{N} \frac{1}{2} \hat{d}_i (\psi_i - \psi_*)^2.
\end{aligned}$$

The verification of this identity is very involved, so we leave it to our dedicated readers (who should use the help of their favourite CAS, for the sake of their sanity).

Conclusion Assuming $h_i > 0$, the only possibly negative term in the expansion above is $1 - \frac{a_{i+1} h_{i+1}}{2a_i + b_i}$, hence as long as it is nonnegative for every i, the point is feasible and the following theorem holds.

Theorem 2.7. *Suppose $h_i > 0$ and $\frac{a_{i+1} h_{i+1}}{2a_i + b_i} \leq 1$ for every i, then*

$$f(x_N) - f(x_*) \leq \frac{L \|x_0 - x^*\|^2}{4 \sum_{k=0}^{N-1} h_k}.$$

It can be easily shown that when $h_i \equiv h$ for some $h > 0$, the condition above is reduced to $0 < h \leq 1$, hence we get the following bound on the projected gradient method:

Corollary 2.8. *Suppose $h_i \equiv h$, $0 < h \leq 1$ then*

$$f(x_N) - f(x_*) \leq \frac{L\|x_0 - x^*\|^2}{4Nh}.$$

Note that the bound above is the best possible bound, as shown by the following result.

Theorem 2.9. *There exists a function $f_0 \in C_L^{1,1}(\mathbb{R})$ and a set $C \subset \mathbb{R}$ such that if $x_0, \ldots, x_N$ are generated by applying the projected gradient method on f_0, then*

$$f_0(x_N) - f_0^* = \frac{L\|x_0 - x^*\|^2}{4\sum_{k=0}^{N-1} h_k}.$$

Proof. We assume WLOG that $L = 1$, $R = 1$, and $x_0 = 1$. Take $C = \{x \in \mathbb{R} \mid x \geq 0\}$ and $f_0(x) = cx$, for some $c > 0$. Then as long as $1 - c\sum_{k=0}^{N-1} h_k \geq 0$, we have

$$x_N = 1 - c\sum_{k=0}^{N-1} h_k$$

and

$$f_0(x_N) - f_0^* = c\left(1 - c\sum_{k=0}^{N-1} h_k\right).$$

Hence, by taking $c = (2\sum_{k=0}^{N-1} h_k)^{-1}$, we reach the desired result. □

2.9 Appendix III: A PEP for the Class of Strongly Convex Functions

In this appendix, we use the PEP approach to demonstrate a gap in the theoretical analysis of the FGM: we show that the worst-case behavior of the (classical) FGM, when provided with strongly convex functions with an unknown strong convexity parameter, is considerably better than predicted by the classical efficiency estimate (2.4.1).

Suppose f is a strongly convex function with constant μ and has Lipschitz-continuous gradient with constant L. Applying (2.3.2) on $f(x) - \frac{1}{2}\mu\|x\|^2$, which has a Lipschitz-continuous gradient with constant $L - \mu$, we get the following property for any $x, y \in \mathbb{R}^d$

$$\frac{1}{2(L-\mu)}\|f'(x) - \mu x - f'(y) + \mu y\|^2 \leq f(x) - \frac{1}{2}\mu\|x\|^2 - f(y) + \frac{1}{2}\mu\|y\|^2 - \langle f'(y) - \mu y, x - y\rangle,$$

or, after some manipulations, we reach

$$\begin{aligned}\frac{1}{2(L-\mu)}\|f'(x) - f'(y)\|^2 + \frac{\mu L}{2(L-\mu)}\|x - y\|^2 \\ \leq f(x) - f(y) - \frac{1}{L-\mu}\langle Lf'(y) - \mu f'(x), x - y\rangle.\end{aligned} \tag{2.9.1}$$

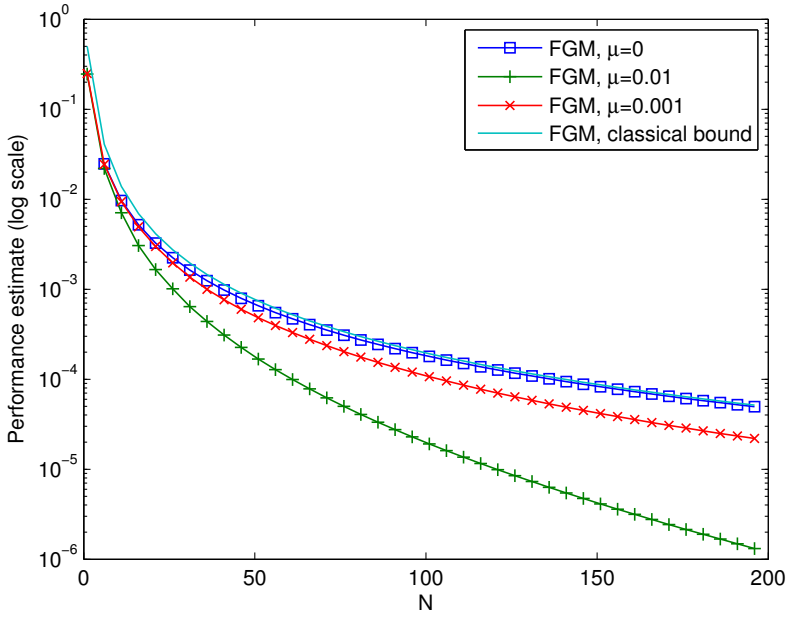

Figure 2.2: The computed worst-case bounds for the classical FGM, when applied on strongly convex functions with $L = 1$, $R = 1$, and various values of μ.

A PEP can now be formed in the usual way using (2.9.1) instead of (2.3.2) and bounds on the worst-case performance of the FGM (and Algorithm FO, in general) can then be found numerically as detailed in the chapter above. A MATLAB code for calculating these bound is available at Listing 2.1.

Figure 2.2 summarizes the obtained numerical bounds on the FGM method, as described in Example 2.4.2, for various values of μ with $L = 1$ and $R = 1$. Note that this version of the FGM does not assume any prior knowledge on the strong convexity parameter, μ, and is *not* an optimal method for the class of strongly convex functions. However, as can be seen from the figure, the method preforms significantly better than the known analytical bound, and even when the condition number of the problem, L/μ, is as large as 1000, the computed worst-case performance of the method after 200 iterations is more than two times better than predicted by the analytical bound. When the condition number of the problem is 100, the computed performance of the method after 200 iterations becomes more than 100 times better than predicted by the analytical bound.

To the best of our knowledge, these results provide the first complexity-based evidence for the well-known superior practical performance of the FGM.

We leave the question of finding a more accurate analytical bound for this problem for future research.

Listing 2.1: A PEP for the FGM

```
1 function bound = Nesterov83PEP2_sc(N, iL, mu, R)
2 %A PEP for Nesterov's '83 FGM
3 %Returns an efficiency estimate for the N'th iteration of the FGM method
      assuming it is given a function in the class of strongly convex
```

```
    functions with Lipschitz-continuous gradient
%N: The number of iterations
%iL: The Lipschitz constant of the gradient
%mu: The strong convexity parameter
%R: An upper bound on the distance from the initial iterate to an
    optimal solution

if (nargin<=1)
    iL=1;
    mu=1/16;
    R=1;
end
if (nargin==0)
    N=9;
end
global mem; mem=[];   %for caching the x seq.
global mem2; mem2=[]; %for caching the a coeff.
global n;
global L;
n=N;
L=iL;

problemdim=totalvars();
cvx_precision best;
cvx_solver sedumi;
cvx_begin quiet
    variable Z(problemdim,problemdim);
    variable delta_vary(N+1);

    global gZ;gZ=Z;
    global dvy;dvy=delta_vary;

    maximize (deltay(N));
    subject to
        Z==semidefinite(problemdim);
        %The strong convexity property (2.9.1), applied on the pair of
            points y(i), y(i+1)
        for i=0:N-1
            1/(L-mu)*(0.5*mu*L*Znorm2(y(i)-y(i+1))+0.5*Znorm2(g(i)-g(i
                +1))-Zdot(y(i)-y(i+1),mu*g(i)-L*g(i+1)))<=deltay(i)-
                deltay(i+1);
        end
        %The strong convexity property (2.9.1), applied on the pair of
            points y(i), x*
        for j=0:N
            1/(L-mu)*(0.5*mu*L*Znorm2(xopt()-y(j))+0.5*Znorm2(gopt()-g(j
                ))-Zdot(xopt()-y(j),mu*gopt()-L*g(j)))<=-deltay(j);
        end
        %Bounded start
        Znorm2(y(0)-xopt())<=R*R;
cvx_end
bound=cvx_optval;
end
```

```
50
51 %*****************************************
52 %Helper functions:
53 %*****************************************
54 function n=Znorm2(v)
55     global gZ;
56     n=v'*gZ*v;
57 end
58 function d=Zdot(u,v)
59     global gZ;
60     d=u'*gZ*v;
61 end
62 function a=lowa(i)
63     global mem2;
64     if isempty(mem2)
65         mem2={};
66     end
67     if i==0
68         a=1;
69     else
70         if (length(mem2)<i) || isempty(mem2{i})
71             mem2{i}=0.5*(1+sqrt(1+4*lowa(i-1)^2));
72         end
73         a=mem2{i};
74     end
75 end
76 function nv=gvars
77     global n;
78     nv=n+1;
79 end
80 function nv=xoptvars
81     nv=1;
82 end
83 function nv=goptvars
84     nv=1;
85 end
86 function t=totalvars
87     t=gvars()+xoptvars()+goptvars();
88 end
89 function v=g(i)
90     global n;
91     if (i<0 || i>n)
92         error('argument error in g(i)');
93     end
94     v=sparse(totalvars(),1);
95     v(i+1)=1;
96 end
97
98 function v=xopt
99     v=sparse(totalvars(),1);
100    v(gvars()+1)=1;
101 end
102 function v=gopt
```

```
103     v=sparse(totalvars(),1);
104     v(gvars()+xoptvars()+1)=1;
105 end
106 function v=x(i)
107     global n;
108     global L;
109     global mem;
110     if (i<-1 || i>n)
111         error('argument error in x(i)');
112     end
113     if isempty(mem)
114         mem={};
115     end
116
117     if i==-1
118         v=sparse(totalvars(),1);
119     else
120         if (size(mem,2)<i+2) || isempty(mem{i+2})
121             v=y(i)-g(i)/L; %The definition of the x sequence
122             mem{i+2}=v;
123         else
124             v=mem{i+2};
125         end
126
127     end
128 end
129 function v=y(i)
130     global n;
131     if (i<0 || i>n)
132         error('argument error in y(i)');
133     end
134
135     if i==0
136         v=sparse(totalvars(),1);
137     else
138         v=(1+(lowa(i-1)-1)/lowa(i))*x(i-1)-(lowa(i-1)-1)/lowa(i)*x(i-2);
                %The definition of the y sequence
139     end
140 end
141 function d=deltay(i)
142     global dvy;
143     global n;
144     if (i<0 || i>n)
145         error('argument error in deltay(i)');
146     end
147     if (i~=n+1)
148         d=dvy(i+1);
149     else
150         d=0;
151     end
152 end
```

Chapter 3

An optimal variant of Kelley's cutting-plane method

We propose a new variant of Kelley's cutting-plane method for minimizing a nonsmooth convex Lipschitz-continuous function over the Euclidean space. We derive the method through a constructive approach and prove that it attains the optimal rate of convergence for this class of problems. In addition, we present an aggregation strategy for obtaining a memory-limited version of the method and discuss some other situations where the approach presented here is applicable.

3.1 Introduction

In this chapter, we focus on unconstrained nonsmooth convex minimization problems, where information on the objective can only be gained through a first-order oracle, which returns the value of the objective and an element in its subgradient at any point in the problem's domain. Problems of this type often arise in real-life applications either as the result of a transformation that was applied on a problem (such as Benders' decomposition [25]) or by some inherent property of the problem (e.g., in an eigenvalue optimization problem).

One of the earliest and most fundamental methods for solving nonsmooth convex problems is Kelley's cutting plane method (or, the Kelley method, for short), which was introduced by Kelley in [53] and also independently by Cheney and Goldstein [37]. The method maintains a polyhedral model of the objective, and at each iteration updates this model according to the first-order information at a point where the model predicts that the objective is minimal. Despite the elegant and intuitive nature of this method, the Kelley method suffers from very poor performance, both in practice and in theory [72]. The source of the poor performance seems to be the instability of the solution, where the iterates of the method tend to be far apart and at locations where the accuracy of the model is poor.

The main objective of this work is to present a new method for minimizing a nonsmooth convex Lipschitz-continuous function over the Euclidean space, which is surprisingly similar to the Kelley method, yet attains the optimal rate of convergence for this class of problems. We derive this method and its rate of convergence through a constructive approach which further develops and extends the framework introduced in Chapter 2. In particular, here the derivation

of a tractable complexity bound leads itself to the construction of the proposed algorithm.

Although the main contribution of this work is entirely theoretical, it should be noted that the resulting method also offers some practical advantages over existing bundle methods. One of the main advantages is that the method allows the implementation to choose at each iteration between two types of steps: a "standard" step, which, as in all bundle methods, requires solving an auxiliary convex optimization program, and an "easy" step which involves only a subgradient step with a predetermined step size. The efficiency estimate of the method remains valid regardless of the choices a specific implementation makes, thereby allowing the implementation to find a balance between accuracy and speed (without performing aggregation on the iterates, which affects the accuracy of the model).

One limitation of the method is that it requires choosing the number of iterations to be performed in advance. However, this limitation is not severe since the "standard" steps provide as a by-product a bound on the worst-case absolute inaccuracy at the *end* of the method's run, hence once the desired accuracy has been achieved, the implementation can choose to perform only "easy" steps thereby quickly ending the execution of the method.

Literature The first successful approach for overcoming the instability in the Kelley method, known as the bundle method, was introduced by Lemaréchal [61] and also independently by Wolfe [100]. In the bundle approach, the instability in the Kelley method is tackled by introducing a regularizing quadratic term in the objective, thereby forcing the next iterate to remain in close proximity to the previous iterates, where the model is more accurate. The bundle approach proved to be very fruitful, and yielded many variations on the idea, see for instance [8, 56, 63] and references therein. The bundle method and its variants also proved to perform very well in practice; however, a theoretical rate of convergence is not available for most variants, and for the variants where a rate of convergence was established, it was shown to be suboptimal [58].

Another fundamental approach is the level bundle method, introduced by Lemaréchal et al. [62]. The idea behind this approach is that the level sets of the polyhedral model of the objective are "stable", and therefore they should be used instead of the complete model. Building on this idea, at each iteration the method performs a projection of the previous iterate on a carefully selected level set of the model, then updates the model according to the first-order information at the resulting point. Several extensions to the method were proposed, including a restricted memory variant [57] and a variant for handling non-Euclidean metrics [24]. The method was shown to possess an optimal rate of convergence, however, note that the constant factor in the bound is not optimal, and leaves room for improvement.

Finally, let us mention that quite a few additional approaches were proposed. Among them are trust-region bundle methods [91] and the bundle-newton method [65], where the objective is approximated by a combination of polyhedral and quadratic functions. For a comprehensive survey, we refer the reader to [66].

Outline. The chapter is organized as follows. In Section 3.2, we present the new Kelley-Like Method (KLM), and state our main result: an optimal rate of convergence (Theorem 3.1). The motivation for the method and our approach is described in Section 3.3. In Sections 3.4–3.6, we provide a detailed description of the construction of the proposed method and prove its rate of convergence. We conclude the main body of the work, in Section 3.7, where we discuss a

limited-memory version of the method and present some additional cases where the approach presented here is applicable. Finally, in Appendix 3.8, we give a new lower-complexity bound for the class of convex and Lipschitz-continuous minimization problems, which shows that the KLM attains the best possible rate of convergence for this class of problems.

Notation. For a convex function f, its subdifferential at x is denoted by $\partial f(x)$ and we use $f'(x)$ to denote some element in $\partial f(x)$. We also denote $f^* = \min_x f(x)$ and $x^* = x_f^* \in \operatorname{argmin}_x f(x)$. The Euclidean norm of a vector x is denoted as $\|x\|$. We use e_i for the i-th canonical basis vector, which consists of all zero components, except for its i-th entry which is equal to one. For an optimization problem (P), $\mathrm{val}(P)$ stands for its optimal value. For a symmetric matrix A, $A \succeq 0$ means A is positive semidefinite (PSD).

To simplify some expressions, we often write $A \succeq 0$ for a non-symmetric matrix A: this should be interpreted as $\frac{1}{2}(A + A^T) \succeq 0$.

3.2 The Algorithm and its Rate of Convergence

In this section we present our main results, namely the new proposed algorithm and its rate of convergence.

3.2.1 The Algorithm: a Kelley-Like Method (KLM)

Consider the minimization problem $\min\{f(x) : x \in \mathbb{R}^p\}$, where $f : \mathbb{R}^p \to \mathbb{R}$ is convex and Lipschitz-continuous with constant $L > 0$. The method described below assumes that $x^* \in \operatorname{argmin}_x f(x)$ is located inside a ball of radius $R > 0$ around a given point $x_0 \in \mathbb{R}^p$ and requires knowing in advance the number of iterations to be performed, N. The method proceeds as follows:

Algorithm KLM

Initialization: (The zeroth iteration.) Set

$$x_1 := x_0,\ s := 0,\ \tau := 1, \text{ and } \mu := \frac{R}{L\sqrt{N}}.$$

Iteration #M: At the Mth iteration ($1 \le M \le N-1$), the method *arbitrarily* chooses between two types of steps:

In the first type (the "standard step"), we set $m \in \operatorname{argmin}_{1 \le i \le M} f(x_i)$ and solve

$$\begin{aligned}(B_M) \qquad \max_{y \in \mathbb{R}^p, \zeta, t \in \mathbb{R}} \ & f(x_m) - t \\ \text{s.t. } & f(x_i) + \langle y - x_i, f'(x_i) \rangle \le t, \quad i = 1, \dots, M, \\ & f(x_m) - L\zeta \le t, \\ & \|y - x_0\|^2 + (N - M)\zeta^2 \le R^2.\end{aligned}$$

Let y^*, ζ^* and t^* be an optimal solution to the primal variables of problem (B_M), and let β^* be the optimal dual multiplier that corresponds to the constraint $f(x_m) - L\zeta \leq t$. The step then proceeds by setting

$$\text{(standard step)} \quad x_{M+1} := y^*,$$

and updating

$$s := M,\ \tau := \beta^*,\ \mu := \frac{\zeta^*}{L}.$$

The second type of step (the "easy step") is a subgradient step with the previously selected step size μ:

$$\text{(easy step)} \quad x_{M+1} := x_M - \mu f'(x_M).$$

Output: The output is given by a convex combination of the best step from the first s steps and the ergodic combination of the last $N - s$ steps:

$$\bar{x}_N := (1-\tau)x_m + \frac{\tau}{N-s}\sum_{j=s+1}^{N} x_j,$$

here $m \in \operatorname{argmin}_{1\leq i\leq s} f(x_i)$.

Note that if the method chooses to perform an "easy" step at every iteration, it simply reduces to the subgradient method with a constant step size. Also note that the "standard" step shares the computational simplicity of the main step in the Kelley method (cf. next section), where the two iteration rules differ only in the introduction of the optimization variable ζ and in the inclusion of the second constraint in (B_M).

3.2.2 An Optimal Rate of Convergence for KLM

We now state the efficiency estimate of the method, which shows that the new method is optimal for the class of nonsmooth minimization with convex and Lipschitz-continuous functions (see Appendix 3.8 and also [72, 74]).

Theorem 3.1. *Suppose $\bar{x}_N$ is generated by Algorithm KLM, and let s be the index of the last iteration where a "standard" step was taken (or zero, when no such step was taken), then*

$$f(\bar{x}_N) - f^* \leq \operatorname{val}(B_s) \leq \frac{LR}{\sqrt{N}}. \tag{3.2.1}$$

Note that although the rate of convergence is of same order as for the level bundle method [62], which to the best of our knowledge has the best known efficiency estimate on a bundle method, the constant term here is smaller by a factor of two. Hence, the proposed method requires a quarter of the steps in order to the reach the same worst-case absolute inaccuracy.

The rest of this chapter is devoted to the detailed construction of the proposed Algorithm KLM and to the proof of Theorem 3.1.

3.3 Motivation

3.3.1 A New Look at the Kelley Method

Consider the problem

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

where $f(x)$ is convex, nonsmooth, and Lipschitz-continuous with constant L. For a given set of trial points, $\mathscr{J}_M := \{(x_j, f(x_j), f'(x_j))\}_{j=1}^M$, denote by $f_M(x)$ the polyhedral model of the function f, defined by

$$f_M(x) = \max\{f(x_j) + \langle f'(x_j), x - x_j\rangle \mid 1 \le j \le M\}. \tag{3.3.1}$$

Assuming that $x_f^* \in \operatorname{argmin}_x f(x)$ lies inside a compact set, which we take here as $\{x : \|x - x_0\| \le R\}$ for some $x_0 \in \mathbb{R}^p$ and $R > 0$, the Kelley method chooses the next iterate, x_{M+1}, by solving

$$\text{(Kelley)} \quad x_{M+1} \in \operatorname{argmin}_{\|x-x_0\|\le R} f_M(x).$$

Alternatively, we can write the previous rule as the following functional optimization problem:

$$\begin{aligned} \text{(Kelley}') \quad x_{M+1} \in \operatorname{argmin}_{\|x-x_0\|\le R} \ & \min_{\varphi\in C_L, \varphi \text{ is convex}} \ \varphi(x) \\ & \text{s.t. } \varphi(x_i) = f(x_i), \quad i = 1,\dots,M, \\ & \quad f'(x_i) \in \partial\varphi(x_i), \quad i = 1,\dots,M, \\ & \quad \|x_\varphi^* - x_0\| \le R, \end{aligned}$$

where the two formulations are equivalent since the solution to the inner minimization problem reduces exactly to f_M inside the ball $\|x - x_0\| \le R$.

The well-known inefficient nature of the method is now apparent: the method chooses the next iterate as one that minimizes the *best-case function value*, which is not a natural strategy when we are interested in obtaining a bound on the *worst-case absolute inaccuracy*, $f(x_{M+1}) - f^*$. This motivates us to consider the following alternative strategy.

3.3.2 The Proposed Approach

Since we are interested in deriving a bound on the worst-case behavior of the absolute inaccuracy, a natural approach, given a set of trial points, $\mathscr{J}_M := \{(x_j, f(x_j), f'(x_j))\}_{j=1}^M$, might be to choose the next iterate in a way such that the worst-case absolute inaccuracy is minimized, i.e.,

$$\begin{aligned} x_{M+1} \in \operatorname{argmin}_{x\in\mathbb{R}^p} \ & \max_{\varphi\in C_L, \varphi \text{ is convex}} \ \varphi(x) - \varphi^* \\ & \text{s.t. } \varphi(x_i) = f(x_i), \quad i = 1,\dots,M, \\ & \quad f'(x_i) \in \partial\varphi(x_i), \quad i = 1,\dots,M, \\ & \quad \|x_\varphi^* - x_0\| \le R. \end{aligned}$$

It appears, however, that this *greedy* approach forces the resulting iterates to be too conservative. In fact, numerical tests show that in some cases the sequence generated by this approach does not even converge to a minimizer of f!

We therefore take a *global* approach and attempt to minimize a bound on the worst-case behavior of the entire sequence, i.e., instead of choosing only the next iterate x_{M+1}, given some $N > M$, we look for a sequence $x_{M+1}, \ldots, x_N$ for which the absolute inaccuracy at the last iterate, x_N, is minimized. In order to accomplish this, we need to assume some form of structure on the sequence $\{x_1, \ldots, x_N\}$.

Let $\{v_1, \ldots, v_r\}$ be an orthonormal set that spans $\{f'(x_1), \ldots, f'(x_M), x_1 - x_0, \ldots, x_M - x_0\}$. Hereafter, we consider sequences $x_{M+1}, \ldots, x_N$ that are generated according to a first-order method of the form

$$x_i = x_0 + \sum_{k=1}^{i-1} h_{1,k}^{(i)}(x_k - x_0) - \sum_{k=1}^{r} h_{2,k}^{(i)} v_k - \sum_{k=M+1}^{i-1} h_{3,k}^{(i)} f'(x_k), \quad i = M+1, \ldots, N, \tag{3.3.2}$$

for step sizes $h_{j,k}^{(i)} \in \mathbb{R}$ that depend only on the data available at the current stage (i.e., L, R and $\mathscr{J}_M$). Note that the first summation is redundant here and can be expressed using the other terms, however, including it will significantly simplify the following analysis.

For sequences of this form, given $h = (h_{j,k}^{(i)})$, the worst-case absolute inaccuracy at x_N is, by definition, the solution to

$$\begin{aligned} P_M(h) := \max_{\varphi \in C_L, \varphi \text{ is convex}} \quad & \varphi(x_N) - \varphi^* \\ \text{s.t. } & x_i = x_0 + \sum_{k=1}^{i-1} h_{1,k}^{(i)}(x_k - x_0) - \sum_{k=1}^{r} h_{2,k}^{(i)} v_k - \sum_{k=M+1}^{i-1} h_{3,k}^{(i)} \varphi'(x_k), \\ & \qquad i = M+1, \ldots, N, \\ & \varphi(x_i) = f(x_i), \quad i = 1, \ldots, M, \\ & f'(x_i) \in \partial\varphi(x_i), \quad i = 1, \ldots, M, \\ & \|x_\varphi^* - x_0\| \le R. \end{aligned}$$

Therefore, the problem of finding step sizes h such that the worst-case absolute inaccuracy at x_N is minimized can be expressed by

$$(P_M) \quad \min_h P_M(h).$$

Note that obtaining an optimal solution for (P_M) is not necessary. Indeed, suppose that for any h we can find a (preferably easy) upper bound $Q_M(h)$ for $P_M(h)$, then it follows that

$$f(x_N) - f^* \le P_M(h) \le Q_M(h),$$

hence a method with a "good" worst-case absolute inaccuracy might be found by minimizing $Q_M(h)$ with respect to h instead of $P_M(h)$. The analysis developed in the forthcoming two sections show how to achieve this, and serves two main goals:

- Derive a tractable upper-bound for the worst-case absolute inaccuracy expressed via problem (P_M).
- Show that the derivation of this bound leads itself to the construction of Algorithm KLM.

3.4 A Tractable Upper-Bound for (P_M)

Problem $(P_M(h))$ (and hence problem (P_M)) is a difficult abstract optimization problem in infinite dimension through the functional constraint on φ. Inspired by the approach developed in Chapter 2, we start by formulating a finite dimensional relaxation of the problem.

3.4.1 A Finite Dimensional Relaxation of (P_M)

To relax (P_M) into a finite dimensional problem, we need to tackle the constraint "$\varphi \in C_L$, φ is convex", which states that for all $u, v \in \mathbb{R}^p$

$$\text{[subgradient inequality]} \qquad \varphi(v) - \varphi(u) \leq \langle \varphi'(v), v-u \rangle, \tag{3.4.1}$$

$$\text{[Lipschitz continuity]} \qquad \|\varphi'(u)\| \leq L, \tag{3.4.2}$$

where $\varphi'(v)$ is an element of $\partial\varphi(v)$. For that purpose, we introduce the variables

$$\begin{aligned} &x_* \in \operatorname{argmin}_x \varphi(x), \\ &\delta_i = \varphi(x_i), \quad i = M+1, \ldots, N, *, \\ &g_i \in \partial\varphi(x_i), \quad i = M+1, \ldots, N, *, \end{aligned}$$

and for ease of notation, we set

$$\begin{aligned} &\delta_j = f(x_j), \quad j = 1, \ldots, M, \\ &g_j = f'(x_j), \quad j = 1, \ldots, M. \end{aligned}$$

We now relax $P_M(h)$ by replacing the function variable φ with the new variables and by introducing constraints that follow from the application of the subgradient inequality (3.4.1) and the Lipschitz-continuity of φ (3.4.2) at the points $x_1, \ldots, x_N, x_*$. Minimizing the resulting problem with respect to h, we reach the following minimax problem in finite dimension:

$$\begin{aligned} \min_h \max_{\substack{g_{M+1},\ldots,g_N,g_*,x_* \in \mathbb{R}^p, \\ \delta_{M+1},\ldots,\delta_N,\delta_* \in \mathbb{R}}} & \delta_N - \delta_* \\ \text{s.t. } x_i = x_0 + \sum_{k=1}^{i-1} h_{1,k}^{(i)}(x_k - x_0) - \sum_{k=1}^{r} h_{2,k}^{(i)} v_k - \sum_{k=M+1}^{i-1} h_{3,k}^{(i)} g_k, & \quad i = M+1, \ldots, N, \\ \delta_i - \delta_j \leq \langle g_i, x_i - x_j \rangle, & \quad i, j = 1, \ldots, N, *, \\ \|g_i\|^2 \leq L^2, & \quad i = 1, \ldots, N, * \\ \|x_* - x_0\|^2 \leq R^2. & \end{aligned}$$

Recall that δ_j, g_j and x_j, $j = 1, \ldots, M$, are given in advance (these are the trial points) and are considered as the problem data.

It appears that this minimax problem (which clearly is not convex-concave) remains non-trivial to tackle. We therefore consider a relaxation obtained by removing some constraints:

$$
\begin{aligned}
(P_M^I) \quad \min_{h} \max_{\substack{g_{M+1},\ldots,g_N,x_*\in\mathbb{R}^p,\\ \delta_{M+1},\ldots,\delta_N,\delta_*\in\mathbb{R}}} \; & \delta_N-\delta_* \\
\text{s.t. } & x_i = x_0+\sum_{k=1}^{i-1} h_{1,k}^{(i)}(x_k-x_0)-\sum_{k=1}^{r} h_{2,k}^{(i)}v_k-\sum_{k=M+1}^{i-1} h_{3,k}^{(i)}g_k, \quad i=M+1,\ldots,N,\\
& \delta_i-\delta_j \le \langle g_i, x_i-x_j\rangle, \quad i=M+1,\ldots,N,\ j=1,\ldots,i-1,\\
& \delta_i-\delta_* \le \langle g_i, x_i-x_*\rangle, \quad i=1,\ldots,N,\\
& \|g_i\|^2 \le L^2, \quad i=M+1,\ldots,N,\\
& \|x_*-x_0\|^2 \le R^2.
\end{aligned}
$$

The omitted constraints can be shown to be inactive. However, this is not necessary for the following arguments as we are currently only interested in *finding an upper bound* on the absolute inaccuracy.

As before, the inner maximization problem is denoted by $(P_M^I(h))$, and we have

$$\mathrm{val}(P_M) \le \mathrm{val}(P_M^I) = \min_h P_M^I(h).$$

Our first main objective is now to derive a tractable convex minimization problem which is an upper-bound for the minimax problem (P_M^I). The first step in that direction is the derivation of a semidefinite programming relaxation of the inner maximization problem $P_M^I(h)$. At this juncture, the reader might naturally be wondering why we do not derive directly a dual problem of the inner maximization to reduce our minimax problem to a minimization problem. It turns out that the SDP relaxation derived below enjoys a fundamental monotonicity property (see Lemma 3.9), which will play a crucial role in the proof of the main complexity result Theorem 3.1.

3.4.2 Relaxing The Inner Maximization Problem to an SDP

We proceed by performing a semidefinite relaxation on $P_M^I(h)$, the inner maximization problem of (P_M^I). Let $X \in \mathbb{S}^{1+r+N-M}$ be

$$
X = \begin{pmatrix}
\langle x_*-x_0, x_*-x_0\rangle & \langle x_*-x_0, v_1\rangle & \cdots & \langle x_*-x_0, v_r\rangle & \langle x_*-x_0, g_{M+1}\rangle & \cdots & \langle x_*-x_0, g_N\rangle \\
\langle v_1, x_*-x_0\rangle & \langle v_1, v_1\rangle & \cdots & \langle v_1, v_r\rangle & \langle v_1, g_{M+1}\rangle & \cdots & \langle v_1, g_N\rangle \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\langle v_r, x_*-x_0\rangle & \langle v_r, v_1\rangle & \cdots & \langle v_r, v_r\rangle & \langle v_r, g_{M+1}\rangle & \cdots & \langle v_r, g_N\rangle \\
\langle g_{M+1}, x_*-x_0\rangle & \langle g_{M+1}, v_1\rangle & \cdots & \langle g_{M+1}, v_r\rangle & \langle g_{M+1}, g_{M+1}\rangle & \cdots & \langle g_{M+1}, g_N\rangle \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\langle g_N, x_*-x_0\rangle & \langle g_N, v_1\rangle & \cdots & \langle g_N, v_r\rangle & \langle g_N, g_{M+1}\rangle & \cdots & \langle g_N, g_N\rangle
\end{pmatrix},
$$

and let $\mathbf{v}_i, \mathbf{g}_i, \mathbf{x}_i \in \mathbb{R}^{1+r+N-M}$ be such that

$$\begin{aligned}
&\mathbf{v}_i = e_{1+i}, \quad i = 1,\dots,r,\\
&\mathbf{g}_i = \begin{cases} \sum_{k=1}^{r} \langle g_i, v_k\rangle \mathbf{v}_k, & i = 1,\dots,M,\\ e_{1+r+i-M}^T, & i = M+1,\dots,N, \end{cases}\\
&\mathbf{x}_i = \begin{cases} \sum_{k=1}^{r} \langle x_i - x_0, v_k\rangle \mathbf{v}_k, & i = 1,\dots,M,\\ \sum_{k=1}^{i-1} h_{1,k}^{(i)} \mathbf{x}_k - \sum_{k=1}^{r} h_{2,k}^{(i)} \mathbf{v}_k - \sum_{k=M+1}^{i-1} h_{3,k}^{(i)} \mathbf{g}_k, & i = M+1,\dots,N,\\ e_1, & i = *, \end{cases}
\end{aligned} \tag{3.4.3}$$

then it is straightforward to verify that the following identities hold

$$\begin{aligned}
&\mathbf{v}_i^T X \mathbf{v}_j = \langle v_i, v_j\rangle, \quad i,j = 1,\dots,r,\\
&\mathbf{g}_i^T X \mathbf{g}_j = \langle g_i, g_j\rangle, \quad i,j = 1,\dots,N,\\
&\mathbf{g}_i^T X \mathbf{x}_j = \langle g_i, x_j - x_0\rangle, \quad i = 1,\dots,N,\ j = 1,\dots,N,*,\\
&\mathbf{x}_i^T X \mathbf{x}_j = \langle x_i - x_0, x_j - x_0\rangle, \quad i,j = 1,\dots,N,*.
\end{aligned} \tag{3.4.4}$$

Now, by using (3.4.4) in (P_M^I) and by relaxing the definition of X to $\mathbf{v}_i^T X \mathbf{v}_j = \langle v_i, v_j\rangle$ and $X \succeq 0$, we reach the following problem, whose inner maximization problem is an SDP:

$$\begin{aligned}
(P_M^{II}) \quad \min_{h} \max_{\substack{X \in \mathbb{S}^{1+r+N-M},\\ \delta_i, \delta_* \in \mathbb{R}}} \ & \delta_N - \delta_*\\
\text{s.t.} \quad & \delta_i - \delta_j \le \mathbf{g}_i^T X (\mathbf{x}_i - \mathbf{x}_j), \quad i = M+1,\dots,N,\ j = 1,\dots,i-1,\\
& \delta_i - \delta_* \le \mathbf{g}_i^T X (\mathbf{x}_i - \mathbf{x}_*), \quad i = 1,\dots,N,\\
& \mathbf{g}_i^T X \mathbf{g}_i \le L^2, \quad i = M+1,\dots,N,\\
& \mathbf{x}_*^T X \mathbf{x}_* \le R^2,\\
& \mathbf{v}_i^T X \mathbf{v}_j = \langle v_i, v_j\rangle, \quad i,j = 1,\dots,r,\\
& X \succeq 0.
\end{aligned}$$

3.4.3 Transforming the Minimax SDP to a Minimization Problem

To transform the minimax problem (P_M^{II}) into a minimization problem, we now use duality. More precisely, as shown below, by using Lagrangian duality for the inner maximization problem in (P_M^{II}) we derive a nonconvex (bilinear) semidefinite minimization problem whose optimal value coincides with the one of (P_M^{II}).

Lemma 3.2. *The minimax problem (P_M^{II}) reduces to the bilinear semi-definite minimization*

problem (P_M^{III}) *defined by*

$$
\begin{aligned}
(P_M^{III}) \min_h \min_{a,b,c,d,\Phi} & \sum_{i=M+1}^{N}\sum_{j=1}^{M} a_{i,j}\delta_j + \sum_{i=1}^{M} b_i(\langle g_i, x_i - x_0\rangle - \delta_i) + L^2 \sum_{i=M+1}^{N} c_i + R^2 d + \sum_{i=1}^{r} \Phi_{i,i} \\
s.t. \quad & -\sum_{i=M+1}^{N}\left(\sum_{j=1}^{i-1} a_{i,j}(\mathbf{x}_i - \mathbf{x}_j) + b_i\mathbf{x}_i\right)\mathbf{g}_i^T + \sum_{i=1}^{N} b_i\mathbf{x}_*\mathbf{g}_i^T \\
& + \sum_{i=M+1}^{N} c_i\mathbf{g}_i\mathbf{g}_i^T + d\mathbf{x}_*\mathbf{x}_*^T + \sum_{i,j=1}^{r}\Phi_{i,j}\mathbf{v}_i\mathbf{v}_j^T \succeq 0, \\
& (a,b) \in \Lambda,\ a_{i,j} \geq 0,\ b_i \geq 0,\ c_i \geq 0,\ d \geq 0,
\end{aligned}
$$

where

$$
\Lambda = \{(a,b) : \sum_{j=1}^{N-1} a_{N,j} + b_N = 1,\ \sum_{j=1}^{N} b_j = 1,\ \sum_{j=i+1}^{N} a_{j,i} - \sum_{j=1}^{i-1} a_{i,j} = b_i, \quad i = M+1,\ldots,N-1\}.
$$

Moreover, we have $\mathrm{val}(P_M^{II}) = \mathrm{val}(P_M^{III})$.

Proof. Consider the inner maximization problem in (P_M^{II}). We attach the dual variables to each of its constraints as follows:

$$
\begin{aligned}
a_{i,j} \in \mathbb{R}_+ &: \delta_i - \delta_j \leq \mathbf{g}_i^T X(\mathbf{x}_i - \mathbf{x}_j), \quad i = M+1,\ldots,N,\ j = 1,\ldots,i-1, \\
b_i \in \mathbb{R}_+ &: \delta_i - \delta_* \leq \mathbf{g}_i^T X(\mathbf{x}_i - \mathbf{x}_*), \quad i = 1,\ldots,N, \\
c_i \in \mathbb{R}_+ &: \mathbf{g}_i^T X\mathbf{g}_i \leq L^2, \quad i = M+1,\ldots,N, \\
d \in \mathbb{R}_+ &: \mathbf{x}_*^T X\mathbf{x}_* \leq R^2, \\
\Phi_{i,j} \in \mathbb{R} &: \mathbf{v}_i^T X\mathbf{v}_j = \langle v_i, v_j\rangle, \quad i,j = 1,\ldots,r.
\end{aligned}
$$

Recalling that δ_i and $\mathbf{g}_i^T X\mathbf{x}_i = \langle g_i, x_i - x_0\rangle$ are fixed for $i = 1,\ldots,M$, and that the set $\{v_1,\ldots,v_r\}$ is orthonormal, the Lagrangian for this maximization problem is given by

$$
\begin{aligned}
L(X,\delta;a,b,c,d,\Phi) &= \delta_N - \delta_* + \sum_{i=M+1}^{N} D_i\delta_i + D_*\delta_* + \mathrm{tr}(XW) + \mathscr{C}, \\
&\equiv L_1(\delta;a,b) + \mathrm{tr}(XW) + \mathscr{C},
\end{aligned}
$$

with

$$
\begin{aligned}
D_i &= -\sum_{j=1}^{i-1} a_{i,j} + \sum_{j=i+1}^{N} a_{j,i} - b_i, \quad i = M+1,\ldots,N, \\
D_* &= \sum_{j=1}^{N} b_j,
\end{aligned}
$$

$$
\begin{aligned}
W &= \sum_{i=M+1}^{N}\sum_{j=1}^{i-1} a_{i,j}(\mathbf{x}_i-\mathbf{x}_j)\mathbf{g}_i^T + \sum_{i=M+1}^{N} b_i\mathbf{x}_i\mathbf{g}_i^T - \sum_{i=1}^{N} b_i\mathbf{x}_*\mathbf{g}_i^T - \sum_{i=M+1}^{N} c_i\mathbf{g}_i\mathbf{g}_i^T \\
&\quad -d\mathbf{x}_*\mathbf{x}_*^T - \sum_{i,j=1}^{r}\Phi_{i,j}\mathbf{v}_i\mathbf{v}_j^T, \\
\mathscr{C} &= \sum_{i=M+1}^{N}\sum_{j=1}^{M} a_{i,j}\delta_j + \sum_{i=1}^{M} b_i(\langle g_i, x_i-x_0\rangle - \delta_i) + L^2\sum_{i=M+1}^{N} c_i + R^2 d + \sum_{i=1}^{r}\Phi_{i,i}.
\end{aligned}
$$

The dual objective function is then defined by

$$
H(a,b,c,d,\Phi) = \max_{\delta,X} L(X,\delta;a,b,c,d,\Phi) = \mathscr{C} + \max_{\delta} L_1(\delta;a,b) + \max_{X\succeq 0}\operatorname{tr}(XW).
$$

Since $L_1(\delta;a,b)$ is linear in the variables δ_i, $i = M+1,\ldots,N,*$, the first maximization problem is equal to zero whenever

$$
\begin{cases}
D_i = -\sum_{j=1}^{i-1} a_{i,j} + \sum_{j=i+1}^{N} a_{j,i} - b_i = 0, \quad i = M+1,\ldots,N-1, \\
1 + D_N = 1 - \sum_{j=1}^{N-1} a_{N,j} - b_N = 0, \\
-1 + D_* = -1 + \sum_{j=1}^{N} b_j = 0,
\end{cases}
$$

i.e., when $(a,b)\in\Lambda$, and is equal to infinity otherwise. Likewise, the second maximization is equal to zero whenever $W \preceq 0$, and is equal to infinity otherwise. Therefore, the dual problem of the inner maximization $P_M^{II}(h)$ reads as

$$
\min_{a,b,c,d,\Phi} H(a,b,c,d,\Phi) = \min_{a,b,c,d,\Phi}\{\mathscr{C} : W\preceq 0,\ (a,b)\in\Lambda,\ a_{i,j}\geq 0,\ b_i\geq 0,\ c_i\geq 0,\ d\geq 0\},
$$

and hence it follows that by minimizing the latter with respect to h, the minimax problem (P_M^{II}) reduces to the minimization problem (P_M^{III}), and the proof of the first claim is completed.

Now, as a consequence of weak duality for the pair of problems $(P_M^{II}(h))$–$(P_M^{III}(h))$ it immediately follows that

$$
\operatorname{val}(P_M^{II}) = \min_h P_M^{II}(h) \leq \min_h P_M^{III}(h) = \operatorname{val}(P_M^{III}).
$$

Furthermore, observing that the inner maximization problem in (P_M^{II}) is feasible and that the inner minimization problem in (P_M^{III}) is strictly feasible (since the elements in the diagonal of the SDP constraint, i.e., c_i, d, and $\Phi_{i,i}$, can be chosen to be arbitrarily large), then by invoking the conic duality theorem [23, Theorem 2.4.1], strong duality holds, and therefore it follows that $\operatorname{val}(P_M^{II}) = \operatorname{val}(P_M^{III})$. □

3.4.4 A Tight Convex SDP Relaxation for (P_M^{III})

At this stage, the minimization problem (P_M^{III}) we have just derived remains a nonconvex (bilinear) problem. Indeed, note that the vectors $\mathbf{x}_i$ depend on the optimization variable h, hence the terms $a_{i,j}(\mathbf{x}_i - \mathbf{x}_j)$ and $b_i\mathbf{x}_i$ in (P_M^{III}) are bilinear. We will now show that it is possible to derive a *tight convex relaxation* for this problem. This will be achieved through two main steps as follows.

Step I: Linearizing the bilinear SDP. As just noted, the terms $a_{i,j}(\mathbf{x}_i - \mathbf{x}_j)$ and $b_i\mathbf{x}_i$ in (P_M^{III}) are bilinear. Here we linearize these terms by introducing new variables $\xi_{i,j}$ and $\psi_{i,j}$ such that

$$-\left(\sum_{j=1}^{i-1} a_{i,j}(\mathbf{x}_i - \mathbf{x}_j) + b_i\mathbf{x}_i\right) = \sum_{j=1}^{r} \xi_{i,j}\mathbf{v}_j + \sum_{j=M+1}^{i-1} \psi_{i,j}\mathbf{g}_j, \quad i = M+1,\ldots,N. \tag{3.4.5}$$

Using (3.4.5) to eliminate the bilinear terms in (P_M^{III}) yields the following linear SDP:

$$\begin{aligned}
(P_M^{IV}) \quad \min_{a,b,c,d,\xi,\psi,\Phi} \; & \sum_{i=M+1}^{N}\sum_{j=1}^{M} a_{i,j}\delta_j + \sum_{i=1}^{M} b_i(\langle g_i, x_i - x_0\rangle - \delta_i) + L^2\sum_{i=M+1}^{N} c_i + R^2 d + \sum_{i=1}^{r}\Phi_{i,i} \\
\text{s.t.} \; & \sum_{i=M+1}^{N}\left(\sum_{j=1}^{r}\xi_{i,j}\mathbf{v}_j + \sum_{j=M+1}^{i-1}\psi_{i,j}\mathbf{g}_j\right)\mathbf{g}_i^T + \sum_{i=1}^{N} b_i\mathbf{x}_*\mathbf{g}_i^T \\
& \qquad + \sum_{i=M+1}^{N} c_i\mathbf{g}_i\mathbf{g}_i^T + d\mathbf{x}_*\mathbf{x}_*^T + \sum_{i,j=1}^{r}\Phi_{i,j}\mathbf{v}_i\mathbf{v}_j^T \succeq 0, \\
& (a,b) \in \Lambda,\ a_{i,j} \geq 0,\ b_i \geq 0,\ c_i \geq 0,\ d \geq 0.
\end{aligned}$$

Since any feasible point for (P_M^{III}) can be transformed using (3.4.5) to a feasible point for (P_M^{IV}) without affecting the objective value, we have

$$\mathrm{val}(P_M^{IV}) \leq \mathrm{val}(P_M^{III}). \tag{3.4.6}$$

As a first step in establishing inequality in the other direction (and therefore equality), we introduce the following lemma, which shows how to recover a feasible point for (P_M^{III}) from a feasible point for (P_M^{IV}) provided that the point satisfies a certain condition.

Lemma 3.3. *Suppose that (a,b,c,d,ξ,ψ,Φ) is feasible for (P_M^{IV}) and satisfies*

$$\sum_{j=1}^{i-1} a_{i,j} + b_i = 0 \Rightarrow \xi_{i,k} = \psi_{i,k} = 0,\ \forall k < i. \tag{3.4.7}$$

Then by taking[1]

$$h_{1,k}^{(i)} = \frac{a_{i,k}}{\sum_{j=1}^{i-1} a_{i,j} + b_i}, \quad h_{2,k}^{(i)} = \frac{\xi_{i,k}}{\sum_{j=1}^{i-1} a_{i,j} + b_i}, \quad h_{3,k}^{(i)} = \frac{\psi_{i,k}}{\sum_{j=1}^{i-1} a_{i,j} + b_i},$$

we get that (h,a,b,c,d,Φ) is feasible for (P_M^{III}) and attains the same objective value.

[1] In order to avoid overly numerous special cases, we adopt the convention $\frac{0}{0} = 0$.

Proof. It is enough to verify that the linearization identity (3.4.5) is satisfied for the chosen values of h. First, when $\sum_{j=1}^{i-1} a_{i,j} + b_i = 0$, recalling that we use the convention $\frac{0}{0} = 0$, the identity (3.4.5) follows immediately from the assumption (3.4.7) and since the step sizes are all zeros. Suppose $\sum_{j=1}^{i-1} a_{i,j} + b_i > 0$, then substituting the term $\mathbf{x}_i$ in (3.4.5) by its definition in (3.4.3), we get that for every $i = M+1, \ldots, N$

$$\begin{aligned}
&-\left(\sum_{j=1}^{i-1} a_{i,j}(\mathbf{x}_i - \mathbf{x}_j) + b_i \mathbf{x}_i\right) = \sum_{j=1}^{i-1} a_{i,j}\mathbf{x}_j - \left(\sum_{j=1}^{i-1} a_{i,j} + b_i\right)\mathbf{x}_i \\
&= \sum_{j=1}^{i-1} a_{i,j}\mathbf{x}_j - \left(\sum_{j=1}^{i-1} a_{i,j} + b_i\right)\left(\sum_{k=1}^{i-1} h_{1,k}^{(i)}\mathbf{x}_k - \sum_{k=1}^{r} h_{2,k}^{(i)}\mathbf{v}_k - \sum_{k=M+1}^{i-1} h_{3,k}^{(i)}\mathbf{g}_k\right) \\
&= \sum_{j=1}^{r} \xi_{i,j}\mathbf{v}_j + \sum_{j=M+1}^{i-1} \psi_{i,j}\mathbf{g}_j,
\end{aligned}$$

where the last equality follows from the choice of h. □

In order to establish that the relaxation performed in this step is indeed tight, it is enough to show that condition (3.4.7) holds for an optimal solution of (P_M^{IV}). However, before we can show how to obtain an optimal solution with the required property, we need to perform an additional transformation on the problem, which in turn will also be very useful when deriving the steps of Algorithm KLM in Section 3.5.

Step II: Simplifying the problem (P_M^{IV}). An equivalent and significantly simpler form of problem (P_M^{IV}) can be derived using the matrix completion theorem.

Consider the PSD constraint in (P_M^{IV}) in its explicit form,

$$Q := \begin{pmatrix}
d & \frac{1}{2}\sum_{k=1}^{M} b_k\langle g_k, v_1\rangle & \cdots & \frac{1}{2}\sum_{k=1}^{M} b_k\langle g_k, v_r\rangle & \frac{1}{2}b_{M+1} & \cdots & \frac{1}{2}b_N \\
\frac{1}{2}\sum_{k=1}^{M} b_k\langle g_k, v_1\rangle & \Phi_{1,1} & \cdots & \Phi_{1,r} & \frac{1}{2}\xi_{M+1,1} & \cdots & \frac{1}{2}\xi_{N,1} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\frac{1}{2}\sum_{k=1}^{M} b_k\langle g_k, v_r\rangle & \Phi_{r,1} & \cdots & \Phi_{r,r} & \frac{1}{2}\xi_{M+1,r} & \cdots & \frac{1}{2}\xi_{N,r} \\
\frac{1}{2}b_{M+1} & \frac{1}{2}\xi_{M+1,1} & \cdots & \frac{1}{2}\xi_{M+1,r} & & & \\
\vdots & \vdots & \ddots & \vdots & & R & \\
\frac{1}{2}b_N & \frac{1}{2}\xi_{N,1} & \cdots & \frac{1}{2}\xi_{N,r} & & &
\end{pmatrix} \succeq 0,$$

with

$$R := \begin{pmatrix}
c_{M+1} & \frac{1}{2}\psi_{M+2,M+1} & \cdots & \frac{1}{2}\psi_{N,M+1} \\
\frac{1}{2}\psi_{M+2,M+1} & c_{M+2} & \ddots & \vdots \\
\vdots & \ddots & \ddots & \frac{1}{2}\psi_{N,N-1} \\
\frac{1}{2}\psi_{N,M+1} & \cdots & \frac{1}{2}\psi_{N,N-1} & c_N
\end{pmatrix}.$$

Then by the properties of PSD matrices, $Q \succeq 0$ implies that the principal minors of Q are also

PSD. As a result, we get that the problem

$$
\begin{aligned}
(P_M^V) \quad \min_{a,b,c,d,\Phi} \; & \sum_{i=M+1}^{N}\sum_{j=1}^{M} a_{i,j}\delta_j + \sum_{i=1}^{M} b_i(\langle g_i, x_i - x_0\rangle - \delta_i) + L^2 \sum_{i=M+1}^{N} c_i + R^2 d + \sum_{i=1}^{r} \Phi_{i,i} \\
\text{s.t.} \quad & \begin{pmatrix} d & \frac{1}{2}\sum_{k=1}^{M} b_k\langle g_k, v_i\rangle \\ \frac{1}{2}\sum_{k=1}^{M} b_k\langle g_k, v_i\rangle & \Phi_{i,i} \end{pmatrix} \succeq 0, \quad i = 1,\dots,r, \\
& \begin{pmatrix} d & \frac{1}{2}b_i \\ \frac{1}{2}b_i & c_i \end{pmatrix} \succeq 0, \quad i = M+1,\dots,N, \\
& (a,b) \in \Lambda,\ a_{i,j} \geq 0,\ b_i \geq 0,\ c_i \geq 0,\ d \geq 0,
\end{aligned}
$$

obtained by replacing $Q \succeq 0$ with constraints of the form $Q_{\{1,i\}\times\{1,i\}} \succeq 0$, is a relaxation of (P_M^{IV}), and thus $\mathrm{val}(P_M^V) \leq \mathrm{val}(P_M^{IV})$. As we shall prove below, it turns out that this relaxation is, in fact tight, i.e., $\mathrm{val}(P_M^V) = \mathrm{val}(P_M^{IV})$. To establish this result, we need the following lemma, which is a special case of the matrix completion theorem [50].

Lemma 3.4. *Suppose $q_{1,i} = q_{i,1}$ and $q_{i,i}$ $(i = 1,\dots,n)$ are numbers such that*

$$
\begin{pmatrix} q_{1,1} & q_{1,i} \\ q_{i,1} & q_{i,i} \end{pmatrix} \succeq 0, \quad i = 2,\dots,n.
$$

Then by taking

$$
q_{i,j} = q_{j,i} = \frac{q_{1,i}q_{1,j}}{q_{1,1}}, \tag{3.4.8}
$$

for $i,j = 2,\dots,n$, $i \neq j$, we get that the $n \times n$ matrix $(q_{i,j})$ is positive semidefinite.

Proof. Suppose $q_{1,1} = 0$, then by the properties of PSD matrices, $q_{1,i}$ and $q_{i,1}$ must also be equal to zero. By adopting the convention $\frac{0}{0} = 0$, we get that $q_{i,j} = q_{j,i} = 0$ for $i,j = 2,\dots,n$, hence the matrix $(q_{i,j})$ is diagonal and the result is trivial.

Now assume $q_{1,1} > 0$ and let $\gamma = (q_{1,1},\dots,q_{1,n})^T$, then the claim follows immediately by observing that the matrix $(q_{i,j})$ is the sum of the positive semidefinite rank-one matrix $q_{1,1}^{-1}\gamma\gamma^T$ and the nonnegative diagonal matrix $\mathrm{diag}(0, q_{2,2} - q_{1,2}^2/q_{1,1},\dots,q_{n,n} - q_{1,n}^2/q_{1,1})$. □

The promised tightness of the relaxation performed in this step now follows.

Corollary 3.5. *Suppose $(a^*,b^*,c^*,d^*,\Phi_{i,i}^*)$ is an optimal solution for (P_M^V), then taking*

$$
\begin{aligned}
\Phi_{i,j}^* &= \frac{\sum_{k=1}^{M} b_k^*\langle g_k, v_i\rangle \sum_{k=1}^{M} b_k^*\langle g_k, v_j\rangle}{2d^*}, \quad i,j = 1,\dots,r,\ i \neq j, \\
\xi_{i,j}^* &= \frac{b_i^* \sum_{k=1}^{M} b_k^*\langle g_k, v_j\rangle}{2d^*}, \quad i = M+1,\dots,N,\ j = 1,\dots,r, \\
\psi_{i,j}^* &= \frac{b_i^* b_j^*}{2d^*}, \qquad i = M+1,\dots,N,\ j = M+1,\dots,i-1.
\end{aligned} \tag{3.4.9}
$$

we get that $(a^*,b^*,c^*,d^*,\xi^*,\psi^*,\Phi^*)$ is an optimal solution for (P_M^{IV}). In particular, we have $\mathrm{val}(P_M^{IV}) = \mathrm{val}(P_M^V)$.

Proof. Observing that the minors of Q selected in (P_M^V) have the same form as in the premise of Lemma 3.4 with $n = 1 + r + (N - M)$,

$$\begin{aligned}
&q_{1,1} = d,\\
&q_{1+i,1+i} = \Phi_{i,i}, \quad i = 1,\dots,r,\\
&q_{1+r+i,1+r+i} = c_i, \quad i = M+1,\dots,N,\\
&q_{1,1+i} = q_{1+i,1} = \tfrac{1}{2}\textstyle\sum_{k=1}^{M} b_k\langle g_k, v_1\rangle, \quad i = 1,\dots,r,\\
&q_{1,1+r+i} = q_{1+r+i,1} = \tfrac{1}{2}b_i, \quad i = M+1,\dots,N,
\end{aligned}$$

we get that using the choice (3.4.9), the relations (3.4.8) are satisfied, hence Q is PSD and the first constraint in (P_M^{IV}) is satisfied for $(a^*, b^*, c^*, d^*, \xi^*, \psi^*, \Phi^*)$. Now, examining (P_M^{IV}), we see that the variables $\Phi_{i,j}$ for $i \neq j$, $\xi_{i,j}$ and $\psi_{i,j}$, do not participate in constraints beside the first constraint or in the objective, hence we conclude that $(a^*, b^*, c^*, d^*, \xi^*, \psi^*, \Phi^*)$ is feasible for (P_M^{IV}) and furthermore $\mathrm{val}(P_M^{IV}) \leq \mathrm{val}(P_M^V)$. Since we have already established that $\mathrm{val}(P_M^V) \leq \mathrm{val}(P_M^{IV})$, the proof is complete. □

Another consequence of Lemma 3.4 is the tightness of the relaxation performed in Step I, allowing us to complete our main goal of this section.

Corollary 3.6. *The following equality holds:*

$$\mathrm{val}(P_M^{IV}) = \mathrm{val}(P_M^{III}).$$

Proof. Let $(a^*, b^*, c^*, d^*, \Phi_{i,i}^*)$ be an optimal solution for (P_M^V). Then from Corollary 3.5 we get that by taking ξ^*, ψ^*, and Φ^* as in (3.4.9), the point $(a^*, b^*, c^*, d^*, \xi^*, \psi^*, \Phi^*)$ is optimal for (P_M^{IV}). Observing that from (3.4.9) we get that $b_i^* = 0$ implies $\xi_{i,j}^* = 0$ and $\psi_{i,j}^* = 0$, then it follows that assumption (3.4.7) is satisfied, hence Lemma 3.3 is applicable on $(a^*, b^*, c^*, d^*, \xi^*, \psi^*, \Phi^*)$. As a result, the optimal value of (P_M^{IV}) is attainable by (P_M^{III}), and since we also have $\mathrm{val}(P_M^{IV}) \leq \mathrm{val}(P_M^{III})$ (see (3.4.6)), we conclude that $\mathrm{val}(P_M^{III}) = \mathrm{val}(P_M^{IV})$, proving the desired claim. □

Summary. To summarize the results up to this point, by performing a series of relaxations and transformations on (P_M), which defined the worst-case absolute inaccuracy at x_N, we obtained a sequence of problems (P_M^I)–(P_M^V) that satisfy

$$\mathrm{val}(P_M) \leq \mathrm{val}(P_M^I) \leq \mathrm{val}(P_M^{II}) = \cdots = \mathrm{val}(P_M^V),$$

where the solution of (P_M^V) provides a tractable upper bound. We are now left with our second main goal, namely to derive the steps of algorithm KLM as defined through problem (B_M) in Section 3.2.

3.5 Derivation of Algorithm KLM

At first glance, problem (P_M^V) does not seem to share much resemblance to problem (B_M). We now proceed to show that this convex SDP problem admits a pleasant equivalent convex minimization reformulation over a simplex in $\mathbb{R}^{M+1}$, and that this representation is, in fact, the dual of problem (B_M).

3.5.1 Reducing (P_M^V) to a Convex Minimization Problem Over the Unit Simplex

The form (P_M^V) allows us to derive analytical optimal solutions to some of the optimization variables. First, for any fixed (a,b,d), it is easy to see that the minimization with respect to Φ and c yields the optimal solutions

$$\Phi_{i,i}^* = \frac{(\sum_{k=1}^M b_k \langle g_k, v_i\rangle)^2}{4d}, \quad i = 1,\ldots,r, \tag{3.5.1}$$

$$c_i^* = \frac{b_i^2}{4d}, \quad i = M+1,\ldots,N. \tag{3.5.2}$$

Therefore, recalling that $\{v_1,\ldots,v_r\}$ is an orthonormal set that spans $g_1,\ldots,g_M$, we get

$$\sum_{j=1}^r \Phi_{j,j}^* = \sum_{j=1}^r \frac{(\sum_{i=1}^M b_i\langle g_i, v_j\rangle)^2}{4d} = \frac{\|\sum_{j=1}^r \sum_{i=1}^M b_i\langle g_i, v_j\rangle v_j\|^2}{4d} = \frac{\|\sum_{i=1}^M b_i g_i\|^2}{4d},$$

and (P_M^V) becomes

$$\begin{aligned} \min_{a,b,d} \ & \sum_{i=M+1}^N \sum_{j=1}^M a_{i,j}\delta_j + \sum_{i=1}^M b_i(\langle g_i, x_i - x_0\rangle - \delta_i) + R^2 d + \frac{L^2\sum_{k=M+1}^N b_i^2 + \|\sum_{i=1}^M b_i g_i\|^2}{4d} \\ \text{s.t. } & (a,b) \in \Lambda,\ a_{i,j} \geq 0,\ b_i \geq 0,\ d \geq 0. \end{aligned}$$

Next, observe that for any fixed (a,b) the minimization with respect to d is also immediate and yields

$$d^* = \frac{\sqrt{\|\sum_{i=1}^M b_i g_i\|^2 + L^2\sum_{i=M+1}^N b_i^2}}{2R}. \tag{3.5.3}$$

Plugging this in the last form of the problem, we reach

$$\begin{aligned} \min_{a,b} \ & \sum_{i=M+1}^N \sum_{j=1}^M a_{i,j}\delta_j + \sum_{i=1}^M b_i(\langle g_i, x_i - x_0\rangle - \delta_i) + R\sqrt{\|\sum_{i=1}^M b_i g_i\|^2 + L^2\sum_{i=M+1}^N b_i^2} \\ \text{s.t. } & (a,b) \in \Lambda,\ a_{i,j} \geq 0,\ b_i \geq 0. \end{aligned} \tag{3.5.4}$$

Now, fixing b, the above minimization problem is a linear program in the variable a, which, as shown by the following lemma, can be solved analytically.

Lemma 3.7. *Suppose $b \in \Delta_N$, where Δ_N denotes the N-dimensional unit simplex, i.e., $\Delta_N := \{b \in \mathbb{R}^N : \sum_{i=1}^N b_i = 1, b_i \geq 0\}$. Then,*

$$\min_a \left\{ \sum_{i=M+1}^N \sum_{j=1}^M a_{i,j}\delta_j : (a,b) \in \Lambda,\ a_{i,j} \geq 0 \right\} = \sum_{i=1}^M b_i \delta_m,$$

where an optimal solution is given by

$$a_{i,j}^* = \begin{cases} \sum_{i=1}^M b_i & i = N,\ j = m, \\ b_j, & i = N,\ j \in \{M+1,\ldots,N-1\}, \\ 0, & \textit{otherwise,} \end{cases} \tag{3.5.5}$$

with

$$m \in \operatorname{argmin}_{1\le i\le M}\delta_i. \tag{3.5.6}$$

Proof. Observe that if we fix $a_{i,j}$ for $j > M$, the constraints in Λ have the form

$$\sum_{j=1}^{M} a_{i,j} = \text{constant}, \quad i = M+1,\dots,N,$$

and we get that the problem is separable into $N-M$ minimization problems over a simplex. This implies that the optimal solution can be attained by setting $a^*_{i,j} = 0$ for all $j \in \{1,\dots,M\}\setminus\{m\}$ (i.e., for all indices except for an index for which δ_j is minimal). Using this assignment, the objective now reads

$$\sum_{i=M+1}^{N} a_{i,m}\delta_m,$$

and Λ is reduced to (taking into account all variables):

$$\begin{aligned}
&-a_{i,m} - \sum_{j=M+1}^{i-1} a_{i,j} + \sum_{k=i+1}^{N} a_{k,i} - b_i = 0, \quad i = M+1,\dots,N-1,\\
&1 - a_{N,m} - \sum_{j=M+1}^{N-1} a_{N,j} - b_N = 0,\\
&-1 + \sum_{i=1}^{N} b_i = 0.
\end{aligned}$$

Summing up the constraints in Λ, we get

$$\begin{aligned}
\sum_{i=M+1}^{N} a_{i,m} &= -\sum_{i=M+1}^{N-1}\left(\sum_{j=M+1}^{i-1} a_{i,j} - \sum_{k=i+1}^{N} a_{k,i}\right) - \sum_{j=M+1}^{N-1} a_{N,j} + \sum_{i=1}^{M} b_i\\
&= \sum_{i=M+1}^{N-1}\sum_{k=i+1}^{N} a_{k,i} - \sum_{i=M+1}^{N}\sum_{j=M+1}^{i-1} a_{i,j} + \sum_{i=1}^{M} b_i = \sum_{i=1}^{M} b_i,
\end{aligned}$$

which means that the optimal value for the objective is $\sum_{i=1}^{M} b_i\delta_m$. It is now straightforward to verify that the given solution (3.5.5) is feasible and attains the optimal value of the problem, hence the proof is complete. □

Invoking Lemma 3.7, we can write problem (3.5.4) in the following form:

$$\min_{b\in\Delta_N} \quad \sum_{i=1}^{M} b_i(\langle g_i, x_i - x_0\rangle + \delta_m - \delta_i) + R\sqrt{\|\textstyle\sum_{i=1}^{M} b_i g_i\|^2 + L^2\sum_{i=M+1}^{N} b_i^2}. \tag{3.5.7}$$

To complete this step, note that if b^* is an optimal solution of the last convex problem then optimality conditions imply that we must have $b^*_{M+1} = \cdots = b^*_N$. We can therefore assume,

without affecting the optimal value of the problem, that $b_{M+1} = \cdots = b_N$, hence, by introducing the variable $\beta = \sum_{i=M+1}^{N} b_i$, we get

$$b_{M+1} = \cdots = b_N = \frac{\beta}{N-M}, \tag{3.5.8}$$

and hence

$$\sum_{i=M+1}^{N} b_i^2 = (N-M)b_N^2 = (N-M)\left(\frac{\beta}{N-M}\right)^2 = \frac{\beta^2}{N-M}.$$

Therefore, using this in (3.5.7), we have shown

Proposition 3.5.1. The convex SDP problem (P_M^V) admits the equivalent convex minimization formulation

$$(P_M^{VI}) \qquad \min_{(b_1,\ldots,b_M,\beta)\in\Delta_{M+1}} \quad \sum_{i=1}^{M} b_i(\langle x_i - x_0, g_i\rangle + \delta_m - \delta_i) + R\sqrt{\|\textstyle\sum_{i=1}^{M} b_i g_i\|^2 + \frac{L^2\beta^2}{N-M}},$$

and we have $\mathrm{val}(P_M^V) = \mathrm{val}(P_M^{VI})$.

3.5.2 Completing the Derivation of KLM

We are now ready to complete the main goal of this section, namely the derivation of Algorithm KLM. Indeed, as shown below, it turns out that the convex problem (P_M^{VI}) is nothing else but a dual representation of problem (B_M) defined in Section 3.2. More precisely, we establish that strong duality holds for the pair of convex problems (P_M^{VI})–(B_M). Furthermore, as a by-product, we derive the desired output of the method as described in Section 3.2. To prove this result, we first recall the following elementary fact.

Lemma 3.8. *Let $D \in \mathbb{S}^l_{++}, q \in \mathbb{R}^l$ and $R > 0$ be given. Then,*

$$\max_{u\in\mathbb{R}^l}\{\langle q,u\rangle : u^T Du \le R^2\} = R\|D^{-1/2}q\| \text{ with optimal } u^* = R\frac{D^{-1}q}{\|D^{-1/2}q\|}. \tag{3.5.9}$$

Proof. The claim is an immediate consequence of Cauchy-Schwartz inequality and can also be derived by simple calculus. □

The first main result of this section now follows.

Proposition 3.5.2. The pair of convex problems (P_M^{VI})–(B_M) are dual to each other, and strong duality holds[2], i.e., $\mathrm{val}(P_M^{VI}) = \mathrm{val}(B_M)$. Moreover, given an optimal solution $(b_1^*,\ldots,b_M^*,\beta^*)$ for (P_M^{VI}), an optimal solution (y^*,ζ^*) for (B_M) is recovered via

$$y^* = x_0 - \frac{1}{2d^*}\sum_{j=1}^{M} b_j^* g_j \text{ and } \zeta^* = \frac{L\beta^*}{2(N-M)d^*}, \tag{3.5.10}$$

with

$$d^* = \frac{\sqrt{\|\sum_{i=1}^{M} b_i^* g_i\|^2 + \frac{L^2(\beta^*)^2}{N-M}}}{2R}.$$

[2]Note that since both problems admit a compact feasible set, attainment of both values is warranted.

Proof. Invoking Lemma 3.8 with $u := (y - x_0, \zeta)$ and $q := (-\sum_{i=1}^{M} b_i g_i, L\beta)$, both in $\mathbb{R}^p \times \mathbb{R}$, and with the block diagonal matrix $D := [I_p; (N-M)^{-1}] \in \mathbb{S}^{p+1}_{++}$, it easily follows that problem (P_M^{VI}) reads as the convex-concave minimax problem:

$$V_* := \min_{(b_1,\ldots,b_M,\beta)\in\Delta_{M+1}} \max_{\|y-x_0\|^2+(N-M)\zeta^2\le R^2} \sum_{i=1}^{M} b_i(\langle x_i - y, g_i\rangle + \delta_m - \delta_i) + \beta L\zeta.$$

Applying the minimax theorem [44], we can reverse the min-max operations, and hence by using the simple fact $\min_{\alpha\in\Delta_l} \sum_{i=1}^{l} \alpha_i v_i = \min_{1\le i\le l} v_i$ it follows that

$$V_* = \max_{\|y-x_0\|^2+(N-M)\zeta^2\le R^2} \min\ \{\delta_m - \delta_1 + \langle x_1 - y, g_1\rangle, \ldots, \delta_m - \delta_M + \langle x_M - y, g_M\rangle, L\zeta\},$$

which is an obvious equivalent reformulation of the problem (B_M), defined in Section 3.2. This establishes the strong duality claim $\mathrm{val}(P_M^{VI}) = \mathrm{val}(B_M)$. Furthermore, if $(b^*, \beta^*) \in \Delta_{M+1}$ is optimal for (P_M^{VI}), again thanks to Lemma 3.8, (with (q, u, D) as defined above), one immediately recovers an optimal solution (y^*, ζ^*) of (B_M) as given in (3.5.10) and the proof is completed. □

As we now show, Proposition 3.5.2 paves the way to determine the iterative steps of Algorithm KLM. For that purpose, we first derive an expression for $x_{M+1}, \ldots, x_N$ in terms an optimal solution $(b_1^*, \ldots, b_M^*, \beta^*)$ for (P_M^{VI}). First, recall that $(a^*, b^*, c^*, d^*, \xi^*, \psi^*, \Phi^*)$ with a^*, b^*, c^*, $\Phi^*_{i,i}$, d^*, ξ^*, ψ^*, and Φ^* defined according to (3.5.5), (3.5.8), (3.5.2), (3.5.1), and (3.4.9), is optimal for (P_M^{IV}) and satisfies the assumption (3.4.7). Thus, as a result of Lemma 3.3 and the definition of the sequence x_i in (3.3.2), the corresponding sequence $x_{M+1}, \ldots, x_N$ can be found via the rule

$$x_i = x_0 + \frac{1}{\sum_{j=1}^{i-1} a^*_{i,j} + b^*_i} \left(\sum_{j=1}^{i-1} a^*_{i,j}(x_j - x_0) - \sum_{j=1}^{r} \xi^*_{i,j} v_j - \sum_{j=M+1}^{i-1} \psi^*_{i,j} g_j \right). \tag{3.5.11}$$

From definitions of ξ^* and ψ^* in (3.4.9) we get that

$$\sum_{j=1}^{r} \xi^*_{i,j} v_j = \frac{b^*_i}{2d^*} \sum_{j=1}^{r} \sum_{k=1}^{M} b^*_k \langle g_k, v_j\rangle v_j = \frac{b^*_i}{2d^*} \sum_{k=1}^{M} b^*_k g_k,$$

and

$$\sum_{j=1}^{r} \xi^*_{i,j} v_k + \sum_{j=M+1}^{i-1} \psi^*_{i,j} g_j = \frac{b^*_i}{2d^*} \sum_{j=1}^{i-1} b^*_j g_j,$$

which, together with (3.5.11), yields an expression for x_i that is independent of $\xi^*_{i,j}$ and $\psi^*_{i,j}$:

$$x_i = \frac{1}{\sum_{j=1}^{i-1} a^*_{i,j} + b^*_i} \left(\sum_{j=1}^{i-1} a^*_{i,j} x_j + b^*_i \left(x_0 - \frac{1}{2d^*} \sum_{j=1}^{i-1} b^*_j g_j \right) \right), \quad i = M+1, \ldots, N. \tag{3.5.12}$$

Now, using the definition of a^* from (3.5.5), we reach the expression

$$x_i = \begin{cases} x_0 - \dfrac{1}{2d^*} \displaystyle\sum_{j=1}^{i-1} b^*_j g_j, & i = M+1, \ldots, N-1, \\ \displaystyle\sum_{j=1}^{M} b^*_j x_m + \sum_{j=M+1}^{N-1} b^*_j x_j + b^*_N \left(x_0 - \frac{1}{2d^*} \sum_{j=1}^{N-1} b^*_j g_j \right), & i = N, \end{cases}$$

where m as in (3.5.6).

This rule can be written in a more convenient form using a solution to the pair of convex problems (P_M^{VI})–(B_M). For that, note that by writing x_i in terms of x_{i-1}, breaking the computation of the last step, x_N into two parts x_N and $\bar{x}_N$, and applying (3.5.10) of Proposition 3.5.2, we obtain

$$x_i = \begin{cases} x_0 - \dfrac{1}{2d^*}\sum_{j=1}^{M} b_j^* g_j = y^*, & i = M+1, \\ x_{i-1} - \dfrac{\beta^*}{2(N-M)d^*} g_{i-1} = x_{i-1} - \dfrac{\zeta^*}{L} g_{i-1}, & i = M+2,\ldots,N, \end{cases} \tag{3.5.13}$$

$$\bar{x}_N = (1-\beta^*)x_m + \frac{\beta^*}{N-M}\sum_{j=M+1}^{N} x_j,$$

which is precisely the output of Algorithm KLM after performing a "standard" step followed by $N-M-1$ "easy" steps.

3.6 The Rate of Convergence: Proof of Theorem 3.1

Before we proceed with the proof of Theorem 3.1, we need the following lemma, which establishes that the optimal value of (P_M^{II}) is non-increasing during the run of the method.

Lemma 3.9. *Let $l \in \mathbb{N}$ be such that $M+l \leq N$ and suppose $x_{M+1},\ldots,x_{M+l}$ satisfy the recursion* (3.3.2) *with $h = \bar{h}$, where $\bar{h}$ is optimal for the outer minimization problem in (P_M^{II}). Then* $\mathrm{val}(P_{M+l}^{II}) \leq \mathrm{val}(P_M^{II})$.

Proof. Denote by $\hat{h}$ the steps sizes in $\bar{h}$ which correspond to the last $N-M-l$ steps performed by the method, $x_{M+l+1},\ldots,x_N$, (i.e., $\hat{h}_{j,k}^{(i)} = \bar{h}_{j,k}^{(i)}$ for $i = M+l+1,\ldots,N$), and let $(\hat{X},\hat{\delta})$ be optimal for the inner maximization problem in (P_{M+l}^{II}) when fixing $h = \hat{h}$. We proceed by constructing a matrix $\bar{X}$ and a vector $\bar{\delta}$ such that $(\bar{h};\bar{X},\bar{\delta})$ is feasible to (P_M^{II}) and achieves the same objective value as $(\hat{h};\hat{X},\hat{\delta})$ achieves for (P_{M+l}^{II}).

Denote by $\bar{\mathbf{v}}_i$, $\bar{\mathbf{g}}_i$ and $\bar{\mathbf{x}}_i$ the vectors $\mathbf{v}_i$, $\mathbf{g}_i$ and $\mathbf{x}_i$ as defined for (P_M^{II}) in (3.4.3), and let $\hat{\mathbf{v}}_i$, $\hat{\mathbf{g}}_i$ and $\hat{\mathbf{x}}_i$ be the vectors $\mathbf{v}_i$, $\mathbf{g}_i$ and $\mathbf{x}_i$ that correspond to (P_{M+l}^{II}), i.e.,

$$\bar{\mathbf{v}}_i = e_{1+i}, \quad i = 1,\ldots,r,$$

$$\bar{\mathbf{g}}_i = \begin{cases} \sum_{k=1}^{r}\langle g_i, v_k\rangle \mathbf{v}_k, & i = 1,\ldots,M, \\ e_{1+r+i-M}^T, & i = M+1,\ldots,N, \end{cases}$$

$$\bar{\mathbf{x}}_i = \begin{cases} \sum_{k=1}^{r}\langle x_i - x_0, v_k\rangle \mathbf{v}_k, & i = 1,\ldots,M, \\ \sum_{k=1}^{i-1} h_{1,k}^{(i)}\mathbf{x}_k - \sum_{k=1}^{r} h_{2,k}^{(i)}\mathbf{v}_k - \sum_{k=M+1}^{i-1} h_{3,k}^{(i)}\mathbf{g}_k, & i = M+1,\ldots,N, \\ e_1, & i = *, \end{cases}$$

and

$$\hat{\mathbf{v}}_i = e_{i+1}, \quad i = 1,\dots,r,$$
$$\hat{\mathbf{g}}_i = \begin{cases} \sum_{k=1}^{r} \langle g_i, v_k\rangle \hat{\mathbf{v}}_k, & i = 1,\dots,M+l, \\ e^T_{1+r+i-M}, & i = M+l+1,\dots,N, \end{cases}$$
$$\hat{\mathbf{x}}_i = \begin{cases} \sum_{k=1}^{r} \langle x_i - x_0, v_k\rangle \hat{\mathbf{v}}_k, & i = 1,\dots,M+l, \\ \sum_{k=1}^{i-1} h^{(i)}_{1,k} \hat{\mathbf{x}}_k - \sum_{k=1}^{r} h^{(i)}_{2,k} \hat{\mathbf{v}}_k - \sum_{k=M+1}^{i-1} h^{(i)}_{3,k} \hat{\mathbf{g}}_k, & i = M+l+1,\dots,N, \\ e_1 & i = *. \end{cases}$$

Now, by taking V as the $(1+r+N-M-l)\times(1+r+N-M)$ matrix

$$V = (\hat{\mathbf{x}}_*, \hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_r, \hat{\mathbf{g}}_{M+1}, \dots, \hat{\mathbf{g}}_N),$$

it follows from the construction above that

$$\hat{\mathbf{v}}_i = V\bar{\mathbf{v}}_i, \quad i = 1,\dots,r,$$
$$\hat{\mathbf{g}}_i = V\bar{\mathbf{g}}_i, \quad i = 1,\dots,N,$$
$$\hat{\mathbf{x}}_i = V\bar{\mathbf{x}}_i, \quad i = 1,\dots,N,*.$$

Hence, by setting

$$\bar{X} = V^T \hat{X} V,$$
$$\bar{\delta}_i = \begin{cases} f(x_i), & i = M+1,\dots,M+l, \\ \hat{\delta}_i, & i = M+l+1,\dots,N,*, \end{cases}$$

we get that the equalities

$$\bar{\mathbf{g}}_i^T \bar{X} \bar{\mathbf{g}}_j = \hat{\mathbf{g}}_i^T \hat{X} \hat{\mathbf{g}}_j, \quad i,j = 1,\dots,N,$$
$$\bar{\mathbf{g}}_i^T \bar{X} \bar{\mathbf{x}}_j = \hat{\mathbf{g}}_i^T \hat{X} \hat{\mathbf{x}}_j, \quad i = 1,\dots,N,\ j = 1,\dots,N,*.$$

are satisfied, and therefore $(\bar{h};\bar{X},\bar{\delta})$ satisfies all the constraints in (P^{II}_M) that also appear in (P^{II}_{M+l}). Note, however, that (P^{II}_M) includes some additional constraints that do not appear in (P^{II}_{M+l}), namely

$$\bar{\delta}_i - \bar{\delta}_j \le \bar{\mathbf{g}}_i^T X (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j),$$

for $i = M+1,\dots,M+l-1$, and $j = 1,\dots,i-1$, and

$$\bar{\mathbf{g}}_i^T X \bar{\mathbf{g}}_i \le L^2,$$

for $i = M+1,\dots,M+l-1$. Nevertheless, since for $i,j \le M+l$ the values of $\bar{\delta}_i$, $\bar{\mathbf{g}}_i^T \bar{X} \bar{\mathbf{g}}_j$ and $\bar{\mathbf{g}}_i^T \bar{X} \bar{\mathbf{x}}_j$ originate from the convex function f, i.e.,

$$\bar{\mathbf{g}}_i^T \bar{X} \bar{\mathbf{g}}_j = \hat{\mathbf{g}}_i^T \hat{X} \hat{\mathbf{g}}_j = \langle f'(x_i), f'(x_j)\rangle, \quad i,j = 1,\dots,M+l,$$
$$\bar{\mathbf{g}}_i^T \bar{X} \bar{\mathbf{x}}_j = \hat{\mathbf{g}}_i^T \hat{X} \hat{\mathbf{x}}_j = \langle f'(x_i), x_j\rangle, \quad i = 1,\dots,M+l,\ j = 1,\dots,M+l,$$

we immediately get from the subgradient inequality and the Lipschitz-continuity of f that these additional constraints hold. We conclude that $(\bar{h};\bar{X},\bar{\delta})$ is feasible for (P_M^{II}) and attains the same objective value as does $(\hat{h};\hat{X},\hat{\delta})$ for (P_{M+l}^{II}).

For a feasible point $(h;X,\delta)$, denote by $P_M^{II}(h;X,\delta)$ the value of the objective in (P_M^{II}) at the given point, then we have just shown that $P_{M+l}^{II}(\hat{h};\hat{X},\hat{\delta}) = P_M^{II}(\bar{h};\bar{X},\bar{\delta})$. As an immediate consequence, we get

$$\mathrm{val}(P_{M+l}^{II}) \leq P_{M+l}^{II}(\hat{h};\hat{X},\hat{\delta}) = P_M^{II}(\bar{h};\bar{X},\bar{\delta}) \leq \mathrm{val}(P_M^{II}),$$

where the first inequality follow since $(\hat{X},\hat{\delta})$ is optimal for the inner maximization problem in (P_{M+l}^{II}) and the last inequality follows since $\bar{h}$ is optimal for the outer minimization problem in (P_M^{II}). □

We are now ready to give the proof of Theorem 3.1.

Proof. (Theorem 3.1.) First, we need to establish that the initialization step corresponds to the solution of (B_0). Indeed, observing that $y^* = x_0$, it is straightforward to verify that $\mathrm{val}(B_0) = LR/\sqrt{N}$ is attained for $\zeta^* = R/\sqrt{N}$, and that β^*, the dual variable that corresponds to the constraint $f(x_m) - L\zeta \leq t$, is equal to one.

Recalling that s is the index of the last step where a "standard" step was taken, then by the definition of the "easy" steps, the sequence $x_{s+1},\ldots,x_N,\bar{x}_N$ satisfies (3.5.13), where y^*, ζ^* and β^* are given by a solution of (B_s). Let $\bar{h}$ be the vector of step sizes in (3.3.2) that matches $x_{s+1},\ldots,x_{N-1},\bar{x}_N$, then by the construction of (B_s) from (P_s^{II}), we get that $\bar{h}$ is optimal for (P_M^{II}), i.e., $\mathrm{val}(P_s^{II}) = P_s^{II}(\bar{h})$ (we use $P_s^{II}(\bar{h})$ to denote the optimal value of the inner maximization problem in (P_s^{II}) with h set to $\bar{h}$). We therefore have

$$f(\bar{x}_N) - f^* \leq P_s(\bar{h}) \leq P_s^{II}(\bar{h}) = \mathrm{val}(P_s^{II}) \leq \mathrm{val}(P_0^{II}),$$

where the first two inequalities follow from the construction of (P_s^{II}) and last inequality follow from Lemma 3.9 by a simple inductive argument.

Finally, since we have already established during the construction and analysis of Section 3.4 that the series of relaxations and transformations preserve the optimal value of the problem, we have $\mathrm{val}(P_M^{II}) = \cdots = \mathrm{val}(P_M^{VI}) = \mathrm{val}(B_M)$ for every M, and the claim immediately follows. □

3.7 Concluding remarks

Through a constructive approach, we have derived a new method for non-smooth convex minimization, which is surprisingly similar to the Kelley method, yet it attains the optimal rate of convergence. We conclude by outlining a refined version of the method, and by briefly discussing how the construction derived in this work can be extended onto some other situations as well, which often arise in nonsmooth optimization schemes/models.

A memory-limited version of Algorithm KLM. The current form of the method requires storing all of the past iterates, which can translate to a significant amount of memory for large values of N. This requirement can be eliminated, as in the aggregation technique described in [55], by observing that Lemma 3.9 makes no assumptions on the way the steps $x_1, \ldots, x_M$ are generated, hence Theorem 3.1 still holds if, at any iteration M where a "standard" step is taken, the trial point set is replaced by another set of points in a way that maintains the solution of (B_M). In fact, by a well-known result from convex optimization, if $b_1^*, \ldots, b_M^*$ are the optimal dual variables corresponding to the constraints

$$f(x_i) + \langle y - x_i, f'(x_i) \rangle \leq t, \quad i = 1, \ldots, M,$$

by replacing these constraint with the conical combination

$$\sum_{i=1}^{M} b_i^* \left(f(x_i) + \langle y - x_i, f'(x_i) \rangle \right) \leq \sum_{i=1}^{M} b_i^* t,$$

we reach a problem that has the same optimal solution as the original problem. Hence, the trial points set can be aggregated to one scalar, $\sum_{i=1}^{M} b_i^* (f(x_i) - \langle x_i, f'(x_i) \rangle)$, and one vector, $\sum_{i=1}^{M} b_i^* f'(x_i)$, without affecting the efficiency estimate of the method. The same technique can be applied to any subset of the trial points, hence the cardinality of the trial points set can be maintained at any desired level.

Knowledge of Lower Bound on f^*. When a lower bound, $\underline{f}$, on f^* is known, (e.g., though a dual bound), the constraint $\underline{f} \leq \varphi^*$ can be added to (P_M) and the analysis can continue with only little change. The resulting method turns out to be nearly the same as the method described above, where the only change is the introduction of the constraint $\underline{f} \leq t$ to (B_M). Furthermore, the resulting efficiency estimate remains unchanged.

Extension with Inexact Subgradients. Another situation is the case where, instead of an exact subgradient, an ε-subgradient $f'(x) \in \partial_\varepsilon f(x)$ is available for some given $\varepsilon \geq 0$, i.e., for any y, instead of the usual subgradient inequality, we have

$$f(x) - f(y) \leq \langle f'(x), x - y \rangle + \varepsilon.$$

The use of ε-subgradients instead of exact subgradients has some practical advantages, see e.g., [10, 38] and references therein for motivating examples and for some recent work in this setting. As in the previous case, only minor changes are needed in the analysis we developed, and the resulting method turns out to be identical to the method presented in Section 3.2, except for the first set of constraint in (B_M), which becomes

$$f(x_i) + \langle y - x_i, f'(x_i) \rangle - \varepsilon \leq t, \quad i = 1, \ldots, M,$$

and for the efficiency estimate of the method (3.2.1), which turns out to be

$$f(\bar{x}_N) - f^* \leq \mathrm{val}(B_s) + \varepsilon \leq LR/\sqrt{N} + \varepsilon.$$

3.8 Appendix: A Tight Lower-Complexity Bound

In this appendix, we refine the proof in [74, Section 3.2] to obtain a new lower-complexity bound on the class of nonsmooth, convex, and Lipschitz-continuous functions, which together with the results discussed above form a *tight* complexity result for this class of problems. More precisely, under the setting of §3.2.1, we show that for any first-order method, the worst-case absolute inaccuracy after N steps cannot be better than $\frac{LR}{\sqrt{N}}$, which is exactly the bound attained by Algorithm KLM.

In order to simplify the presentation, and following [74, Section 3.2], we restrict our attention to first-order methods that generate sequences that satisfy the following assumption:

Assumption A. The sequence $\{x_i\}$ satisfies

$$x_i \in x_1 + \operatorname{span}\{f'(x_1), \ldots, f'(x_{i-1})\},$$

where $f'(x_i) \in \partial f(x_i)$ is obtained by evaluating a first-order oracle at x_i.

As noted by Nesterov [74, Page 59], this assumption is not necessary and can be avoided by some additional reasoning.

The lower-complexity result is stated as follows.

Theorem 3.10. *For any $L, R > 0$, $N, p \in \mathbb{N}$ with $N \leq p$, and any starting point $x_1 \in \mathbb{R}^p$, there exists a convex and Lipschitz-continuous function $f : \mathbb{R}^p \to \mathbb{R}$ with Lipschitz constant L and $\|x_f^* - x_1\| \leq R$, and a first-order oracle $\mathcal{O}(x) = (f(x), f'(x))$, such that*

$$f(x_N) - f^* \geq \frac{LR}{\sqrt{N}}$$

for all sequences $x_1, \ldots, x_N$ that satisfies Assumption A.

Proof. The proof proceeds by constructing a "worst-case" function, on which any first-order method that satisfies Assumption A will not be able to improve its initial objective value during the first N iterations.

Let $f_N : \mathbb{R}^p \to \mathbb{R}$ and $\bar{f}_N : \mathbb{R}^p \to \mathbb{R}$ be defined by

$$\begin{aligned} f_N(x) &= \max_{1 \leq i \leq N} \langle x, e_i \rangle, \\ \bar{f}_N(x) &= L \max(f_N(x), \|x\| - R(1 + N^{-1/2})), \end{aligned}$$

then it is easy to verify that $\bar{f}_N$ is Lipschitz-continuous with constant L and that

$$\bar{f}_N^* = -\frac{LR}{\sqrt{N}}$$

is attained for $x^* \in \mathbb{R}^p$ such that

$$x^* = -\frac{R}{\sqrt{N}} \sum_{i=1}^{N} e_i.$$

We equip $\bar{f}_N$ with the oracle $\mathscr{O}_N(x) = (\bar{f}_N(x), \bar{f}'_N(x))$ by choosing $\bar{f}'_N(x) \in \partial \bar{f}_N(x)$ according to:

$$\bar{f}'_N(x) = \begin{cases} Lf'_N(x), & f_N(x) \geq \|x\| - R(1+N^{-1/2}), \\ L\frac{x}{\|x\|}, & f_N(x) < \|x\| - R(1+N^{-1/2}), \end{cases} \tag{3.8.1}$$

where

$$f'_N(x) = e_{i^*}, \quad i^* = \min\{i : f_N(x) = \langle x, e_i \rangle\}. \tag{3.8.2}$$

We also denote

$$\mathbb{R}^{i,p} := \{x \in \mathbb{R}^d : \langle x, e_j \rangle = 0,\ i+1 \leq j \leq p\}.$$

Now, let $x_1, \ldots, x_N$ be a sequence that satisfies Assumption A with $f = \bar{f}_N$ and the oracle $\mathscr{O}_N$, where without loss of generality we assume $x_1 = 0$. Then $\bar{f}'_N(x_1) = e_1$ and we get $x_2 \in \operatorname{span}\{\bar{f}'_N(x_1)\} = \mathbb{R}^{1,p}$. Now, from $\langle x_2, e_2 \rangle = \cdots = \langle x_2, e_N \rangle = 0$, we get that $\min\{i : f_N(x) = \langle x, e_i \rangle\} \leq 2$ and it follows by (3.8.1) and (3.8.2) that $f'_N(x_2) \in \mathbb{R}^{2,p}$ and $\bar{f}'_N(x_2) \in \mathbb{R}^{2,p}$. Hence, we conclude from Assumption A that $x_3 \in \operatorname{span}\{\bar{f}'_N(x_1), \bar{f}'_N(x_2)\} \subseteq \mathbb{R}^{2,p}$. It is straightforward to continue this argument to show that $x_i \in \mathbb{R}^{i-1,p}$ and $\bar{f}'_N(x_i) \in \mathbb{R}^{i,p}$ for $i = 1, \ldots, N$, thus $x_N \in \mathbb{R}^{N-1,p}$. Finally, since for every $x \in \mathbb{R}^{N-1,p}$ we have $\bar{f}_N(x) \geq \langle x, e_N \rangle = 0$, we immediately get

$$\bar{f}_N(x_N) - \bar{f}^*_N \geq \frac{LR}{\sqrt{N}},$$

which completes the proof. □

Chapter 4

An $O(1/\varepsilon)$ Algorithm for a Class of Nonsmooth Convex-Concave Saddle-Point Problems

We introduce a novel algorithm for solving a class of structured nonsmooth convex-concave saddle-point problems involving a smooth function and the sum of finitely many bilinear terms and nonsmooth functions. The proposed method is simple. It uses only one gradient of the smooth term, one proximal map of each nonsmooth part, and matrix-vector multiplication per iteration. We prove that the proposed algorithm globally converges to a saddle-point with an $O(1/\varepsilon)$ efficiency estimate. We illustrate its relevance for tackling a broad class of composite minimization problems and its performance through numerical examples for the image deblurring problem and for the fused lasso logistic regression problem.

4.1 Introduction

In this chapter, we consider a class of nonsmooth structured convex-concave saddle-point (SP) problems. By structured we mean that the model consists of a saddle-point function that is a sum of a smooth function (i.e., with Lipschitz continuous gradient), with a finite collection of nonsmooth functions and bilinear terms. The precise definition of the model appears in Section 4.2. This model is very rich and encompasses most convex optimization models arising in a wide array of applications in signal/image processing and machine learning, see for instance the two very recent edited volumes [80, 92].

The past and current research activities in the search of methods for solving the alluded class of convex-concave SP problems and their relatives composite nonsmooth minimization problems, have been intensive over the past five decades and has been recently revived due to their relevance in many applications. As a result, the body of literature is rather very large, and clearly this chapter does not intend to review all these developments. For some of earlier representative works see, e.g., [3, 6, 7, 59, 81, 64, 46, 97, 41, 35, 98] and for more recent studies see, e.g., [12, 70, 28, 84] and references therein. The main focus of these works has been on the sequential convergence analysis of algorithms for various types of problems as

well as their extensions through abstract frameworks, see for instance the very recent work [84] which introduces a generalized forward-backward algorithm and also provides a very good synthesis of many methods (old and more recent), including an up-to-date comprehensive list of references. Here, we focus on methods with provable *nonasymptotic* efficiency estimates.

The main goal of this chapter is to present a simple and novel algorithm for the class SP and their relatives composite nonsmooth minimization problems that achieves the nonasymptotic efficiency estimate $O(1/\varepsilon)$, where $\varepsilon > 0$ is the desired accuracy. To the best of our knowledge, this is the best known rate that can be achieved for this class of composite nonsmooth SP with first order methods (FOM) capable of efficiently solving large scale applied problems. The motivation emerges from the current trends of research efforts on FOM which rely on special structures and data information to devise simple schemes that are faster than the more general approaches which rely on classical nonsmooth optimization algorithms, e.g., subgradient methods which are often very slow, sharing an $O(1/\varepsilon^2)$ efficiency estimate [72, 18].

To put our contribution in perspective, let us briefly review some state-of-the-art methods which have led to these improvements in the efficiency estimates for FOM. For the simplest convex composite minimization problem which consists of minimizing the sum of a smooth function with a nonsmooth one, Nesterov [78] and Beck-Teboulle [19] have proven that it is possible to devise schemes with the improved efficiency estimate $O(1/\sqrt{\varepsilon})$, namely like the so-called "optimal gradient method" [73]. Both methods assumes that the proximal map of the nonsmooth function is "easy" to compute. However, when the nonsmooth function is composed with a linear map, the resulting proximal map is generally not an easy task any more. Moreover, for problems involving a finite sum of such terms, which is one of the problem of interest in this chapter, the situation obviously becomes much harder and often leads to intractable problems.

We now briefly describe three main approaches that can overcome this difficulty as well as their limitations and which have motivated the present study. It is well known (see, e.g., [43]) that convex-concave SP problems can be reformulated as special case of the more general variational inequality problem. Korpelevitch [59] proposed the so-called extra-gradient method which can solve the monotone Lipshcitz continuous variational inequality problem, and hence the general but *smooth* convex-concave SP problem. Extra-gradient based algorithms have been recently shown to exhibit an $O(1/\varepsilon)$ efficiency estimate by Nemirovsky [71] and independently by Auslender-Teboulle [11]. However, these extra-gradient type methods cannot in general be applied for nonsmooth SP problems without adequate reformulations or further assumptions, see for instance the recent work [52]. Moreover, they double the amount of computation of projections, due to the needed extra-projected gradient step, and as a result this can often severely affect their performance, see Section 4.5 for more details.

Another approach is to exploit the "max-structure" inherently present in the SP formulation. This was proposed by Nesterov [76], who developed a smoothing method combined with a specific fast gradient scheme to derive a method with an $O(1/\varepsilon)$ efficiency estimate. This method requires knowledge of the smoothing parameter (which depends on the desired accuracy) and on compactness assumption. For a unified framework analysis on smoothing, FOM and their extension, see the recent work [21].

The third approach relies on primal-dual methods. Recently, Chambolle and Pock presented in [33] a primal-dual method that can solve the nonsmooth SP which emerges from the classical

convex minimization model, i.e., the sum of two nonsmooth convex functions, one composed with a linear map. The method was proven to achieve an $O(1/\varepsilon)$ efficiency estimate and has been shown to be successful in solving a wide variety of problems in image sciences. However, an $O(1/\varepsilon)$ efficiency estimate is not known for the method [33] if applied to problems involving the sum of finitely many composite nonsmooth terms. Moreover, the model and algorithm [33] does not distinguish (and hence does not exploit) the possible presence of a smooth term. As a consequence, when a smooth function is present, it requires computing the proximal map of *both* the nonsmooth and the smooth function, and the later can often be a major computational issue, see Section 4.5 for further discussion and details.

Motivated by the above recent developments, we present a novel and simple algorithm to tackle the class of SP problems which is proven to globally converge to a saddle-point with efficiency estimate $O(1/\varepsilon)$, where $\varepsilon > 0$ is the desired accuracy. By simple we mean an algorithm which at each iteration utilizes one gradient and one proximal map operation on the given nonsmooth function, assumed to be easy to compute or/and can be efficiently computed. Moreover, the remaining operations consist only of multiplying a matrix by a vector, that is, no matrix inversion is involved and furthermore we do not rely on nested optimization schemes. To achieve these goals we blend in a peculiar fashion some fundamental and old ideas such as duality, predictor-corrector steps and proximal methods which are reminiscent to [35] and its extension in [98], see Section 4.2 for details and the proposed algorithm. In Section 4.3 we derive the promised global nonasymptotic efficiency estimate, and as an easy by-product the sequential convergence is also obtained. Our approach, which exploits the interplay between optimization problems and their saddle-point representation, allows to efficiently address the important class of structured convex models involving the sum of smooth function with a finite sum of nonsmooth functions composed with linear maps in the objective or in the constraints. In particular, the much more difficult and computationally demanding task which often required to compute the proximal map of the (sum of) composition of the given nonsmooth functions with linear maps is avoided, yet our method allows to preserve the $O(1/\varepsilon)$ efficiency estimate within minimal computational effort, see Section 4.4. To demonstrate the relevance and performance of the proposed algorithm when compared to some recent state-of-the-art schemes sharing the same iteration complexity, numerical illustration on the constrained total-variation image deblurring problem and the fused lasso logistic regression problem are presented in Section 4.5. Lastly, we conclude with two appendices: the first appendix discusses a rate of convergence result for the primal representation of a problem that is solved via its saddle-point representation. The second appendix presents a formulation of the proposed method in the PEP framework introduced in Chapter 2.

Notation. The set of symmetric $p \times p$ positive (semi)-definite matrices is denoted by $\mathbb{S}^p_{++}$ ($\mathbb{S}^p_+$). We also use $M \succ 0$ ($M \succeq 0$). For any vector $z \in \mathbb{R}^p$ and any $M \in \mathbb{S}^p_+$, we define the semi-norm induced by M, as follows $\|z\|_M := \langle z, Mz\rangle^{1/2}$. When $M \equiv I_p$, the $p \times p$ identity matrix, the standard Euclidean norm is recovered, and will simply be denoted by $\|z\|$. Also, recall that for any real matrix $W \in \mathbb{R}^{n\times p}$ and any given vector norm $\|\cdot\|'$, the induced norm of W is defined by $\|W\|' = \max\left\{\|Wz\|' : \ \|z\|' = 1\right\}$. For $h : \mathbb{R}^p \to (-\infty, \infty]$ which is proper, lower-semi-continuous (lsc) and convex, its conjugate is defined by $h^*(y) := \sup_{x\in\mathbb{R}^p}\{\langle x, y\rangle - h(x)\}$.

Other standard convex analysis notations not explicitly defined here can be found in any text, e.g., [86].

4.2 The Saddle-Point Model and The Algorithm

We begin by describing the setting of the nonsmooth convex-concave structured saddle- point problem of interest.

4.2.1 The Saddle-Point Problem

We consider convex-concave saddle-point problems of the form

$$\text{(M)} \qquad \min_{u\in\mathbb{R}^n}\max_{v\in\mathbb{R}^d}\left\{K(u,v) := f(u) + \langle u, \mathscr{A}v\rangle - g(v)\right\},$$

where f and g are convex functions and $\mathscr{A}$ is a linear map such that

(i) $f:\mathbb{R}^n\to\mathbb{R}$ is a convex function which is continuously differentiable and its gradient ∇f is Lipschitz continuous with constant L_f, i.e., for all $u_1,u_2\in\mathbb{R}^n$, we have

$$\|\nabla f(u_1) - \nabla f(u_2)\| \le L_f\|u_1-u_2\|.$$

(ii) $g_i:\mathbb{R}^{d_i}\to(-\infty,+\infty]$, $i=1,2,\ldots,m$, is a proper, lower semicontinuous (lsc) and convex function (possibly nonsmooth). With $v_i\in\mathbb{R}^{d_i}$, we define $v := (v_1,v_2,\ldots,v_m)\in\mathbb{R}^d$ where $d=\sum_{i=1}^m d_i$ and we let $g:\mathbb{R}^d\to(-\infty,+\infty]$ be the proper, lsc and convex function defined by

$$g(v) := \sum_{i=1}^m g_i(v).$$

(iii) $A_i:\mathbb{R}^{d_i}\to\mathbb{R}^n$, $i=1,2,\ldots,m$, is a linear map and we let $\mathscr{A}:\mathbb{R}^d\to\mathbb{R}^n$ be the linear map defined by $\mathscr{A}v=\sum_{i=1}^m A_iv_i$.

Note that the model (M) can easily include constraints on the variable v, thanks to the fact that g is extended valued. On the other hand, our model's formulation does not include constraint on the variable u. We will later show on Section 4.4 how constraints on u can be adequately handled.

The choice of our model (M) is not accidental. As we shall see, the saddle-point approach will offer much flexibility in tackling various composite optimization models arising in many important applications within the proposed algorithm (see Sections 4.4 and 4.5).

4.2.2 The Standing Assumption

Throughout this chapter, our standing assumption is that the convex-concave function $K(\cdot,\cdot)$ has a saddle-point, i.e., there exists $(u^*,v^*)\in\mathbb{R}^n\times\mathbb{R}^d$ such that

$$K(u^*,v)\le K(u^*,v^*)\le K(u,v^*), \quad \forall\, u\in\mathbb{R}^n,\ v\in\mathbb{R}^d.$$

The existence of a saddle point corresponds to zero duality gap for the induced optimization problems

$$(P) \qquad \inf_{u\in\mathbb{R}^n}\left[r(u)=\sup_{v\in\mathbb{R}^d}K(u,v)\right] \qquad \text{and} \qquad (D) \qquad \sup_{v\in\mathbb{R}^d}\left[q(v)=\inf_{u\in\mathbb{R}^n}K(u,v)\right].$$

One always has $\inf_{u\in\mathbb{R}^n} r(u) \geq \sup_{v\in\mathbb{R}^d} q(v)$ (i.e., weak duality). In addition, (u^*,v^*) is a saddle-point of K if and only if u^* is an optimal solution of the primal problem (P), v^* is an optimal solution of the dual problem (D), and

$$\inf_{u\in\mathbb{R}^n}\sup_{v\in\mathbb{R}^d}K(u,v)=\sup_{v\in\mathbb{R}^d}\inf_{u\in\mathbb{R}^n}K(u,v)=K(u^*,v^*),$$

where $K(u^*,v^*)$ is the saddle-point value. For ease of reference, we denote by S_P and S_D the optimal solutions sets of the primal-dual pair (P)-(D), respectively. For standard qualification conditions (as well as more conditions) which warrant this equality, i.e., the existence of saddle-points for convex-concave K, we refer the reader to the monographs [9, Chapter 5] and [89, Chapter 11].

4.2.3 The Algorithm

Before we state our algorithm we need to recall the definition of the Moreau proximal map [68] and to introduce some convenient notations.

Let $h:\mathbb{R}^p\to(-\infty,\infty]$ be a proper, lsc and convex function. For any $x\in\mathbb{R}^p$ and $M\in\mathbb{S}^p_{++}$, the proximal map associated with h is defined by:

$$\text{prox}^h_M(x) := \text{argmin}_y\left\{h(y)+\frac{1}{2}\|y-x\|^2_M\right\}. \tag{4.2.1}$$

Clearly, the proximal mapping is well (uniquely) defined for any $x\in\mathbb{R}^p$ and any $M\in\mathbb{S}^p_{++}$. When $M=\mu^{-1}I_p$, $\mu>0$, where I_p stands for the $p\times p$ identity matrix, we simply use the following notation $\text{prox}^h_\mu(\cdot)$. We also recall the fundamental Moreau proximal identity [68] which states that the proximal mapping of a function can be easily computed from the proximal mapping of its conjugate (and vice-versa), that is, for any $z\in\mathbb{R}^p$

$$\text{prox}^h_M(z)+M\text{prox}^{h^*}_{M^{-1}}\left(M^{-1}z\right)=z, \tag{4.2.2}$$

where M^{-1} is the inverse of the symmetric positive definite matrix M.

For any given real numbers $\sigma_1,\sigma_2,\ldots,\sigma_m>0$, we denote $S_i:=\sigma_i^{-1}I_{d_i}$, $i=1,2,\ldots,m$, where I_{d_i} stands for the $d_i\times d_i$ identity matrix, and define the block diagonal matrix $S:=\text{Diag}[S_1,S_2,\ldots,S_m]\in\mathbb{S}^d_{++}$ with $d=\sum_{i=1}^m d_i$.

The algorithm we propose consists of a predictor-corrector gradient step for handling the smooth part of K and a proximal step for handling the nonsmooth part. The idea of using predictor-corrector steps goes back to the work of [35] in the context of augmented Lagrangian methods, and was further extended in [98] to handle general monotone inclusions, and in particular convex-concave saddle point problems like (M). This will be further discussed below. The main steps of the algorithm now follow.

PAPC: Proximal Alternating Predictor Corrector
Initialization. $\left(u^0, v^0\right) \in \mathbb{R}^n \times \mathbb{R}^d$ and let $\tau > 0, S \succ 0$.
General Step. For $k = 1, 2, \ldots$, compute

$$p^k = u^{k-1} - \tau\left(\mathscr{A} v^{k-1} + \nabla f\left(u^{k-1}\right)\right), \tag{4.2.3}$$

$$v^k = \operatorname{prox}_S^g\left(v^{k-1} + S^{-1}\mathscr{A}^T p^k\right), \tag{4.2.4}$$

$$u^k = u^{k-1} - \tau\left(\mathscr{A} v^k + \nabla f\left(u^{k-1}\right)\right). \tag{4.2.5}$$

The choice of the parameters τ and S will be made precise in Section 4.3.

A few remarks regarding the computational steps involved in the PAPC method are now in order.

- A major computational effort of the method is given in the second step (4.2.4). Since here $g(v) = \sum_{i=1}^m g_i(v_i)$, using the definition of the matrix S we immediately obtain that at any given point $x_i \in \mathbb{R}^{d_i}$, $i = 1, 2, \ldots, m$,

  $$\operatorname{prox}_S^g(x) = \left(\operatorname{prox}_{\sigma_1}^{g_1}(x_1), \operatorname{prox}_{\sigma_2}^{g_2}(x_2), \ldots, \operatorname{prox}_{\sigma_m}^{g_m}(x_m)\right),$$

  and hence the second step of the algorithm (4.2.4) decomposes accordingly and for all $i = 1, 2, \ldots, m$ we have

  $$v_i^k = \operatorname{prox}_{\sigma_i}^{g_i}\left(v_i^{k-1} + \sigma_i A_i^T p^k\right) = \operatorname{argmin}_{v_i \in \mathbb{R}^{d_i}}\left\{g_i(v_i) + \frac{1}{2\sigma_i}\left\|v_i - \left(v_i^{k-1} + \sigma_i A_i^T p^k\right)\right\|^2\right\}.$$

  Thus, the algorithm PAPC achieves full decomposition for the given structure of K in the sense that for each i, it avoids the much more difficult task of computing the proximal map of the composite function $g_i \circ A_i$, and only requires computing the proximal map of $g_i(\cdot)$, $i = 1, 2, \ldots, m$.

- The algorithm uses only *one* evaluation of the gradient of the smooth function f, and a careful implementation requires only one application of the operator $\mathscr{A}$ and one application of the operator $\mathscr{A}^T$ per iteration. Thus, for large scale problems this potentially amounts to a considerable reduction of computation time compared to the straightforward implementation.

Before proceeding with the analysis and convergence properties of the algorithm, we end this section with some remarks discussing the underlying nature of the PAPC method and its relation to well-known methods.

Despite some striking similarities between PAPC and the scheme proposed in [98, Example 4, p. 963], the algorithm PAPC is different. Indeed, a simple computation shows that applying the algorithm in [98] on model (M) would require to compute *two* gradients of f (one at p^k and one at u^{k-1}) per each iteration (as opposed to one in PAPC), and it uses the same step-size

for all iterations, as opposed to PAPC which used (τ, S). Moreover and most importantly, the iteration complexity of the method [98] appears to be unknown.

In fact, it turns out that we can give an interesting dual interpretation of PAPC via the well-known proximal gradient (ProxGrad) (e.g., [20]). For simplicity of exposition, it is enough to consider the saddle-point model with $m = 1$, that is, $A := \mathscr{A} \equiv A_1$, $d_1 \equiv d$ and $g_1 \equiv g$. More precisely we thus consider,

$$\text{(M1)} \quad \min_{u\in\mathbb{R}^n} \max_{v\in\mathbb{R}^d} \{f(u) + \langle u, Av\rangle - g(v)\} = \min_{u\in\mathbb{R}^n} \left\{f(u) + g^*\left(A^T u\right)\right\},$$

and its dual (conveniently rewritten as a minimization problem with appropriate change of sign)

$$\text{(DM1)} \quad \min_{v\in\mathbb{R}^d} \{f^*(-Av) + g(v)\} = \min_{v\in\mathbb{R}^d, \eta\in\mathbb{R}^n} \{f^*(\eta) + g(v) : \ \eta + Av = 0\}.$$

Since f is assumed with L_f-Lipschitz gradient, it means that by [89, Proposition 12.60, page 565], its conjugate f^* is L_f^{-1}-strongly convex. It is then well-known that applying the ProxGrad on (M1) is equivalent to applying the so-called *alternating minimization* (AM) algorithm [46, 97] on (DM1). An easy computation show that this reduces to the following steps: for $k = 1, 2, \ldots$, compute

$$\begin{aligned} v^k &= \operatorname{argmin}_{v\in\mathbb{R}^d} \left\{ g(v) + \frac{\tau}{2} \left\| Av + \nabla f\left(u^{k-1}\right) - \tau^{-1} u^{k-1} \right\|^2 \right\} \equiv \operatorname{argmin}_{v\in\mathbb{R}^d} \{g(v) + P_k(v)\} \\ u^k &= u^{k-1} - \tau\left(Av^k + \nabla f\left(u^{k-1}\right)\right). \end{aligned} \tag{4.2.6}$$

The main difficulty is in the step to compute v^k, which in general will be a computationally a too demanding task[1] due to the least-squares term $P_k(\cdot)$. This step consists of minimizing the sum of nonsmooth function, $g(\cdot)$, with a smooth one P_k (here with Lipschitz constant $L_{P_k} = \tau \left\|A^T A\right\|$). In the spirit of [35], the alluded difficulty can thus be avoided by solving it *approximately*. More precisely, let us apply "one shot" (iteration) of the ProxGrad scheme. That is, given some $\sigma > 0$, with $\sigma\tau \left\|A^T A\right\| \leq 1$, we replace the first step by its approximate version,

$$v^k = \operatorname{argmin}_{v\in\mathbb{R}^d} \left\{ g(v) + \frac{\sigma}{2} \left\| v - \left(v^{k-1} - \sigma^{-1} \nabla_v P_k\left(v^{k-1}\right)\right) \right\|^2 \right\}, \tag{4.2.7}$$

where

$$\nabla_v P_k\left(v^{k-1}\right) = \tau A^T \left(Av^{k-1} + \nabla f\left(u^{k-1}\right) - \tau^{-1} u^{k-1}\right) \equiv -A^T p^k$$

with

$$p^k := u^{k-1} - \tau\left(Av^{k-1} + \nabla f\left(u^{k-1}\right)\right). \tag{4.2.8}$$

It can be immediately seen that the resulting *Approximate Alternating Minimization* (APxAM) scheme just derived and defined via (4.2.6), (4.2.7) and (4.2.8) reduces exactly to the proposed PAPC method.

[1] In the particular case $A^T A = I$, we simply obtain that $v^k = \operatorname{prox}^g_{\tau^{-1}}\left(A(u^{k-1} - \tau\nabla f(u^{k-1}))\right)$.

Remark 4.2.1. As a by-product of the above development, the forthcoming sublinear rate of convergence result for PAPC (see Theorem 4.5) also proves that the same rate of convergence is shared by this *approximate* version of the so-called alternating minimization algorithm, a result which to the best of our knowledge appears to be new. The connection between the 4 discussed algorithms can be conveniently summarized as follows

ProxGrad on (M1)	$\Longleftrightarrow$	AM on (DM1)
PAPC on (M1)	$\Longleftrightarrow$	APxAM on (DM1)

Thus, in the particular case when $\mathscr{A} \equiv I$, the four algorithms coincide and reduce to the ProxGrad method when applied to the standard composite model minimizing the sum of a smooth and nonsmooth function.

Remark 4.2.2. A this juncture, we also note that very recently, a fast version of the alternating minimization was derived in [22]. It was shown there that the dual objective function sequence converges at the rate of $O(1/k^2)$ while the rate of convergence of the primal sequence is of the order of $O(1/k)$. Unfortunately, the results of [22] are not applicable to our model (M1).

4.3 Main Convergence Results for PAPC

In this section, we establish the main convergence properties of the PAPC algorithm. In particular, we prove its global rate of convergence, showing that it shares the claimed $O(1/\varepsilon)$ efficiency estimate. As an easy by-product we also derive a global convergence of the sequence generated by PAPC to a saddle-point of $K(\cdot,\cdot)$. We start with few preliminaries.

4.3.1 Elementary Preliminaries

Let $h : \mathbb{R}^p \to \mathbb{R}$ be a continuously differentiable function whose gradient ∇h is assumed to be L_h-Lipschitz continuous. Then, we have the well-known property (usually referred as the Descent Lemma for smooth functions, see for instance [26]):

$$h(u) \leq h(v) + \langle u - v, \nabla h(v) \rangle + \frac{L_h}{2} \|u - v\|^2, \quad \forall\, u, v \in \mathbb{R}^p. \tag{4.3.1}$$

For convex functions, we can then deduce the following useful inequality.

Lemma 4.1. *Let $h : \mathbb{R}^p \to \mathbb{R}$ be a convex and continuously differentiable function such that its gradient is Lipschitz continuous with constant L_h. Then, for any three points $x, y, z \in \mathbb{R}^p$, we have*

$$h(x) \leq h(y) + \langle \nabla h(z), x - y \rangle + \frac{L_h}{2} \|x - z\|^2.$$

Proof. Since h is convex and differentiable, the gradient inequality holds, i.e.,

$$0 \leq h(y) - h(z) - \langle \nabla h(z), y - z \rangle, \quad \forall\, y, z \in \mathbb{R}^p,$$

and from (4.3.1) we obtain

$$h(x) \leq h(z) + \langle \nabla h(z), x - z \rangle + \frac{L_h}{2} \|x - z\|^2.$$

Adding these two inequalities yields the desired result. □

The next lemma is the well-known proximal inequality (slightly extended with respect to a given matrix $M \in \mathbb{S}^p_+$) that will be systematically used in the forthcoming analysis.

Lemma 4.2. *Let $h : \mathbb{R}^p \to (-\infty, \infty]$ be a proper, lsc and convex function. Given $M \in \mathbb{S}^p_+$ and $x \in \mathbb{R}^p$, let*

$$z \in \operatorname{argmin}_{\xi \in \mathbb{R}^p} \left\{ h(\xi) + \frac{1}{2} \|\xi - x\|_M^2 \right\}.$$

Then, for all $\xi \in \mathbb{R}^p$, we have

$$h(z) - h(\xi) \leq \langle \xi - z, M(z-x) \rangle .$$

Proof. The optimality condition characterizing z yields $\gamma + M(z-x) = 0$ with $\gamma \in \partial h(z)$. Invoking the subgradient inequality for the convex function h we get that for any $z, \xi \in \mathbb{R}^p$,

$$h(z) - h(\xi) \leq \langle \xi - z, -\gamma \rangle = \langle \xi - z, M(z-x) \rangle ,$$

which completes the proof. □

Finally, we recall the Pythagoras identity that will be useful in the analysis. For any matrix $M \in \mathbb{S}^p_+$, we have

$$2 \langle w - v, M(u-v) \rangle = \|w-v\|_M^2 - \|w-u\|_M^2 + \|u-v\|_M^2 , \qquad \forall\, u, v, w \in \mathbb{R}^p. \tag{4.3.2}$$

4.3.2 Global Rate of Convergence of the PAPC Method

To establish the iteration complexity of the PAPC algorithm and the convergence of the generated sequence $\left\{ \left(u^k, v^k\right) \right\}_{k \in \mathbb{N}}$, we consider the following quantity

$$\Gamma_k(u,v) = K\left(u^k, v\right) - K\left(u, v^k\right), \quad \forall\, u \in \mathbb{R}^n,\ v \in \mathbb{R}^d.$$

Our main task is to find an upper-bound for $\Gamma_k(u,v)$, $k \in \mathbb{N}$. Indeed, $\Gamma_k(u,v) \leq 0$ for all $u \in \mathbb{R}^n$ and all $v \in \mathbb{R}^d$ implies

$$\begin{aligned} K\left(u^k, v^k\right) &\leq K\left(u, v^k\right), \quad \forall\, u \in \mathbb{R}^n \\ K\left(u^k, v\right) &\leq K\left(u^k, v^k\right), \quad \forall\, v \in \mathbb{R}^d, \end{aligned}$$

namely, that (u^k, v^k) is a saddle-point of K with saddle-point value $K(u^k, v^k)$. We now proceed to prove two key inequalities which will be the basis for proving our main convergence results.

Lemma 4.3. *Let $\{(p^k, v^k, u^k)\}_{k \in \mathbb{N}}$ be the sequence generated by the PAPC algorithm, then for every $k \in \mathbb{N}$ and every $u \in \mathbb{R}^n$, we have*

$$K\left(u^k, v^k\right) - K\left(u, v^k\right) \leq \frac{1}{2\tau} \left(\left\| u - u^{k-1} \right\|^2 - \left\| u - u^k \right\|^2 \right) - \frac{1}{2} \left(\frac{1}{\tau} - L_f \right) \left\| u^k - u^{k-1} \right\|^2 .$$

Proof. Applying Lemma 4.1 on the convex and differentiable function $h(u) := K\left(u, v^k\right)$ with $x := u^k$, $y := u$ and $z := u^{k-1}$, yields

$$K\left(u^k, v^k\right) - K\left(u, v^k\right) \leq \left\langle \nabla_u K\left(u^{k-1}, v^k\right), u^k - u \right\rangle + \frac{L_f}{2}\left\|u^k - u^{k-1}\right\|^2.$$

Using the fact that $\nabla_u K\left(u^{k-1}, v^k\right) = \mathscr{A} v^k + \nabla f\left(u^{k-1}\right) = \tau^{-1}\left(u^{k-1} - u^k\right)$, where the last equation follows from the definition of step (4.2.5), we get

$$K\left(u^k, v^k\right) - K\left(u, v^k\right) \leq \frac{1}{\tau}\left\langle u^{k-1} - u^k, u^k - u \right\rangle + \frac{L_f}{2}\left\|u^k - u^{k-1}\right\|^2.$$

The desired result follows by using the identity (4.3.2) with $M \equiv I_n$, for the first term in the right hand side of the above inequality. □

Lemma 4.4. *Let $\{(p^k, u^k, v^k)\}_{k\in\mathbb{N}}$ be the sequence generated by the PAPC algorithm and assume that the matrix $G := S - \tau\mathscr{A}^T\mathscr{A}$ is positive semidefinite. Then, for every $k \in \mathbb{N}$ and every $v \in \mathbb{R}^d$, we have*

$$K\left(u^k, v\right) - K\left(u^k, v^k\right) \leq \frac{1}{2}\left(\left\|v - v^{k-1}\right\|_G^2 - \left\|v - v^k\right\|_G^2 - \left\|v^k - v^{k-1}\right\|_G^2\right).$$

Proof. First, note that by the definition of $K(\cdot,\cdot)$ we have

$$-K\left(p^k, v\right) = g(v) - \left\langle \mathscr{A}^T p^k, v \right\rangle - f\left(p^k\right),$$

and hence step (4.2.4) of PAPC can be written (after omitting constant terms) as

$$v^k = \operatorname{prox}_S^{-K(p^k,\cdot)}\left(v^{k-1}\right) = \operatorname{argmin}_{v\in\mathbb{R}^d}\left\{-K\left(p^k, v\right) + \frac{1}{2}\left\|v - v^{k-1}\right\|_S^2\right\}.$$

Applying Lemma 4.2 to the convex function $h(v) := -K(p^k, v)$ with $\xi := v$, $z := v^k$ and $x := v^{k-1}$, yields

$$K\left(p^k, v\right) - K\left(p^k, v^k\right) \leq \left\langle v^k - v^{k-1}, S\left(v - v^k\right)\right\rangle. \tag{4.3.3}$$

Now, from the definition of $K(\cdot,\cdot)$, simple algebra shows that the following identity holds

$$K\left(u^k, v\right) - K\left(u^k, v^k\right) + K\left(p^k, v^k\right) - K\left(p^k, v\right) = \left\langle u^k - p^k, \mathscr{A}\left(v - v^k\right)\right\rangle. \tag{4.3.4}$$

Using the definitions of p^k and u^k given in steps (4.2.3) and (4.2.5), respectively, we have

$$u^k - p^k = \tau\mathscr{A}\left(v^{k-1} - v^k\right),$$

hence, together with (4.3.3) and (4.3.4), we obtain

$$\begin{aligned} K\left(u^k, v\right) - K\left(u^k, v^k\right) &= \left\langle u^k - p^k, \mathscr{A}\left(v - v^k\right)\right\rangle + K\left(p^k, v\right) - K\left(p^k, v^k\right) \\ &\leq \tau\left\langle v^{k-1} - v^k, \mathscr{A}^T\mathscr{A}\left(v - v^k\right)\right\rangle + \left\langle v^k - v^{k-1}, S\left(v - v^k\right)\right\rangle \\ &= \left\langle v^k - v^{k-1}, \left(S - \tau\mathscr{A}^T\mathscr{A}\right)\left(v - v^k\right)\right\rangle. \end{aligned}$$

Thus, with $G := S - \tau\mathscr{A}^T\mathscr{A}$, which assumed to be positive semidefinite, the desired result follows by using the identity (4.3.2) with $M \equiv G$. □

Before we proceed with the convergence results, we need some additional notations. For any sequence $\{x^k\}_{k\in\mathbb{N}}$ and any integer $N \geq 1$, we denote by

$$\bar{x}^N := \frac{1}{N}\sum_{k=1}^{N} x^k,$$

the average (ergodic) sequence associated with $\{x^k\}_{k\in\mathbb{N}}$. For the parameters $\sigma_1, \sigma_2, \ldots, \sigma_m > 0$ used in PAPC, we denote $\sigma := \max_{1\leq i\leq m} \sigma_i$.

The global rate of convergence result for the ergodic sequence now follows.

Theorem 4.5. *Let $\{(p^k, u^k, v^k)\}_{k\in\mathbb{N}}$ be the sequence generated by the PAPC algorithm with $\tau L_f \leq 1$ and $\sigma\tau\sum_{i=1}^m \|A_i\|^2 \leq 1$. Then, $G = S - \tau\mathscr{A}^T\mathscr{A}$ is positive semidefinite and for every $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^d$, we have*

$$K\left(\bar{u}^N, v\right) - K\left(u, \bar{v}^N\right) \leq \frac{\tau^{-1}\left\|u - u^0\right\|^2 + \left\|v - v^0\right\|_G^2}{2N}.$$

Proof. We begin by showing that the condition $\sigma\tau\sum_{i=1}^m \|A_i\|^2 \leq 1$ implies that the symmetric $d \times d$ matrix G is positive semi-definite. First, note that $G = S - \tau\mathscr{A}^T\mathscr{A} \succeq 0$ if $\lambda_{\min}(S) \geq \tau\lambda_{\max}\left(\mathscr{A}^T\mathscr{A}\right)$, where $\lambda_{\min}(\cdot)$ ($\lambda_{\max}(\cdot)$) stands for the minimal (maximal) eigenvalue of the given symmetric matrix. Recalling the definition of S given in Section 4.2.3, we obtain

$$\lambda_{\min}(S) = \min_{1\leq i\leq m} \sigma_i^{-1} = \frac{1}{\max_{1\leq i\leq m}\sigma_i} = \frac{1}{\sigma},$$

and hence the last condition reduces to $\sigma\tau\lambda_{\max}\left(\mathscr{A}^T\mathscr{A}\right) \leq 1$. On the other hand, using the definition of $\mathscr{A}$ we have

$$\lambda_{\max}\left(\mathscr{A}^T\mathscr{A}\right) = \left\|\mathscr{A}^T\mathscr{A}\right\| = \left\|\sum_{i=1}^m A_i^T A_i\right\| \leq \sum_{i=1}^m \left\|A_i^T A_i\right\| = \sum_{i=1}^m \|A_i\|^2,$$

and the first part of the claim follows. Now, let $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^d$. Since we assume that $\tau L_f \leq 1$, using Lemma 4.3 we get that for all $k \in \mathbb{N}$

$$\begin{aligned} K\left(u^k, v^k\right) - K\left(u, v^k\right) &\leq \frac{1}{2\tau}\left(\left\|u - u^{k-1}\right\|^2 - \left\|u - u^k\right\|^2\right) - \frac{1}{2}\left(\frac{1}{\tau} - L_f\right)\left\|u^k - u^{k-1}\right\|^2 \\ &\leq \frac{1}{2\tau}\left(\left\|u - u^{k-1}\right\|^2 - \left\|u - u^k\right\|^2\right). \end{aligned} \tag{4.3.5}$$

On the other hand, with $G \succeq 0$, from Lemma 4.4 we immediately obtain that for all $k \in \mathbb{N}$

$$K\left(u^k, v\right) - K\left(u^k, v^k\right) \leq \frac{1}{2}\left(\left\|v - v^{k-1}\right\|_G^2 - \left\|v - v^k\right\|_G^2\right).$$

Adding this inequality to (4.3.5) we get for every $k \in \mathbb{N}$, and for all $u \in \mathbb{R}^n$, $v \in \mathbb{R}^d$

$$K\left(u^k, v\right) - K\left(u, v^k\right) \leq \frac{1}{2\tau}\left(\left\|u - u^{k-1}\right\|^2 - \left\|u - u^k\right\|^2\right) + \frac{1}{2}\left(\left\|v - v^{k-1}\right\|_G^2 - \left\|v - v^k\right\|_G^2\right).$$

Now, since $K(u,v)$ is convex-concave, using the definition of $\left(\bar{u}^N,\bar{v}^N\right)$, by the Jensen inequality and using the last inequality we get

$$\begin{aligned}
K\left(\bar{u}^N,v\right)-K\left(u,\bar{v}^N\right) &= K\left(\frac{1}{N}\sum_{k=1}^{N}u^k,v\right)-K\left(u,\frac{1}{N}\sum_{k=1}^{N}v^k\right)\\
&\leq \frac{1}{N}\sum_{k=1}^{N}\left(K\left(u^k,v\right)-K\left(u,v^k\right)\right)\\
&\leq \frac{1}{N}\sum_{k=1}^{N}\frac{1}{2\tau}\left(\left\|u-u^{k-1}\right\|^2-\left\|u-u^k\right\|^2\right)\\
&\quad+\frac{1}{N}\sum_{k=1}^{N}\frac{1}{2}\left(\left\|v-v^{k-1}\right\|_G^2-\left\|v-v^k\right\|_G^2\right)\\
&= \frac{1}{2\tau N}\left(\left\|u-u^0\right\|^2-\left\|u-u^N\right\|^2\right)+\frac{1}{2N}\left(\left\|v-v^0\right\|_G^2-\left\|v-v^N\right\|_G^2\right)\\
&\leq \frac{1}{2\tau N}\left\|u-u^0\right\|^2+\frac{1}{2N}\left\|v-v^0\right\|_G^2,
\end{aligned}$$

which proves the claimed result. □

Two important consequences of this global upper-bound established in Theorem 4.5 can be deduced. The first states that the PAPC method possesses a sub-linear global rate of convergence. More precisely, let $\varepsilon>0$, then following Nemirovsky and Yudin [72], a point $(u_\varepsilon,v_\varepsilon)$ is called an ε-saddle-point for K if

$$\sup\left\{K(u_\varepsilon,v)-K(u,v_\varepsilon)\ :\ u\in S_P,v\in S_D\right\}\leq\varepsilon,$$

where S_P is the optimal solutions set of the primal problem and S_D is the optimal solutions set of the dual problem associated to the saddle-point function K (see Section 4.2). Using this definition, we thus immediately obtain from Theorem 4.5 the following efficiency estimate result.

Corollary 4.6. *Let $\left\{\left(p^k,u^k,v^k\right)\right\}_{k\in\mathbb{N}}$ be the sequence generated by the PAPC algorithm with $\tau L_f\leq 1$ and $\sigma\tau\sum_{i=1}^m\|A_i\|^2\leq 1$. Assume that both optimal solutions sets S_P and S_D associated to the saddle-point problem (M) are compact[2]. Then, given a desired accuracy $\varepsilon>0$, the PAPC method produces an ε-saddle-point $(\bar{u}^N,\bar{v}^N)$ of K in $N=O(1/\varepsilon)$ iterations.*

Another easy consequence of Theorem 4.5 is a convergence result of the sequence generated by PAPC to a saddle-point of problem (M).

Corollary 4.7. *Let $\{(p^k,u^k,v^k)\}_{k\in\mathbb{N}}$ be the sequence generated by the PAPC algorithm with $\tau L_f<1$ and $\sigma\tau\sum_{i=1}^m\|A_i\|^2<1$.Then, the sequence $\{(u^k,v^k)\}_{k\in\mathbb{N}}$ converges to a saddle-point $(\widetilde{u},\widetilde{v})$ of K.*

[2]Note that under standard qualification conditions, which is our standing assumption (cf. Section 4.2.2), the optimal set S_D of the dual problem associated to (M) is always compact.

Proof. From Lemmas 4.3 and 4.4 it immediately follows that for any $u \in \mathbb{R}^n$, $v \in \mathbb{R}^d$ and for all $k \in \mathbb{N}$

$$K\left(u^k, v\right) - K\left(u, v^k\right) \leq \frac{1}{2\tau}\left(\left\|u^{k-1} - u\right\|^2 - \left\|u^k - u\right\|^2\right) - \frac{1}{2}\left(\frac{1}{\tau} - L_f\right)\left\|u^k - u^{k-1}\right\|^2 + \frac{1}{2}\left(\left\|v^{k-1} - v\right\|_G^2 - \left\|v^k - v\right\|_G^2 - \left\|v^k - v^{k-1}\right\|_G^2\right). \tag{4.3.6}$$

In particular, let $(u^*, v^*) \in \mathbb{R}^n \times \mathbb{R}^d$ be an arbitrary saddle-point of the function K, then by the saddle-point property, $K(u^k, v^*) - K(u^*, v^k) \geq 0$, and it follows from the last inequality that

$$\left(\frac{1}{\tau} - L_f\right)\left\|u^k - u^{k-1}\right\|^2 + \left\|v^k - v^{k-1}\right\|_G^2 \leq D\left(w^{k-1}, w^*\right) - D\left(w^k, w^*\right), \tag{4.3.7}$$

where $w := (u, v) \in \mathbb{R}^n \times \mathbb{R}^d$ and we define

$$D(w_1, w_2) := \frac{1}{\tau}\|u_1 - u_2\|^2 + \|v_1 - v_2\|_G^2.$$

As a consequence of (4.3.7), the sequence $\{D(w^k, w^*)\}_{k\in\mathbb{N}}$ is non-increasing and therefore the sequence $\{w^k\}_{k\in\mathbb{N}}$ is bounded. On the other hand, summing (4.3.7) for any $k = 1, 2, \ldots, N$ yields

$$D\left(w^N, w^*\right) + \sum_{k=1}^{N}\left[\left(\frac{1}{\tau} - L_f\right)\left\|u^k - u^{k-1}\right\|^2 + \left\|v^k - v^{k-1}\right\|_G^2\right] \leq D\left(w^0, w^*\right),$$

and hence with $G \succ 0$ and $L_f\tau < 1$, we obtain

$$\lim_{k\to\infty}\left\|u^k - u^{k-1}\right\| = 0 \quad \text{and} \quad \lim_{k\to\infty}\left\|v^k - v^{k-1}\right\|_G = 0. \tag{4.3.8}$$

Since the sequence $\{w^k\}_{k\in\mathbb{N}}$ is bounded, it has at least one limit point. Suppose that $\widetilde{w} = (\widetilde{u}, \widetilde{v})$ is a limit point of the sequence $\{w^k\}_{k\in\mathbb{N}}$, then taking the limit in (4.3.6) over the appropriate subsequences and using (4.3.8) yields

$$K(\widetilde{u}, v) - K(u, \widetilde{v}) \leq 0, \quad \forall\, u \in \mathbb{R}^n,\ v \in \mathbb{R}^d,$$

which proves that $(\widetilde{u}, \widetilde{v})$ is a saddle-point of K. To complete the proof it only remains to show that $\{w^k\}_{k\in\mathbb{N}}$ has a unique limit point. This follows by a standard argument, see e.g., [87, page 885]. □

The saddle-point model (M) covers a very broad class of generic convex optimization problems arising in many applications. Below we describe some typical important prototype problems.

4.4 Composite Minimization via Saddle-Point

Our main purpose in this section is, on one hand, to illustrate the flexibility of the model (M), and on the other hand, the simplicity of the resulting PAPC algorithm when applied to these problems. To do so, we first recall the following fundamental result [86] which states that the *bi-conjugate* of a proper, lsc and convex function $h : \mathbb{R}^p \to (-\infty, \infty]$ coincides with itself, i.e., $h^{**} = h$. Thus, any proper, lsc and convex function h admits the following variational max-representation

$$h(x) = \max_{u \in \mathbb{R}^p} \{\langle u, x\rangle - h^*(u)\}.$$

This well-known and fundamental relation is in fact the key player for handling constraints as well as for deriving "full splitting" of most optimization problems involving composition with linear maps through their saddle-point representation in the form (M). This mechanism is described in the next part, and then we illustrate it on various class of optimization models.

4.4.1 The Dual Transportation Trick

Let $U \subset \mathbb{R}^n$ be a closed and convex set, consider the following constrained convex problem

$$(C) \qquad \min_{u \in \mathbb{R}^p} \{F(u) : u \in U\}.$$

Let δ_U denotes the usual *convex indicator function* of the set U (i.e., 0 if $u \in U$ and ∞ otherwise). Recall that the conjugate of δ_U is the so-called *support function* of the set U, denoted by $\sigma_U : \mathbb{R}^p \to (-\infty, \infty]$ and given for any $x \in \mathbb{R}^p$ by

$$\sigma_U(x) := \sup_{u \in \mathbb{R}^p} \{\langle u, x\rangle : u \in U\} = \delta_U^*(x). \tag{4.4.1}$$

The support function $\sigma_U(\cdot)$ is always convex (with U convex or not). Moreover, σ_U is proper, lsc and convex when U is a closed and convex set, thus in this case $\sigma_U^* = \delta_U$. Equipped with these basic objects, problem (C) can be written

$$\min_{u \in \mathbb{R}^p} \{F(u) + \delta_U(u)\} = \min_{u \in \mathbb{R}^p} \max_{v \in \mathbb{R}^p} \{F(u) + \langle u, v\rangle - \delta_U^*(v)\} \tag{4.4.2}$$

$$= \min_{u \in \mathbb{R}^p} \max_{v_1, v_2 \in \mathbb{R}^p} \{\langle u, v_1\rangle - F^*(v_1) - \langle u, v_2\rangle - \delta_U^*(v_2)\}. \tag{4.4.3}$$

Clearly, both saddle-point representations given in (4.4.2) and (4.4.3) can be seen as particular cases of model (M), yet they illustrate two important different goals: (4.4.2) provides a way to reinterpret a constrained optimization problem as an unconstrained saddle-point problem, while (4.4.3) provides a way to continue and further decompose the problem to fit our model (M) in the case where F is also nonsmooth. For ease of reference we call this the *dual transportation trick*. It allows to transport the primal constraint variable $u \in U$ into the objective function, but with an additional nonsmooth convex function $\sigma_U = \delta_U^*$ in the dual space of the saddle-point function.

This elementary transportation trick plays a key role in our way to treat primal constraints in saddle-point problems as well as quite general convex composite optimization problems as described below.

4.4.2 Handling Constrained Saddle-Point Problems

Let $U \subseteq \mathbb{R}^n$ be a closed and convex set, consider the following constrained saddle-point problem (here for simplicity of exposition it is enough to look at $m = 1$, which means that, $d \equiv d_1$, $\mathscr{A} = A_1 \equiv A$, $g_1(v) \equiv g(v)$ and $\sigma_1 = \sigma$):

$$\text{(CM)} \quad \min_{u \in U} \max_{v \in \mathbb{R}^d} \left\{ K(u,v) = f(u) + \langle u, Av \rangle - g(v) \right\}.$$

Since the PAPC method requires the problem to be unconstrained (cf. problem (M)), the method cannot be directly applied here. However, using the dual transportation trick just described above, we obtain the following equivalent unconstrained saddle-point problem which is compatible with the requirements of the PAPC method

$$\text{(CM}') \quad \min_{u \in \mathbb{R}^n} \max_{v \in \mathbb{R}^d, w \in \mathbb{R}^n} \left\{ K'(u;v,w) := f(u) + \langle u, Av \rangle - g(v) + \langle u, w \rangle - \sigma_U(w) \right\}.$$

Observing that the inner maximization problem in (CM$'$) is *separable* in the variables v and w, the PAPC method for solving the constrained saddle-point problem (CM) can be formulated as follows.

PAPC: constrained version
Initialization. $\left(u^0, v^0, w^0\right) \in \mathbb{R}^n \times \mathbb{R}^d \times \mathbb{R}^n$ and $\tau, \sigma > 0$.
General Step ($k = 1, 2, \ldots$)

$$p^k = u^{k-1} - \tau \left(A v^{k-1} + w^{k-1} + \nabla f\left(u^{k-1}\right) \right), \tag{4.4.4}$$

$$v^k = \operatorname{prox}_{\sigma}^{g}\left(v^{k-1} + \sigma A^T p^k \right), \tag{4.4.5}$$

$$w^k = \operatorname{prox}_{\sigma}^{\sigma_U}\left(w^{k-1} + \sigma p^k \right), \tag{4.4.6}$$

$$u^k = u^{k-1} - \tau \left(A v^k + w^k + \nabla f\left(u^{k-1}\right) \right). \tag{4.4.7}$$

Thanks to the Moreau proximal identity, the step (4.4.6) can be readily computed using the proximal mapping of the function δ_U, which is nothing else but the projection onto the set U, and thus reads as:

$$w^k = w^{k-1} + \sigma p^k - \sigma P_U\left(\frac{w^{k-1} + \sigma p^k}{\sigma} \right).$$

It is important to notice that while the support function of the set U was needed to *model* our constrained problem in the form (M), the computation/knowledge of the support itself is not necessary. Theorem 4.5 holds on problem (P$'$) with the parameters $\tau L_f \leq 1$ and $\sigma\tau\left(\|A\|^2 + 1\right) \leq 1$.

4.4.3 Composite Minimization with Sum of Finitely Many Terms

Let $U \subseteq \mathbb{R}^n$ be a closed and convex set. The problem of interest can be described as follows

$$\text{(Gen)} \qquad \min_{u\in\mathbb{R}^p} \left\{ F(u) + \sum_{i=1}^{m} H_i(B_i u) : \; u \in U \right\},$$

where F is a smooth and convex function on $\mathbb{R}^p$ (see (ii) of the problems setting in Section 4.2), H_i, $i = 1,2,\ldots,m$, is a proper, lsc and convex function over $\mathbb{R}^{d_i}$ (extended valued) and $B_i \in \mathbb{R}^{d_i} \times \mathbb{R}^p$. This model is quite general and covers many interesting problems in imaging sciences and machine learning for which numerical results will be presented in Section 4.5. This model also includes convex problems with separable structure in the objective and coupling linear constraints of the form

$$\text{(SC)} \qquad \min_{x_i} \left\{ \sum_{i=1}^{m} \psi_i(x_i) : \; \sum_{i=1}^{m} B_i x_i = b \right\}.$$

Indeed, a direct computation shows that a dual formulation of problem (SC)[3] also fits the model (Gen) with $F := \langle u, b\rangle$, $H_i := \psi_i^*$ and $B_i \leftarrow -B_i^T$.

Using the dual transportation trick for the constraint $u \in U$, and the fact that H_i is proper, lsc and convex, problem (Gen) can be written as

$$\min_{u\in\mathbb{R}^p} \max_{y_i\in\mathbb{R}^{d_i}, w\in\mathbb{R}^p} \left\{ F(u) + \sum_{i=1}^{m} \left\langle B_i^T y_i, u \right\rangle + \langle w, u\rangle - \sum_{i=1}^{m} H_i^*(y_i) - \sigma_U(w) \right\},$$

which clearly reduces to a saddle-point problem in the form (M) with saddle-point function $K(u,v) = F(u) + \langle u, \mathscr{A} v\rangle - g(v)$ through the identification $g(v) := \sum_{i=1}^{m} H_i^*(y_i) + \sigma_U(w)$, $\mathscr{A} := \left[B_1^T, B_2^T, \ldots, B_m^T, I_p\right]$ and $v := (y_1, y_2, \ldots, y_m, w)$.

Observe that this yields a fully separable nonsmooth part in the variables y_i, $i = 1,2,\ldots,m$ and w, which allows for adequate decomposition in the main computational step of PAPC. In particular, this eliminates the difficulty of computing the proximal map of the composition of a convex function with a linear map. Thus, here the resulting proximal mapping in PAPC can be computed separately and reads for each $i = 1,2,\ldots,m$ as follows

$$v_i^k = \operatorname{prox}_{\sigma_i}^{H_i^*}\left(v_i^{k-1} + \sigma_i B_i p^k\right),$$
$$w^k = \operatorname{prox}_{\sigma}^{\sigma_U}\left(w^{k-1} + \sigma p^k\right).$$

As noted earlier, thanks to the Moreau's proximal identity, the proximal map of the conjugate function H_i^*, $i = 1,2,\ldots,m$, can be easily computed from the proximal map of the function H_i (which are assumed to be simple) and the computational step for w amounts to computing a projection onto the set U.

Note that very recently a different algorithm was proposed in [84], where it was assumed that $F = 0$ and $A_i = I$ for all $1 \le i \le m$, and for which no efficiency estimate was established.

[3] For convenience, dual problems will always be re-written as minimiztion problems after an appropriate change of sign.

4.4.4 Constrained Composite Minimization

Another interesting model is the following constrained composite convex minimization problem

$$\text{(C-Gen)} \qquad \min_{u\in\mathbb{R}^p}\left\{F(u):\ \sum_{i=1}^{m} H_i(B_i u)\le \alpha\right\},$$

where F, H_i $(i=1,2,\ldots,m)$ and B_i $(i=1,2,\ldots,m)$ as in the (Gen) model and here $H_i(\cdot)$ is finite valued. To tackle this problem, we first reformulate the constraint set as the intersection of adequate closed and convex sets defined as follows:

$$\begin{aligned}\Delta_m &:= \left\{z\in\mathbb{R}^m:\ \sum_{i=1}^{m} z_i\le\alpha\right\},\\ C_i &:= \{(y,t)\in\mathbb{R}^p\times\mathbb{R}:\ H_i(y)\le t\}\\ D_i &:= \{(u,y)\in\mathbb{R}^p\times\mathbb{R}^m:\ B_i u = y\}.\end{aligned}$$

Then with these sets, problem (C-Gen) can be written as follows:

$$\min\left\{F(u)+\delta_{\Delta_m}(z)+\sum_{i=1}^{m}\delta_{C_i}(y_i,z_i)\right.$$
$$\left.+\sum_{i=1}^{m}\delta_{D_i}(u,y_i):\ u\in\mathbb{R}^p, z_i\in\mathbb{R}, y_i\in\mathbb{R}^{d_i},\ i=1,2\ldots,m\right\}.$$

As previously explained, using the dual transportation trick, it is then easy to see that the later can then be written as a saddle-point problem of the form (M) which will involve a separable sum of support functions. Applying the PAPC algorithm on the resulting minimax formulation of problem (C-Gen), requires in this case (thanks to Moreau proximal identity) the computation of the projection onto each set Δ_m, C_i and D_i, $i=1,2,\ldots,m$. The projection onto Δ_m and D_i admits a closed form solution. Furthermore, the projection onto a set of the form C_i, namely the epigraph of H_i, can also be computed via the following result whose simple proof is left to the reader.

Proposition 4.4.1. Let $H:\mathbb{R}^p\to\mathbb{R}$ be convex and let $C:=\{(y,t)\in\mathbb{R}^p\times\mathbb{R}:\ H(y)\le t\}$. For any $(x,s)\notin C$, let $(\bar{y},\bar{t})=P_C((x,s))$ be the projection of (x,s) onto C. Then,

$$\bar{y}=\operatorname{argmin}_{y\in\mathbb{R}^n}\left\{\|y-x\|^2+(H(y)-s)^2\right\} \quad \text{and} \quad \bar{t}=H(\bar{y}).$$

For example, when H is a norm, i.e., $H(\cdot):=\|\cdot\|$, then C is the second-order cone and a closed form solution can be derived.

4.4.5 Rate of Convergence for the Primal Formulation

Consider the problem:

$$(CC)\quad \min_{u\in\mathbb{R}^n} F(u),$$
$$\text{s.t. } Au\in C,$$

where $A : \mathbb{R}^n \to \mathbb{R}^d$ is a linear transformation and $C \subset \mathbb{R}^d$ is a convex set which admits an efficient projection operator and has an nonempty interior. We now proceed to show that the sequence generated by applying the PAPC method on the saddle-point reformulation converges in terms of the problem (CC).

We start by deriving a saddle-point reformulation for (CC). Let δ_C be the convex delta function of the set C, and let $\delta_C^*(v) = \max_u\{\langle v,u\rangle - \delta_C(u)\}$ be the convex conjugate of δ_C, then by the identity $\delta_C^{**} \equiv \delta_C$, we get that the following problem is equivalent to (CC)

$$(CC') \quad \min_{u\in\mathbb{R}^n}\max_{v\in\mathbb{R}^d}\{F(u) + \langle Au, v\rangle - \delta_C^*(v)\}.$$

Suppose we applied the PAPC method on this problem with the initial point (u^0, v^0) and the output was $(\bar{u}_N, \bar{v}_N)$, then from Theorem 4.5 we get that for every $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^d$ the following bound holds

$$\begin{aligned} F(\bar{u}_N) + \langle A\bar{u}_N, v\rangle - \delta_C^*(v) - (F(u) + \langle Au, \bar{v}_N\rangle - \delta_C^*(\bar{v}_N)) \\ \leq \frac{\tau^{-1}\|u-u^0\|^2 + \|v-v^0\|_M^2}{2N}, \end{aligned} \tag{4.4.8}$$

where $M = \frac{1}{\sigma}I - \tau A^T A$.

The next proposition establishes a bound on the rate of convergence for the absolute inaccuracy of the primal problem, $F(\bar{u}_N) - F(u^*)$. Note that the sequence $\{\bar{u}_N\}$ generated by the PAPC method on problem (CC') in not necessarily feasible for (CC).

Proposition 4.4.2. Suppose $\bar{u}_N$ is generated by applying the PAPC method on problem (CC') then $\{F(\bar{u}_i)\}_{i\in\mathbb{N}}$ converges to $F(u^*)$ at a sublinear rate or, more precisely,

$$-\frac{\tau^{-1}\|u^*-u^0\|^2 + \|2\kappa\mu - v^0\|_M^2}{2\kappa N} \leq F(\bar{u}_N) - F(u^*) \leq \frac{\tau^{-1}\|u^*-u^0\|^2 + \|v^0\|_M^2}{2N}, \tag{4.4.9}$$

where $\kappa > 0$ is some constant and $\mu \equiv \mu(N) \in \mathbb{R}^d$ is some unit vector.

Note that although μ depends on N, the expression $\|2\kappa\mu - v^0\|_M^2$ is bounded since μ is a unit vector.

Proof. The upper bound. Choosing $u = u^*$ and $v = 0$ in (4.4.8) and noting that $\delta_C^*(0) = 0$, we get:

$$F(\bar{u}_N) - F(u^*) - (\langle Au^*, \bar{v}_N\rangle - \delta_C^*(\bar{v}_N)) \leq \frac{\tau^{-1}\|u^*-u^0\|^2 + \|v^0\|_M^2}{2N}.$$

Now, the term $\langle Au^*, \bar{v}_N\rangle - \delta_C^*(\bar{v}_N)$ must be non-positive (since u^* is feasible), and we reach the desired bound

$$F(\bar{u}_N) - F(u^*) \leq \frac{\tau^{-1}\|u^*-u^0\|^2 + \|v^0\|_M^2}{2N}. \tag{4.4.10}$$

The lower bound. From this point on, we assume that our point, $\bar{u}_N$, is infeasible, i.e., $A\bar{u}_N \notin C$, since otherwise $F(\bar{u}_N) - F(u^*)$ is trivially bounded from below by zero. Denote by (CC_t) the problem

$$\begin{aligned} (CC_t) \quad & \min_{u\in\mathbb{R}^n} F(u), \\ & \text{s.t. } \phi_C(Au) \leq t, \end{aligned}$$

where $\phi_C(u)$ is the *signed* distance to the set C, i.e.,

$$\phi_C(u) = \begin{cases} -\inf_{v \notin C} \|u - v\|, & u \in C, \\ \inf_{v \in C} \|u - v\|, & u \notin C \end{cases}$$

(it is well-known that ϕ_C is a convex function [29, Excercise 8.5]). We have

$$\mathrm{val}(CC_0) = F(u^*).$$

Now, denote by ρ the constraint violation of the point $A\bar{u}_N$,

$$\rho := \phi_C(A\bar{u}_N), \tag{4.4.11}$$

then from the definition of ρ, $\bar{u}_N$ is feasible for (CC_ρ) and we have $F(\bar{u}_N) \geq \mathrm{val}(CC_\rho)$ hence

$$F(\bar{u}_N) - F(u^*) \geq \mathrm{val}(CC_\rho) - \mathrm{val}(CC_0). \tag{4.4.12}$$

Furthermore, by the sensitivity analysis theorem [29, Section 5.6] on problem (CC_t), it follows that there exists a constant $\kappa \geq 0$ (which is equal to the value of the optimal dual variable for the constraint in (CC_t)) such that for any $t \geq 0$

$$\mathrm{val}(CC_t) - \mathrm{val}(CC_0) \geq -\kappa t, \tag{4.4.13}$$

which together with (4.4.12) (taking $t = \rho$) yields

$$F(\bar{u}_N) - F(u^*) \geq -\kappa\rho. \tag{4.4.14}$$

Before proceeding, we need to establish another property of ρ. Since $\rho \geq 0$ is the Euclidean distance from $A\bar{u}_N$ to the set C, we can equivalently say that $A\bar{u}_N$ lie at the boundary of the set $C \oplus \rho B$ (where $\oplus$ denotes the Minkowski sum and B is the unit ball). Let μ be a *unit vector*, normal to a supporting plane to the (convex) set $C \oplus \rho B$ that passes through the point $A\bar{u}_N$, then $A\bar{u}_N \in \mathrm{argmax}_u\{\langle \mu, u\rangle - \delta_{C\oplus\rho B}(u)\}$ and we get $\delta^*_{C\oplus\rho B}(\mu) = \langle \mu, A\bar{u}_N\rangle$. From the basic properties of support functions we get

$$\delta^*_{C\oplus\rho B}(\mu) = \delta^*_C(\mu) + \delta^*_{\rho B}(\mu) = \delta^*_C(\mu) + \rho\|\mu\| = \delta^*_C(\mu) + \rho,$$

hence μ is a unit vector such that

$$\langle A\bar{u}_N, \mu\rangle - \delta^*_C(\mu) = \rho.$$

We return to (4.4.8), with $u = u^*$ but this time we take $v = 2\kappa\mu$ in (4.4.8), we get

$$\langle A\bar{u}_N, 2\kappa\mu\rangle - \delta^*_C(2\kappa\mu) = 2\kappa\rho,$$

hence

$$\kappa\rho \leq F(\bar{u}_N) - F(u^*) + 2\kappa\rho \leq \frac{\tau^{-1}\|u^* - u^0\|^2 + \|2\kappa\mu - v^0\|_M^2}{2N}, \tag{4.4.15}$$

where the left inequality follows from (4.4.14). A lower bound on the primal error function $F(\bar{u}_N) - F(u^*)$ immediately follows:

$$F(\bar{u}_N) - F(u^*) \geq P_\rho - P_0 \geq -\rho \geq -\frac{\tau^{-1}\|u^* - u^0\|^2 + \|2\kappa\mu - v^0\|_M^2}{2\kappa N}. \tag{4.4.16}$$

(When $\kappa = 0$, the value of $F(\bar{u}_N) - F(u^*)$ is nonnegative from (4.4.14), and the last inequality holds for any positive value for κ.)

Combining the bounds derived above, we reach the claimed result. □

4.5 Numerical Examples

In this section, we illustrate the behavior of the PAPC method on the image deblurring and the fused logistic regression problems. Our objective is just to demonstrate the flexibility and the potential of the proposed algorithm. For that purpose we also compare PAPC with two state of the art algorithms sharing the same efficiency estimate: the primal-dual method given in [33] and the extra-gradient based method of [71, 11].

4.5.1 Image Deblurring

In the image deblurring problem we seek to recover an unknown image that has undergone some known, but ill-conditioned transformation and was then corrupted by some statistically independent random noise. In order to choose the best image from the possibly large set of solutions to the inverse transformation, we turn to the successful Rudin-Osher-Fatemi (ROF) model for image restoration, introduced in [90]. Under this model the set of feasible solutions is reduced by assuming that the original image has a bounded value of *total variation* (TV), defined in the discrete (and anisotropic) case for $\mathbf{x} \in \mathbb{R}^{m\times n}$ by (e.g., see [32])

$$\begin{aligned} TV(\mathbf{x}) = &\sum_{i=1}^{m-1}\sum_{j=1}^{n-1}\left\{\left|\mathbf{x}_{i,j}-\mathbf{x}_{i+1,j}\right|+\left|\mathbf{x}_{i,j}-\mathbf{x}_{i,j+1}\right|\right\} \\ &+\sum_{i=1}^{m}\left|\mathbf{x}_{i,n}-\mathbf{x}_{i+1,n}\right|+\sum_{j=1}^{n}\left|\mathbf{x}_{m,j}-bx_{m,j+1}\right|. \end{aligned}$$

More precisely, let $\mathbf{x} \in \mathbb{R}^{m\times n}$ be the original image, $\mathscr{M} : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ be a linear (blurring) transformation, and $\mathbf{w} \in \mathbb{R}^{m\times n}$ be the (unknown) random noise. Then given the observed image $\mathbf{b} = \mathscr{M}\mathbf{x} + \mathbf{w}$, the goal is to find an image $\mathbf{x} \in \mathbb{R}^{m\times n}$ which under the transformation $\mathscr{M}$ produces an image that is "close" to the observed image (i.e., has a low value of $\|\mathscr{M}\mathbf{x} - \mathbf{b}\|$) and has TV value of no more than α, for some given $\alpha > 0$. This problem can be formulated as a constrained convex optimization problem:

$$\text{(BTV)} \quad \min_{\mathbf{x}\in\mathbb{R}^{m\times n}} \left\{ \|\mathscr{M}\mathbf{x} - \mathbf{b}\|^2 : \; TV(\mathbf{x}) \le \alpha \right\},$$

and is often solved via the penalty approach, which thus requires the tuning of the penalty parameter. Here we demonstrate the applicability of PAPC on the original constrained formulation.

One of the main difficulties encountered when solving problem (BTV) is the inability to efficiently compute projections onto the feasible set $\{\mathbf{x} : TV(\mathbf{x}) \le \alpha\}$ thus prohibiting the use of many successful first order methods such as the celebrated fast gradient method [73].

One approach for overcoming the difficulty alluded above is by rewriting the problem as a smooth convex-concave saddle-point problem. For that goal we introduce the linear operator $\mathscr{L} : \mathbb{R}^{m\times n} \to \mathbb{R}^{(m-1)\times n} \times \mathbb{R}^{m\times(n-1)}$ defined by

$$\mathscr{L}(\mathbf{x}) = (\mathbf{p}, \mathbf{q})$$

where $\mathbf{p} \in \mathbb{R}^{(m-1)\times n}$ and $\mathbf{q} \in \mathbb{R}^{m\times(n-1)}$ are the matrices defined by

$$\begin{aligned} p_{i,j} &= x_{i,j} - x_{i+1,j}, \quad i = 1,2,\ldots,m-1 \quad \text{and} \quad j = 1,2,\ldots,n, \\ q_{i,j} &= x_{i,j} - x_{i,j+1}, \quad i = 1,2,\ldots,m \quad \text{and} \quad j = 1,2,\ldots,n-1. \end{aligned}$$

Using this notation, we can write (with some abuse of notation with respect to the usual matrix norm) the total variation of the variable $\mathbf{x}$ as the ℓ_1 norm of $\mathscr{L}\mathbf{x}$, i.e.,

$$TV(\mathbf{x}) = \|\mathscr{L}\mathbf{x}\|_1 .$$

Now, using the technique given in Section 4.4.1, problem (BTV) can be rewritten as the following saddle-point problem

$$\text{(BTV}') \quad \min_{\mathbf{x}\in\mathbb{E}} \max_{\mathbf{z}\in\mathbb{F}} \left\{ \|\mathscr{M}\mathbf{x} - \mathbf{b}\|^2 + \langle \mathscr{L}\mathbf{x}, \mathbf{z}\rangle - \alpha \|\mathbf{z}\|_\infty \right\},$$

where $\mathbb{E} := \mathbb{R}^{m\times n}$ and $\mathbb{F} := \mathbb{R}^{(m-1)\times n} \times \mathbb{R}^{m\times(n-1)}$. The PAPC method for problem (BTV$'$) reads as follows.

PAPC for image deblurring – (BTV$'$) formulation
Initialization. $\left(\mathbf{x}^0, \mathbf{z}^0\right) \in \mathbb{E}\times\mathbb{F}$ and $\tau, \sigma > 0$.
General Step ($k = 1,2,\ldots$)

$$p^k = \mathbf{x}^{k-1} - \tau\left(2\mathscr{M}^T\left(\mathscr{M}\mathbf{x}^{k-1} - \mathbf{b}\right) + \mathscr{L}^T\mathbf{z}^{k-1}\right), \tag{4.5.1}$$

$$\mathbf{z}^k = \operatorname{prox}^{\|\cdot\|_\infty}_{1/(\alpha\sigma)}\left(\mathbf{z}^{k-1} + \sigma\mathscr{L}p^k\right) \tag{4.5.2}$$

$$\mathbf{x}^k = \mathbf{x}^{k-1} - \tau\left(2\mathscr{M}^T\left(\mathscr{M}\mathbf{x}^{k-1} - \mathbf{b}\right) + \mathscr{L}^T\mathbf{z}^{k}\right). \tag{4.5.3}$$

Note that the extra-gradient method [59, 71] cannot be applied on problem (BTV$'$) since the objective is not differentiable. Furthermore, the Chambolle-Pock (CP) method [33] is theoretically applicable on (BTV$'$), however, it requires calculating the inverse of the ill conditioned operator $\mathscr{M}$. A natural observation at this point is that the problem (BTV) has alternative saddle-point formulations that are suitable for applying these methods; in the rest of this part we compare (BTV$'$) to two such formulations of the (BTV) problem: one that is suitable for the Chambolle-Pock method, and the other suitable for the extra-gradient method.

Example 4.5.1. We begin by constructing a formulation that satisfies the assumptions of the successful CP method. As mentioned above, the reason that the CP method cannot be applied directly on problem (BTV$'$) is because the proximal mapping of the quadratic term in the objective cannot be efficiently computed, however, by taking advantage of the identity $\|\mathscr{M}\mathbf{x} - \mathbf{b}\|^2 = \max_{\mathbf{y}\in\mathbb{E}} \left\{ 2\langle \mathscr{M}\mathbf{x} - \mathbf{b}, \mathbf{y}\rangle - \|\mathbf{y}\|^2 \right\}$ we arrive to the following problem, which can be readily solved by the CP method

$$\text{(BTV}'') \quad \min_{\mathbf{x}\in\mathbb{E}} \max_{\mathbf{y}\in\mathbb{E},\mathbf{z}\in\mathbb{F}} \left\{ 2\langle \mathscr{M}\mathbf{x} - \mathbf{b}, \mathbf{y}\rangle - \|\mathbf{y}\|^2 + \langle \mathscr{L}\mathbf{x}, \mathbf{z}\rangle - \alpha\|\mathbf{z}\|_\infty \right\}.$$

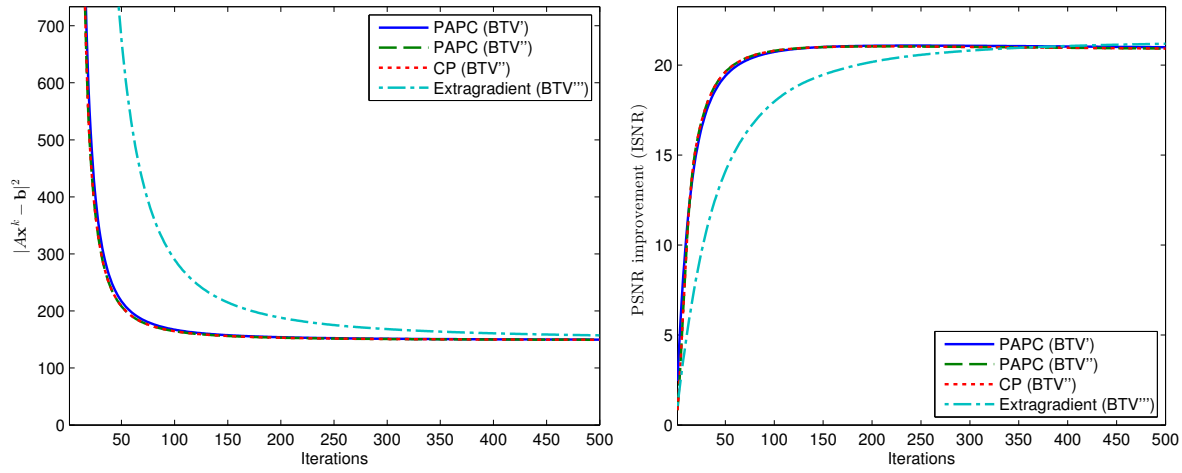

Figure 4.1: Left: the value of the objective function for the ergodic sequences generated by each method. Right: the PSNR improvement for each iteration.

Note that the PAPC method can also be applied on this formulation.

Next, we construct a formulation that is suitable for the extra-gradient method. The desired formulation can be derived from (BTV) by introducing an auxiliary variable $\mathbf{z} = \mathscr{L}\mathbf{x}$ with the constraint $\|\mathbf{z}\|_1 \leq \alpha$, then considering the Lagrangian-based representation of the problem. We arrive to

$$(\text{BTV}''') \qquad \min_{\mathbf{x}\in\mathbb{E},\mathbf{z}\in\mathbb{F}} \max_{\mathbf{y}\in\mathbb{F}} \left\{ \|\mathscr{M}\mathbf{x} - \mathbf{b}\|^2 + \langle \mathbf{y}, \mathscr{L}\mathbf{x} - \mathbf{z}\rangle : \ \|\mathbf{z}\|_1 \leq \alpha \right\},$$

which is amenable to the extra-gradient method since the objective has Lipschitz continuous gradient and the constraint set is "simple" (i.e., a projection onto this set can be easily computed).

We applied the PAPC, CP, and extra-gradient methods on the (BTV$'$), (BTV$''$), and (BTV$'''$) problems (respectively), and also the PAPC method on the (BTV$''$) problem (for the purpose of a fair comparison with the CP method).

For the purpose of this demonstration we took the standard 256×256 Lena test image, where the pixel values were normalized to be in the range $[0,1]$, applied a 5×5 Gaussian blurring operator with standard deviation of 4, and added random Gaussian noise with standard deviation of 0.05. The parameter α was chosen as half the total variation of the original image. The initial point for all methods was chosen as the all-zero vector, and the "free" method parameters (i.e., parameters whose values are not determined by the respective convergence theorem) were hand-tuned to give the best possible performance for the problem instance.

Figure 4.1 summarizes the performance after the first 500 iterations of the ergodic sequences generated by the PAPC, CP and extra-gradient methods on the (BTV$'$), (BTV$''$) and (BTV$'''$) problems (respectively), and by the PAPC method on the (BTV$''$) problem. In all cases the sequences remained feasible throughout the run, and the CPU time of the algorithms was approximately the same in all cases, except for the extra-gradient method, which took about two times longer to complete (as it requires four applications of the operator $\mathscr{A}$ per iteration).

The results show that the performance of the PAPC method was unaffected by the change of problem formulation and was virtually identical to the performance of the CP method, which is often regarded as state of the art. On the other hand, the extra-gradient method on (BTV$'''$) performed significantly worse.

We conclude by noting that PAPC was applicable to the original problem's formulation and its performance appears to be robust with respect to different formulations of the problem (and which were needed to apply the other two schemes). However, adequate reformulation of a given problem is not always easily accessible. The next example exemplifies this situation, whereby PAPC is applicable while the proximal map of the objective function as required by the CP method is a difficult computational task, and we can't reformulate the problem such that the CP method could be effectively applied through simple computations.

4.5.2 Fused Lasso Regression

The main problem in the supervised machine learning framework is defined as follows: given a set of training examples, where each example consists of a vector of features and a label, correctly determine the label for an unseen vector of features. A common approach for solving this problem is by assuming that there is some underlying statistical model which describes the distribution of the label given the feature vector. One such model is the logistic model, under which the probability that the vector of features $a \in \mathbb{R}^n$ will be labeled by $b \in \{-1,+1\}$ is given by

$$P(b|a) = \frac{1}{1+\exp\left(-b\left(w^T a+v\right)\right)}, \tag{4.5.4}$$

where $v \in \mathbb{R}$ and $w \in \mathbb{R}^n$ are the parameters of the model.

Suppose that the training set consists of N training examples $\{(a_i,b_i)\}_{i=1}^N$, where $a_1,\ldots,a_N \in \mathbb{R}^n$ are the feature vectors and $b_1,b_2,\ldots,b_N \in \{-1,+1\}$ are the respective labels, then the likelihood function of the training set is given by

$$L\left(v,w;\{(a_i,b_i)\}_{i=1}^N\right) = \prod_{i=1}^N \frac{1}{1+\exp\left(-b\left(w^T a_i+v\right)\right)}.$$

Following standard practice, we define the *average logistic loss* function by

$$l_{\text{avg}}(v,w) := -\frac{1}{N}\log\left(L\left(v,w;\{(a_i,b_i)\}_{i=1}^N\right)\right) = \frac{1}{N}\sum_{i=1}^N \log\left(1+\exp\left(b_i\left(w^T a_i+v\right)\right)\right),$$

now, using these notations, the *logistic regression problem* (LRP) is defined as the problem of finding the maximum likelihood estimator (MLE) for v and w, i.e., finding v and w that minimize $l_{\text{avg}}(v,w)$. Note that since the average logistic loss is convex, we have reached a convex optimization problem.

When the dimension of the model is large relatively to the number of samples, the LRP quickly becomes ill-conditioned, and additional assumptions are required for finding a meaningful result. For the purpose of this demonstration, we consider the assumptions introduced by Tibshirani et al. in [96] where, similarly to the popular lasso model [95], the vector w is

assumed to have a bounded ℓ_1 norm, and in addition we assume that the vector of differences $w_{i+1} - w_i$, $i = 1, 2, \ldots, n-1$ also has a bounded ℓ_1 norm. These assumptions can be interpreted as a requirement on the model parameters to have a sparse and "staircase-like" pattern. This can happen when the model parameters are given in some natural order. For more discussion on this model see, for example, [36].

Under the assumptions described above, the problem of finding the MLE for v and w become:

$$\text{(FLR)} \qquad \min_{v\in\mathbb{R}, w\in\mathbb{R}^n} \left\{ l_{\text{avg}}(v,w) : \ \|w\|_1 \le s_1, \ \|Dw\|_1 \le s_2 \right\},$$

where $D : \mathbb{R}^n \to \mathbb{R}^{n-1}$ is the linear operator defined by $(Dw)_i = w_{i+1} - w_i$, $i = 1, 2, \ldots, n-1$ and $s_1, s_2 > 0$ are two constants. Using the transportation technique given in Section 4.4.1, we can "transport" the constraints in the primal variables to the dual variable, thereby reaching the following saddle-point formulation of the problem:

$$\text{(FLR}') \qquad \min_{v\in\mathbb{R}, w\in\mathbb{R}^n} \max_{y\in\mathbb{R}^n, z\in\mathbb{R}^{n-1}} \left\{ l_{\text{avg}}(v,w) + \langle w, y\rangle - s_1 \|y\|_\infty + \langle Dw, z\rangle - s_2 \|z\|_\infty \right\}.$$

A straightforward analysis shows that l_{avg} has a Lipscihtz continuous gradient with constant $L_{l_{\text{avg}}} := \sum_{i=1}^N \left(\|a_i\|^2 + 1 \right) / (4N)$, hence the PAPC method can be applied here. The PAPC iterates for this problem then reads as follows.

PAPC for fused lasso regression
Initialization. $\left(v^0, w^0, y^0, z^0\right) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{n-1}$ and $\tau, \sigma > 0$.
General Step ($k = 1, 2, \ldots$)

$$p^k = w^{k-1} - \tau \left(\nabla_w l_{\text{avg}} \left(v^{k-1}, w^{k-1} \right) + y^{k-1} + D^T z^{k-1} \right), \tag{4.5.5}$$

$$y^k = \text{prox}^{\|\cdot\|_\infty}_{1/(s_1\sigma)} \left(y^{k-1} + \sigma p^k \right) \tag{4.5.6}$$

$$z^k = \text{prox}^{\|\cdot\|_\infty}_{1/(s_2\sigma)} \left(z^{k-1} + \sigma D p^k \right) \tag{4.5.7}$$

$$v^k = v^{k-1} - \tau \nabla_v l_{\text{avg}} \left(v^{k-1}, w^{k-1} \right). \tag{4.5.8}$$

$$w^k = w^{k-1} - \tau \left(\nabla_w l_{\text{avg}} \left(v^{k-1}, w^{k-1} \right) + y^k + D^T z^k \right). \tag{4.5.9}$$

We would like to emphasize that this method requires only *one evaluation of the computationally expensive gradient* ∇l_{avg} per iteration. Also, note that thanks to Moreau's idendity, the proximal map computations in steps (4.5.6) and (4.5.7) reduce to projection onto an l_1 ball. It is easy to see that the later problem can be reduced to a projection onto a simplex, and thus as proven in [30] can be efficiently computed (with linear complexity $O(p)$, where p is the problem's dimension).

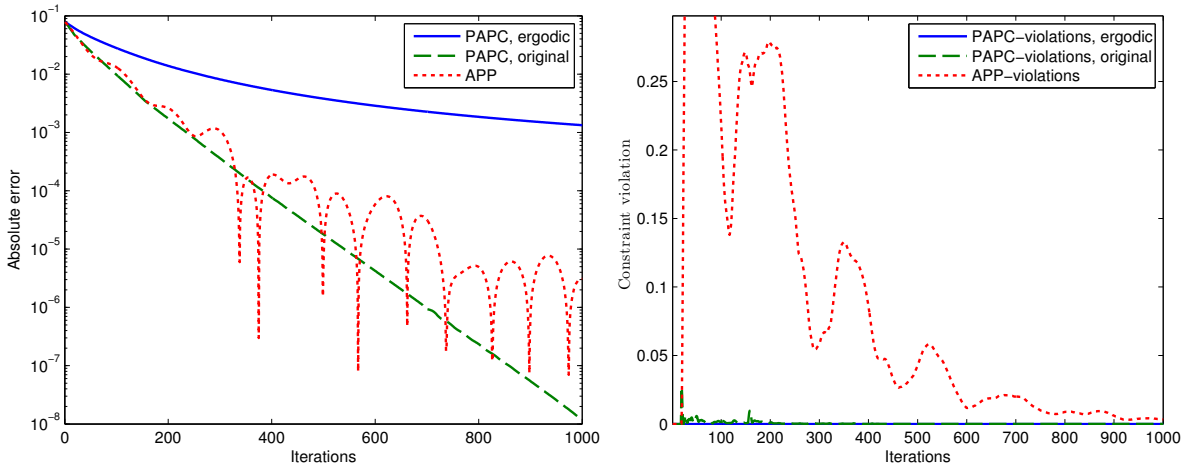

Figure 4.2: Left: the absolute error at each iteration. Right: the constraint violation at each iteration.

Example 4.5.2. In this example we apply the PAPC method to the problem (FLR′). For this purpose we generated 1000 samples from the logistic model that corresponds to $v = 0$ and the vector $w \in \mathbb{R}^{100}$ defined by

$$w_i = \begin{cases} 0.1, & i \in \{11, \dots, 14\}, \\ -0.2, & i \in \{41, \dots, 44\}, \\ 0.3, & i \in \{91, \dots, 94\}, \\ 0, & \text{otherwise.} \end{cases}$$

The features $a_i \in \mathbb{R}^{100}$ $(i = 1, 2, \dots, 1000)$ were generated such that each component was independently drawn from the standard normal distribution, and the labels $b_i \in \{-1, +1\}$ were then randomly chosen according to the logistic model (4.5.4).

We applied the PAPC method with $\tau = 1/L_{l_{\text{avg}}}$, $\sigma = L_{l_{\text{avg}}} / \left(\|D\|^2 + 1 \right)$. The starting point was $v^0 = 0$ and $w^0 = 0$ and the bounds s_1 and s_2 were chosen according to their value in the model, i.e., $s_1 = 3$ and $s_2 = 1.2$. An exact solution for the optimization problem was then obtained using the CVX modelling package for MATLAB [49, 48].

Figure 4.2 shows the absolute error of the objective $l_{\text{avg}}(v, w) - l^*_{\text{avg}}$ and the constraint violation $\max\{\|w\|_1 - s_1, 0\} + \max\{\|Dw\|_1 - s_2, 0\}$ after the first 1000 iterates of the main and the ergodic sequences of the PAPC method. As can be seen, the ergodic sequence generated by the PAPC method performs as expected from the rate of convergence theorem and remains in the feasible region for the entire run. The main sequence $\left\{ \left(v^k, w^k \right) \right\}_{k \in \mathbb{N}}$, on the other hand, performs much better and appears to converge at a linear rate, where the constraints violations for the primal iterates remained very small throughout the run and reached about 10^{-6} after 1000 iterations.

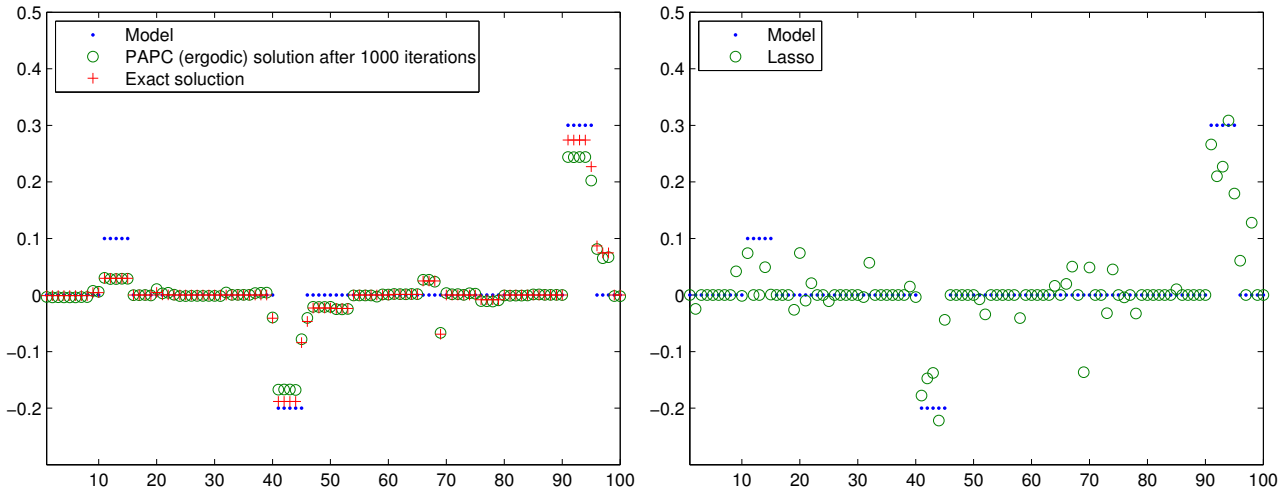

Figure 4.3: Left: the w parameter of the model and its approximation by PAPC. Right: the lasso solution of the problem.

The model parameters estimated by the PAPC method (the ergodic sequence) and the exact solution of problem (FLR) given by the CVX modeling package are shown in the left part of Figure 4.3. The right part of the figure shows the estimated model using the classical lasso model, $\min\left\{l_{\text{avg}}(v,w) : \|w\|_1 \leq s_1\right\}$. As can be seen, adding the "fused" constraints considerably increases the quality and the ability to interpret the recovered model.

4.6 Appendix: A PEP for PAPC

In this section, we show how to apply the PEP approach to structured minimax problem of the form

$$\min_{u\in\mathbb{R}^d}\max_{v\in\mathbb{R}^d}\left\{f(u)+\langle u,Av\rangle - g(v)\right\},$$

where f and g are convex, f is differentiable, with a Lipschitz continuous gradient, and $A \in \mathbb{R}^{d\times d}$ is a matrix with $\|A\| \leq \omega$. Although no new analytical results on the performance of the PAPC method are derived here, the following formulation may open a possibility for finding new methods for this class of problems.

We consider a simplified version of the PAPC method:

Initialization. $\left(u^0, v^0\right) \in \mathbb{R}^d \times \mathbb{R}^d$ and $\tau, \sigma > 0$.
General Step ($k = 1,2,\ldots$)

$$\begin{aligned} p^k &= u^{k-1} - \tau\left(Av^{k-1} + f'(u^{k-1})\right),\\ v^k &= \text{prox}^g_{1/\sigma}(v^{k-1} + \sigma A^T p^k),\\ u^k &= u^{k-1} - \tau\left(Av^k + f'(u^{k-1})\right). \end{aligned}$$

Since we want an upper bound on the ergodic convergence rate of the method, the corresponding PEP is given by:

$$\begin{aligned}
\max_{\phi\in C_L^{1,1},\psi \text{ convex},\ \mathscr{A}\in\mathbb{R}^{d\times d}} \ & \sum_{i=1}^{N}(\varphi(u_i)+\langle \mathscr{A}u_i, v_*\rangle - \psi(v^*)) - (\varphi(u_*)+\langle \mathscr{A}u_*, v_i\rangle - \psi(v_i)) \\
\text{s.t. } & p^k = u^{k-1} - \tau\left(\mathscr{A}v^{k-1} + f'(u^{k-1})\right), \quad k=1,\dots,N, \\
& v^k = \operatorname{prox}^g_{1/\sigma}(v^{k-1} + \sigma\mathscr{A}p^k), \quad k=1,\dots,N, \\
& u^k = u^{k-1} - \tau\left(\mathscr{A}v^{k} + f'(u^{k-1})\right), \quad k=1,\dots,N, \\
& \phi'(u_k) = f'(u_k), \quad k=0,\dots,N, \\
& (u_*, v_*) \text{ is a saddle point}, \\
& \|u_0 - u_*\| \le R,\ \|v_0 - v_*\| \le R, \\
& \|\mathscr{A}\| \le \omega.
\end{aligned}$$

Alternatively, we can choose the objective to correspond to the best point encountered in the first N iterations, i.e.,

$$\min_{0\le i\le N}\{(\varphi(u_i)+\langle Au_i, v_*\rangle - \psi(v^*)) - (\varphi(u_*)+\langle Au_*, v_i\rangle - \psi(v_i))\}.$$

There are three challenges in developing a finite-dimensional relaxation for this problem, compared to the PEP for unconstrained first-order methods:

- The PAPC method employs a proximal mapping step.
- The applications of the linear mappings $\mathscr{A}$ and $\mathscr{A}^T$ need to be expressed.
- The bound on the matrix norm of $\mathscr{A}$ must be imposed.

In order to reach a finite-dimensional relaxation, we use the following properties:

$$\begin{aligned}
& \frac{1}{2L}\|f'(x)-f'(y)\|^2 \le F(x) - F(y) - \langle f'(y), x-y\rangle, \text{ for all } x,y\in\mathbb{R}^d, \\
& 0 \le g(x) - g(y) - \langle g'(y), x-y\rangle, \text{ for all } x,y\in\mathbb{R}^d,\ g'(y)\in\partial g(y), \\
& u = \operatorname{prox}^g_{1/\sigma}(x) \Leftrightarrow x - u \in \sigma\partial g(u), \\
& \omega^2 V^T V - V^T A^T A V \succeq 0, \text{ for any matrix } V\ .
\end{aligned}$$

We denote:

$$\begin{aligned}
& y_k = \varphi(u_k) - \varphi(u_*), \\
& z_k = \psi(v_k) - \psi(v_*), \\
& g_k = \nabla\varphi(u_k), \\
& h_k \in \partial\psi(v_k), \\
& v_k^A = \mathscr{A}v^k, \\
& u_k^A = \mathscr{A}^T u^k, \\
& p_k^A = \mathscr{A}^T p^k,
\end{aligned}$$

then from these definitions, we get the following relations

$$\begin{aligned}
&\langle p_i^A, v_j\rangle = \langle p_i, v_j^A\rangle, \quad i,j = 0,\dots,N,*,\\
&\langle u_i^A, v_j\rangle = \langle u_i, v_j^A\rangle, \quad i,j = 0,\dots,N,*.
\end{aligned}$$

The properties of f and g induce the following inequalities:

$$\begin{aligned}
&\frac{1}{2L}\|g_i - g_j\|^2 \le y_i - y_j - \langle g_j, u_i - u_j\rangle, \quad i,j = 0,\dots,N,*,\\
&0 \le h_i - h_j - \langle h_j, v_i - v_j\rangle, \quad i,j = 0,\dots,N,*.
\end{aligned}$$

Finally, from the properties of $\mathscr{A}$, we get that

$$\begin{aligned}
&\omega^2 p_I^T p_I - (p_I^A)^T p_I^A \succeq 0,\\
&\omega^2 u_I^T u_I - (u_I^A)^T u_I^A \succeq 0,\\
&\omega^2 v_I^T v_I - (v_I^A)^T v_I^A \succeq 0,
\end{aligned}$$

where $I = \{0,\dots,N,*\}$ and u_I (v_I) stands for the matrix whose columns are u_i (v_i) for $i \in I$.

The PAPC method can now be expressed as follows:

$$\begin{aligned}
p_k &= u_{k-1} - \tau\left(v_{k-1}^A + g_{k-1}\right),\\
v_k &= v^{k-1} + \sigma p_k^A - \sigma h_k,\\
u_k &= u_{k-1} - \tau\left(v_k^A + g_{k-1}\right).
\end{aligned}$$

and the corresponding finite-dimensional PEP for the PAPC method, is given by

$$\begin{aligned}
\max_{\substack{u_i,u_i^{A^T},v_i,v_i^A,v_i^{A^T A},\\ g_i,g_i^A,h_i,y_i,z_i}} \ & \frac{1}{N}\sum_{i=1}^{N}\left(y_i + \langle v_*^A, u_i\rangle - z_* - \left(y_* + \langle v_i^A, u_*\rangle - z_i\right)\right)\\
\text{s.t. } & p_k = u_{k-1} - \tau\left(v_{k-1}^A + g_{k-1}\right),\\
& v_k = v^{k-1} + \sigma p_k^A - \sigma h_k,\\
& u_k = u_{k-1} - \tau\left(v_k^A + g_{k-1}\right),\\
& \langle p_i^A, v_j\rangle = \langle p_i, v_j^A\rangle, \quad i,j = 0,\dots,N,*,\\
& \langle u_i^A, v_j\rangle = \langle u_i, v_j^A\rangle, \quad i,j = 0,\dots,N,*,\\
& \frac{1}{2L}\|g_i - g_j\|^2 \le y_i - y_j - \langle g_j, u_i - u_j\rangle, \quad i,j = 0,\dots,N,*,\\
& 0 \le h_i - h_j - \langle h_j, v_i - v_j\rangle, \quad i,j = 0,\dots,N,*.\\
& \omega^2 u_I^T u_I - (u_I^A)^T u_I^A \succeq 0, \quad I = \{0,\dots,N,*\}\\
& \omega^2 v_I^T v_I - (v_I^A)^T v_I^A \succeq 0, \quad I = \{0,\dots,N,*\}\\
& \|u_0 - u_*\|^2 \le R,\ \|v_0 - v_*\|^2 \le R,
\end{aligned}$$

An efficient SDP relaxation can be devised in the usual way, by defining a matrix

$$Z = (g_i, h_i, u_i, p_i, v_i, u_k^A, p_k^A, v_k^A)^T (g_i, h_i, u_i, p_i, v_i, u_k^A, p_k^A, v_k^A)$$

and expressing the inequalities in the problem above through Z. A MATLAB code that implements this SDP relaxation is available at Listing 4.1.

Numerical experiments suggest that the bound in Theorem 4.5 can be slightly improved, where the constant in the denominator can be increased from N to $N+1$.

Listing 4.1: A PEP of PAPC

```
function bound = PAPC_pep(N, L, normA, R)
%Returns the efficiency estimate of the PAPC method on problems of the form
%min_u max_v f(u)+<Au,v>-g(v),
%where f has a Lipchitz constant L, |A|=normA, |u-u*|<=R, |v-v*|<=R
if (nargin<=4)
    N=4;
    L=1;
    normA=1;
    R=1;
end

global n; n=N;
global gL; gL=L;
global gSigma; gSigma=L/normA^2;

problemdim=totalvars();
cvx_precision best;
cvx_solver sedumi;
cvx_begin quiet
    variable Z(problemdim,problemdim);
    variable deltay_var(N+1);
    variable deltaz_var(N+1);

    global gZ;gZ=Z;
    global dyuv;dyuv=deltay_var;
    global dzv;dzv=deltaz_var;

    expression objective(N);
    for i=1:N
        objective(i)=deltay(i)+Zdot(u(i),Av(N+1))+deltaz(i)-deltay(N+1)-Zdot(u(N+1),Av(i))-deltaz(N+1);
    end
    maximize (sum(objective)/N); %Alternatively:maximize (min(objective))

    subject to
        Z==semidefinite(problemdim);
        %Lipschitz cont. of f' and convexity of g
        for i=0:N+1 %N+1 stands for *
            for j=0:N+1
                if i~=j
```

```
40                      1/(2*L)*Znorm2(gu(i)-gu(j))+Zdot(gu(j),u(i)-u(j))<=
                            deltay(i)-deltay(j);
41                      deltaz(i)-deltaz(j)<=Zdot(h(i),v(i)-v(j));
42                  end
43              end
44          end
45          %<p, Av> = <A'p, v>
46          for i=1:N
47              for j=0:N+1
48                  Zdot(p(i),Av(j))==Zdot(ATp(i),v(j));
49              end
50          end
51          %Bounded start
52          Znorm2(u(0)-u(N+1))<=R*R;
53          Znorm2(v(0)-v(N+1))<=R*R;
54          %Definition of normA
55          (normA^2)*p(1:N)'*Z*p(1:N)-ATp(1:N)'*Z*ATp(1:N)==semidefinite(N)
                ;
56          (normA^2)*v(0:N+1)'*Z*v(0:N+1)-Av(0:N+1)'*Z*Av(0:N+1)==
                semidefinite(N+2);
57  cvx_end
58
59  bound=cvx_optval;
60  end
61  %*****************************************
62  %Helper functions:
63  %*****************************************
64  function n=Znorm2(v)
65      global gZ;
66      n=v'*gZ*v;
67  end
68  function d=Zdot(u,v)
69      global gZ;
70      d=u'*gZ*v;
71  end
72  function p=gupos
73      p=0;
74  end
75  function nv=guvars
76      global n;
77      nv=n+2;
78  end
79  function v=gu(idx)
80      global n;
81      if (min(idx)<0 || max(idx)>n+1)
82          error('argument error in gu(i)');
83      end
84      v=sparse(totalvars(),length(idx));
85      for k=1:length(idx)
86          i=idx(k);
87          v(gupos()+i+1,k)=1;
88      end
89  end
```

```
function p=hpos
    p=gupos()+guvars();
end
function nv=hvars
    global n;
    nv=n+2;
end
function v=h(idx)
    global n;
    if (min(idx)<0 || max(idx)>n+1)
        error('argument error in h(i)');
    end
    v=sparse(totalvars(),1);
    for k=1:length(idx)
        i=idx(k);
        v(hpos()+i+1,k)=1;
    end
end
function p=upos
    p=hpos()+hvars();
end
function nv=uvars
    nv=1;
end
function rv=u(idx)
    global gL;
    global n;
    if (min(idx)<0 || max(idx)>n+1)
        error('argument error in u(idx)');
    end
    rv=sparse(totalvars(),length(idx));
    for k=1:length(idx)
        i=idx(k);
        if (i==0)
            %Leave the vector to be 0
        elseif (i~=n+1)
            rv(:,k)=u(i-1)-(1/gL)*(Av(i)+gu(i-1)); %The definition of
                the PAPC method
        else
            rv(upos()+1,k)=1;
        end
    end
end
function v=p(idx)
    global gL;
    global n;
    if (min(idx)<1 || max(idx)>n)
        error('argument error in p(idx)');
    end
    for k=1:length(idx)
        i=idx(k);
        v(:,k)=u(i-1)-(1/gL)*(Av(i-1)+gu(i-1)); %The definition of the
            PAPC method
```

```
141     end
142 end
143 function p=vpos
144     p=upos()+uvars();
145 end
146 function nv=vvars
147     nv=1;
148 end
149 function rv=v(idx)
150     global n;
151     if (min(idx)<0 || max(idx)>n+1)
152         error('argument error in v(idx)');
153     end
154     global gSigma;
155     rv=sparse(totalvars(),length(idx));
156     for k=1:length(idx)
157         i=idx(k);
158         if (i==0)
159             %Leave the vector to be 0
160         elseif (i~=n+1)
161             rv(:,k)=v(i-1)+gSigma*(ATp(i)-h(i)); %The definition of the
                    PAPC method
162         else
163             rv(vpos()+1,k)=1;
164         end
165     end
166 end
167 function p=Avpos
168     p=vpos()+vvars();
169 end
170 function nv=Avvars
171     global n;
172     nv=n+2;
173 end
174 function v=Av(idx)
175     global n;
176     if (min(idx)<0 || max(idx)>n+1)
177         error('argument error in Av(idx)');
178     end
179     v=sparse(totalvars(),length(idx));
180     for k=1:length(idx)
181         i=idx(k);
182         v(Avpos()+i+1,k)=1;
183     end
184 end
185 function p=ATppos
186     p=Avpos()+Avvars();
187 end
188 function nv=ATpvars
189     global n;
190     nv=n;
191 end
192 function v=ATp(idx)
```

```
193     global n;
194     if (min(idx)<0 || max(idx)>n)
195         error('argument error in ATp(i)');
196     end
197     v=sparse(totalvars(),length(idx));
198     for k=1:length(idx)
199         i=idx(k);
200         v(ATppos()+i,k)=1;
201     end
202 end
203 function t=totalvars
204     t=ATppos()+ATpvars();
205 end
206 function d=deltay(i)
207     global dyuv;
208     global n;
209     if (i<0 || i>n+1)
210         error('argument error in deltay(i)');
211     end
212     if (i~=n+1)
213         d=dyuv(i+1);
214     else
215         d=0;
216     end
217 end
218 function d=deltaz(i)
219     global dzv;
220     global n;
221     if (i<0 || i>n+1)
222         error('argument error in deltaz(i)');
223     end
224     if (i~=n+1)
225         d=dzv(i+1);
226     else
227         d=0;
228     end
229 end
```

Chapter 5

A new SDP relaxation scheme for a class of quadratic matrix problems

We consider a special class of quadratic matrix optimization problems which often arise in applications. By exploiting the special structure of these problems, we derive a new semidefinite relaxation, which under mild assumptions is proven to be tight for a larger number of constraints than could be achieved via a direct approach. We show the potential usefulness of these results when applied to the robust least squares problem, the sphere packing problems, and to problems over the complex domain.

This chapter is based on the published paper [16].

5.1 Introduction

The class of nonconvex quadratically constrained quadratic programming (QCQP) problems plays a key role in both subproblems arising in optimization algorithms such as trust region methods (see for example [31, 45]) and is also a bridge to the analysis of many combinatorial optimization problems that can be formulated as such. In principle, nonconvex QCQP problems are hard to solve, and as a result many approximation techniques were devised in order to tackle them. Many of these techniques rely on the so-called semidefinite relaxation (SDR), which is a related convex problem over the matrix space that can be solved efficiently, see e.g., [51, 99].

A key issue in the analysis of QCQPs is to determine under which conditions the semidefinite relaxation is tight, meaning that it has the same optimal value as the original QCQP problem. In these cases, one can construct the global optimal solution of the QCQP problem from the optimal solution of the SDR via a rank reduction procedure. There are several classes of QCQP problems which posses this "tight semidefinite relaxation" result; among them are the class of generalized trust region subproblems [45, 67] which are QCQPs with a single quadratic constraint, problems with two constraints over the complex number field [17] as well as problems arising in the context of quadratic assignment problem [2, 1].

Another class of QCQP problems is the class of *Quadratic Matrix Programming* (QMP)

problems whose general form is given by

$$\text{(QMP)} \quad \begin{aligned} \min_{X\in\mathbb{R}^{n\times r}} \ & \operatorname{tr}(X^TA_0X)+2\operatorname{tr}(\tilde{B}_0^TX)+c_0 \\ \text{s.t. } & \operatorname{tr}(X^TA_iX)+2\operatorname{tr}(\tilde{B}_i^TX)+c_i \le \alpha_i, \quad i\in\mathscr{I}, \\ & \operatorname{tr}(X^TA_jX)+2\operatorname{tr}(\tilde{B}_j^TX)+c_j = \alpha_j, \quad j\in\mathscr{E}, \end{aligned}$$

where n, r are positive integers, $\mathscr{I}$ and $\mathscr{E}$ are sets of indices such that $\mathscr{I}\cap\mathscr{E}=\emptyset$, $A_i\in\mathbb{S}^n$, $\tilde{B}_i\in\mathbb{R}^{n\times r}$ and $c_i,\alpha_i\in\mathbb{R}$. This class of problems was introduced and studied in [15] where it was also shown that it encompasses a broad class of important problems both in theory and in applications. The main result in [15] is that problem (QMP) with at most r constraints has a tight SDR property. In the homogeneous case (i.e., when $\tilde{B}_i=0$ for all i) this question was already studied by Barvinok [13, 14] for the problem of determining the feasibility of this problem; Barvinok's results were then extended by Pataki [82] to include any homogeneous quadratic objective function. In both cases it was shown that it is possible to use the SDP relaxation to solve the original nonconvex problem when the number of constraints is at most $\binom{r+2}{2}-1$.

In this chapter we concentrate on a special type of QMP problems defined by

$$\text{(sQMP)} \quad \begin{aligned} \min_{X\in\mathbb{R}^{n\times r}} \ & \operatorname{tr}(X^TA_0X)+2\operatorname{tr}(V^TB_0^TX)+c_0 \\ \text{s.t. } & \operatorname{tr}(X^TA_iX)+2\operatorname{tr}(V^TB_i^TX)+c_i \le \alpha_i, \quad i\in\mathscr{I}, \\ & \operatorname{tr}(X^TA_jX)+2\operatorname{tr}(V^TB_j^TX)+c_j = \alpha_j, \quad j\in\mathscr{E}, \end{aligned} \tag{5.1.1}$$

with $A_i\in\mathbb{S}^n$, $B_i\in\mathbb{R}^{n\times s}$ $(i\in\{0\}\cup\mathscr{I}\cup\mathscr{E})$ and $0\neq V\in\mathbb{R}^{s\times r}$, $s\le r$. Essentially, this type of QMP problems is characterized by the property that the matrices $\tilde{B}_i$ are of the special form $\tilde{B}_i=B_iV$; for the case $n>r>s$, this means that the range spaces of the $n\times r$ matrices $\tilde{B}_i$, $(i\in\{0\}\cup\mathscr{I}\cup\mathscr{E})$ are all contained in the same s-dimensional subspace, which is the range space of V. Note that when $s=r$ and $V=I_r$ we are back to the original QMP setting.
At a first glance, it seems that this property of the matrices $\tilde{B}_i$ is quite restrictive, however, it naturally appears in applications as the example below demonstrates.

Example 5.1.1 (Robust Least Squares). Consider the *robust least squares problem* which seeks to minimize $\|Ax-b\|^2$ when the matrix $A\in\mathbb{R}^{r\times n}$ is perturbed by an unknown matrix $\Delta\in\mathscr{U}$. This problem was defined and studied in [42] and [34] and was later inspected via the QMP framework in [15]. The problem can be formulated as

$$\min_x \max_{\Delta\in\mathscr{U}} \|b-(A+\Delta^T)x\|^2, \tag{5.1.2}$$

where in the following we assume that the set $\mathscr{U}$ has the following form:

$$\mathscr{U}=\{\Delta\in\mathbb{R}^{n\times r} : \|L_i\Delta\|^2\le\rho_i, i=1,\dots,m\}$$

for some $L_i\in\mathbb{R}^{k_i\times n}$ and where the norm used is the Frobenius norm. Under these assumptions we can rewrite the robust least squares problem (5.1.2) as follows:

$$\text{(RLS)} \quad \begin{aligned} \min_x \max_{\Delta\in\mathbb{R}^{n\times r}} \ & \operatorname{tr}(\Delta^Txx^t\Delta)+2\operatorname{tr}((b-Ax)x^T\Delta)+\operatorname{tr}((b-Ax)(b-Ax)^T) \\ \text{s.t. } & \operatorname{tr}(\Delta^TL_i^TL_i\Delta)\le\rho_i, \quad i=1,\dots,m. \end{aligned}$$

The inner maximization problem is an sQMP with $s = 1$ since here we can take $V = (b - Ax)^T, B_0 = x, B_i = 0, i = 1, \ldots, m$.

The main result of this chapter, developed in Section 5.3, is that a specially devised SDR of problem sQMP is tight as long as the number of constraints does not exceed $\binom{r+2}{2} - \binom{s+1}{2} - 1$, which is an improvement of the result from [15] that allows only r constraints. To do so, we use a rank reduction argument which can be traced back to Barvinok and Pataki (see the beginning of the introduction). Further analysis of the robust least squares example along with an additional sphere packing application are given in Section 5.4.

Notation. We use the following notations: Suppose (P) is an optimization problem that attains its optimal value (e.g., (P) $\min_{x \in C} f(x)$), then we denote (P)'s optimal value by $\mathrm{val}(P)$. We use $\mathbb{S}^n$ to denote the set of $n \times n$ symmetric matrices over $\mathbb{R}$, and for two matrices A, B, $A \succeq B$, $(A \succ B)$ means $A - B$ is positive semidefinite (positive definite). The $n \times m$ matrix of zeros is denoted by $0_{n \times m}$; I_r is the $r \times r$ identity matrix and $e_i \in \mathbb{R}^n$, $i = 1, \ldots, n$ stands for the i-th canonical unit vector.

5.2 Preliminaries

We record here some results that will be useful to our analysis. We begin with a fundamental result on existence of low rank solutions to general semidefinite programming (SDP) problems which was established by Pataki [82].

Consider the general SDP problem:

$$\begin{array}{ll} \min\limits_{X \in \mathbb{S}^n} & \mathrm{tr}(C_0 X) \\ \text{s.t.} & \mathrm{tr}(C_i X) \leq b_i, \qquad i \in \mathscr{I}, \\ & \mathrm{tr}(C_i X) = b_i, \qquad i \in \mathscr{E}, \\ & X \succeq 0, \end{array} \tag{5.2.1}$$

where $C_i \in \mathbb{S}^n, i \in \{0\} \cup \mathscr{I} \cup \mathscr{E}$. We state here a slightly different (but equivalent) version of Pataki's result which was given in [15, Theorem 3.1].

Theorem 5.1. *Suppose that the SDP problem (5.2.1) attains its optimal value. Then if* $|\mathscr{I}| + |\mathscr{E}| \leq \binom{r+2}{2} - 1$, *there exists an optimal solution* $X^* \in \mathbb{S}^n$ *satisfying* $\mathrm{rank} X^* \leq r$.

The next result recalls the so-called Schur-Complement Lemma, see e.g., [23].

Lemma 5.2. *Consider a square matrix in the block form:*

$$M = \begin{pmatrix} F & G^T \\ G & H \end{pmatrix},$$

where F is a square matrix assumed to be positive definite. Then,

$$M \succeq 0 \ (\succ 0) \text{ if and only if } H - GF^{-1}G^T \succeq 0 \ (\succ 0).$$

Finally, we need the following result that plays an important role in the forthcoming analysis.

Lemma 5.3. *Let $A,B \in \mathbb{R}^{m\times n}$ be two matrices satisfying $AA^T = BB^T$. Then there exists an orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ such that $A = BQ$.*

Proof. Since $AA^T = BB^T$, it follows that A and B have the same singular values. Let U be an orthogonal matrix diagonalizing $AA^T = BB^T$, namely, $U^TAA^TU = U^TBB^TU$ is diagonal. The matrices A and B have the following singular value decomposition (SVD):

$$A = U\Sigma V_1^T, B = U\Sigma V_2^T,$$

where $V_1, V_2 \in \mathbb{R}^{n\times n}$ are orthogonal matrices and Σ is an $m \times n$ diagonal matrix containing the singular values of A (which are also the singular values of B). Thus, $A = BV_2V_1^T$, and the result is established with $Q = V_2V_1^T$. □

5.3 A Tight SDR Result for (sQMP)

Consider the problem (sQMP) (given in (5.1.1)). For $i \in \{0\} \cup \mathscr{I} \cup \mathscr{E}$, define

$$M_i = \begin{pmatrix} A_i & B_i \\ B_i^T & \frac{c_i}{\operatorname{tr} VV^T} I_s \end{pmatrix} \in \mathbb{S}^{n+s},$$

and consider the following homogenized program

$$(\text{sQMP}_2) \qquad \begin{aligned} \min_{Z\in\mathbb{S}^{n+s}} \ & \operatorname{tr}(M_0Z) \\ \text{s.t.} \ & \operatorname{tr}(M_iZ) \le \alpha_i, \quad i \in \mathscr{I}, \\ & \operatorname{tr}(M_jZ) = \alpha_j, \quad j \in \mathscr{E}, \\ & Z \succeq 0, \\ & \operatorname{rank} Z \le r, \\ & Z_{n+i,n+j} = (VV^T)_{i,j}, \quad i,j = 1,\dots,s, \end{aligned}$$

where the last set of constraints essentially state that the bottom right $s \times s$ submatrix of Z is VV^T. Note that these constraints can be expressed using $\binom{s+1}{2}$ trace constraints. As the following lemma shows, (sQMP) and (sQMP_2) are essentially the same problem.

Lemma 5.4. *Problem* (sQMP) *attains its optimal value if and only if* (sQMP_2) *attains its optimal value. Furthermore, if either* val(sQMP) *or* val(sQMP_2) *is finite, then* val(sQMP) = val(sQMP_2).

Proof. We will show that any feasible point for one problem can be transformed into a feasible point for the other problem without affecting the objective value.

Suppose that X is feasible for $(sQMP)$, then define

$$Z = \begin{pmatrix} XX^T & XV^T \\ VX^T & VV^T \end{pmatrix}.$$

Since

$$Z = \begin{pmatrix} X \\ V \end{pmatrix} \begin{pmatrix} X^T & V^T \end{pmatrix},$$

we get that $\operatorname{rank} Z \leq r$. In addition,

$$\operatorname{tr}(M_i Z) = \operatorname{tr}(A_i X X^T) + 2\operatorname{tr}(B_i^T X V^T) + c_i, \quad i \in \{0\} \cup \mathscr{I} \cup \mathscr{E}, \tag{5.3.1}$$

which immediately implies that Z is feasible for (sQMP_2) and has the same objective function value as X for (sQMP). In the reverse direction, suppose that Z is feasible for (sQMP_2). Since the rank of Z is at most r and Z is positive semidefinite, there exists a matrix $W \in \mathbb{R}^{(n+s)\times r}$ such that $Z = WW^T$. Denote the first n rows of W by $Y \in \mathbb{R}^{n\times r}$ and the last s rows of W by $U \in \mathbb{R}^{s\times r}$ (i.e., $W = (Y;U)$ in "Matlab notation"); we can therefore write

$$Z = \begin{pmatrix} YY^T & YU^T \\ UY^T & UU^T \end{pmatrix}$$

From the constraints on Z we obtain that $UU^T = VV^T$, and thus it follows from Lemma 5.3 that there exists an orthogonal matrix $Q \in \mathbb{S}^r$ such that $U = VQ$. Now, define $X = YQ^T$, then since $YU^T = XQQ^TV^T = XV^T$, we get

$$Z = \begin{pmatrix} XX^T & XV^T \\ VX^T & VV^T \end{pmatrix}$$

and therefore, following the same argument as in the first part or the proof, X is feasible for (sQMP) and achieves the same objective value. □

We now omit the hard rank constraint and consider the SDP relaxation of (sQMP_2) given by

$$\text{(sQMP-R)} \qquad \begin{aligned} \min_{Z\in\mathbb{S}^{n+s}} \ & \operatorname{tr}(M_0 Z) \\ \text{s.t.} \ & \operatorname{tr}(M_i Z) \leq \alpha_i, \quad i \in \mathscr{I} \\ & \operatorname{tr}(M_j Z) = \alpha_j, \quad j \in \mathscr{E} \\ & Z \succeq 0 \\ & Z_{n+i,n+j} = (VV^T)_{i,j}, \quad i,j = 1,\ldots,s \end{aligned}$$

Remark 5.3.1. Note that when $n+s \leq r$ the relaxation (sQMP-R) is exact since the rank constraint in (sQMP_2) is trivially satisfied.

We now proceed to give a condition, similar to Theorem 3.2 in [15], under which (sQMP) can be solved via (sQMP-R). Note that the number of trace constraints in (sQMP-R) is $|\mathscr{I}| + |\mathscr{E}| + \binom{s+1}{2}$ instead of $|\mathscr{I}| + |\mathscr{E}| + \binom{r+1}{2}$ in the corresponding setting of Theorem 3.2 in [15]. This property of the new SDP relaxation allows us to improve and extend the result of Theorem 3.2 as follows:

Theorem 5.5. *Suppose that problem* (sQMP-R) *attains its optimal value and that either* $n+s \leq r$ *or* $|\mathscr{I}| + |\mathscr{E}| \leq \binom{r+2}{2} - \binom{s+1}{2} - 1$. *Then* val(sQMP) *is finite and* val(sQMP) $=$ val(sQMP-R).

Proof. Suppose that problem (sQMP-R) attains its optimal value and that $|\mathscr{I}|+|\mathscr{E}| \leq \binom{r+2}{2} - \binom{s+1}{2} - 1$. Then the number of constraints in (sQMP-R) is $\binom{r+2}{2} - 1$. Hence, by Theorem 5.1, problem (sQMP-R) has a an optimal solution with rank at most r. This solution is therefore feasible and optimal for (sQMP$_2$), and by Lemma 5.4 val(sQMP) = val(sQMP$_2$) = val(sQMP-R).

When $n+s \leq r$, the claim follows immediately from Lemma 5.4 and Remark 5.3.1. □

In particular, note that when $s = r$ the SDP relaxation is tight when the number of constraints is at most r, thus we recover [15, Theorem 3.2].

In the following we need the dual of (sQMP-R) which is given by

$$\text{(sQMP-D)} \qquad \begin{aligned} \max_{\lambda_i, \Phi \in \mathbb{S}^s} \; & -\sum_{i \in \mathscr{I} \cup \mathscr{E}} \lambda_i \alpha_i - \operatorname{tr}(VV^T \Phi) \\ \text{s.t. } & M_0 + \sum_{i \in \mathscr{I} \cup \mathscr{E}} \lambda_i M_i + \begin{pmatrix} 0_{n\times n} & 0_{n\times s} \\ 0_{s\times n} & \Phi \end{pmatrix} \succeq 0, \\ & \Phi \in \mathbb{S}^s, \\ & \lambda_i \geq 0, i \in \mathscr{I}. \end{aligned}$$

From the conic duality theorem [23], if (sQMP-D) is strictly feasible and bounded from above, then (sQMP-R) and (sQMP-D) have the same optimal value. The next claim immediately follows.

Corollary 5.6. *Suppose that* (sQMP-D) *is strictly feasible and bounded from above. Then if either* $n+s \leq r$ *or* $|\mathscr{I}|+|\mathscr{E}| \leq \binom{r+2}{2} - \binom{s+1}{2} - 1$, *we have* val(sQMP) = val(sQMP-D).

A simple condition given in [15, Lemma 3.2] that ensures the strict feasibility and boundedness of (sQMP-D) is the following: there exist numbers $\lambda_i \in \mathbb{R}, i \in \{0\} \cup \mathscr{I} \cup \mathscr{E}$ for which

$$A_0 + \sum_{i \in \mathscr{I} \cup \mathscr{E}} \lambda_i A_i \succ 0 \text{ and } \lambda_i \geq 0 \; \forall i \in \mathscr{I}.$$

5.4 Applications

5.4.1 Robust Least Squares

Consider robust least squares problem (RLS) discussed in Example 5.1.1. Recall that the problem is formulated as:

$$\text{(RLS)} \qquad \begin{aligned} \min_x \max_{\Delta \in \mathbb{R}^{n\times r}} \; & \operatorname{tr}(\Delta^T x x^t \Delta) + 2\operatorname{tr}((b-Ax)x^T\Delta) + \operatorname{tr}((b-Ax)(b-Ax)^T) \\ \text{s.t. } & \operatorname{tr}(\Delta^T L_i^T L_i \Delta) \leq \rho_i, \quad i = 1, \ldots, m. \end{aligned}$$

We begin our analysis by deriving the dual of the inner maximization problem in (RLS). Suppose that $Ax = b$. Then in this case the inner maximization problem in (RLS) is a homogeneous

quadratic problem; performing the standard SDP relaxation technique for homogeneous problems and taking the dual, we reach the following problem

$$\text{(RLS-D}') \quad \begin{aligned} &\min_{\lambda_i,t} \sum_{i=1}^{m} \lambda_i \rho_i \\ &\text{s.t. } -xx^T + \sum_{i=1}^{m} \lambda_i L_i^T L_i \succeq 0, \\ &\qquad \lambda_i \geq 0, \quad i = 1,\ldots,m. \end{aligned}$$

In the case $Ax \neq b$, the inner maximization problem in (RLS) is of the form of problem (sQMP) with $s = 1$ and

$$\begin{aligned} A_0 &= -xx^T, \\ B_0 &= -\|b-Ax\|x, \\ c_0 &= -\|b-Ax\|^2, \\ A_i &= L_i^T L_i, \quad i = 1,\ldots,m, \\ B_i &= 0, \quad i = 1,\ldots,m, \\ c_i &= 0, \quad i = 1,\ldots,m, \\ \alpha_i &= \rho_i, \quad i = 1,\ldots,m, \\ V &= \frac{1}{\|b-Ax\|}(b-Ax)^T. \end{aligned}$$

By taking the dual form (sQMP-D) we get

$$\text{(RLS-D)} \quad \begin{aligned} &\min_{\lambda_i,t} \sum_{i=1}^{m} \lambda_i \rho_i + t \\ &\text{s.t. } \begin{pmatrix} -xx^T + \sum_{i=1}^{m} \lambda_i L_i^T L_i & -\|b-Ax\|x \\ -\|b-Ax\|x^T & -\|b-Ax\|^2 + t \end{pmatrix} \succeq 0 \\ &\qquad \lambda_i \geq 0, \quad i = 1,\ldots,m. \end{aligned}$$

Note that if we set $Ax = b$ in (RLS-D), the optimal value for t becomes 0 and we are left with the problem (RLS-D$'$); hence, (RLS-D) can be used as the dual problem for both cases.

Now, if we further assume that (RLS-D) is strictly feasible and bounded (e.g., when the linear combination $\sum_{i=1}^{m} \lambda_i L_i^T L_i \succ 0$ for some $\lambda_i \geq 0$) and that either $r \geq n+1$ or that the number of constraints satisfies $m \leq \binom{r+2}{2} - 2$, then (RLS-D) and the inner maximization problem in (RLS) have the same optimal solution for every x. Therefore, in order to solve (RLS) it is sufficient to solve the following problem:

$$\text{(RLS}_2\text{)} \quad \begin{aligned} &\min_{x,\lambda_i,t} \sum_{i=1}^{m} \lambda_i \rho_i + t \\ &\text{s.t. } \mathscr{A}(x) = \begin{pmatrix} -xx^T + \sum_{i=1}^{m} \lambda_i L_i^T L_i & -\|b-Ax\|x \\ -\|b-Ax\|x^T & -\|b-Ax\|^2 + t \end{pmatrix} \succeq 0, \\ &\qquad \lambda_i \geq 0, \quad i = 1,\ldots,m. \end{aligned}$$

We will now show that it is possible to rewrite (RLS$_2$) as a standard SDP.

Proposition 5.4.1. The point (x^*, λ^*, t^*) is an optimal solution for (RLS_2) if and only if it is an optimal solution for the following SDP problem:

$$\begin{aligned}
&\min_{x,\lambda_i,t} \sum_{i=1}^{m} \lambda_i \rho_i + t \\
&\text{s.t.} \begin{pmatrix} 1 & x^T & (b-Ax)^T \\ x & \sum_{i=1}^m \lambda_i L_i^T L_i & 0 \\ b-Ax & 0 & tI_r \end{pmatrix} \succeq 0, \\
&\quad \lambda_i \geq 0, \quad i = 1, \ldots, m.
\end{aligned}$$

Proof. The matrix inequality in (RLS_2) is given by

$$\mathscr{A}(x) \succeq 0,$$

which is equivalent to

$$\begin{pmatrix} \mathscr{A}(x) & 0_{n\times(r-1)} \\ 0_{(r-1)\times n} & I_{r-1} \end{pmatrix} \succeq 0.$$

The later inequality can be rewritten as

$$\begin{pmatrix} -xx^T + \sum_{i=1}^m \lambda_i L_i^T L_i & -\|b-Ax\| x e_1^T \\ -\|b-Ax\| e_1 x^T & -\|b-Ax\|^2 e_1 e_1^T + tI_r \end{pmatrix} \succeq 0. \tag{5.4.1}$$

Now, let $Q \in \mathbb{S}^r$ be an orthogonal matrix such that $Qe_1 = \frac{b-Ax}{\|b-Ax\|}$ (when $Ax = b$, one can choose $Q = I_m$), then

$$\begin{aligned}
&\begin{pmatrix} I_n & 0 \\ 0 & Q \end{pmatrix} \begin{pmatrix} -xx^T + \sum_{i=1}^m \lambda_i L_i^T L_i & -xe_1^T \|b-Ax\| \\ -\|b-Ax\| e_1 x^T & -\|b-Ax\|^2 e_1 e_1^T + tI_r \end{pmatrix} \begin{pmatrix} I_n & 0 \\ 0 & Q^T \end{pmatrix} \\
&= \begin{pmatrix} -xx^T + \sum_{i=1}^m \lambda_i L_i^T L_i & -x(b-Ax)^T \\ -(b-Ax)x^T & -(b-Ax)(b-Ax)^T + tI_r \end{pmatrix}.
\end{aligned}$$

Hence, (5.4.1) is equivalent to

$$\begin{pmatrix} -xx^T + \sum_{i=1}^m \lambda_i L_i^T L_i & -x(b-Ax)^T \\ -(b-Ax)x^T & -(b-Ax)(b-Ax)^T + tI_r \end{pmatrix} \succeq 0. \tag{5.4.2}$$

Finally, writing the last constraint in the form

$$\begin{pmatrix} \sum_{i=1}^m \lambda_i L_i^T L_i & 0 \\ 0 & tI_r \end{pmatrix} - \begin{pmatrix} x \\ b-Ax \end{pmatrix} \begin{pmatrix} x \\ b-Ax \end{pmatrix}^T \succeq 0$$

and applying the Schur complement Lemma (see Lemma 5.2), we obtain the desired equivalent SDP formulation. □

Note that the dimension of the matrix constraint is $n+r+1$ instead of $nr+r+1$ in the standard formulation [15]. Thus, assuming strong duality holds, this new formulation can handle much more complex sets of uncertainty, with $\binom{r+2}{2} - 2$ constraints if $r \leq n$ and an arbitrary number of quadratic constraints if $r \geq n+1$.

5.4.2 The Sphere Packing Problem

In the *sphere packing problem* we are interested in determining a feasible configuration of non-overlapping spheres bounded within a given shape. This problem was extensively studied in various settings over the years. See for example [27, 69, 93, 94] and many more.

Consider the problem of finding a packing of n spheres with given radii within the intersection of k balls with known centers and radii in $\mathbb{R}^d$ ($k \leq d+1$). This problem can be formulated as determining whether the following set of constraints is feasible:

$$\begin{aligned} \|X^T e_i - c_j\| &\leq R_j - r_i, \quad i = 1,\dots,n,\ j = 1,\dots,k, \\ \|X^T e_i - X^T e_j\| &\geq r_i + r_j, \quad i,j = 1,\dots,n, \\ X &\in \mathbb{R}^{n\times d}. \end{aligned}$$

where $c_1,\dots,c_k \in \mathbb{R}^d$ are the centers of the containing balls, $R_1,\dots,R_k > 0$ are the respective radii and $r_1,\dots,r_n > 0$ are the radii of the inner spheres. The radii are assumed to satisfy the relation $\min_{j=1,\dots,k} R_j \geq \max_{i=1,\dots,n} r_i$ which is necessary in order to make the problem feasible. The rows of the decision variables matrix X represent the centers of the spheres to be determined.

Since we can assume without loss of generality that $c_1 = 0$, by choosing

$$V = \begin{pmatrix} c_2^T \\ \vdots \\ c_k^T \end{pmatrix} = \sum_{j=2}^{k} e_{j-1} c_j^T$$

and for the first kn constraints taking

$$\begin{aligned} B_{i,1} &= 0_{d\times k-1}, \quad i = 1,\dots,n \\ B_{i,j} &= \underbrace{e_i}_{\in\mathbb{R}^{d\times 1}} \underbrace{e_{j-1}^T}_{\in\mathbb{R}^{1\times(k-1)}}, \quad i = 1,\dots,n,\ j = 2,\dots,k, \end{aligned}$$

it can be readily seen that this problem is of the form $(sQMP)$ discussed above with $s = k-1$ and $kn + \binom{n}{2}$ constraints. According to Theorem 5.5, the SDP relaxation is tight when $kn + \binom{n}{2} \leq \binom{d+2}{2} - \binom{k}{2} - 1$ or when $d \geq n+k-1$. The first condition is equivalent to

$$n \leq -k + \frac{1}{2} + \sqrt{d^2 + 3d + \frac{1}{4}}$$

and since

$$d - k + 1 = -k + \frac{1}{2} + \sqrt{d^2 + d + \frac{1}{4}} < -k + \frac{1}{2} + \sqrt{d^2 + 3d + \frac{1}{4}},$$

it follows that the validity of the second condition implies the validity of the first condition. Thus we have proved the following:

Proposition 5.4.2. The problem of finding the feasibility of packing n spheres in the intersection of k balls in d dimensions can be solved by an SDP when $n \leq d - k + 1$.

Note that the standard homogenization scheme can be applied when $kn + \binom{n}{2} \leq d$, hence for a fixed k only $O(\sqrt{d})$ spheres can be handled this way and thus the technique presented in this chapter provides a major improvement.

5.4.3 A Strong Duality Result for QCQP Problems Over the Complex Domain

Here we consider the following optimization problem over the complex domain

$$(P) \quad \min_{z\in\mathbb{C}^n}\{f_0(z) : f_i(z) \le 0,\ i=1,\dots,m\} \tag{5.4.3}$$

for

$$f_i(z) = z^*Q_iz + \mathbb{R}(u_i^*z) + d_i,$$

where $Q_i \in \mathbb{S}^n$ is a symmetric *real* matrix, $u_i \in \mathbb{R}^n$ and $d_i \in \mathbb{R}$. z^* denotes the conjugate transpose and $\mathbb{R}(\cdot)$ denotes the real part of a complex number.

Lemma 5.7. *Problem (P) is an sQMP with $s = 1$ and $r = 2$.*

Proof. Consider the decomposition $z = x + iy$, where $x, y \in \mathbb{R}^n$. Then

$$\begin{aligned} f_i(z) &= z^*Q_iz + \mathbb{R}(u_i^*z) + d_i \\ &= (x-iy)^TQ_i(x+iy) + \mathbb{R}(u_i^T(x+iy)) + d_i \\ &= x^TQ_ix + y^TQ_iy + u_i^Tx + d_i. \end{aligned}$$

Now, let $e_1 = (1,0)^T \in \mathbb{R}^2$, $e_2 = (0,1)^T \in \mathbb{R}^2$, and set

$$X = xe_1^T + ye_2^T = \begin{pmatrix} x & y \end{pmatrix} \in \mathbb{R}^{n\times 2}.$$

Then

$$\begin{aligned} &\operatorname{tr}(X^TQ_iX) + \operatorname{tr}(e_1u_i^TX) + d_i \\ &= \operatorname{tr}((xe_1^T + ye_2^T)^TQ_i(xe_1^T + ye_2^T)) + \operatorname{tr}(e_1u_i^T(xe_1^T + ye_2^T)) + d_i \\ &= x^TQ_ix + y^TQ_iy + \operatorname{tr}(u_i^Tx) + d_i \\ &= f_i(z), \end{aligned}$$

thus, by taking $A_i = Q_i$, $B_i = u_i$, $c_i = d_i$ and $V = e_1^T$, we get that (P) is an sQMP with the claimed parameters. □

As an immediate result of the previous lemma and the discussion at the end of Section 5.3, we have the following corollary.

Corollary 5.8. *Suppose there exist numbers $\lambda_i \ge 0$, $i = 1,\dots,m$ such that*

$$Q_0 + \sum_{i=1}^m \lambda_iQ_i \succ 0.$$

Then if either $n = 1$ or the number of constraints, m, is at most $\binom{4}{2} - \binom{2}{2} - 1 = 4$, problem (P) admits strong duality.

Note that if we assume that $u_i \in \mathbb{C}^n$, then we can still perform the above analysis, but get an sQMP with $s = 2$ and $r = 2$, hence strong duality is attained when the number of constraints is at most $\binom{4}{2} - \binom{3}{2} - 1 = 2$. This recovers the result by Beck and Eldar for strong duality in nonconvex quadratic optimization with two quadratic constraints [17].

Bibliography

[1] K. Anstreicher, X. Chen, H. Wolkowicz, and Y. Yuan. Strong duality for a trust-region type relaxation of the quadratic assignment problem. *Linear Algebra Appl.*, 301(1-3):121–136, 1999.

[2] K. Anstreicher and H. Wolkowicz. On Lagrangian relaxation of quadratic matrix constraints. *SIAM J. Matrix Anal. Appl.*, 22(1):41–55, 2000.

[3] K. J. Arrow, L. Hurwicz, and H. Uzawa. *Studies in linear and non-linear programming*. With contributions by H. B. Chenery, S. M. Johnson, S. Karlin, T. Marschak, R. M. Solow. Stanford Mathematical Studies in the Social Sciences, vol. II. Stanford University Press, Stanford, Calif., 1958.

[4] H. Attouch, J. Bolte, and P. Redont. Optimizing properties of an inertial dynamical system with geometric damping. Link with proximal methods. *Control Cybernet.*, 31(3):643–657, 2002. Well-posedness in optimization and related topics (Warsaw, 2001).

[5] H. Attouch, X. Goudou, and P. Redont. The heavy ball with friction method. I. The continuous dynamical system: global exploration of the local minima of a real-valued function by asymptotic analysis of a dissipative dynamical system. *Commun. Contemp. Math.*, 2(1):1–34, 2000.

[6] A. Auslender. Méthodes numériques pour la décomposition et la minimisation de fonctions non différentiables. *Numer. Math.*, 18:213–223, 1971.

[7] A. Auslender. *Optimisation: méthodes numériques*. Masson, 1976.

[8] A. Auslender. Numerical methods for nondifferentiable convex optimization. In B. Cornet, V. Nguyen, and J. Vial, editors, *Nonlinear Analysis and Optimization*, volume 30 of *Mathematical Programming Studies*, pages 102–126. Springer Berlin Heidelberg, 1987.

[9] A. Auslender and M. Teboulle. *Asymptotic Cones and Functions in Optimization and Variational Inequalities*. Springer Monographs in Mathematics. New York: Springer, 2003.

[10] A. Auslender and M. Teboulle. Interior gradient and epsilon-subgradient descent methods for constrained convex minimization. *Math. Oper. Res.*, 29(1):1–26, 2004.

[11] A. Auslender and M. Teboulle. Interior projection-like methods for monotone variational inequalities. *Math. Programming*, 104(1):39–68, 2005.

[12] A. Auslender and M. Teboulle. Projected subgradient methods with non-Euclidean distances for non-differentiable convex minimization and variational inequalities. *Math. Programming*, 120(1, Ser. B):27–48, 2009.

[13] A. Barvinok. Problems of distance geometry and convex properties of quadratic maps. *Discrete Comput. Geom.*, 13:189–202, 1995.

[14] A. Barvinok. A remark on the rank of positive semidefinite matrices subject to affine constraints. *Discrete Comput. Geom.*, 25(1):23–31, 2001.

[15] A. Beck. Quadratic matrix programming. *SIAM J. Optim.*, 17:1224–1238, December 2006.

[16] A. Beck, Y. Drori, and M. Teboulle. A new semidefinite programming relaxation scheme for a class of quadratic matrix problems. *Oper. Res. Lett.*, 40(4):298–302, 2012.

[17] A. Beck and Y. C. Eldar. Strong duality in nonconvex quadratic optimization with two quadratic constraints. *SIAM J. Optim.*, 17(3):844–860, 2006.

[18] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.*, 31:167–175, 2003.

[19] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2:183–202, March 2009.

[20] A. Beck and M. Teboulle. Gradient-based algorithms with applications to signal-recovery problems. In *Convex optimization in signal processing and communications*, pages 42–88. Cambridge Univ. Press, Cambridge, 2010.

[21] A. Beck and M. Teboulle. Smoothing and first order methods: a unified framework. *SIAM J. Optim.*, 22(2):557–580, 2012.

[22] A. Beck and M. Teboulle. A fast dual proximal gradient algorithm for convex minimization and applications. *Oper. Res. Lett.*, 42:1–6, 2014.

[23] A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. SIAM, 2001.

[24] A. Ben-Tal and A. Nemirovski. Non-euclidean restricted memory level method for large-scale convex optimization. *Math. Programming*, 102(3):407–456, 2005.

[25] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numer. Math.*, 4(1):238–252, 1962.

[26] D. P. Bertsekas. *Nonlinear Programming*. Belmont MA: Athena Scientific, second edition, 1999.

[27] E. G. Birgin and F. N. C. Sobral. Minimizing the object dimensions in circle and sphere packing problems. *Comput. Oper. Res.*, 35(7):2357–2375, 2008.

[28] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[29] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Mar. 2004.

[30] P. Brucker. An o(n) algorithm for quadratic knapsack problems. *Oper. Res. Lett.*, 3(3):163 – 166, 1984.

[31] M. R. Celis, J. E. Dennis, and R. A. Tapia. A trust region strategy for nonlinear equality constrained optimization. In *Numerical optimization, 1984 (Boulder, Colo., 1984)*, pages 71–82. SIAM, Philadelphia, PA, 1985.

[32] A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vision*, 20:89–97, 2004.

[33] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vision*, 40(1):120–145, 2011.

[34] S. Chandrasekaran, G. H. Golub, M. Gu, and A. H. Sayed. Parameter estimation in the presence of bounded data uncertainties. *SIAM J. Matrix Anal. Appl.*, 19:235–252, January 1998.

[35] G. Chen and M. Teboulle. A proximal-based decomposition method for convex minimization problems. *Math. Programming*, 64(1, Ser. A):81–101, 1994.

[36] X. Chen, Q. Lin, S. Kim, J. G. Carbonell, and E. P. Xing. Smoothing proximal gradient method for general structured sparse regression. *Ann Appl Stat.*, 6(2):719–752, 2012.

[37] E. W. Cheney and A. A. Goldstein. Newton's method for convex programming and tchebycheff approximation. *Numer. Math.*, 1(1):253–268, 1959.

[38] W. de Oliveira and C. Sagastizábal. Bundle methods in the XXIst century: A birds-eye view. *Optimization Online Report*, 4088, 2013.

[39] J. E. Dennis, Jr and J. J. Moré. Quasi-newton methods, motivation and theory. *SIAM review*, 19(1):46–89, 1977.

[40] Y. Drori and M. Teboulle. Performance of first-order methods for smooth convex minimization: a novel approach. *Math. Programming, Series A*, 145:451–482, 2014.

[41] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Programming*, 55(1):293–318, 1992.

[42] L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM J. Matrix Anal. Appl.*, 18(4):1035–1064, 1997.

[43] F. Facchinei and J. S. Pang. *Finite-dimensional variational inequalities and complementarity problems, Vol. II*. Springer Series in Operations Research. Springer-Verlag, New York, 2003.

[44] K. Fan. Minimax theorems. *Proc. Natl. Acad. Sci. USA*, 39(1):42, 1953.

[45] C. Fortin and H. Wolkowicz. The trust region subproblem and semidefinite programming. *Optim. Method. Softw.*, 19(1):41–67, 2004.

[46] D. Gabay. Applications of the method of multipliers to variational inequalities. In M. Fortin and R. Glowinski, editors, *Augmented Lagrangian Methods: Applications to the Solution of Boundary-Valued Problems*, pages 299–331. North-Holland, Amsterdam, 1983, 1983.

[47] C. Gonzaga and E. Karas. Fine tuning Nesterov's steepest descent algorithm for differentiable convex programming. *Math. Program.*, pages 1–26, 2012.

[48] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

[49] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, Mar. 2014.

[50] R. Grone, C. R. Johnson, E. M. Sá, and H. Wolkowicz. Positive definite completions of partial hermitian matrices. *Linear Algebra Appl.*, 58:109–124, 1984.

[51] C. Helmberg, F. Rendl, R. Vanderbei, and H. Wolkowicz. An interior-point method for semidefinite programming. *SIAM J. Optim.*, 6:342–361, 1996.

[52] A. Juditsky and A. Nemirovsky. First order methods for nonsmooth convex large-scale optimization, ii: Utilizing problems structure. In S. Sra, S. Nowozin, and S. J. Wright, editors, *Optimization for Machine Learning*, pages 29–63. The MIT Press, Cambridge, 2012.

[53] J. E. Kelley, Jr. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial & Applied Mathematics*, 8(4):703–712, 1960.

[54] D. Kim and J. A. Fessler. Optimized first-order methods for smooth convex minimization. *arXiv preprint arXiv:1406.5468*, 2014.

[55] K. C. Kiwiel. *Methods of descent for nondifferentiable optimization*, volume 1133. Springer-Verlag Berlin, 1985.

[56] K. C. Kiwiel. Proximity control in bundle methods for convex nondifferentiable minimization. *Math. Programming*, 46(1-3):105–122, 1990.

[57] K. C. Kiwiel. Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities. *Math. Programming*, 69(1-3):89–109, 1995.

[58] K. C. Kiwiel. Efficiency of proximal bundle methods. *J. Optim. Theory Appl.*, 104(3):589–603, 2000.

[59] G. M. Korpelevitch. The extra gradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.

[60] G. Lan, Z. Lu, and R. Monteiro. Primal-dual first-order methods with iteration-complexity for cone programming. *Math. Program.*, 126(1):1–29, 2011.

[61] C. Lemaréchal. An extension of davidon methods to non differentiable problems. In *Nondifferentiable optimization*, pages 95–109. Springer, 1975.

[62] C. Lemaréchal, A. Nemirovskii, and Y. Nesterov. New variants of bundle methods. *Math. Programming*, 69(1-3):111–147, 1995.

[63] C. Lemaréchal and C. Sagastizábal. Variable metric bundle methods: from conceptual to implementable forms. *Math. Programming*, 76(3):393–410, 1997.

[64] P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.*, 16(6):964–979, 1979.

[65] L. Lukšan and J. Vlček. A bundle-newton method for nonsmooth unconstrained minimization. *Math. Programming*, 83(1-3):373–391, 1998.

[66] M. Mäkelä. Survey of bundle methods for nonsmooth optimization. *Optim. Methods Softw.*, 17(1):1–29, 2002.

[67] J. J. Moré. Generalizations of the trust region subproblem. *Optim. Method. Softw.*, 2:189–209, 1993.

[68] J. J. Moreau. Proximité et dualité dans un espace hilbertien. *Bull. Soc. Math. France*, 93:273–299, 1965.

[69] G. E. Mueller. Numerically packing spheres in cylinders. *Powder technol.*, 159(2):105–110, 2005.

[70] A. Nedić and A. Ozdaglar. Subgradient methods for saddle-point problems. *J. Optim. Theory Appl.*, 142(1):205–228, 2009.

[71] A. Nemirovski. Prox-method with rate of convergence O(1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM J. Optim.*, 15(1):229–251, 2004.

[72] A. S. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization*. A Wiley-Interscience Publication. John Wiley & Sons Inc., New York, 1983. Translated from the Russian and with a preface by E. R. Dawson, Wiley-Interscience Series in Discrete Mathematics.

[73] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Math. Dokl.*, 27(2):372–376, 1983.

[74] Y. Nesterov. *Introductory lectures on convex optimization: a basic course*. Applied optimization. Kluwer Academic Publishers, 2004.

[75] Y. Nesterov. Excessive gap technique in nonsmooth convex minimization. *SIAM J. Optim.*, 16(1):235–249, 2005.

[76] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Programming*, 103(1, Ser. A):127–152, 2005.

[77] Y. Nesterov. Dual extrapolation and its applications to solving variational inequalities and related problems. *Math. Programming*, 109(2-3):319–344, 2007.

[78] Y. Nesterov. Gradient methods for minimizing composite objective function. 2007. CORE Report. Available at http://www.ecore.be/DPs/dp_1191313936.pdf.

[79] J. Nocedal and S. J. Wright. *Conjugate gradient methods*. Springer, 2006.

[80] D. P. Palomar and Y. C. Eldar, editors. *Convex Optimization in Signal Processing and Communications*. Cambridge University Press, 2010.

[81] G. B. Passty. Ergodic convergence to a zero of the sum of monotone operators in Hilbert space. *J. Math. Anal. Appl.*, 72(2):383–390, 1979.

[82] G. Pataki. *The Geometry of Semidefinite Programs*, pages 29–65. Springer, 2000.

[83] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Comp. Math. Math. Phys.*, 4(5):1–17, 1964.

[84] H. Raguet, J. Fadili, and G. Peyré. A generalized forward-backward splitting. *SIAM J. Imag. Sci.*, 6(3):1199–1226, 2013.

[85] P. Richtárik. Improved algorithms for convex minimization in relative scale. *SIAM J. Optim.*, 21(3):1141–1167, 2011.

[86] R. T. Rockafellar. *Convex Analysis*. Princeton NJ: Princeton Univ. Press, 1970.

[87] R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control Optim.*, 14(5):877–898, 1976.

[88] R. T. Rockafellar and J. B. W. Roger. *Variational analysis*, volume 317 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1998.

[89] R. T. Rockafellar and J. B. R. Wets. *Variational analysis*. Springer, 2004.

[90] L. I. Rudin, S. J. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60:259–268, 1992.

[91] H. Schramm and J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: Conceptual idea, convergence analysis, numerical results. *SIAM J. Optim.*, 2(1):121–152, 1992.

[92] S. Sra, S. Nowozin, and S. J. Wright, editors. *Optimization for Machine Learning*. MIT Press, Cambridge, MA., 2011.

[93] Y. Stoyan and G. Yaskov. Packing identical spheres into a rectangular parallelepiped. In *Intelligent Decision Support*, pages 47–67. Gabler, 2008.

[94] A. Sutou and Y. Dai. Global optimization approach to unequal sphere packing problems in 3d. *J. Optimiz. Theory App.*, 114:671–694, 2002.

[95] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. C*, pages 267–288, 1996.

[96] R. Tibshirani, M. Saunders, R. Saharon, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *J. Roy. Statist. Soc. Ser. C*, 67(1):pp. 91–108, 2005.

[97] P. Tseng. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM J. Control Optim.*, 29(1):119–138, 1991.

[98] P. Tseng. Alternating projection-proximal methods for convex programming and variational inequalities. *SIAM J. Optim.*, 7(4):951–965, 1997.

[99] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Rev.*, 38(1):49–95, 1996.

[100] P. Wolfe. A method of conjugate subgradients for minimizing nondifferentiable functions. In *Nondifferentiable optimization*, pages 145–173. Springer, 1975.

תקציר

החלק המרכזי בעבודה זו מתמקד בהצגת גישה חדשה לניתוח קצב התכנסות של אלגוריתמי אופטימיזציה לבעיות קמורות. הגישה מבוססת על ההבחנה כי ניתן להציג את החסם על קצב ההתכנסות של אלגוריתם נתון כבעית אופטימיזציה בפני עצמה (אשר מחפשת את "הקלט הרע ביותר" לאלגוריתם) ולפתור בעיה זו בעזרת כלים מתחום האופטימיזציה. בפרק 2, אנו מתמקדים בבעיות אופטימיזציה קמורות וגזירות, ומראים כיצד ניתן ליישם את הגישה הנ"ל על שיטת הגרדיאנט וכתוצאה מכך משיגים חסם חדש והדוק על אלגוריתם זה. בנוסף, אנו מראים כיצד להפעיל את הגישה על מחלקה רחבה של אלגוריתמים הכוללת את שיטת הגרדיאנט המהירה ושיטת הכדור הכבד, ומראים כי במקרים בהם פתרון אנליטי לבעית האופטימיזציה המתקבלת אינו ידוע, ניתן להשתמש בכלים נומריים על מנת לקבל חסם עליון על קצב ההתכנסות של האלגוריתם הנתון. כמו כן, אנו מראים כיצד למצוא באופן נומרי את האלגוריתם הטוב ביותר במחלקת האלגוריתמים המדוברת וכי קצב ההתכנסות של אלגוריתם זה טוב כפליים מחסמים ידועים על קצבי ההתכנסות של אלגוריתמים קלאסיים.

בפרק 3 של העבודה, אנו מרחיבים את הגישה הנ"ל ומראים כיצד ניתן בעזרתה למצוא אלגוריתם חדש לאופטימיזציה קמורה במקרה הלא גזיר. אנו מראים בפירוט את בניית האלגוריתם ומוכיחים כי קצב ההתכנסות שלו אופטימלי על מחלקת הפונקציות הקמורות שמקיימות את תנאי ליפשיץ. באופן מפתיע, האלגוריתם המתקבל דומה לאלגוריתם המישור החותך של קלי.

בפרק 4, אנו מציעים אלגוריתם חדש לפתרון בעיות מיני-מקס בעלות מבנה. האלגוריתם פשוט במיוחד ובעל מספר יתרונות טכניים על אלגוריתמים קיימים לבעיה למשל בכך שאינו דורש מהמשתמש לקבוע את הדיוק המבוקש לפני הפעלת האלגוריתם. אנו מציגים את האלגוריתם, מוכיחים את קצב ההתכנסות שלו ומדגימים את היעילות שלו על מספר בעיות מעשיות. כמו כן, אנו מראים כיצד ניתן ליישם את הגישה החדשה שהוצגה בפרקים הקודמים על מנת לקבל חסם נומרי על קצב ההתכנסות של אלגוריתם זה.

לבסוף, בפרק 5, אנו מתמקדים בבעיות אופטימיזציה לא קמורה של פונקציות ריבועיות עם אילוצים ריבועיים. באמצעות בחינה מעודנת של מבנה הבעיה אנו מציגים תנאים חדשים בהם קיים פתרון יעיל לבעיות אלו באמצעות בעיות תכנות קמור מוגדר אי-שלילי. אנו מדגימים את שימושיות התוצאות, הן במישור התיאורטי והן במישור המעשי, באמצעות מספר דוגמאות.

אוניברסיטת תל-אביב
הפקולטה למדעים מדוייקים ע"ש ריימונד ובברלי סאקלר
בית-הספר למדעי המתמטיקה
המחלקה לסטטיסטיקה וחקר ביצועים

תרומות לניתוח הסיבוכיות של אלגוריתמי אופטימיזציה

הוגש לסנט של אוניברסיטת תל-אביב
לשם קבלת תואר "דוקטור לפילוסופיה"

על-ידי

יואל דרורי

עבודה זו נעשתה בהדרכת פרופ' מרק טבול

דצמבר 2014
טבת תשע"ה