GRADIENT METHODS FOR LARGE-SCALE NONLINEAR OPTIMIZATION

By

HONGCHAO ZHANG

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2006

To my parents

ACKNOWLEDGMENTS

First of all, I would like to express my gratitude to my advisor, Professor William W. Hager. His enthusiasm, encouragement, consistent support, and guidance made my graduate experience fruitful and thoroughly enjoyable. I am grateful to have had the opportunity to study under such a caring, intelligent, and energetic advisor. His confidence in me will always stimulate me to move forward on my research.

Second, I would also like to thank Dr. Panos M. Pardalos, Dr. Shari Moskow, Dr. Sergei S. Pilyugin, and Dr. Jay Gopalakrishnan, for serving on my supervisory committee. Their valuable suggestions have been very helpful to my research.

Third, thanks go to my officemates (Dr. Shu-jen Huang, Beyza, Sukanya), and all colleagues and friends in the Department of Mathematics at the University of Florida; their company alleviated the stress and frustration of this time.

Last, but not least, I wish to express my special thanks to my parents for their immeasurable support and love. Without their support and encouragement, this dissertation could not have been completed successfully.

This work was supported by the National Science Foundation Grant 0203270.

TABLE OF CONTENTS

LIST OF TABLES

Table page

LIST OF FIGURES

KEY TO ABBREVIATIONS

ASA: active set algorithm

CBB: cyclic Barzilai-Borwein

CG: conjugate gradient

LP: linear programming

NCG: nonlinear conjugate gradient

NGPA: nonmonotone gradient projection algorithm

NLP: nonlinear programming

SSOSC: strong second order sufficient condition

KEY TO SYMBOLS

The list shown below gives a brief description of the major mathematical symbols defined in this work. For each symbol, the page number corresponds to the place where the symbol is first used.

: 11

$\mathbb{R}^n$: The space of real n-dimensional vectors 11

k : Integer, often used to denote the iteration number in an algorithm 11

$\mathbf{x}$: $\mathbb{R}^n$ vector of unknown variables 11

$\mathbf{x}_{ki}$: Stands for the i-th component of the iterate $\mathbf{x}_k$ 11

$f(\mathbf{x})$: The objective function: $\mathbb{R}^n \to \mathbb{R}$ 11

f_k : $f_k = f(\mathbf{x}_k)$ 11

$\mathbf{g}(\mathbf{x})$: A collum vector, the transpose of the gradient of objective function at $\mathbf{x}$, i.e. $\mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x})^{\mathsf{T}}$ 11

$\mathbf{g}_k$: $\mathbf{g}_k = \mathbf{g}(\mathbf{x}_k)$ 11

$H(\mathbf{x})$: The hessian of objective function at $\mathbf{x}$, i.e. $H(\mathbf{x}) = \nabla^2 f(\mathbf{x})$ 11

H_k : $H_k = H(\mathbf{x}_k)$ 11

$\|\cdot\|$: The Euclidean norm of a vector 11

$\mathcal{B}(\mathbf{x}, \rho)$: The ball with center $\mathbf{x}$ and radius $\rho > 0$ 11

$|\mathcal{S}|$: Stands for the number of elements (cardinality) of $\mathcal{S}$, for any set $\mathcal{S}$ 11

$\mathcal{S}^c$: The complement of $\mathcal{S}$ 11

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

GRADIENT METHODS FOR LARGE-SCALE NONLINEAR OPTIMIZATION

By

Hongchao Zhang

August 2006

Chair: William W. Hager
Major Department: Mathematics

In this dissertation, we develop theories and efficient algorithmic approaches on gradient methods to solve large-scale nonlinear optimization problems.

The first part of this dissertation discusses new gradient methods and techniques for dealing with large-scale unconstrained optimization. We first propose a new nonlinear CG method (CG_DESCENT), which satisfies the strong descent condition $\mathbf{g_k^{\mathsf{T}} d_k} \leq -\frac{\mathbf{7}}{\mathbf{8}}\|\mathbf{g_k}\|^{\mathbf{2}}$ independent of linesearch. This new CG method is one member of a one parameter family of nonlinear CG method with guaranteed descent. We also develop a new "Approximate Wolfe" linesearch which is both efficient and highly accurate. CG_DESCENT is the first nonlinear CG method which satisfies the sufficient descent condition, independent of linesearch. Moreover, global convergence is established under the standard (not strong) Wolfe conditions. The CG_DESCENT software turns out to be a benchmark software for solving unconstrained optimization. Then, we propose a so-called cyclic Baizilai-Borwein (CBB) method. It is proved that CBB is locally linearly convergent at a local minimizer with positive definite Hessian. Numerical evidence indicates that when $m > n/2 \geq 3$, CBB is locally superlinearly convergent, where m is the cycle length and n is the dimension. However, in the special case $m = 3$ and $n = 2$, we give an example which shows that the convergence rate is in general no better than linear. Combining a nonmonotone line search and an adaptive choice for the cycle length, an implementation of the CBB method, called adaptive CBB (ACBB), is proposed. The adaptive CBB (ACBB) performs much better than the traditional BB methods and is even competitive with some established nonlinear conjugate gradient methods. Finally, we propose a class of self-adaptive proximal point methods suitable for degenerate optimization problems where multiple minimizers may exist, or where the Hessian may be singular at a local minimizer. Two different acceptance criteria for an approximate solution to the proximal problem is analyzed and the convergence rate are analogous to those of exact iterates.

The second part of this dissertation discusses using gradient methods to solve large-scale box constrained optimization. We first discuss the gradient projection methods. Then, an active set algorithm (ASA) for box constrained optimization is developed. The algorithm consists of a nonmonotone gradient projection step, an unconstrained optimization step, and a set of rules for branching between the two steps. Global convergence to a stationary point is established. Under the strong second-order sufficient optimality condition, without assuming strict complementarity, the algorithm eventually reduces to unconstrained optimization without restarts. For strongly convex quadratic box constrained optimization, ASA is shown to have finite convergence when a conjugate gradient method is used in the unconstrained optimization step. A specific implementation of ASA is given, which exploits the cyclic Barzilai-Borwein algorithm for the gradient projection step and CG_DESCENT for unconstrained optimization. Numerical experiments using the box constrained problems in the CUTEr and MINPACK test problem libraries show that this new algorithm outperforms benchmark softwares such as GENCAN, L-BFGS-B, and TRON.

CHAPTER 1
INTRODUCTION

1.1 Motivation

Although computational optimization can be dated back to the maximum value problems, it became one branch of computational science only after appearance of the simplex method proposed by Dantzig in the 1940s for linear programming. Loosely speaking, optimization method seeks to answer the question "What is best?" for problems in which the quality of any answer can be expressed as a numerical value. Now as computer power increases so much, it makes possible for researchers to tackle really large nonlinear problems in many practical applications. Such problems arise in all areas of mathematics, the physical, chemical and biological sciences, engineering, architecture, economics, and management. However, to take advantage of these powers, good algorithms must also be developed. So developing theories and efficient methods to solve large-scale optimization problems is a very important and active research area. This is essentially the goal of this dissertation.

Throughout the dissertation, the nonlinear program (NLP) that we are trying to solve has the following general formulations:

$$\min\{f(\mathbf{x}) : \mathbf{x} \in \mathcal{S}\}, \tag{1.1}$$

where f is a real-valued, continuous function defined on a nonempty set $\mathcal{S} \subset \mathbb{R}^n$. f is often called the objective function and $\mathcal{S}$ is called the feasible set. In Chapter 2, we consider the case where $\mathcal{S} = \mathbb{R}^n$, i.e., the unconstrained optimization problem; while in Chapter 3, we study the case where $\mathcal{S}$ is a box set defined on $\mathbb{R}^n$, i.e., $\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}$ and $\mathbf{l} < \mathbf{u}$ are vectors in $\mathbb{R}^n$. For general nonlinear optimization, people often consider $\mathcal{S}$ is defined by a finite sequence of equality and inequality constraints. More specifically, problem (1.1) can be reformulated as the following:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f(\mathbf{x}) \\ s.t. \quad & c_i(\mathbf{x}) = 0, \qquad i = 1, 2, \ldots, m_e; \\ & c_i(\mathbf{x}) \geq 0, \qquad i = m_e + 1, \ldots, m, \end{aligned}$$

where $c_i : \mathbb{R}^n \to \mathbb{R}, \quad i = 1, 2, \cdots, m$ are smooth functions and at least one of them is nonlinear. We often denote $E = \{1, 2, \ldots, m_e\}$, $I = \{m_e + 1, \ldots, m\}$ and $I(\mathbf{x}) = \{i | c_i(\mathbf{x}) \leq 0, i \in I\}$.

1.2 Optimality Conditions

First, we give the concepts of global minimum and local minimum.

Definition 1. Given $\mathbf{x}^* \in \mathcal{S}$, if $f(\mathbf{x}) \geq f(\mathbf{x}^*)$ for all $\mathbf{x} \in \mathcal{S}$, then $\mathbf{x}^*$ is called a global minimum of the problem (1.1). If $f(\mathbf{x}) > f(\mathbf{x}^*)$ for all $\mathbf{x} \in \mathcal{S}$ and $\mathbf{x} \neq \mathbf{x}^*$, then $\mathbf{x}^*$ is called a strictly global minimum of the problem (1.1).

Definition 2. Given $\mathbf{x}^* \in \mathcal{S}$, if there exists a $\delta > 0$ such that $f(\mathbf{x}) \geq f(\mathbf{x}^*)$ for all $\mathbf{x} \in \mathcal{S} \cap \mathcal{B}(\mathbf{x}^*, \delta)$, then $\mathbf{x}^*$ is called a local minimum of the problem (1.1). If $f(\mathbf{x}) > f(\mathbf{x}^*)$ for all $\mathbf{x} \in \mathcal{S} \cap \mathcal{B}(\mathbf{x}^*, \delta)$ and $\mathbf{x} \neq \mathbf{x}^*$, then $\mathbf{x}^*$ is called a strictly local minimum of the problem (1.1).

When f is a convex function, we know (strictly) local minimum is also a (strictly) global minimum. However, it is hard in advance to know whether the objective function is convex or not and in many cases it is very hard or even impossible to find a global minimum on a feasible region. So in the context of nonlinear optimization it is often found that a local minimum is a solution of problem (1.1), and many algorithms are trying to find a feasible point which satisfies some necessary conditions of a local minimum. In the following, we list some of these necessary conditions.

Theorem 1 *(First order necessary condition for unconstrained optimization) Suppose $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function. If $\mathbf{x}^*$ is a local minimum of problem (1.1), then*

$$\nabla \mathbf{g}(\mathbf{x}^*) = \mathbf{0}.$$

Theorem 2 *(Second order necessary condition for unconstrained optimization) Suppose $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ is a twice continuously differentiable function. If $\mathbf{x}^*$ is a local minimum of problem (1.1), then*

$$\nabla \mathbf{g}(\mathbf{x}^*) = \mathbf{0} \quad and \quad H(\mathbf{x}^*) \geq 0.$$

Because for the general nonlinear optimization, the necessary conditions become more complicated and have many variants, we only state one first order first order necessary condition here which is most often used in practice.

Theorem 3 *(First order necessary condition for constrained optimization) Suppose $f(\mathbf{x})$ and $c_i(\mathbf{x})(i = 1, \cdots, m)$ in problem (1.2) are continuously differentiable functions. If $\mathbf{x}^*$ is a local minimum of problem (1.2) and $\nabla c_i(\mathbf{x}^*)(i \in E \cup I(\mathbf{x}^*))$ are linearly independent, then there exists $\lambda_i^*(i = 1, \cdots, m)$ such that*

$$\begin{aligned} &\nabla f(\mathbf{x}^*) = \sum_{i=1}^{m} \lambda_i^* \nabla c_i(\mathbf{x}^*) \\ &\lambda_i^* \geq 0, \quad \lambda_i^* c_i(\mathbf{x}^*) = 0, \quad i \in I. \end{aligned}$$

For the proof of these theorems and many other necessary optimality conditions, one may refer to R. Fletcher's books [55, 56].

In this dissertation, we mainly focus on developing gradient methods to generate iteration points which satisfy some first order condition for large-scale unconstrained and box constrained optimization.

CHAPTER 2
UNCONSTRAINED OPTIMIZATION

In this chapter, we consider to solve (1.1) with $\mathcal{S} = \mathbb{R}^n$, i.e. the following unconstrained optimization problem:

$$\min \{f(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\}, \tag{2.1}$$

where f is a real-valued, continuous function.

2.1 A New Conjugate Gradient Method with Guaranteed Descent

2.1.1 Introduction to Nonlinear Conjugate Gradient Method

Conjugate gradient (CG) methods comprise a class of unconstrained optimization algorithms which are characterized by low memory requirements and strong local and global convergence properties. CG history, surveyed by Golub and O'leary [65], begins with research of Cornelius Lanczos and Magnus Hestenes and others (Forsythe, Motzkin, Rosser, Stein) at the Institute for Numerical Analysis (National Applied Mathematics Laboratories of the United States National Bureau of Standards in Los Angeles), and with independent research of Eduard Stiefel at Eidg. Technische Hochschule Zürich. In the seminal 1952 paper [81] of Hestenes and Stiefel, the algorithm is presented as an approach to solve symmetric, positive-definite linear systems.

When applied to the nonlinear problem (2.1), a nonlinear conjugate gradient method generates a sequence $\mathbf{x}_k$, $k \geq 1$, starting from an initial guess $\mathbf{x}_0 \in \mathbb{R}^n$, using the recurrence

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k, \tag{2.2}$$

Table 2–1: Various choices for the CG update parameter

$\beta_k^{HS} = \dfrac{\mathbf{g}_{k+1}^\mathsf{T}\mathbf{y}_k}{\mathbf{d}_k^\mathsf{T}\mathbf{y}_k}$	(1952)	in the original (linear) CG paper of Hestenes and Stiefel [81]
$\beta_k^{FR} = \dfrac{\lVert\mathbf{g}_{k+1}\rVert^2}{\lVert\mathbf{g}_k\rVert^2}$	(1964)	first nonlinear CG method, proposed by Fletcher and Reeves [57]
$\beta_k^{D} = \dfrac{\mathbf{g}_{k+1}^\mathsf{T}\nabla^2 f(\mathbf{x}_k)\mathbf{d}_k}{\mathbf{d}_k^\mathsf{T}\nabla^2 f(\mathbf{x}_k)\mathbf{d}_k}$	(1967)	proposed by Daniel [40], requires evaluation of the Hessian $\nabla^2 f(\mathbf{x})$
$\beta_k^{PRP} = \dfrac{\mathbf{g}_{k+1}^\mathsf{T}\mathbf{y}_k}{\lVert\mathbf{g}_k\rVert^2}$	(1969)	proposed by Polak and Ribière [106] and by Polyak [107]
$\beta_k^{CD} = \dfrac{\lVert\mathbf{g}_{k+1}\rVert^2}{-\mathbf{d}_k^\mathsf{T}\mathbf{g}_k}$	(1987)	proposed by Fletcher [55], CD stands for “Conjugate Descent”
$\beta_k^{LS} = \dfrac{\mathbf{g}_{k+1}^\mathsf{T}\mathbf{y}_k}{-\mathbf{d}_k^\mathsf{T}\mathbf{g}_k}$	(1991)	proposed by Liu and Storey [93]
$\beta_k^{DY} = \dfrac{\lVert\mathbf{g}_{k+1}\rVert^2}{\mathbf{d}_k^\mathsf{T}\mathbf{y}_k}$	(1999)	proposed by Dai and Yuan [35]
$\beta_k^{N} = \left(\mathbf{y}_k - 2\mathbf{d}_k\dfrac{\lVert\mathbf{y}_k\rVert^2}{\mathbf{d}_k^\mathsf{T}\mathbf{y}_k}\right)^\mathsf{T}\dfrac{\mathbf{g}_{k+1}}{\mathbf{d}_k^\mathsf{T}\mathbf{y}_k}$	(2005)	proposed by Hager and Zhang [73]

where the positive step size α_k is obtained by a line search, and the directions $\mathbf{d}_k$ are generated by the rule:

$$\mathbf{d}_{k+1} \quad = \quad -\mathbf{g}_{k+1} + \beta_k\mathbf{d}_k, \quad \mathbf{d}_0 = -\mathbf{g}_0. \tag{2.3}$$

Table 2–1 provides a chronological list of some choices for the CG update parameter. The 1964 formula of Fletcher and Reeves is usually considered the first nonlinear CG algorithm since their paper [57] focuses on nonlinear optimization, while the 1952 paper [81] of Hestenes and Stiefel focuses on symmetric, positive-definite linear systems. Daniel’s choice for the update parameter, which is fundamentally different from the other choices, is not discussed in this dissertation. For large-scale problems, choices for the update parameter that do not require the evaluation of the Hessian matrix are often preferred in practice over methods that require the Hessian in each iteration. In the remaining methods of Table 2–1, except for the new method at the end, the numerator of the update parameter β_k is either $\|\mathbf{g}_{k+1}\|^2$ or $\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{y}_k$ and the denominator is either $\|\mathbf{g}_k\|^2$ or $\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k$ or $-\mathbf{d}_k^{\mathsf{T}}\mathbf{g}_k$. The 2 possible choices for the numerator and the 3 possible choices for the denominator lead to 6 different choices for β_k shown in Table 2–1.

If f is a strongly convex quadratic, then in theory, all 8 choices for the update parameter in Table 2–1 are equivalent with an exact line search. However, for non-quadratic cost functions, each choice for the update parameter leads to quite different performance under inexact line searches.

2.1.2 CG_DESCENT

The method corresponding to the last parameter β_k^N in Table 2–1 is a recently developed NCG method [73] with guaranteed descent, named CG_DESCENT. It has close connections to memoryless quasi-Newton scheme of Perry [105] and Shanno [115]. To prove the global convergence for a general nonlinear function, similar to the approach [60, 79, 121] taken for the Polak-Ribière-Polyak [106, 107] version of the conjugate gradient method, we restrict the lower value of β_k^N. In our restricted scheme, unlike the Polak-Ribière-Polyak method, we dynamically adjust the lower bound on β_k^N in order to make the lower bound smaller as the iterates converge:

$$\begin{aligned} \mathbf{d}_{k+1} &= -\mathbf{g}_{k+1} + \bar{\beta}_k^N \mathbf{d}_k, \quad \mathbf{d}_0 = -\mathbf{g}_0, && (2.4) \\ \bar{\beta}_k^N &= \max\left\{\beta_k^N, \eta_k\right\}, \quad \eta_k = \frac{-1}{\|\mathbf{d}_k\| \min\{\eta, \|\mathbf{g}_k\|\}}, && (2.5) \end{aligned}$$

where $\eta > 0$ is a constant; we took $\eta = .01$ in the experiments

With conjugate gradient methods, the line search typically requires sufficient accuracy to ensure that the search directions yield descent. Moreover, it has been shown [36] that for the Fletcher-Reeves [57] and the Polak-Ribière-Polyak [106, 107] conjugate gradient methods, a line search that satisfies the strong Wolfe conditions may not yield a direction of descent, for a suitable choice of the Wolfe line search parameters, even for the function $f(\mathbf{x}) = \lambda\|\mathbf{x}\|^2$, where $\lambda > 0$ is a constant. An attractive feature of the new conjugate gradient scheme, which we now establish, is that the search directions always yield descent when $\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k \neq 0$, a condition which is satisfied when f is strongly convex, or the line search satisfies the Wolfe conditions.

Theorem 4 *If* $\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k \neq 0$ *and*

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \tau\mathbf{d}_k, \quad \mathbf{d}_0 = -\mathbf{g}_0, \tag{2.6}$$

for any $\tau \in [\beta_k^N, \max\{\beta_k^N, 0\}]$, *then*

$$\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k \leq -\frac{7}{8}\|\mathbf{g}_k\|^2. \tag{2.7}$$

Proof. Since $\mathbf{d}_0 = -\mathbf{g}_0$, we have $\mathbf{g}_0^{\mathsf{T}}\mathbf{d}_0 = -\|\mathbf{g}_0\|^2$, which satisfies (2.7). Suppose $\tau = \beta_k^N$. Multiplying (2.6) by $\mathbf{g}_{k+1}^{\mathsf{T}}$, we have

$$\begin{aligned}
\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_{k+1} &= -\|\mathbf{g}_{k+1}\|^2 + \beta_k^N \mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k \\
&= -\|\mathbf{g}_{k+1}\|^2 + \mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k \left(\frac{\mathbf{y}_k^{\mathsf{T}}\mathbf{g}_{k+1}}{\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k} - 2\frac{\|\mathbf{y}_k\|^2\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k}{(\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k)^2} \right) \\
&= \frac{\mathbf{y}_k^{\mathsf{T}}\mathbf{g}_{k+1}(\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k)(\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k) - \|\mathbf{g}_{k+1}\|^2(\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k)^2 - 2\|\mathbf{y}_k\|^2(\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k)^2}{(\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k)^2}
\end{aligned} \tag{2.8}$$

We apply the inequality

$$\mathbf{u}^{\mathsf{T}}\mathbf{v} \leq \frac{1}{2}(\|\mathbf{u}\|^2 + \|\mathbf{v}\|^2)$$

to the first term in (2.8) with

$$\mathbf{u} = \frac{1}{2}(\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k)\mathbf{g}_{k+1} \quad \text{and} \quad \mathbf{v} = 2(\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k)\mathbf{y}_k$$

to obtain (2.7). On the other hand, if $\tau \neq \beta_k^N$, then $\beta_k^N \leq \tau \leq 0$. After multiplying (2.6) by $\mathbf{g}_{k+1}^{\mathsf{T}}$, we have

$$\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_{k+1} = -\|\mathbf{g}_{k+1}\|^2 + \tau\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k.$$

If $\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k \geq 0$, then (2.7) follows immediately since $\tau \leq 0$. If $\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k < 0$, then

$$\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_{k+1} = -\|\mathbf{g}_{k+1}\|^2 + \tau\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k \leq -\|\mathbf{g}_{k+1}\|^2 + \beta_k^N\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k$$

since $\beta_k^N \leq \tau \leq 0$. Hence, (2.7) follows by our previous analysis. □

By taking $\tau = \beta_k^N$, we see that the directions generated by (2.2)–(2.3) are descent directions. Since η_k in (2.5) is negative, it follows that

$$\bar{\beta}_k^N = \max\left\{\beta_k^N, \eta_k\right\} \in [\beta_k^N, \max\{\beta_k^N, 0\}].$$

Hence, the direction given by (2.4) and (2.5) is a descent direction. Dai and Yuan [35, 37] present conjugate gradient schemes with the property that $\mathbf{d}_k^{\mathsf{T}}\mathbf{g}_k < 0$ when $\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k > 0$. If f is strongly convex or the line search satisfies the Wolfe conditions, then $\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k > 0$ and the Dai/Yuan schemes yield descent. Note that in (2.7) we bound $\mathbf{d}_k^{\mathsf{T}}\mathbf{g}_k$ by $-(7/8)||\mathbf{g}_k||^2$, while for the schemes [35, 37], the negativity of $\mathbf{d}_k^{\mathsf{T}}\mathbf{g}_k$ is established.

2.1.3 Global Convergence

Convergence analysis for strongly convex functions. Although the search directions generated by either (2.2)–(2.3) with $\beta_k = \beta_k^N$ or (2.4)–(2.5) are always descent directions, we need to constrain the choice of α_k to ensure convergence. We consider line searches that satisfy either Goldstein's conditions [64]:

$$\delta_1\alpha_k\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k \leq f(\mathbf{x}_k + \alpha_k\mathbf{d}_k) - f(\mathbf{x}_k) \leq \delta_2\alpha_k\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k, \tag{2.9}$$

where $0 < \delta_2 < \frac{1}{2} < \delta_1 < 1$ and $\alpha_k > 0$, or the Wolfe conditions [122, 123]:

$$\begin{aligned} f(\mathbf{x}_k + \alpha_k\mathbf{d}_k) - f(\mathbf{x}_k) &\leq \delta\alpha_k\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k, & (2.10)\\ \mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k &\geq \sigma\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k, & (2.11)\end{aligned}$$

where $0 < \delta \leq \sigma < 1$. As in in Dai and Yuan [35], we do not require the "strong Wolfe" condition $|\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k| \leq -\sigma \mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k$, which is often used to prove convergence of nonlinear conjugate gradient methods.

Lemma 1 *Suppose that $\mathbf{d}_k$ is a descent direction and ∇f satisfies the Lipschitz condition*

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}_k)\| \leq L\|\mathbf{x} - \mathbf{x}_k\|$$

for all $\mathbf{x}$ on the line segment connecting $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$, where L is a constant. If the line search satisfies the Goldstein conditions, then

$$\alpha_k \geq \frac{(1-\delta_1)}{L}\frac{|\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k|}{\|\mathbf{d}_k\|^2}. \tag{2.12}$$

If the line search satisfies the Wolfe conditions, then

$$\alpha_k \geq \frac{1-\sigma}{L}\frac{|\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k|}{\|\mathbf{d}_k\|^2}. \tag{2.13}$$

Proof. The proof is standard and we omit its proof here (for example similar proofs can be found [73, 131]). □

We now prove convergence of the unrestricted scheme (2.2)–(2.3) with $\beta_k = \beta_k^N$ when f is strongly convex.

Theorem 5 *Suppose that f is strongly convex and Lipschitz continuous on the level set*

$$\mathcal{L} = \{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) \leq f(\mathbf{x}_0)\}. \tag{2.14}$$

That is, there exists constants L and $\mu > 0$ such that

$$\begin{aligned} \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| &\leq L\|\mathbf{x} - \mathbf{y}\| \text{ and} \\ \mu\|\mathbf{x} - \mathbf{y}\|^2 &\leq (\nabla f(\mathbf{x}) - \nabla f(\mathbf{y}))(\mathbf{x} - \mathbf{y}) \end{aligned} \tag{2.15}$$

for all $\mathbf{x}$ and $\mathbf{y} \in \mathcal{L}$. If the conjugate gradient method (2.2)–(2.3) is implemented using a line search that satisfies either the Wolfe or the Goldstein conditions in *each step, then either* $\mathbf{g}_k = \mathbf{0}$ *for some* k*, or*

$$\lim_{k\to\infty} \; \mathbf{g}_k = \mathbf{0}. \tag{2.16}$$

Proof. Suppose that $\mathbf{g}_k \neq \mathbf{0}$ for all k. By the strong convexity assumption,

$$\mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k = (\mathbf{g}_{k+1} - \mathbf{g}_k)^{\mathsf{T}}\mathbf{d}_k \geq \mu\alpha_k\|\mathbf{d}_k\|^2. \tag{2.17}$$

Theorem 4 and the assumption $\mathbf{g}_k \neq \mathbf{0}$ imply that $\mathbf{d}_k \neq \mathbf{0}$. Since $\alpha_k > 0$, it follows from (2.17) that $\mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k > 0$. Since f is strongly convex over $\mathcal{L}$, f is bounded from below. After summing over k the upper bound in (2.9) or (2.10), we conclude that

$$\sum_{k=0}^{\infty} \alpha_k\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k > -\infty.$$

Combining this with the lower bound for α_k given in Lemma 1 and the descent property (2.7) gives

$$\sum_{k=0}^{\infty} \frac{\|\mathbf{g}_k\|^4}{\|\mathbf{d}_k\|^2} < \infty. \tag{2.18}$$

By Lipschitz continuity (2.15),

$$\|\mathbf{y}_k\| = \|\mathbf{g}_{k+1} - \mathbf{g}_k\| = \|\nabla f(\mathbf{x}_k + \alpha_k\mathbf{d}_k) - \nabla f(\mathbf{x}_k)\| \leq L\alpha_k\|\mathbf{d}_k\|. \tag{2.19}$$

Utilizing (2.17) and (2.3), we have

$$\begin{aligned} |\beta_k^N| &= \left|\frac{\mathbf{y}_k^{\mathsf{T}}\mathbf{g}_{k+1}}{\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k} - 2\frac{\|\mathbf{y}_k\|^2\mathbf{d}_k^{\mathsf{T}}\mathbf{g}_{k+1}}{(\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k)^2}\right| \\ &\leq \frac{\|\mathbf{y}_k\|\|\mathbf{g}_{k+1}\|}{\mu\alpha_k\|\mathbf{d}_k\|^2} + 2\frac{\|\mathbf{y}_k\|^2\|\mathbf{d}_k\|\|\mathbf{g}_{k+1}\|}{\mu^2\alpha_k^2\|\mathbf{d}_k\|^4} \\ &\leq \frac{L\alpha_k\|\mathbf{d}_k\|\|\mathbf{g}_{k+1}\|}{\mu\alpha_k\|\mathbf{d}_k\|^2} + 2\frac{L^2\alpha_k^2\|\mathbf{d}_k\|^3\|\mathbf{g}_{k+1}\|}{\mu^2\alpha_k^2\|\mathbf{d}_k\|^4} \\ &\leq \left(\frac{L}{\mu} + \frac{2L^2}{\mu^2}\right)\frac{\|\mathbf{g}_{k+1}\|}{\|\mathbf{d}_k\|}. \end{aligned} \tag{2.20}$$

Hence, we have

$$\|\mathbf{d}_{k+1}\| \leq \|\mathbf{g}_{k+1}\| + |\beta_k^N|\|\mathbf{d}_k\| \leq \left(1 + \frac{L}{\mu} + \frac{2L^2}{\mu^2}\right)\|\mathbf{g}_{k+1}\|.$$

Inserting this upper bound for $\mathbf{d}_k$ in (2.18) yields

$$\sum_{k=1}^{\infty} \|\mathbf{g}_k\|^2 < \infty,$$

which completes the proof. □

We now observe that the directions generated by the new conjugate gradient update (2.2) approximately point along the Perry/Shanno direction,

$$\mathbf{d}_{k+1}^{PS} = \frac{\mathbf{y}_k^{\mathsf{T}}\mathbf{s}_k}{\|\mathbf{y}_k\|^2}\left(\mathbf{d}_{k+1} + \frac{\mathbf{d}_k^{\mathsf{T}}\mathbf{g}_{k+1}}{\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k}\mathbf{y}_k\right), \tag{2.21}$$

where $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, when f is strongly convex and the cosine of the angle between $\mathbf{d}_k$ and $\mathbf{g}_{k+1}$ is sufficiently small. By (2.17) and (2.19), we have

$$\frac{|\mathbf{d}_k^{\mathsf{T}}\mathbf{g}_{k+1}|}{|\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k|}\|\mathbf{y}_k\| \le \frac{L}{\mu}|\mathbf{u}_k^{\mathsf{T}}\mathbf{g}_{k+1}| = c_1\epsilon\|\mathbf{g}_{k+1}\|, \tag{2.22}$$

where $\mathbf{u}_k = \mathbf{d}_k/\|\mathbf{d}_k\|$ is the unit vector in the direction $\mathbf{d}_k$, ϵ is the cosine of the angle between $\mathbf{d}_k$ and $\mathbf{g}_{k+1}$, and $c_1 = L/\mu$. By the definition of $\mathbf{d}_{k+1}$ in (2.2), we have

$$\|\mathbf{d}_{k+1}\|^2 \ge \|\mathbf{g}_{k+1}\|^2 - 2\beta_k^N\mathbf{d}_k^{\mathsf{T}}\mathbf{g}_{k+1}. \tag{2.23}$$

By the bound for β_k^N in (2.20),

$$|\beta_k^N\mathbf{d}_k^{\mathsf{T}}\mathbf{g}_{k+1}| \le c_2|\mathbf{u}_k^{\mathsf{T}}\mathbf{g}_{k+1}|\|\mathbf{g}_{k+1}\| = c_2\epsilon\|\mathbf{g}_{k+1}\|^2, \tag{2.24}$$

where c_2 is the constant appearing in (2.20). Combining (2.23) and (2.24), we have

$$\|\mathbf{d}_{k+1}\| \ge \sqrt{1-2c_2\epsilon}\|\mathbf{g}_{k+1}\|.$$

This lower bound for $\|\mathbf{d}_{k+1}\|$ and the upper bound (2.22) for the $\mathbf{y}_k$ term in (2.21) imply that the ratio between them is bounded by $c_1\epsilon/\sqrt{1-2c_2\epsilon}$. As a result, when ϵ is small, the direction generated by (2.2) is approximately a multiple of the Perry/Shanno direction (2.21).

Convergence analysis for general nonlinear functions. Our analysis of (2.4)–(2.5) for general nonlinear functions exploits insights developed by Gilbert and Nocedal in their analysis [60] of the PRP+ scheme. Similar to the approach taken in [60], we establish a bound for the change $\mathbf{u}_{k+1} - \mathbf{u}_k$ in the normalized direction $\mathbf{u}_k = \mathbf{d}_k/\|\mathbf{d}_k\|$, which we use to conclude, by contradiction, that the gradients cannot be bounded away from zero. The following theorem is the analogue of [60, Lemma 4.1], it differs in the treatment of the direction update formula (2.4).

Lemma 2 *If the level set (2.14) is bounded and the Lipschitz condition (2.15) holds, then for the scheme (2.4)–(2.5) and a line search that satisfies the Wolfe conditions (2.10)–(2.11), we have*

$$\mathbf{d}_k \neq \mathbf{0} \quad \textit{for each } k \textit{ and} \quad \sum_{k=0}^{\infty} \|\mathbf{u}_{k+1} - \mathbf{u}_k\|^2 < \infty$$

whenever $\inf\{\|\mathbf{g}_k\| : k \geq 0\} > 0$.

Proof. Define $\gamma = \inf\{\|\mathbf{g}_k\| : k \geq 0\}$. Since $\gamma > 0$ by assumption, it follows from the descent property Theorem 4 that $\mathbf{d}_k \neq \mathbf{0}$ for each k. Since $\mathcal{L}$ is bounded, f is bounded from below, and by (2.10) and (2.13),

$$\sum_{k=0}^{\infty} \frac{(\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k)^2}{\|\mathbf{d}_k\|^2} < \infty.$$

Again, the descent property yields

$$\gamma^4 \sum_{k=0}^{\infty} \frac{1}{\|\mathbf{d}_k\|^2} \leq \sum_{k=0}^{\infty} \frac{\|\mathbf{g}_k\|^4}{\|\mathbf{d}_k\|^2} \leq \frac{64}{49} \sum_{k=0}^{\infty} \frac{(\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k)^2}{\|\mathbf{d}_k\|^2} < \infty. \tag{2.25}$$

Define the quantities:

$$\beta_k^+ = \max\{\bar{\beta}_k^N, 0\}, \quad \beta_k^- = \min\{\bar{\beta}_k^N, 0\}, \quad \mathbf{r}_k = \frac{-\mathbf{g}_k + \beta_{k-1}^- \mathbf{d}_{k-1}}{\|\mathbf{d}_k\|}, \quad \delta_k = \beta_{k-1}^+ \frac{\|\mathbf{d}_{k-1}\|}{\|\mathbf{d}_k\|}.$$

By (2.4)–(2.5), we have

$$\mathbf{u}_k = \frac{\mathbf{d}_k}{\|\mathbf{d}_k\|} = \frac{-\mathbf{g}_k + (\beta_{k-1}^+ + \beta_{k-1}^-)\mathbf{d}_{k-1}}{\|\mathbf{d}_k\|} = \mathbf{r}_k + \delta_k \mathbf{u}_{k-1}.$$

Since the $\mathbf{u}_k$ are unit vectors,

$$\|\mathbf{r}_k\| = \|\mathbf{u}_k - \delta_k \mathbf{u}_{k-1}\| = \|\delta_k \mathbf{u}_k - \mathbf{u}_{k-1}\|.$$

Since $\delta_k > 0$, it follows that

$$\begin{aligned} \|\mathbf{u}_k - \mathbf{u}_{k-1}\| &\leq \|(1+\delta_k)(\mathbf{u}_k - \mathbf{u}_{k-1})\| \\ &\leq \|\mathbf{u}_k - \delta_k \mathbf{u}_{k-1}\| + \|\delta_k \mathbf{u}_k - \mathbf{u}_{k-1}\| \\ &= 2\|\mathbf{r}_k\|. \end{aligned} \tag{2.26}$$

By the definition of β_k^- and the fact that $\eta_k < 0$ and $\bar{\beta}_k^N \geq \eta_k$ in (2.5), we have the following bound for the numerator of $\mathbf{r}_k$:

$$\begin{aligned} \|-\mathbf{g}_k + \beta_{k-1}^- \mathbf{d}_{k-1}\| &\leq \|\mathbf{g}_k\| - \min\{\bar{\beta}_{k-1}^N, 0\}\|\mathbf{d}_{k-1}\| \\ &\leq \|\mathbf{g}_k\| - \eta_{k-1}\|\mathbf{d}_{k-1}\| \\ &\leq \|\mathbf{g}_k\| + \frac{1}{\|\mathbf{d}_{k-1}\| \min\{\eta, \gamma\}}\|\mathbf{d}_{k-1}\| \\ &\leq \Gamma + \frac{1}{\min\{\eta, \gamma\}}, \end{aligned} \tag{2.27}$$

where

$$\Gamma = \max_{\mathbf{x} \in \mathcal{L}} \|\nabla f(\mathbf{x})\|. \tag{2.28}$$

Let c denote the expression $\Gamma + 1/\min\{\eta, \gamma\}$ in (2.27). This bound for the numerator of $\mathbf{r}_k$ coupled with (2.26) gives

$$\|\mathbf{u}_k - \mathbf{u}_{k-1}\| \leq 2\|\mathbf{r}_k\| \leq \frac{2c}{\|\mathbf{d}_k\|}. \tag{2.29}$$

Finally, squaring (2.29), summing over k, and utilizing (2.25), the proof is complete. □

Theorem 6 *If the level set (2.14) is bounded and the Lipschitz condition (2.15) holds, then for the scheme (2.4)–(2.5) and a line search that satisfies the Wolfe* *conditions (2.10)–(2.11), either* $\mathbf{g}_k = \mathbf{0}$ *for some* k *or*

$$\liminf_{k\to\infty} \|\mathbf{g}_k\| = 0. \tag{2.30}$$

Proof. We suppose that $\mathbf{g}_k \neq \mathbf{0}$ for all k, and $\liminf_{k\to\infty} \|\mathbf{g}_k\| > 0$, and we obtain a contradiction. Defining $\gamma = \inf\, \{\|\mathbf{g}_k\| : k \geq 0\}$, we have $\gamma > 0$ due to (2.30) and the fact that $\mathbf{g}_k \neq \mathbf{0}$ for all k. The proof is divided into 3 steps.

I. *A bound for* $\bar{\beta}_k^N$:

By the Wolfe condition $\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k \geq \sigma \mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k$, we have

$$\mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k = (\mathbf{g}_{k+1} - \mathbf{g}_k)^{\mathsf{T}}\mathbf{d}_k \geq (\sigma - 1)\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k = -(1-\sigma)\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k. \tag{2.31}$$

By Theorem 4,

$$-\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k \geq \frac{7}{8}\|\mathbf{g}_k\|^2 \geq \frac{7}{8}\gamma^2.$$

Combining this with (2.31) gives

$$\mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k \geq (1-\sigma)\frac{7}{8}\gamma^2. \tag{2.32}$$

Also, observe that

$$\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k = \mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k + \mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k < \mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k. \tag{2.33}$$

Again, the Wolfe condition gives

$$\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k \geq \sigma \mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k = -\sigma \mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k + \sigma \mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k. \tag{2.34}$$

Since $\sigma < 1$, we can rearrange (2.34) to obtain

$$\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k \geq \frac{-\sigma}{1-\sigma}\mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k.$$

Combining this lower bound for $\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k$ with the upper bound (2.33) yields

$$\left|\frac{\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k}{\mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k}\right| \leq \max\left\{\frac{\sigma}{1-\sigma}, 1\right\}. \tag{2.35}$$

By the definition of $\bar{\beta}_k^N$ in (2.5), we have

$$\bar{\beta}_k^N = \beta_k^N \text{ if } \beta_k^N \geq 0 \quad \text{and} \quad 0 \geq \bar{\beta}_k^N \geq \beta_k^N \text{ if } \beta_k^N < 0.$$

Hence, $|\bar{\beta}_k^N| \leq |\beta_k^N|$ for each k. We now insert the upper bound (2.35) for $|\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k|/|\mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k|$, the lower bound (2.32) for $\mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k$, and the Lipschitz estimate (2.19) for $\mathbf{y}_k$ into the expression (2.3) to obtain:

$$\begin{aligned} |\bar{\beta}_k^N| &\leq |\beta_k^N| \\ &\leq \frac{1}{|\mathbf{d}_k^{\mathsf{T}}\mathbf{y}_k|}\left(|\mathbf{y}_k^{\mathsf{T}}\mathbf{g}_{k+1}| + 2\|\mathbf{y}_k\|^2\frac{|\mathbf{g}_{k+1}^{\mathsf{T}}\mathbf{d}_k|}{|\mathbf{y}_k^{\mathsf{T}}\mathbf{d}_k|}\right) \\ &\leq \frac{8}{7}\frac{1}{(1-\sigma)\gamma^2}\left(L\Gamma\|\mathbf{s}_k\| + 2L^2\|\mathbf{s}_k\|^2\max\left\{\frac{\sigma}{1-\sigma}, 1\right\}\right) \\ &\leq C\|\mathbf{s}_k\|, \end{aligned} \tag{2.36}$$

where Γ is defined in (2.28),

$$C = \frac{8}{7}\frac{1}{(1-\sigma)\gamma^2}\left(L\Gamma + 2L^2 D\max\left\{\frac{\sigma}{1-\sigma}, 1\right\}\right), \tag{2.37}$$

$$D = \max\{\|\mathbf{y}-\mathbf{z}\| : \mathbf{y}, \mathbf{z} \in \mathcal{L}\}. \tag{2.38}$$

Here D is the diameter of $\mathcal{L}$.

II. *A bound on the steps* $\mathbf{s}_k$:

This is a modified version of [60, Thm. 4.3]. Observe that for any $l \geq k$,

$$\mathbf{x}_l - \mathbf{x}_k = \sum_{j=k}^{l-1} \mathbf{x}_{j+1} - \mathbf{x}_j = \sum_{j=k}^{l-1}\|\mathbf{s}_j\|\mathbf{u}_j = \sum_{j=k}^{l-1}\|\mathbf{s}_j\|\mathbf{u}_k + \sum_{j=k}^{l-1}\|\mathbf{s}_j\|(\mathbf{u}_j - \mathbf{u}_k).$$

By the triangle inequality:

$$\sum_{j=k}^{l-1}\|\mathbf{s}_j\| \leq \|\mathbf{x}_l - \mathbf{x}_k\| + \sum_{j=k}^{l-1}\|\mathbf{s}_j\|\|\mathbf{u}_j - \mathbf{u}_k\| \leq D + \sum_{j=k}^{l-1}\|\mathbf{s}_j\|\|\mathbf{u}_j - \mathbf{u}_k\|. \tag{2.39}$$

Let Δ be a positive integer, chosen large enough that

$$\Delta \geq 4CD, \tag{2.40}$$

where C and D appear in (2.37) and (2.38). Choose k_0 large enough that

$$\sum_{i \geq k_0} \|\mathbf{u}_{i+1} - \mathbf{u}_i\|^2 \leq \frac{1}{4\Delta}. \tag{2.41}$$

By Lemma 2, k_0 can be chosen in this way. If $j > k \geq k_0$ and $j - k \leq \Delta$, then by (2.41) and the Cauchy-Schwarz inequality, we have

$$\begin{aligned} \|\mathbf{u}_j - \mathbf{u}_k\| &\leq \sum_{i=k}^{j-1} \|\mathbf{u}_{i+1} - \mathbf{u}_i\| \\ &\leq \sqrt{j-k} \left(\sum_{i=k}^{j-1} \|\mathbf{u}_{i+1} - \mathbf{u}_i\|^2 \right)^{1/2} \\ &\leq \sqrt{\Delta} \left(\frac{1}{4\Delta} \right)^{1/2} = \frac{1}{2}. \end{aligned}$$

Combining this with (2.39) yields

$$\sum_{j=k}^{l-1} \|\mathbf{s}_j\| \leq 2D, \tag{2.42}$$

when $l > k \geq k_0$ and $l - k \leq \Delta$.

III. *A bound on the directions* $\mathbf{d}_l$:

By (2.4) and the bound on $\bar{\beta}_k^N$ given in Step I, we have

$$\|\mathbf{d}_l\|^2 \leq (\|\mathbf{g}_l\| + |\bar{\beta}_{l-1}^N| \|\mathbf{d}_{l-1}\|)^2 \leq 2\Gamma^2 + 2C^2 \|\mathbf{s}_{l-1}\|^2 \|\mathbf{d}_{l-1}\|^2,$$

where Γ is the bound on the gradient given in (2.28). Defining $S_i = 2C^2 \|\mathbf{s}_i\|^2$, we conclude that for $l > k_0$,

$$\|\mathbf{d}_l\|^2 \leq 2\Gamma^2 \left(\sum_{i=k_0+1}^{l} \prod_{j=i}^{l-1} S_j \right) + \|\mathbf{d}_{k_0}\|^2 \prod_{j=k_0}^{l-1} S_j. \tag{2.43}$$

Above, the product is defined to be one whenever the index range is vacuous. Let us consider a product of Δ consecutive S_j, where $k \geq k_0$:

$$\begin{aligned}\prod_{j=k}^{k+\Delta-1} S_j &= \prod_{j=k}^{k+\Delta-1} 2C^2\|\mathbf{s}_j\|^2 = \left(\prod_{j=k}^{k+\Delta-1} \sqrt{2}C\|\mathbf{s}_j\|\right)^2 \\ &\leq \left(\frac{\sum_{j=k}^{k+\Delta-1} \sqrt{2}C\|\mathbf{s}_j\|}{\Delta}\right)^{2\Delta} \leq \left(\frac{2\sqrt{2}CD}{\Delta}\right)^{2\Delta} \leq \frac{1}{2^\Delta}\end{aligned}$$

The first inequality above is the arithmetic-geometric mean inequality, the second is due to (2.42), and the third comes from (2.40). Since the product of Δ consecutive S_j is bounded by $1/2^\Delta$, it follows that the sum in (2.43) is bounded, independent of l. This bound for $\|\mathbf{d}_l\|$, independent of $l > k_0$, contradicts (2.25). Hence, $\gamma = \liminf_{k\to\infty} \|\mathbf{g}_k\| = 0$. □

2.1.4 Line Search

The line search is an important factor in the overall efficiency of any optimization algorithm. Papers focusing on the development of efficient line search algorithms include [1, 85, 100, 101]. The algorithm [101] of Moré and Thuente is used widely; it is incorporated in the L-BFGS limited memory quasi-Newton code of Nocedal and in the PRP+ conjugate gradient code of Liu, Nocedal, and Waltz. However, there is a fundamental numerical problem associated with the first condition (2.10) in the standard Wolfe conditions (for detail explanations, please refer the paper [73]). Based on this observation, in practice we proposed the the *approximate Wolfe conditions*:

$$(2\delta - 1)\phi'(0) \geq \phi'(\alpha_k) \geq \sigma\phi'(0), \tag{2.44}$$

where $\delta < \min\{.5, \sigma\}$ and $\phi(\alpha) = f(\mathbf{x}_k + \alpha\mathbf{d}_k)$. The second inequality in (2.44) is identical to the second Wolfe condition (2.11). The first inequality in (2.44) is identical to the first Wolfe condition (2.10) when f is quadratic. For general f, we now show that the first inequality in (2.44) and the first Wolfe condition agree to order α_k^2. The interpolating (quadratic) polynomial q that matches $\phi(\alpha)$ at $\alpha = 0$ and $\phi'(\alpha)$ at $\alpha = 0$ and $\alpha = \alpha_k$ is

$$q(\alpha) = \frac{\phi'(\alpha_k) - \phi'(0)}{2\alpha_k}\alpha^2 + \phi'(0)\alpha + \phi(0).$$

For such an interpolating polynomial, $|q(\alpha) - \phi(\alpha)| = O(\alpha^3)$. After replacing ϕ by q in the first Wolfe condition, we obtain the first inequality in (2.44) (with an error term of order α_k^2). We emphasize that this first inequality is an approximation to the first Wolfe condition. On the other hand, this approximation can be evaluated with greater precision than the original condition, when the iterates are near a local minimizer, since the approximate Wolfe conditions are expressed in terms of a derivative, not the difference of function values.

With these insights, we terminate the line search when either of the following conditions holds:

T1. The original Wolfe conditions (2.10)–(2.11) are satisfied.

T2. The approximate Wolfe conditions (2.44) are satisfied and

$$\phi(\alpha_k) \le \phi(0) + \epsilon_k, \tag{2.45}$$

where $\epsilon_k \ge 0$ is an estimate for the error in the value of f at iteration k. In the experiments section, we took

$$\epsilon_k = \epsilon|f(\mathbf{x}_k)|, \tag{2.46}$$

where ϵ is a (small) fixed parameter.

We satisfy the termination criterion by constructing a nested sequence of (bracketing) intervals which converge to a point satisfying either T1 or T2. A typical interval $[a, b]$ in the nested sequence satisfies the following *opposite slope condition*:

$$\phi(a) \le \phi(0) + \epsilon_k, \quad \phi'(a) < 0, \quad \phi'(b) \ge 0. \tag{2.47}$$

Given a parameter $\theta \in (0, 1)$. We also develop the *interval update rules* which can be found as the procedure “interval update” in the paper [73]. And during the “interval update” procedure, a new so called “Double Secant Step” is used. We prove implementing this new “Double Secant Step” in the “interval update”, an asymptotic root convergence order $1 + \sqrt{2} \approx 2.4$ can be obtained with is slightly less the the square of the convergence speed of the traditional second method $((1 + \sqrt{5})^2/4 \approx 2.6)$. More specifically, we have the following theorem. For the detail proof, please refer the paper [73].

Theorem 7 *Suppose that ϕ is three times continuously differentiable near a local minimizer α^*, with $\phi''(\alpha^*) > 0$ and $\phi'''(\alpha^*) \neq 0$. Then for a_0 and b_0 sufficiently close to α^* with $a_0 \leq \alpha^* \leq b_0$, the iteration*

$$[a_{k+1}, b_{k+1}] = \text{ secant}^2(a_k, b_k)$$

converges to α^*. Moreover, the interval width $|b_k - a_k|$ tends to zero with root convergence order $1 + \sqrt{2}$.

2.1.5 Numerical Comparisons

In this section we compare the CPU time performance of the new conjugate gradient method, denoted CG_DESCENT, to the L-BFGS limited memory quasi-Newton method of Nocedal [103] and Liu and Nocedal [91] and to other conjugate gradient methods as well. Comparisons based on other metrics, such as the number of iterations or number of function/gradient evaluations, can be found in paper [74], where extensive numerical testing of the methods is done. We considered both the PRP+ version of the conjugate gradient method developed by Gilbert and Nocedal [60], where the β_k associated with the Polak-Ribière-Polyak conjugate gradient method [106, 107] is kept nonnegative, and versions of the conjugate gradient method developed by Dai and Yuan in [35, 37], denoted CGDY and CGDYH, which achieve descent for any line search that satisfies the Wolfe conditions (2.10)–(2.11). The hybrid conjugate gradient method CGDYH uses

$$\beta_k = \max\{0, \min\{\beta_k^{HS}, \beta_k^{DY}\}\},$$

where β_k^{HS} is the choice of Hestenes-Stiefel [81] and β_k^{DY} appears in [35]. The test problems are the unconstrained problems in the CUTE [12] test problem library.

The L-BFGS and PRP+ codes were obtained from Jorge Nocedal's web page. The L-BFGS code is authored by Jorge Nocedal, while the PRP+ code is co-authored by Guanghui Liu, Jorge Nocedal, and Richard Waltz. In the documentation for the L-BFGS code, it is recommended that between 3 and 7 vectors be used for the memory. Hence, we chose 5 vectors for the memory. The line search in both codes is a modification of subroutine CSRCH of Moré and Thuente [101], which employs various polynomial interpolation schemes and safeguards in satisfying the strong Wolfe line search conditions.

We also manufactured a new L-BFGS code by replacing the Moré/Thuente line search by the new line search presented in our paper. We call this new code L-BFGS*. The new line search would need to be modified for use in the PRP+ code to ensure descent. Hence, we retained the Moré/Thuente line search in the PRP+ code. Since the conjugate gradient algorithms of Dai and Yuan achieves descent for any line search that satisfies the Wolfe conditions, we are able to use the new line search in our experiments with CGDY and with CGDYH. All codes were written in Fortran and compiled with f77 (default compiler settings) on a Sun workstation.

For our line search algorithm, we used the following values for the parameters:

$$\delta = .1, \quad \sigma = .9, \quad \epsilon = 10^{-6}, \quad \theta = .5, \quad \gamma = .66, \quad \eta = .01$$

Our rationale for these choices was the following: The constraints on δ and σ are $0 < \delta \le \sigma < 1$ and $\delta < .5$. As δ approaches 0 and σ approaches 1, the line search terminates quicker. The chosen values $\delta = .1$ and $\sigma = .9$ represent a compromise between our desire for rapid termination and our desire to improve the function value. When using the approximate Wolfe conditions, we would like to achieve decay in the function value, if numerically possible. Hence, we made the small choice $\epsilon = 10^{-6}$ in (2.46). When restricting β_k in (2.5), we would like to avoid truncation if possible, since the fastest convergence for a quadratic function is obtained when there is no truncation at all. The choice $\eta = .01$ leads to infrequent truncation of β_k. The choice $\gamma = .66$ ensures that the length of the interval $[a, b]$ decreases by a factor of 2/3 in each iteration of the line search algorithm. The choice $\theta = .5$ in the update procedure corresponds to the use of bisection. Our starting guess for the step α_k in the line search was obtained by minimizing a quadratic interpolant.

In the first set of experiments, we stopped whenever

$$\text{(a) } \|\nabla f(\mathbf{x}_k)\|_\infty \le 10^{-6} \quad \text{or} \quad \text{(b) } \alpha_k \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k \le 10^{-20} |f(\mathbf{x}_{k+1})|, \tag{2.48}$$

where $\|\cdot\|_\infty$ denotes the maximum absolute component of a vector. In all but 3 cases, the iterations stopped when (a) was satisfied – the second criterion essentially says that the estimated change in the function value is insignificant compared to the function value itself.

The cpu time in seconds and the number of iterations, function evaluations, and gradient evaluations, for each of the methods are posted at the author's web site. In running the numerical experiments, we checked whether different codes converged to different local minimizers; we only provide data for problems where all six codes converged to the same local minimizer. The numerical results are now analyzed.

The performance of the 6 algorithms, relative to cpu time, was evaluated using the profiles of Dolan and Moreé [43]. That is, for each method, we plot the fraction

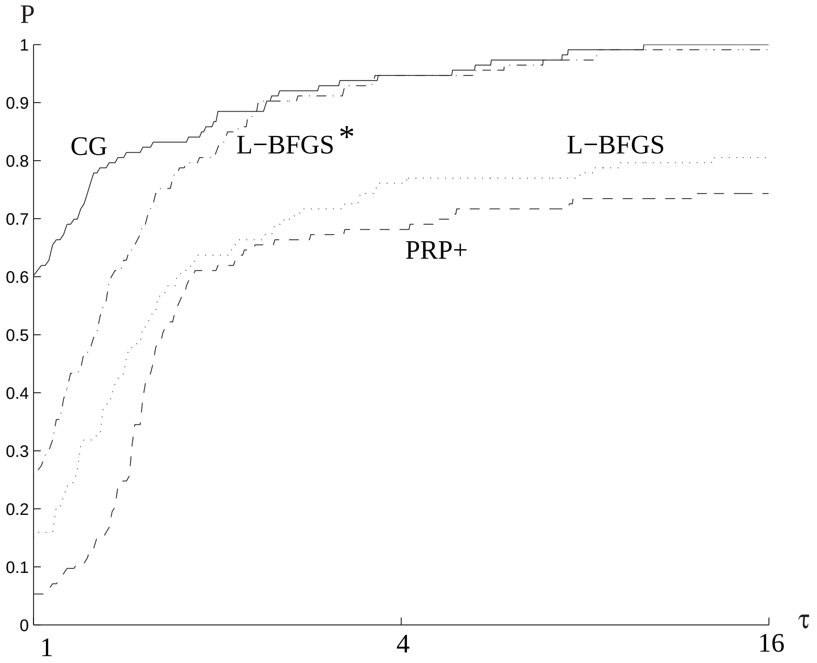

Figure 2–1: Performance profiles

P of problems for which the method is within a factor τ of the best time. In Figure 2–1, we compare the performance of the 4 codes CG, L-BFGS*, L-BFGS, and PRP+. The left side side of the figure gives the percentage of the test problems for which a method is the fastest; the right side gives the percentage of the test problems that were successfully solved by each of the methods. The top curve is the method that solved the most problems in a time that was within a factor τ of the best time. Since the top curve in Figure 2–1 corresponds to CG, this algorithm is clearly fastest for this set of 113 test problems with dimensions ranging from 50 to 10,000. In particular, CG is fastest for about 60% (68 out of 113) of the test problems, and it ultimately solves 100% of the test problems. Since L-BFGS* (fastest for 29 problems) performed better than L-BFGS (fastest for 17 problems), the new line search led to improved performance. Nonetheless, L-BFGS* was still dominated by CG.

In Figure 2–2 we compare the performance of the four conjugate gradient

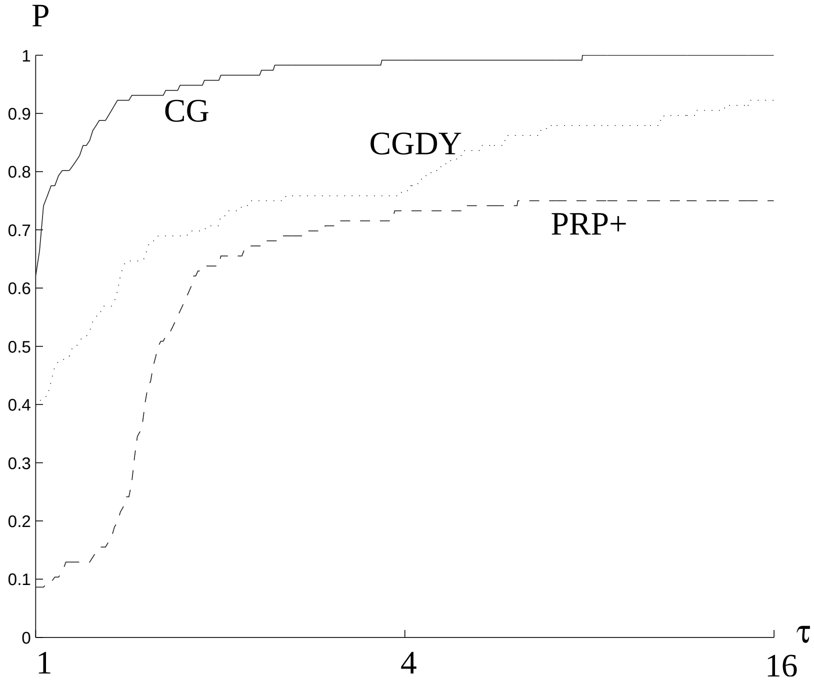

Figure 2–2: Performance profiles of conjugate gradient methods

algorithms. Observe that CG is the fastest of the four algorithm. Since CGDY, CGDYH, and CG use the same line search, Figure 2–2 indicates that the search direction of CG yields quicker descent than the search directions of CGDY and CGDYH. Also, CGDYH is more efficient than CGDY. Since each of these six codes differs in the amount of linear algebra required in each iteration and in the relative number of function and gradient evaluations, different codes will be superior in different problem sets. In particular, the fourth ranked PRP+ code in Figure 2–1 still achieved the fastest time in 6 of the 113 test problems.

In our next series of experiments, shown in Table 2–2, we explore the ability of the algorithms and line search to accurately solve the test problems. In this series of experiments, we repeatly solve six test problems, increasing the specified accuracy in each run. For the initial run, the stopping condition was $\|\mathbf{g}_k\|_\infty \le 10^{-2}$, and in the last run, the stopping condition was $\|\mathbf{g}_k\|_\infty \le 10^{-12}$. The test problems used in these experiments, and their dimensions, were the following:

Table 2–2: Solution time versus tolerance

Tolerance	Algorithm	Problem Number					
$\|\mathbf{g}_k\|_\infty$	Name	#1	#2	#3	#4	#5	#6
10^{-2}	CG	5.22	2.32	0.86	0.00	1.57	10.04
	L-BFGS*	4.19	1.57	0.75	0.01	1.81	14.80
	L-BFGS	4.24	2.01	0.99	0.00	2.46	16.48
	PRP+	6.77	3.55	1.43	0.00	3.04	17.80
10^{-3}	CG	9.20	5.27	2.09	0.00	2.26	17.13
	L-BFGS*	6.72	6.18	2.42	0.01	2.65	19.46
	L-BFGS	6.88	7.46	2.65	0.00	3.30	22.63
	PRP+	12.79	7.16	3.61	0.00	4.26	24.13
10^{-4}	CG	10.79	5.76	5.04	0.00	3.23	25.26
	L-BFGS*	11.56	10.87	6.33	0.01	3.49	31.12
	L-BFGS	12.24	10.92	6.77	0.00	4.11	33.36
	PRP+	15.97	11.40	8.13	0.00	5.01	F
10^{-5}	CG	14.26	7.94	7.97	0.00	4.27	27.49
	L-BFGS*	17.14	16.05	10.21	0.01	4.33	36.30
	L-BFGS	16.60	16.99	10.97	0.00	4.90	F
	PRP+	21.54	12.09	12.31	0.00	6.22	F
10^{-6}	CG	16.68	8.49	9.80	5.71	5.42	32.03
	L-BFGS*	21.43	19.07	14.58	9.01	5.08	46.86
	L-BFGS	21.81	21.08	13.97	7.78	5.83	F
	PRP+	24.58	12.81	15.33	8.07	7.95	F
10^{-7}	CG	20.31	11.47	11.93	5.81	5.93	39.79
	L-BFGS*	26.69	25.74	17.30	12.00	6.10	54.43
	L-BFGS	26.47	F	17.37	9.98	6.39	F
	PRP+	31.17	F	17.34	8.50	9.50	F
10^{-8}	CG	23.22	12.88	14.09	9.68	6.49	47.50
	L-BFGS*	28.18	33.19	20.16	16.58	6.73	63.42
	L-BFGS	32.23	F	20.48	14.85	7.67	F
	PRP+	33.75	F	19.83	F	10.86	F
10^{-9}	CG	27.92	13.32	16.80	12.34	7.46	56.68
	L-BFGS*	32.19	38.51	26.50	26.08	7.67	72.39
	L-BFGS	33.64	F	F	F	8.50	F
	PRP+	F	F	F	F	11.74	F
10^{-10}	CG	33.25	13.89	21.18	13.21	8.11	65.47
	L-BFGS*	34.16	50.60	29.79	33.60	8.22	79.08
	L-BFGS	39.12	F	F	F	9.53	F
	PRP+	F	F	F	F	13.56	F
10^{-11}	CG	38.80	14.38	25.58	13.39	9.12	77.03
	L-BFGS*	36.78	55.70	34.81	39.02	9.14	88.86
	L-BFGS	F	F	F	F	9.99	F
	PRP+	F	F	F	F	14.44	F
10^{-12}	CG	42.51	15.62	27.54	13.38	9.77	78.31
	L-BFGS*	41.73	60.89	39.29	43.95	9.97	101.36
	L-BFGS	F	F	F	F	10.54	F
	PRP+	F	F	F	F	15.96	F

1. FMINSURF (5625)
2. NONCVXU2 (1000)
3. DIXMAANE (6000)
4. FLETCBV2 (1000)
5. SCHMVETT (10000)
6. CURLY10 (1000)

These problems were chosen somewhat randomly except that we did not include any problem for which the optimal cost was zero. When the optimal cost is zero while the minimizer $\mathbf{x}$ is not zero, the estimate $\epsilon|f(\mathbf{x}_k)|$ for the error in function value (which we used in the previous experiments) can be very poor as the iterates approach the minimizer (where f vanishes). These six problems all have nonzero optimal cost. In Table 2–2, F means that the line search terminated before the convergence tolerance for $\|\mathbf{g}_k\|$ was satisfied. According to the documentation for the line search in the L-BFGS and PRP+ codes, "Rounding errors prevent further progress. There may not be a step which satisfies the sufficient decrease and curvature conditions. Tolerances may be too small."

As can be seen in Table 2–2, the line search based on the Wolfe conditions (used in the L-BFGS and PRP+ codes) fails much sooner than the line search based on both the Wolfe and the approximate Wolfe conditions (used in CG and L-BFGS*). Roughly, a line search based on the Wolfe conditions can compute a solution with accuracy on the order of the square root of the machine epsilon, while a line search that also includes the approximate Wolfe conditions can compute a solution with accuracy on the order of the machine epsilon.

2.2 A Cyclic Barzilai-Borwein (CBB) Method

2.2.1 Introduction to Nonmonotone Line Search

We know many iterative methods for (2.1) produce a sequence $\mathbf{x}_0$, $\mathbf{x}_1$, $\mathbf{x}_2$, . . ., where $\mathbf{x}_{k+1}$ is generated from $\mathbf{x}_k$, the current direction $\mathbf{d}_k$, and the stepsize $\alpha_k > 0$ by the rule

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k. \tag{2.49}$$

In monotone line search methods, α_k is chosen so that $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$. In nonmonotone line search methods, some growth in the function value is permitted. As pointed out by many researchers (for example, see [28, 117]), nonmonotone schemes can improve the likelihood of finding a global optimum; also, they can improve convergence speed in cases where a monotone scheme is forced to creep along the bottom of a narrow curved valley. Encouraging numerical results have been reported [27, 67, 94, 104, 108, 117, 130] when nonmonotone schemes were applied to difficult nonlinear problems.

The earliest nonmonotone line search framework was developed by Grippo, Lampariello, and Lucidi in [66] for Newton's methods. Their approach was roughly the following: Parameters λ_1, λ_2, σ, and δ are introduced where $0 < \lambda_1 < \lambda_2$ and $\sigma, \delta \in (0,1)$, and they set $\alpha_k = \bar{\alpha}_k \sigma^{h_k}$ where $\bar{\alpha}_k \in (\lambda_1, \lambda_2)$ is the "trial step" and h_k is the smallest nonnegative integer such that

$$f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) \le \max_{0 \le j \le m_k} f(\mathbf{x}_{k-j}) + \delta \alpha_k \nabla f(\mathbf{x}_k) \mathbf{d}_k. \tag{2.50}$$

The memory m_k at step k is a nondecreasing integer, bounded by some fixed integer M. More precisely,

$$m_0 = 0 \text{ and for } k > 0,\ 0 \le m_k \le \min\{m_{k-1} + 1, M\}.$$

Many subsequent papers, such as [10, 27, 67, 94, 108, 134], have exploited nonmonotone line search techniques of this nature. Recently, a completely new nonmonotone technique is proposed and analyzed in [131]. In that scheme, the authors require that an average of the successive function values decreases, while the above traditional nonmonotone approach (2.50) requires that a maximum of recent function values decreases. They also proved global convergence for nonconvex, smooth functions, and R-linear convergence for strongly convex functions. For the L-BFGS method and the unconstrained optimization problems in the

CUTE library, this new nonmonotone line search algorithm used fewer function and gradient evaluations, on average, than either the monotone or the traditional nonmonotone scheme.

The original method of Barzilai and Borwein was developed in [3]. The basic idea of Barzilai and Borwein is to regard the matrix $D(\alpha_k) = \frac{1}{\alpha_k} I$ as an approximation of the Hessian $\nabla^2 f(\mathbf{x}_k)$ and impose a quasi-Newton property on $D(\alpha_k)$:

$$\alpha_k = \arg\min_{\alpha \in \Re} \ \|D(\alpha)\mathbf{s}_{k-1} - \mathbf{y}_{k-1}\|_2, \tag{2.51}$$

where $\mathbf{s}_{k-1} = \mathbf{x}_k - \mathbf{x}_{k-1}$, $\mathbf{y}_{k-1} = \mathbf{g}_k - \mathbf{g}_{k-1}$, and $k \geq 2$. The proposed stepsize, obtained from (2.51),is

$$\alpha_k^{BB} = \frac{\mathbf{s}_{k-1}^{\mathsf{T}} \mathbf{s}_{k-1}}{\mathbf{s}_{k-1}^{\mathsf{T}} \mathbf{y}_{k-1}}. \tag{2.52}$$

Other possible choices for the stepsize α_k include [29, 31, 34, 38, 58, 68, 109, 114]. In this dissertation, we refer to (2.52) as the Barzilai-Borwein (BB) formula. The iterative method (2.49), corresponding to the step size to be the BB stepsize (2.52) and $\mathbf{d}_k = -\mathbf{g}_k$, is called the BB method. Due to their simplicity, efficiency and low memory requirements, BB-like methods have been used in many applications. Glunt, Hayden, and Raydan [62] present a direct application of the BB method in chemistry. Birgin *et al.* [8] use a globalized BB method to estimate the optical constants and the thickness of thin films, while in Birgin *et al.* [10] further extensions are given, leading to more efficient projected gradient methods. Liu and Dai [92] provide a powerful scheme for solving noisy unconstrained optimization problems by combining the BB method and a stochastic approximation method. The projected BB-like method turns out to be very useful in machine learning for training support vector machines (see Serafini *et al* [114] and Dai and Fletcher [31]). Empirically, good performance is observed on a wide variety of classification problems. All the above good performances of the BB-like method are based on combining the method with some nonmonotone line search technique. In the following of this section, we are going to discuss a so called cyclic BB method (CBB). By using a modified version of the adaptive nonmonotone line search developed in [39], a globally adaptive convergent scheme for CBB method is also developed.

2.2.2 Method and Local Linear Convergence

The superior performance of cyclic steepest descent, compared to the ordinary steepest descent, as shown in [30], leads us to consider the cyclic BB method (CBB):

$$\alpha_{m\ell+i} = \alpha_{m\ell+1}^{BB} \quad \text{for } i = 1, \ldots, m, \tag{2.53}$$

where $m \geq 1$ is again the cycle length. An advantage of the CBB method is that for general nonlinear functions, the stepsize is given by the simple formula (2.51) in contrast to the nontrivial optimization problem associated with the steepest descent step.

In this section we prove R-linear convergence for the CBB method. In [92], it is proposed that R-linear convergence for the BB method applied to a general nonlinear function could be obtained from the R-linear convergence results for a quadratic by comparing the iterates associated with a quadratic approximation to the general nonlinear iterates. In our R-linear convergence result for the CBB method, we make such a comparison.

The CBB iteration can be expressed as

$$x_{k+1} = x_k - \alpha_k g_k, \tag{2.54}$$

where

$$\alpha_k = \frac{s_i^{\mathsf{T}} s_i}{s_i^{\mathsf{T}} y_i}, \quad i = \nu(k), \quad \text{and} \quad \nu(k) = m\lfloor (k-1)/m \rfloor, \tag{2.55}$$

$k \geq 1$. For $r \in \Re$, $\lfloor r \rfloor$ denotes the largest integer j such that $j \leq r$. We assume that f is two times Lipschitz continuously differentiable in a neighborhood of a local minimizer x^* where the Hessian $H = \nabla^2 f(x^*)$ is positive definite. The second-order Taylor approximation $\hat{f}$ to f around x^* is given by

$$\hat{f}(x) = f(x^*) + \frac{1}{2}(x - x^*)^\mathsf{T} H(x - x^*). \tag{2.56}$$

We will compare an iterate x_{k+j} generated by (2.54) to a CBB iterate $\hat{x}_{k,j}$ associated with $\hat{f}$ and the starting point $\hat{x}_{k,0} = x_k$. More precisely, we define:

$$\begin{aligned} \hat{x}_{k,0} &= x_k \\ \hat{x}_{k,j+1} &= \hat{x}_{k,j} - \hat{\alpha}_{k,j}\hat{g}_{k,j}, \quad j \geq 0, \end{aligned} \tag{2.57}$$

where

$$\hat{\alpha}_{k,j} = \begin{cases} \alpha_k & \text{if} \quad \nu(k+j) = \nu(k) \\ \dfrac{\hat{s}_i^\mathsf{T}\hat{s}_i}{\hat{s}_i^\mathsf{T}\hat{y}_i}, & i = \nu(k+j), \ \ \text{otherwise.} \end{cases}$$

Here $\hat{s}_{k+j} = \hat{x}_{k,j+1} - \hat{x}_{k,j}$, $\hat{g}_{k,j} = H(\hat{x}_{k,j} - x^*)$, and $\hat{y}_{k+j} = \hat{g}_{k,j+1} - \hat{g}_{k,j}$.

We exploit the following result established in [29, Thm. 3.2]:

Lemma 3 *Let $\{\hat{x}_{k,j} : j \geq 0\}$ be the CBB iterates associated with the starting point $\hat{x}_{k,0} = x_k$ and the quadratic $\hat{f}$ in (2.56), where H is positive definite. Given two arbitrary constants $C_2 > C_1 > 0$, there exists a positive integer N with the following property: For any $k \geq 1$ and*

$$\hat{\alpha}_{k,0} \in [C_1, C_2], \tag{2.58}$$

$$\|\hat{x}_{k,N} - x^*\| \leq \frac{1}{2}\|\hat{x}_{k,0} - x^*\|.$$

In our next lemma, we estimate the distance between $\hat{x}_{k,j}$ and x_{k+j}. Let $B_\rho(x)$ denote the ball with center x and radius ρ. Since f is two times Lipschitz continuously differentiable and $\nabla^2 f(x^*)$ is positive definite, there exists positive constants ρ, λ, and $\Lambda_2 > \Lambda_1$ such that

$$\|\nabla f(x) - H(x - x^*)\| \leq \lambda\|x - x^*\|^2 \quad \text{for all } x \in B_\rho(x^*) \tag{2.59}$$

and

$$\Lambda_1 \le \frac{y^{\mathsf{T}} \nabla^2 f(x) y}{y^{\mathsf{T}} y} \le \Lambda_2 \quad \text{for all } y \in \Re^n \text{ and } x \in B_\rho(x^*). \tag{2.60}$$

Notice that if x_i and $x_{i+1} \in B_\rho(x^*)$, then the fundamental theorem of calculus applied to $y_i = g_{i+1} - g_i$ yields

$$\frac{1}{\Lambda_2} \le \frac{s_i^{\mathsf{T}} s_i}{s_i^{\mathsf{T}} y_i} \le \frac{1}{\Lambda_1}. \tag{2.61}$$

Hence, when the CBB iterates lie in $B_\rho(x^*)$, the condition (2.58) of Lemma 3 is satisfied with $C_1 = 1/\Lambda_2$ and $C_2 = 1/\Lambda_1$. If we define $g(x) = \nabla f(x)^{\mathsf{T}}$, then the fundamental theorem of calculus can also be used to deduce that

$$\|g(x)\| = \|g(x) - g(x^*)\| \le \Lambda_2 \|x - x^*\| \tag{2.62}$$

for all $x \in B_\rho(x^*)$.

Lemma 4 *Let $\{x_j : j \ge k\}$ be a sequence generated by the CBB method applied to a function f with a local minimizer x^*, and assume that the Hessian $H = \nabla^2 f(x^*)$ is positive definite with (2.60) satisfied. Then for any fixed positive integer N, there exist positive constants δ and γ with the following property: For any $x_k \in B_\delta(x^*)$, $\alpha_k \in [\Lambda_2^{-1}, \Lambda_1^{-1}]$, $\ell \in [0, N]$ with*

$$\|\hat{x}_{k,j} - x^*\| \ge \frac{1}{2} \|\hat{x}_{k,0} - x^*\| \quad \textit{for all } j \in [0, \max\{0, \ell - 1\}], \tag{2.63}$$

we have

$$x_{k+j} \in B_\rho(x^*) \textit{ and } \|x_{k+j} - \hat{x}_{k,j}\| \le \gamma \|x_k - x^*\|^2 \tag{2.64}$$

for all $j \in [0, \ell]$.

Proof. Throughout the proof, we let c denote a generic positive constant, which depends on fixed constants such as N or Λ_1 or Λ_2 or λ, but not on either k or the choice of $x_k \in B_\delta(x^*)$ or the choice of $\alpha_k \in [\Lambda_2^{-1}, \Lambda_1^{-1}]$. To facilitate the proof, we also show that

$$\begin{aligned}\|g(x_{k+j}) - \hat{g}(\hat{x}_{k,j})\| &\leq c\|x_k - x^*\|^2, & (2.65)\\ \|s_{k+j}\| &\leq c\|x_k - x^*\|, & (2.66)\\ |\alpha_{k+j} - \hat{\alpha}_{k,j}| &\leq c\|x_k - x^*\|, & (2.67)\end{aligned}$$

for all $j \in [0, \ell]$, where $\hat{g}(x) = \nabla\hat{f}(x)^{\mathsf{T}} = H(x - x^*)$.

The proof of (2.64)–(2.67) is by induction on ℓ. For $\ell = 0$, we take $\delta = \rho$. The relation (2.64) is trivial since $\hat{x}_{k,0} = x_k$. By (2.59), we have

$$\|g(x_k) - \hat{g}(\hat{x}_{k,0})\| = \|g(x_k) - \hat{g}(x_k)\| \leq \lambda\|x_k - x^*\|^2,$$

which gives (2.65). Since $\delta = \rho$ and $x_k \in B_\delta(x^*)$, it follows from (2.62) that

$$\|s_k\| = \|\alpha_k g_k\| \leq \frac{\Lambda_2}{\Lambda_1}\|x_k - x^*\|,$$

which gives (2.66). The relation (2.67) is trivial since $\hat{\alpha}_{k,0} = \alpha_k$.

Now, proceeding by induction, suppose that there exist $L \in [1, N)$ and $\delta > 0$ with the property that if (2.63) holds for any $\ell \in [0, L-1]$, then (2.64)–(2.67) are satisfied for all $j \in [0, \ell]$. We wish to show that for a smaller choice of $\delta > 0$, we can replace L by $L + 1$. Hence, we suppose that (2.63) holds for all $j \in [0, L]$. Since (2.63) holds for all $j \in [0, L-1]$, it follows from the induction hypothesis and (2.66) that

$$\begin{aligned}\|x_{k+L+1} - x^*\| &\leq \|x_k - x^*\| + \sum_{i=0}^{L}\|s_{k+i}\| \\ &\leq c\|x_k - x^*\|. & (2.68)\end{aligned}$$

Consequently, by choosing δ smaller if necessary, we have $x_{k+L+1} \in B_\rho(x^*)$ when $x_k \in B_\delta(x^*)$.

By the triangle inequality,

$$\begin{aligned}
&\|x_{k+L+1} - \hat{x}_{k,L+1}\| \\
&\quad = \|x_{k+L} - \alpha_{k+L} g(x_{k+L}) - [\hat{x}_{k,L} - \hat{\alpha}_{k,L}\hat{g}(\hat{x}_{k,L})]\| \\
&\quad \leq \|x_{k+L} - \hat{x}_{k,L}\| + |\hat{\alpha}_{k,L}|\|g(x_{k+L}) - \hat{g}(\hat{x}_{k,L})\| \\
&\qquad + |\alpha_{k+L} - \hat{\alpha}_{k,L}|\|g(x_{k+L})\|. \qquad (2.69)
\end{aligned}$$

We now analyze each of the terms in (2.69). By the induction hypothesis, the bound (2.64) with $j = L$ holds, which gives

$$\|x_{k+L} - \hat{x}_{k,L}\| \leq c\|x_k - x^*\|^2. \qquad (2.70)$$

By the definition of $\hat{\alpha}$, either $\hat{\alpha}_{k,L} = \alpha_k \in [\Lambda_2^{-1}, \Lambda_1^{-1}]$, or

$$\hat{\alpha}_{k,L} = \frac{\hat{s}_i^{\mathsf{T}}\hat{s}_i}{\hat{s}_i^{\mathsf{T}}\hat{y}_i}, \quad i = \nu(k+L).$$

In this latter case,

$$\frac{1}{\Lambda_2} \leq \frac{\hat{s}_i^{\mathsf{T}}\hat{s}_i}{\hat{s}_i^{\mathsf{T}}H\hat{s}_i} = \frac{\hat{s}_i^{\mathsf{T}}\hat{s}_i}{\hat{s}_i^{\mathsf{T}}\hat{y}_i} \leq \frac{1}{\Lambda_1}.$$

Hence, in either case $\hat{\alpha}_{k,L} \in [\Lambda_2^{-1}, \Lambda_1^{-1}]$. It follows from (2.65) with $j = L$ that

$$\begin{aligned}
|\hat{\alpha}_{k,L}|\|g(x_{k+L}) - \hat{g}(\hat{x}_{k,L})\| &\leq \frac{1}{\Lambda_1}\|g(x_{k+L}) - \hat{g}(\hat{x}_{k,L})\| \\
&\leq c\|x_k - x^*\|^2. \qquad (2.71)
\end{aligned}$$

Also, by (2.67) with $j = L$ and (2.62), we have

$$|\alpha_{k+L} - \hat{\alpha}_{k,L}|\|g(x_{k+L})\| \leq c\|x_k - x^*\|\|x_{k+L} - x^*\|.$$

Utilizing (2.68) (with L replaced by $L-1$) gives

$$|\alpha_{k+L} - \hat{\alpha}_{k,L}|\|g(x_{k+L})\| \leq c\|x_k - x^*\|^2. \qquad (2.72)$$

We combine (2.69)–(2.72) to obtain (2.64) for $j = L+1$. Notice that in establishing (2.64), we exploited (2.65)–(2.67). Consequently, to complete the induction step, each of these estimates should be proved for $j = L+1$.

Focusing on (2.65) for $j = L+1$, we have

$$\begin{aligned}
&\|g(x_{k+L+1}) - \hat{g}(\hat{x}_{k,L+1})\| \\
&\quad \le \ \|g(x_{k+L+1}) - \hat{g}(x_{k+L+1})\| + \|\hat{g}(x_{k+L+1}) - \hat{g}(\hat{x}_{k,L+1})\| \\
&\quad = \ \|g(x_{k+L+1}) - \hat{g}(x_{k+L+1})\| + \|H(x_{k+L+1} - \hat{x}_{k,L+1})\| \\
&\quad \le \ \|g(x_{k+L+1}) - H(x_{k+L+1} - x^*)\| + \Lambda_2\|x_{k+L+1} - \hat{x}_{k,L+1}\| \\
&\quad \le \ \|g(x_{k+L+1}) - H(x_{k+L+1} - x^*)\| + c\|x_k - x^*\|^2,
\end{aligned}$$

since $\|H\| \le \Lambda_2$ by (2.60). The last inequality is due to (2.64) for $j = L+1$, which was just established. Since we chose δ small enough that $x_{k+L+1} \in B_\rho(x^*)$ (see (2.68)), (2.59) implies that

$$\|g(x_{k+L+1}) - H(x_{k+L+1} - x^*)\| \le \lambda\|x_{k+L+1} - x^*\|^2 \le c\|x_k - x^*\|^2.$$

Hence, $\|g(x_{k+L+1}) - \hat{g}(\hat{x}_{k,L+1})\| \le c\|x_k - x^*\|^2$, which establishes (2.65) for $j = L+1$.

Observe that α_{k+L+1} either equals $\alpha_k \in [\Lambda_2^{-1}, \Lambda_1^{-1}]$, or $(s_i^{\mathsf{T}} s_i)/(s_i^{\mathsf{T}} y_i)$, where $k + L \ge i = \nu(k+L+1) > k$. In this latter case, since $x_{k+j} \in B_\rho(x^*)$ for $0 \le j \le L+1$, it follows from (2.61) that

$$\alpha_{k+L+1} \le \frac{1}{\Lambda_1}.$$

Combining this with (2.62), (2.68), and the bound (2.66) for $j \le L$, we obtain

$$\|s_{k+L+1}\| = \|\alpha_{k+L+1} g(x_{k+L+1})\| \le \frac{\Lambda_2}{\Lambda_1}\|x_{k+L+1} - x^*\| \le c\|x_k - x^*\|.$$

Hence, (2.66) is established for $j = L+1$.

Finally, we focus on (2.67) for $j = L+1$. If $\nu(k+L+1) = \nu(k)$, then $\hat{\alpha}_{k,L+1} = \alpha_{k+L+1} = \alpha_k$, so we are done. Otherwise, $\nu(k+L+1) > \nu(k)$, and there exists an index $i \in (0, L]$ such that

$$\alpha_{k+L+1} = \frac{s_{k+i}^{\mathsf{T}} s_{k+i}}{s_{k+i}^{\mathsf{T}} y_{k+i}} \quad \text{and} \quad \hat{\alpha}_{k,L+1} = \frac{\hat{s}_{k+i}^{\mathsf{T}} \hat{s}_{k+i}}{\hat{s}_{k+i}^{\mathsf{T}} \hat{y}_{k+i}}.$$

By (2.64) and the fact that $i \leq L$, we have

$$\|s_{k+i} - \hat{s}_{k+i}\| \leq c\|x_k - x^*\|^2.$$

Combining this with (2.66), and choosing δ smaller if necessary, gives

$$|s_{k+i}^{\mathsf{T}} s_{k+i} - \hat{s}_{k+i}^{\mathsf{T}} \hat{s}_{k+i}| = \left|2s_{k+i}^{\mathsf{T}}(s_{k+i} - \hat{s}_{k+i}) - \|\hat{s}_{k+i} - s_{k+i}\|^2\right| \leq c\|x_k - x^*\|^3. \quad (2.73)$$

Since $\hat{\alpha}_{k,i} \in [\Lambda_2^{-1}, \Lambda_1^{-1}]$, we have

$$\begin{aligned} \|\hat{s}_{k+i}\| &= \|\hat{\alpha}_{k,i}\hat{g}_{k,i}\| \geq \frac{1}{\Lambda_2}\|H(\hat{x}_{k,i} - x^*)\| \\ &\geq \frac{\Lambda_1}{\Lambda_2}\|\hat{x}_{k,i} - x^*\|. \end{aligned}$$

Furthermore, by (2.63) it follows that

$$\|\hat{s}_{k+i}\| \geq \frac{\Lambda_1}{2\Lambda_2}\|\hat{x}_{k,0} - x^*\| = \frac{\Lambda_1}{2\Lambda_2}\|x_k - x^*\|. \quad (2.74)$$

Hence, combining (2.73) and (2.74) gives

$$\left|1 - \frac{s_{k+i}^{\mathsf{T}} s_{k+i}}{\hat{s}_{k+i}^{\mathsf{T}} \hat{s}_{k+i}}\right| = \frac{|s_{k+i}^{\mathsf{T}} s_{k+i} - \hat{s}_{k+i}^{\mathsf{T}} \hat{s}_{k+i}|}{\hat{s}_{k+i}^{\mathsf{T}} \hat{s}_{k+i}} \leq c\|x_k - x^*\|. \quad (2.75)$$

Now let us consider the denominators of α_{k+i} and $\hat{\alpha}_{k,i}$. Observe that

$$\begin{aligned} s_{k+i}^{\mathsf{T}} y_{k+i} - \hat{s}_{k+i}^{\mathsf{T}} \hat{y}_{k+i} &= s_{k+i}^{\mathsf{T}}(y_{k+i} - \hat{y}_{k+i}) + (s_{k+i} - \hat{s}_{k+i})^{\mathsf{T}} \hat{y}_{k+i} \\ &= s_{k+i}^{\mathsf{T}}(y_{k+i} - \hat{y}_{k+i}) + (s_{k+i} - \hat{s}_{k+i})^{\mathsf{T}} H \hat{s}_{k+i}. \quad (2.76) \end{aligned}$$

By (2.64) and (2.66), we have

$$\begin{aligned} |(s_{k+i} - \hat{s}_{k+i})^{\mathsf{T}} H \hat{s}_{k+i}| &= |(s_{k+i} - \hat{s}_{k+i})^{\mathsf{T}} H s_{k+i} - (s_{k+i} - \hat{s}_{k+i})^{\mathsf{T}} H (s_{k+i} - \hat{s}_{k+i})| \\ &\leq c\|x_k - x^*\|^3 \quad (2.77) \end{aligned}$$

for δ sufficiently small. Also, by (2.65) and (2.66), we have

$$|s_{k+i}^{\mathsf{T}}(y_{k+i} - \hat{y}_{k+i})| \le \|s_{k+i}\|(\|g_{k+i+1} - \hat{g}_{k,i+1}\| + \|g_{k+i} - \hat{g}_{k,i}\|) \le c\|x_k - x^*\|^3. \quad (2.78)$$

Combining (2.76)–(2.78) yields

$$\left|s_{k+i}^{\mathsf{T}} y_{k+i} - \hat{s}_{k+i}^{\mathsf{T}} \hat{y}_{k+i}\right| \le c\|x_k - x^*\|^3. \quad (2.79)$$

Since x_{k+i} and $x_{k+i+1} \in B_\rho(x^*)$, it follows from (2.60) that

$$s_{k+i}^{\mathsf{T}} y_{k+i} = s_{k+i}^{\mathsf{T}}(g_{k+i+1} - g_{k+i}) \ge \Lambda_1\|s_{k+i}\|^2 = \Lambda_1|\alpha_{k+i}|^2\|g_{k+i}\|^2. \quad (2.80)$$

By (2.61) and (2.60), we have

$$|\alpha_{k+i}|^2\|g_{k+i}\|^2 \ge \frac{1}{\Lambda_2^2}\|g_{k+i}\|^2 = \frac{1}{\Lambda_2^2}\|g(x_{k+i}) - g(x^*)\|^2 \ge \frac{\Lambda_1^2}{\Lambda_2^2}\|x_{k+i} - x^*\|^2. \quad (2.81)$$

Finally, (2.63) gives

$$\|x_{k+i} - x^*\|^2 \ge \frac{1}{4}\|x_k - x^*\|^2. \quad (2.82)$$

Combining (2.80)–(2.82) yields

$$s_{k+i}^{\mathsf{T}} y_{k+i} \ge \frac{\Lambda_1^3}{4\Lambda_2^2}\|x_k - x^*\|^2. \quad (2.83)$$

Combining (2.79) and (2.83) gives

$$\left|1 - \frac{\hat{s}_{k+i}^{\mathsf{T}}\hat{y}_{k+i}}{s_{k+i}^{\mathsf{T}} y_{k+i}}\right| = \frac{|s_{k+i}^{\mathsf{T}} y_{k+i} - \hat{s}_{k+i}^{\mathsf{T}}\hat{y}_{k+i}|}{s_{k+i}^{\mathsf{T}} y_{k+i}} \le c\|x_k - x^*\|. \quad (2.84)$$

Observe that

$$\begin{aligned}
|\alpha_{k+L+1} - \hat{\alpha}_{k,L+1}| &= \left|\frac{s_{k+i}^{\mathsf{T}} s_{k+i}}{s_{k+i}^{\mathsf{T}} y_{k+i}} - \frac{\hat{s}_{k+i}^{\mathsf{T}}\hat{s}_{k+i}}{\hat{s}_{k+i}^{\mathsf{T}}\hat{y}_{k+i}}\right| \\
&= \hat{\alpha}_{k,L+1}\left|1 - \left(\frac{s_{k+i}^{\mathsf{T}} s_{k+i}}{\hat{s}_{k+i}^{\mathsf{T}}\hat{s}_{k+i}}\right)\left(\frac{\hat{s}_{k+i}^{\mathsf{T}}\hat{y}_{k+i}}{s_{k+i}^{\mathsf{T}} y_{k+i}}\right)\right| \\
&\le \frac{1}{\Lambda_1}\left|1 - \left(\frac{s_{k+i}^{\mathsf{T}} s_{k+i}}{\hat{s}_{k+i}^{\mathsf{T}}\hat{s}_{k+i}}\right)\left(\frac{\hat{s}_{k+i}^{\mathsf{T}}\hat{y}_{k+i}}{s_{k+i}^{\mathsf{T}} y_{k+i}}\right)\right| \\
&= \frac{1}{\Lambda_1}|a(1-b) + b| \le \frac{1}{\Lambda_1}(|a| + |b| + |ab|), \quad (2.85)
\end{aligned}$$

where

$$a = 1 - \frac{s_{k+i}^{\mathsf{T}} s_{k+i}}{\hat{s}_{k+i}^{\mathsf{T}} \hat{s}_{k+i}} \quad \text{and} \quad b = 1 - \frac{\hat{s}_{k+i}^{\mathsf{T}} \hat{y}_{k+i}}{s_{k+i}^{\mathsf{T}} y_{k+i}}.$$

Together, (2.75), (2.84), and (2.85) yield

$$|\alpha_{k+L+1} - \hat{\alpha}_{k,L+1}| \leq c\|x_k - x^*\|$$

for δ sufficiently small. This completes the proof of (2.64)–(2.67). □

Theorem 8 *Let x^* be a local minimizer of f, and assume that the Hessian $\nabla^2 f(x^*)$ is positive definite. Then there exist positive constants δ and γ, and a positive constant $c < 1$ with the property that for all starting points $x_0, x_1 \in B_\delta(x^*)$, $x_0 \neq x_1$, the CBB iterates generated by (2.54)-(2.55) satisfy*

$$\|x_k - x^*\| \leq \gamma c^k \|x_1 - x^*\|.$$

Proof. Let $N > 0$ be the integer given in Lemma 3, corresponding to $C_1 = \Lambda_1^{-1}$ and $C_2 = \Lambda_2^{-1}$, and let δ_1 and γ_1 denote the constants δ and γ given in Lemma 4r. Let γ_2 denote the constant c in (2.66). In other words, these constant δ_1, γ_1, and γ_2 have the property that whenever $\|x_k - x^*\| \leq \delta_1$, $\alpha_k \in [\Lambda_2^{-1}, \Lambda_1^{-1}]$, and

$$\|\hat{x}_{k,j} - x^*\| \geq \frac{1}{2}\|\hat{x}_{k,0} - x^*\| \quad \text{for } 0 \leq j \leq \ell - 1 < N,$$

we have

$$\begin{aligned} \|x_{k+j} - \hat{x}_{k,j}\| &\leq \gamma_1 \|x_k - x^*\|^2, & (2.86) \\ \|s_{k+j}\| &\leq \gamma_2 \|x_k - x^*\|, & (2.87) \\ x_{k+j} &\in B_\rho(x^*), & (2.88) \end{aligned}$$

for all $j \in [0, \ell]$. Moreover, by the triangle inequality and (2.87), it follows that

$$\begin{aligned} \|x_{k+j} - x^*\| &\leq (N\gamma_2 + 1)\|x_k - x^*\| \\ &= \gamma_3 \|x_k - x^*\|, \quad \gamma_3 = (N\gamma_2 + 1), \qquad (2.89) \end{aligned}$$

for all $j \in [0, \ell]$. We define

$$\delta = \min\{\delta_1, \rho, (4\gamma_1)^{-1}\}. \tag{2.90}$$

For any x_0 and $x_1 \in B_\delta(x^*)$, we define a sequence $1 = k_1 < k_2 < \ldots$ in the following way: Starting with the index $k_1 = 1$, let $j_1 > 0$ be the smallest integer with the property that

$$\|\hat{x}_{k_1,j_1} - x^*\| \le \frac{1}{2}\|\hat{x}_{k_1,0} - x^*\| = \frac{1}{2}\|x_1 - x^*\|.$$

Since x_0 and $x_1 \in B_\delta(x^*) \subset B_\rho(x^*)$, it follows from (2.61) that

$$\hat{\alpha}_{1,0} = \alpha_1 = \frac{s_0^\mathsf{T} s_0}{s_0^\mathsf{T} y_0} \in [\Lambda_2^{-1}, \Lambda_1^{-1}].$$

By Lemma 3, $j_1 \le N$. Define $k_2 = k_1 + j_1 > k_1$. By (2.86) and (2.90), we have

$$\begin{aligned}
\|x_{k_2} - x^*\| &= \|x_{k_1+j_1} - x^*\| \le \|x_{k_1+j_1} - \hat{x}_{k_1,j_1}\| + \|\hat{x}_{k_1,j_1} - x^*\| \\
&\le \gamma_1\|x_{k_1} - x^*\|^2 + \frac{1}{2}\|\hat{x}_{k_1,0} - x^*\| \\
&= \gamma_1\|x_{k_1} - x^*\|^2 + \frac{1}{2}\|x_{k_1} - x^*\| \\
&\le \frac{3}{4}\|x_{k_1} - x^*\|.
\end{aligned} \tag{2.91}$$

Since $\|x_1 - x^*\| \le \delta$, it follows that $x_{k_2} \in B_\delta(x^*)$. By (2.88), $x_j \in B_\rho(x^*)$ for $1 \le j \le k_1$.

Now, proceed by induction. Assume that k_i has been determined with $x_{k_i} \in B_\delta(x^*)$ and $x_j \in B_\rho(x^*)$ for $1 \le j \le k_i$. Let $j_i > 0$ be the smallest integer with the property that

$$\|\hat{x}_{k_i,j_i} - x^*\| \le \frac{1}{2}\|\hat{x}_{k_i,0} - x^*\| = \frac{1}{2}\|x_{k_i} - x^*\|.$$

Set $k_{i+1} = k_i + j_i > k_i$. Exactly as in (2.91), we have

$$\|x_{k_{i+1}} - x^*\| \le \frac{3}{4}\|x_{k_i} - x^*\|.$$

Again, $x_{k_{i+1}} \in B_\delta(x^*)$ and $x_j \in B_\rho(x^*)$ for $j \in [1, k_{i+1}]$.

For any $k \in [k_i, k_{i+1})$, we have $k \le k_i + N - 1 \le Ni$, since $k_i \le N(i-1)+1$. Hence, $i \ge k/N$. Also, (2.89) gives

$$\begin{array}{rcl} \|x_k - x^*\| & \le & \gamma_3 \|x_{k_i} - x^*\| \\ & \le & \gamma_3 \left(\frac{3}{4}\right)^{i-1} \|x_{k_1} - x^*\| \\ & \le & \gamma_3 \left(\frac{3}{4}\right)^{(k/N)-1} \|x_1 - x^*\| \\ & = & \gamma c^k \|x_1 - x^*\|, \end{array}$$

where

$$\gamma = \left(\frac{4}{3}\right)\gamma_3 \quad \text{and} \quad c = \left(\frac{3}{4}\right)^{1/N} < 1.$$

This completes the proof. □

2.2.3 Method for Convex Quadratic Programming

In this section, we give numerical evidence which indicates that when m is sufficiently large, the CBB method is superlinearly convergent for a quadratic function

$$f(x) = \frac{1}{2} x^{\mathsf{T}} A x - b^{\mathsf{T}} x, \tag{2.92}$$

where $A \in \Re^{n \times n}$ is symmetric, positive definite and $b \in \Re^n$. Since CBB is invariant under an orthogonal transformation and since gradient components corresponding to identical eigenvalues can be combined (see for example Dai and Fletcher [30]), we assume without loss of generality that A is diagonal:

$$A = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) \quad \text{with } 0 < \lambda_1 < \lambda_2 < \cdots < \lambda_n. \tag{2.93}$$

In the quadratic case, it follows from (2.49) and (2.92) that

$$g_{k+1} = (I - \alpha_k A) g_k. \tag{2.94}$$

Table 2–3: Transition to superlinear convergence

n	2	3	4	5	6	8	10	12	14
superlinear m	1	3	2	4	4	5	6	7	8
linear m		2	1	3	3	4	5	6	7

If $g_k^{(i)}$ denotes the i-th component of the gradient g_k, then by (2.94) and (2.93), we have

$$g_{k+1}^{(i)} = (1 - \alpha_k \lambda_i) g_k^{(i)} \qquad i = 1, 2, \ldots, n. \tag{2.95}$$

We assume that $g_k^{(i)} \neq 0$ for all sufficiently large k. If $g_k^{(i)} = 0$, then by (2.95) component i remains zero during all subsequent iterations; hence it can be discarded. In the BB method, starting values are needed for x_0 and x_1 in order to compute α_1. In our study of CBB, we treat α_1 as a free parameter. In our numerical experiments, we choose α_1 to be the exact stepsize.

For different choices of the diagonal matrix (2.93) and the starting point x_1, we have evaluated the convergence rate of CBB. By the analysis given in [58] for positive definite quadratics, or by the result given in Theorem 8 for general nonlinear functions, the convergence rate of the iterates is at least linear. On the other hand, for m sufficiently large, we observe experimentally, that the convergence rate is superlinear. For fixed dimension n, the value of m where the convergence rate makes a transition between linear and nonlinear is shown in Table 2–3. More precisely, for each value of n, the convergence rate is superlinear when m is great than or equal to the integer given in the second row of the Table 2–3. The convergence is linear when m is less than or equal to the integer given in the third row of Table 2–3.

The limiting integers appearing in Table 2–3 are computed in the following way: For each dimension, we randomly generate 30 problems, with eigenvalues uniformly distributed on $(0, n]$, and 50 starting points – a total of 1500 problems. For each test problem, we perform $1000n$ CBB iterations, and we plot $\log(\log(\|g_k\|_\infty))$

versus the iteration number. We fit the data with a least squares line, and we compute the correlation coefficient to determine how well the linear regression model fits the data. If the correlation coefficient is 1 (or -1), then the linear fit is perfect, while a correlation coefficient of 0 means that the data is uncorrelated. A linear fit in a plot of $\log(\log(\|g_k\|_\infty))$ versus the iteration number indicates superlinear convergence. For m large enough, the correlation coefficients are between -1.0 and -0.98, indicating superlinear convergence. As we decrease m, the correlation coefficient abruptly jumps to the order of -0.8. The integers shown in Table 2–3 reflect the values of m where the correlation coefficient jumps.

Based on Table 2–3, the convergence rate is conjectured to be superlinear for $m > n/2 \geq 3$. For $n < 6$, the relationship between m and n at the transition between linear and superlinear convergence is more complicated, as seen in Table 2–3. Graphs illustrating the convergence appear in Figure 2–3. The horizontal axis in these figures is the iteration number, while the vertical axis gives $\log(\log(\|g_k\|_\infty))$. Here $\|\cdot\|_\infty$ represents the sup-norm. In this case, straight lines correspond to superlinear convergence – the slope of the line reflects the convergence order. In Figure 2–3, the bottom two graphs correspond to superlinear convergence, while the top two graphs correspond to linear convergence – for these top two examples, a plot of $\log(\|g_k\|_\infty)$ versus the iteration number is linear. For the theoretical verification of the experimental results given in Table 2–3 is not easy. However, some partial result in connection with $n = 3$ and $m = 2$ can be theoretically verified. For details, one may refer the paper [32].

2.2.4 An Adaptive CBB Method

In this section, we examine the convergence speed of CBB for different values of $m \in [1, 7]$, using quadratic programming problems of the form:

$$f(x) = \frac{1}{2}x^{\mathsf{T}}Ax, \quad A = \mathrm{diag}(\lambda_1, \cdots, \lambda_n). \tag{2.96}$$

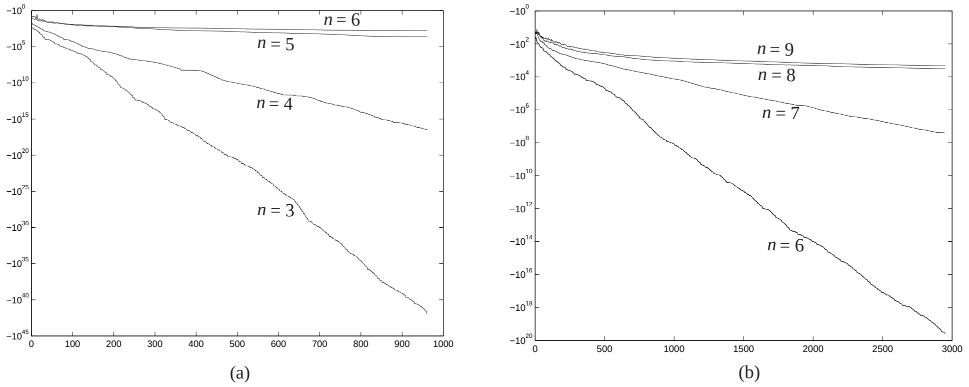

Figure 2–3: Graphs of $\log(\log(\|g_k\|_\infty))$ versus k, (a) $3 \le n \le 6$ and $m = 3$, (b) $6 \le n \le 9$ and $m = 4$.

We will see that the choice for m has a significant impact on performance. This leads us to propose an adaptive choice for m. The BB algorithm with this adaptive choice for m and a nonmonotone line search is called ACBB. Numerical comparisons with SPG2 and with conjugate gradient codes using the CUTEr test problem library are given in the numerical comparisons section.

A numerical investigation of CBB method. We consider the test problem (2.96) with four different condition numbers C for the diagonal matrix: $C = 10^2$, $C = 10^3$, $C = 10^4$, and $C = 10^5$; and with three different dimensions $n = 10^2$, $n = 10^3$, and $n = 10^4$. We let $\lambda_1 = 1$, $\lambda_n = C$, the condition number. The other diagonal elements λ_i, $2 \le i \le n-1$, are randomly generated on the interval $(1, \lambda_n)$. The starting points $x_1^{(i)}$, $i = 1, \cdots, n$, are randomly generated on the interval $[-5, 5]$. The stopping condition is

$$\|g_k\|_2 \le 10^{-8}.$$

For each case, 10 runs are made and the average number of iterations required by each algorithm is listed in Table 2–4 (under the columns labeled BB and CBB). The upper bound for the number of iterations is 9999. If this upper bound is exceeded, then the corresponding entry in Table 2–4 is F.

Table 2–4: Comparing CBB(m) method with an adaptive CBB method

		BB	CBB						*adaptive*	
n	*cond*		$m{=}2$	$m{=}3$	$m{=}4$	$m{=}5$	$m{=}6$	$m{=}7$	$\overline{M}{=}5$	$\overline{M}{=}10$
10^2	10^2	147	219	156	145	150	160	166	136	134
	10^3	505	2715	468	364	376	395	412	367	349
	10^4	1509	*F*	1425	814	852	776	628	878	771
	10^5	5412	*F*	5415	3074	1670	1672	1157	2607	1915
10^3	10^2	147	274	160	158	162	166	181	150	145
	10^3	505	1756	548	504	493	550	540	481	460
	10^4	1609	*F*	1862	1533	1377	1578	1447	1470	1378
	10^5	5699	*F*	6760	4755	3506	3516	2957	4412	3187
10^4	10^2	156	227	162	166	167	170	187	156	156
	10^3	539	3200	515	551	539	536	573	497	505
	10^4	1634	*F*	1823	1701	1782	1747	1893	1587	1517
	10^5	6362	*F*	6779	5194	4965	4349	4736	4687	4743

In Table 2–4 we see that $m = 2$ gives the worst numerical results – in the previous subsection, we saw that as m increases, convergence became superlinear. For each case, a suitably chosen m drastically improves the efficiency the BB method. For example, in case of $n = 10^2$ and *cond* $= 10^5$, CBB with $m = 7$ only requires one fifth of the iterations of the BB method. The optimal choice of m varies from one test case to another. If the problem condition is relatively small (*cond* $= 10^2$, 10^3), a smaller value m (3 or 4) is preferred. If the problem condition is relatively large (*cond* $= 10^4$, 10^5), a larger value of m is more efficient. This observation is the motivation for introducing an adaptive choice for m in the CBB method.

Our adaptive idea arises from the following considerations. If a stepsize is used infinitely often in the gradient method; namely, $\alpha_k \equiv \alpha$, then under the assumption that the function Hessian A has no multiple eigenvalues, the gradient g_k must approximate an eigenvector of A, and $g_k^{\mathsf{T}} A g_k / g_k^{\mathsf{T}} g_k$ tends to the corresponding eigenvalue of A, see [29]. Thus, it is reasonable to assume that repeated use of a BB stepsize leads to good approximations of eigenvectors of A. First, we define

$$\nu_k = \frac{g_k^{\mathsf{T}} A g_k}{\|g_k\| \, \|A g_k\|}. \tag{2.97}$$

If g_k is exactly an eigenvector of A, we know that $\nu_k = 1$. If $\nu_k \approx 1$, then g_k can be regarded as an approximation of an eigenvector of A and $\alpha_k^{BB} \approx \alpha_k^{SD}$. In this case, it is worthwhile to calculate a new BB stepsize α_k^{BB} so that the method accepts a step close to the steepest descent step. Therefore, we test the condition

$$\nu_k \geq \beta, \tag{2.98}$$

where $\beta \in (0, 1)$ is constant. If the above condition holds, we calculate a new BB stepsize. We also introduce a parameter $\overline{M}$, and if the number of cycles $m > \overline{M}$, we calculate a new BB stepsize. Numerical results for this adaptive CBB with $\beta = 0.95$ are listed under the column *adaptive* of Table 2–4, where two values $\overline{M} = 5$, 10 are tested.

From Table 2–4, we see that the adaptive strategy makes sense. The performance with $\overline{M} = 5$ or $\overline{M} = 10$ is often better than that of the BB method. This motivates the use of a similar strategy for designing an efficient gradient algorithms for unconstrained optimization.

Nonmonotone line search and cycle number. As we saw in previous sections, the choice of the stepsize α_k is very important for the performance of a gradient method. For the BB method, function values do not decrease monotonically. Hence, when implementing BB or CBB, it is important to use a nonmonotone line search.

Assuming that d_k is a descent direction at the k-th iteration ($g_k^\mathsf{T} d_k < 0$), a common termination condition for the steplength algorithm is

$$f(x_k + \alpha_k d_k) \leq f_r + \delta \alpha_k g_k^\mathsf{T} d_k, \tag{2.99}$$

where f_r is the so-called *reference function value* and $\delta \in (0, 1)$ a constant. If $f_r = f(x_k)$, then the line search is monotone since $f(x_{k+1}) < f(x_k)$. The nonmonotone line search proposed in [67] chooses f_r to be the maximum function value for the M most recent iterates. That is, at the k-th iteration, we have

$$f_r = f_{\max} = \max_{0 \le i \le \min\{k, M-1\}} f(x_{k-i}). \tag{2.100}$$

This nonmonotone line search is used by Raydan [108] to obtain GBB. Dai and Schittkowski [33] extended the same idea to a sequential quadratic programming method for general constrained nonlinear optimization. An even more adaptive way of choosing f_r is proposed by Toint [118] for trust region algorithms and then extended by Dai and Zhang [39]. Compared with (2.100), the new adaptive way of choosing f_r allows big jumps in function values, and is therefore very suitable for the BB algorithm (see [30], [31], and [39]).

The numerical results which we report in this section are based on the nonmonotone line search algorithm given in [39]. The line search in this paper differs from the line search in [39] in the initialization of the stepsize. Here, the starting guess for the stepsize coincides with the prior BB step until the cycle length has been reached; at which point we recompute the step using the BB formula. In each subsequent subiteration, after computing a new BB step, we replace (2.99) by

$$f(x_k + \bar{\alpha}_k d_k) \le \min\{f_{\max}, f_r\} + \delta \bar{\alpha}_k g_k^{\mathsf{T}} d_k,$$

where f_r is the reference value given in [39] and $\bar{\alpha}_k$ is the initial trial stepsize (the previous BB step). It is proved in [39, Thm. 3.2] that the criteria given in [39] for choosing the nonmonotone stepsize ensures convergence in the sense that

$$\liminf_{k \to \infty} \|g_k\| = 0.$$

We now explain how we decided to terminate the current cycle, and recompute the stepsize using the BB formula. Notice that the reinitialization of the stepsize has no effect on convergence, it only effects the initial stepsize used in the line search. Loosely, we would like to compute a new BB step in any of the following cases:

R1. The number of times m the current BB stepsize has been reused is sufficiently large: $m \geq \overline{M}$, where $\overline{M}$ is a constant.

R2. The following nonquadratic analogue of (2.98) is satisfied:

$$\frac{s_k^{\mathsf{T}} y_k}{\|s_k\|_2 \|y_k\|_2} \geq \beta, \tag{2.101}$$

where $\beta < 1$ is near 1. We feel that the condition (2.101) should only be used in a neighborhood a local minimizer, where f is approximately quadratic. Hence, we only use the condition (2.101) when the stepsize is sufficiently small:

$$\|s_k\|_2 < \min\{\frac{c_1 f_{k+1}}{\|g_{k+1}\|_\infty}, 1\}, \tag{2.102}$$

where c_1 is a constant.

R3. The current step s_k is sufficiently large:

$$\|s_k\|_2 \geq \max\{c_2 \frac{f_{k+1}}{\|g_{k+1}\|_\infty}, 1\}, \tag{2.103}$$

where c_2 is a constant.

R4. In the previous iteration, the BB step was truncated in the line search. That is, the BB step had to be modified by the nonmonotone line search routine to ensure convergence.

Nominally, we recompute the BB stepsize in any of the cases R1–R4. One case where we prefer to retain the current stepsize is the case where the iterates lie in a region where f is not strongly convex. Notice that if $s_k^{\mathsf{T}} y_k < 0$, then there exists a point between x_k and x_{k+1} where the Hessian of f has negative eigenvalues. In detail, our rules for terminating the current cycle and reinitializing the BB stepsize are the following:

Cycle termination/Stepsize initialization.

T1. If any of the condition R1 through R4 are satisfied and $s_k^{\mathsf{T}} y_k > 0$, then the current cycle is terminated and the initial stepsize for the next cycle is given by

$$\alpha_{k+1} = \max\{\alpha_{\min}, \min\{\frac{s_k^{\mathsf{T}} s_k}{s_k^{\mathsf{T}} y_k}, \alpha_{\max}\},$$

where $\alpha_{\min} < \alpha_{\max}$ are fixed constants.

T2. If the length m of the current cycle satisfies $m \geq 1.5\overline{M}$, then the current cycle is terminated and the initial stepsize for the next cycle is given by

$$\alpha_{k+1} = \max\{1/\|g_{k+1}\|_\infty, \alpha_k\}.$$

Condition T2 is a safeguard for the situation where $s_k^{\mathsf{T}} y_k < 0$ in a series of iterations.

2.2.5 Numerical Comparisons

In this subsection, we compare the performance of our adaptive cyclic BB stepsize algorithm, denoted ACBB, with the SPG2 algorithm of Birgin, Martínez, and Raydan [10, 11], with the PRP+ conjugate gradient code developed by Gilbert and Nocedal [60], and with the CG_DESCENT code of Hager and Zhang [73, 74]. The SPG2 algorithm is an extension of Raydan's [108] GBB algorithm which was downloaded from the TANGO web page maintained by Ernesto Birgin. In our tests, we set the bounds in SPG2 to infinity. The PRP+ code is available at the Nocedal's web page. The CG_DESCENT code is found at the author's web page. The line search in the PRP+ code is a modification of subroutine CSRCH of Moré and Thuente [101], which employs various polynomial interpolation schemes and safeguards in satisfying the strong Wolfe conditions. CG_DESCENT employs an "approximate Wolfe" line search. All codes are written in Fortran and compiled with f77 under the default compiler settings on a Sun workstation. The parameters used by CG_DESCENT are the default parameter value given in [74] for version 1.1

of the code. For SPG2, we use parameter values recommended on the TANGO web page. In particular, the step length was restricted to the interval $[10^{-30}, 10^{30}]$, while the memory in the nonmonotone line search was 10.

The parameters of the ACBB algorithm are $\alpha_{\min} = 10^{-30}$, $\alpha_{\max} = 10^{30}$, $c_1 = c_2 = 0.1$, and $\overline{M} = 4$. For the initial iteration, the starting stepsize for the line search was $\alpha_1 = 1/\|g_1\|_\infty$. The parameter values for the nonmonotone line search routine from [39] were $\delta = 10^{-4}$, $\sigma_1 = 0.1$, $\sigma_2 = 0.9$, $\beta = 0.975$, $L = 3$, $M = 8$, and $P = 40$.

Our numerical experiments are based on the entire set of 160 unconstrained optimization problem available from CUTEr in the Fall, 2004. As explained in [74], we deleted problems that were small, or problems where different solvers converged to different local minimizers. After the deletion process, we were left with 111 test problems with dimension ranging from 50 to 10^4.

Nominally, our stopping criterion was the following:

$$\|\nabla f(x_k)\|_\infty \le \max\{10^{-6}, 10^{-12}\|\nabla f(x_0)\|_\infty\}. \tag{2.104}$$

In a few cases, this criterion was too lenient. For example, with the test problem PENALTY1, the computed cost still differs from the optimal cost by a factor of 10^5 when the criterion (2.104) is satisfied. As a result, different solvers obtain completely different values for the cost, and the test problem would be discarded. By changing the convergence criterion to $\|\nabla f(x_k)\|_\infty \le 10^{-6}$, the computed costs all agreed to 6 digits. The problems for which the convergence criterion was strengthened were DQRTIC, PENALTY1, POWER, QUARTC, and VARDIM.

The CPU time in seconds and the number of iterations, function evaluations, and gradient evaluations for each of the methods are posted at the author's web site. Here we analyze the performance data using the profiles of Dolan and Moré [43]. That is, we plot the fraction p of problems for which any given method is

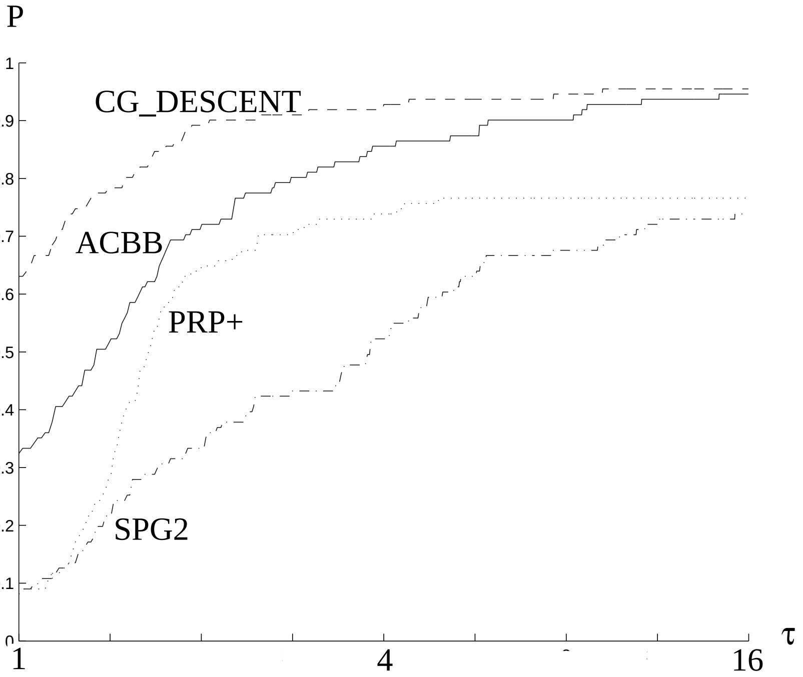

Figure 2–4: Performance based on CPU time

within a factor τ of the best time. In a plot of performance profiles, the top curve is the method that solved the most problems in a time that was within a factor τ of the best time. The percentage of the test problems for which a method is the fastest is given on the left axis of the plot. The right side of the plot gives the percentage of the test problems that were successfully solved by each of the methods. In essence, the right side is a measure of an algorithm's robustness.

In Figure 2–4, we use CPU time to compare the performance of the four codes ACBB, SPG2, PRP+, and CG_DESCENT. Note that the horizontal axis in Figure 2–4 is scaled proportional to $\log_2(\tau)$. The best performance, relative to the CPU time metric, was obtained by CG_DESCENT, the top curve in Figure 2–4, followed by ACBB. The horizontal axis in the figure stops at $\tau = 16$ since the plots are essentially flat for larger values of τ. For this collection of methods, the number of times any method achieved the best time is shown in Table 2–5. The column total in Table 2–5 exceeds 111 due to ties for some test problems.

Table 2–5: Number of times each method was fastest (time metric, stopping criterion (2.104))

Method	Fastest
CG_DESCENT	70
ACBB	36
PRP+	9
SPG2	9

The results of Figure 2–4 indicate that ACBB is much more efficient than SPG2, while it performed better than PRP+, but not as well as CG_DESCENT. ¿From the experience in [108], the GBB algorithm, with a traditional nonmonotone line search [66], may be affected significantly by nearly singular Hessians at the solution. We observe that nearly singular Hessians do not affect ACBB significantly. In fact, Table 2–4 also indicates that ACBB becomes more efficient as the problem becomes more singular. Furthermore, since ACBB does not need to calculate the BB stepsize at every iteration, CPU time is saved, which can be significant when the problem dimension is large. For this test set, we found that the average cycle length for ACBB was 2.59. In other words, the BB step is reevaluated after 2 or 3 iterations, on average. This memory length is smaller than the memory length that works well for quadratic function. When the iterates are far from a local minimizer of a general nonlinear function, the iterates may not behave like the iterates of a quadratic. In this case, better numerical results are obtained when the BB-stepsize is updated more frequently.

Even though ACBB did not perform as well as CG_DESCENT for the complete set of test problems, there were some cases where it performed exceptionally well (see Table 2–6). One important advantage of the ACBB scheme over conjugate gradient routines such as PRP+ or CG_DESCENT is that in many cases, the stepsize for ACBB is either the previous stepsize or the BB sizesize (2.51). In contrast, with conjugate gradient routines, each iteration requires a line search. Due to

Table 2–6: CPU times for selected problems

Problem	Dimension	ACBB	CG_DESCENT
FLETCHER	5000	9.14	989.55
FLETCHER	1000	1.32	27.27
BDQRTIC	1000	.37	3.40
VARDIM	10000	.05	2.13
VARDIM	5000	.02	.92

the simplicity of the ACBB stepsize, it can be more efficient when the iterates are in a regime where the function is irregular and the asymptotic convergence properties of the conjugate gradient method are not in effect. One such application is bound constrained optimization problems – as components of x reach the bounds, these components are often held fixed, and the associated partial derivative change discontinuously. In Chapter 3, ACBB is combined with CG_DESCENT to obtain a very efficient active set algorithm for box constrained optimization problems.

2.3 Self-adaptive Inexact Proximal Point Methods

2.3.1 Motivation and the Algorithm

The convergence rate of algorithms for solving an unconstrained optimization problem (2.1) often depends on the eigenvalues of the Hessian matrix at a local minimizer. As the ratio between largest and smallest eigenvalues grows, convergence rates can degrade. In this section, we propose a class of self-adaptive proximal point methods for ill conditioned problems where the smallest eigenvalue of the Hessian can be zero, and where the solution set $\mathbf{X}$ may have more than one element [77].

The proximal point method is one strategy for dealing with degeneracy at a minimum. The iterates $\mathbf{x}_k$, $k \geq 1$, are generated by the rule:

$$\mathbf{x}_{k+1} \in \arg\ \min\ \{F_k(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\}, \tag{2.105}$$

where

$$F_k(\mathbf{x}) = f(\mathbf{x}) + \frac{1}{2}\mu_k\|\mathbf{x} - \mathbf{x}_k\|^2.$$

Here $\mathbf{x}_0 \in \mathbb{R}^n$ is an initial guess for a minimizer and the parameters μ_k, $k \geq 0$, are positive scalars. Due to the quadratic term, F_k is strictly convex at a local minimizer. Hence, the proximal point method improves the conditioning at the expense of replacing the single minimization (2.1) by a sequence of minimizations (2.105).

The proximal point method, first proposed by Martinet [96, 97], has been studied in many papers including [69, 95, 84, 111, 112]. In [112] Rockafellar shows that if f is strongly convex at a solution of (2.1), then the proximal point method converges linearly when μ_k is bounded away from zero, and superlinearly when μ_k tends to zero. Here we develop linear and superlinear convergence results for problems that are not necessarily strongly convex.

Since the solution to (2.105) approximates a solution to (2.1), we do not need to solve (2.105) exactly. We analyze two criteria for the accuracy with which we solve (2.105). The first criterion is that an iterate $\mathbf{x}_{k+1}$ is acceptable when

$$\text{(C1)} \quad F_k(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) \quad \text{and} \quad \|\nabla F_k(\mathbf{x}_{k+1})\| \leq \mu_k\|\nabla f(\mathbf{x}_k)\|.$$

The second acceptance criterion is

$$\text{(C2)} \quad F_k(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k) \quad \text{and} \quad \|\nabla F_k(\mathbf{x}_{k+1})\| \leq \theta\mu_k\|\mathbf{x}_{k+1} - \mathbf{x}_k\|,$$

where $\theta < 1/\sqrt{2}$. In either case, we show, for μ_k sufficiently small, that

$$\|\mathbf{x}_{k+1} - \bar{\mathbf{x}}_{k+1}\| \leq C\mu_k\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|, \tag{2.106}$$

where C is a constant depending on local properties of f and $\bar{\mathbf{x}}$ is the projection of $\mathbf{x} \in \mathbb{R}^n$ onto $\mathbf{X}$:

$$\|\bar{\mathbf{x}} - \mathbf{x}\| = \min_{\boldsymbol{\chi} \in \mathbf{X}} \|\boldsymbol{\chi} - \mathbf{x}\|.$$

Since f is continuous, the set of minimizers $\mathbf{X}$ of (2.1) is closed, and the projection exists. By taking $\mu_k = \|\nabla f(\mathbf{x}_k)\|$ in (2.106), we obtain quadratic convergence of the approximate proximal iterates to the solution set $\mathbf{X}$, while the sequence of iterates approaches a limit at least linearly.

In [112] Rockafellar studies the acceptance condition

$$\|\nabla F_k(\mathbf{x}_{k+1})\| \le \epsilon_k \mu_k \|\mathbf{x}_{k+1} - \mathbf{x}_k\|$$

where $\sum_k \epsilon_k < \infty$ (see his condition B'). Our criterion (C2) corresponds to the case $\sum_k \epsilon_k = \infty$. (C2) is also studied in [82] where the authors give an introduction to proximal point algorithm, borrowing ideas from descent methods for unconstrained optimization. In [82] global convergence for convex functions is estabished, while here we obtain local convergence rates for general nonlinear functions.

2.3.2 Local Error Bound Condition

Our analysis of the approximate proximal iterates makes use of a local error bound condition employed when the Hessian is singular at a minimizer of f – see [54, 119, 127, 126]. Referring to [54] and [127], we have the following terminology: ∇f provides a local error bound at $\hat{\mathbf{x}} \in \mathbf{X}$ if there exist positive constants α and ρ such that

$$\|\nabla f(\mathbf{x})\| \ge \alpha \|\mathbf{x} - \bar{\mathbf{x}}\| \text{ whenever } \|\mathbf{x} - \hat{\mathbf{x}}\| \le \rho. \tag{2.107}$$

Using this condition, Yamashita and Fukushima [127], and Fan and Yuan [54], study the Levenberg-Marquardt method to solve a system of nonlinear equations. When their approach is applied to (1.1), the following linear system is solved in each subproblem:

$$(\mathbf{H}(\mathbf{x}_k)^2 + \mu_k \mathbf{I})\mathbf{d} + \mathbf{H}(\mathbf{x}_k)\mathbf{g}(\mathbf{x}_k) = \mathbf{0}, \tag{2.108}$$

where $\mu_k > 0$ is the regularization parameter and $\mathbf{d}$ is the search direction at step k. In [54] and [127], the authors choose $\mu_k = \|\mathbf{g}(\mathbf{x}_k)\|$ and $\mu_k = \|\mathbf{g}(\mathbf{x}_k)\|^2$, respectively. They show that if $\nabla f(\mathbf{x})$ provided a local error bound, then the iterates associated with (2.108) are locally quadratically convergent.

Li, Fukushima, Qi and Yamashita point out in [88] that the linear system of equations (2.108) may lose sparsity when $\mathbf{H}(\mathbf{x}_k)$ is squared; moreover, squaring the matrix squares the condition number of $\mathbf{H}(\mathbf{x}_k)$. Hence, they consider a search direction $\mathbf{d}$ chosen to satisfy:

$$(\mathbf{H}(\mathbf{x}_k) + \mu_k \mathbf{I})\mathbf{d} + \mathbf{g}(\mathbf{x}_k) = \mathbf{0}, \tag{2.109}$$

where $\mu_k = c\|\mathbf{g}(\mathbf{x}_k)\|$ for some constant $c > 0$. When $f(\mathbf{x})$ is convex and $\nabla f(\mathbf{x})$ provides a local error bound, they establish a local quadratic convergence result for iterates generated by the approximate solution of (2.109) followed by a line search along the approximate search direction.

When the problem dimension is large, computing the Hessian and solving (2.109) can be expensive. In our approach, we use our newly developed conjugate gradient routine CG_DESCENT [73, 72] to solve (2.105) with either stopping criterion (C1) or (C2).

First, by observing that the minimum of F_k is bounded from above by $F_k(\bar{\mathbf{x}}_k)$, we can easily get the following proposition.

Proposition 1 *If f is continuous and its set of minimizers $\mathbf{X}$ is nonempty, then for each k, F_k has a minimizer $\mathbf{x}_{k+1}$ and we have*

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \le \|\bar{\mathbf{x}}_k - \mathbf{x}_k\|.$$

Proof. The proposition follows directly from

$$\begin{aligned} F_k(\mathbf{x}_{k+1}) = f(\mathbf{x}_{k+1}) + \frac{1}{2}\mu_k\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 &\le F_k(\bar{\mathbf{x}}_k) = f(\bar{\mathbf{x}}_k) + \frac{1}{2}\mu_k\|\bar{\mathbf{x}}_k - \mathbf{x}_k\|^2 \\ &\le f(\mathbf{x}) + \frac{1}{2}\mu_k\|\bar{\mathbf{x}}_k - \mathbf{x}_k\|^2. \end{aligned}$$

□

In our approach, it is more convenient to employ a local error bound based on the function value rather than the function gradient used in (2.107). We say that f provides a local error bound at $\hat{\mathbf{x}} \in \mathbf{X}$ if there exist positive constants α and ρ such that

$$f(\mathbf{x}) - f(\bar{\mathbf{x}}) \geq \alpha \|\bar{\mathbf{x}} - \mathbf{x}\|^2 \text{ whenever } \|\mathbf{x} - \hat{\mathbf{x}}\| \leq \rho. \tag{2.110}$$

We now show that under certain smooth assumption, these two conditions are equivalent:

Lemma 5 *If f is twice continuously differentiable in a neighborhood of $\hat{\mathbf{x}} \in \mathbf{X}$, then f provides a local error bound at $\hat{\mathbf{x}}$ is equivalent to ∇f provides a local error bound at $\hat{\mathbf{x}}$.*

Proof. Suppose f provides a local error bound at $\hat{\mathbf{x}} \in \mathbf{X}$ with positive scalars α and ρ satisfying (2.110). Choose ρ smaller if necessary so that f is twice continuously differentiable in $\mathcal{B}_\rho(\hat{\mathbf{x}})$. Define $r = \rho/2$. Given $\mathbf{x} \in \mathcal{B}_r(\hat{\mathbf{x}})$, the triangle inequality implies that

$$\|\bar{\mathbf{x}} - \hat{\mathbf{x}}\| \leq \|\bar{\mathbf{x}} - \mathbf{x}\| + \|\mathbf{x} - \hat{\mathbf{x}}\| \leq 2r = \rho. \tag{2.111}$$

Since both $\mathbf{x}$ and $\bar{\mathbf{x}} \in \mathcal{B}_\rho(\hat{\mathbf{x}})$, we can expand $f(\mathbf{x})$ in a Taylor series around $\bar{\mathbf{x}}$ and apply (2.110) to obtain:

$$\begin{aligned} \alpha \|\mathbf{x} - \bar{\mathbf{x}}\|^2 &\leq f(\mathbf{x}) - f(\bar{\mathbf{x}}) \\ &= \frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^{\mathsf{T}} \bar{\mathbf{H}} (\mathbf{x} - \bar{\mathbf{x}}) + R_2(\mathbf{x}, \bar{\mathbf{x}}), \end{aligned} \tag{2.112}$$

where R_2 is the remainder term and $\bar{\mathbf{H}} = \nabla^2 f(\bar{\mathbf{x}})$ is the Hessian at $\bar{\mathbf{x}}$. Choose ρ small enough that

$$|R_2(\mathbf{x}, \bar{\mathbf{x}})| \leq \frac{\alpha}{3} \|\mathbf{x} - \bar{\mathbf{x}}\|^2 \text{ whenever } \mathbf{x} \in \mathcal{B}_\rho(\hat{\mathbf{x}}).$$

In this case, (2.112) gives

$$\frac{4\alpha}{3}\|\mathbf{x}-\bar{\mathbf{x}}\| \le \|\bar{\mathbf{H}}(\mathbf{x}-\bar{\mathbf{x}})\|. \tag{2.113}$$

Now expand $\nabla f(\mathbf{x})$ in a Taylor series around $\bar{\mathbf{x}}$ to obtain

$$\nabla f(\mathbf{x}) = \nabla f(\mathbf{x}) - \nabla f(\bar{\mathbf{x}}) = \bar{\mathbf{H}}(\mathbf{x}-\bar{\mathbf{x}}) + \mathbf{R}_1(\mathbf{x}, \bar{\mathbf{x}}), \tag{2.114}$$

where $\mathbf{R}_1$ is the remainder term. Choose ρ smaller if necessary so that

$$\|R_1(\mathbf{x}, \bar{\mathbf{x}})\| \le \frac{\alpha}{3}\|\mathbf{x}-\bar{\mathbf{x}}\| \text{ whenever } \mathbf{x} \in \mathcal{B}_\rho(\hat{\mathbf{x}}). \tag{2.115}$$

Combining (2.113)–(2.115) yields

$$\|\nabla f(\mathbf{x})\| \ge \alpha\|\mathbf{x}-\bar{\mathbf{x}}\|.$$

Hence, ∇f provides a local error bound at $\hat{\mathbf{x}}$ with constants α and $\rho/2$.

Conversely, suppose ∇f provides a local error bound at $\hat{\mathbf{x}} \in \mathbf{X}$ with positive scalars α and ρ satisfying (2.107). Choose ρ smaller if necessary so that f is twice continuously differentiable in $\mathcal{B}_\rho(\hat{\mathbf{x}})$ and (2.115) holds. Combining the Taylor expansion (2.114), the fact that $\bar{\mathbf{H}}$ is positive semidefinite, the bound (2.115) on the remainder, and the local error bound condition (2.107), we obtain

$$\frac{2\alpha}{3}\|\mathbf{x}-\bar{\mathbf{x}}\| \le \|\bar{\mathbf{H}}(\mathbf{x}-\bar{\mathbf{x}})\| \le \|\bar{\mathbf{H}}^{1/2}\|\|\bar{\mathbf{H}}^{1/2}(\mathbf{x}-\bar{\mathbf{x}})\|.$$

Squaring both sides gives

$$\frac{4\alpha^2}{9}\|\mathbf{x}-\bar{\mathbf{x}}\|^2 \le \|\bar{\mathbf{H}}\|\|\bar{\mathbf{H}}^{1/2}(\mathbf{x}-\bar{\mathbf{x}})\|^2 = \|\bar{\mathbf{H}}\|(\mathbf{x}-\bar{\mathbf{x}})^{\mathsf{T}}\bar{\mathbf{H}}(\mathbf{x}-\bar{\mathbf{x}}).$$

Consequently,

$$(\mathbf{x}-\bar{\mathbf{x}})^{\mathsf{T}}\bar{\mathbf{H}}(\mathbf{x}-\bar{\mathbf{x}}) \ge \left(\frac{4\alpha^2}{9\|\bar{\mathbf{H}}\|}\right)\|\mathbf{x}-\bar{\mathbf{x}}\|^2 \ge \left(\frac{4\alpha^2}{9\lambda}\right)\|\mathbf{x}-\bar{\mathbf{x}}\|^2, \tag{2.116}$$

where λ is a bound for the Hessian of f over $\mathcal{B}_\rho(\hat{\mathbf{x}})$. Similar to (2.112), we expand f in a Taylor expansion around $\bar{\mathbf{x}}$ and utilize (2.116) to obtain

$$f(\mathbf{x}) - f(\bar{\mathbf{x}}) - R_2(\mathbf{x}, \bar{\mathbf{x}}) = \frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^{\mathsf{T}}\bar{\mathbf{H}}(\mathbf{x} - \bar{\mathbf{x}}) \geq \beta\|\mathbf{x} - \bar{\mathbf{x}}\|^2, \quad \text{where} \quad \beta = \frac{2\alpha^2}{9\lambda}.$$

To complete the proof, choose ρ smaller if necessary so that

$$|R_2(\mathbf{x}, \bar{\mathbf{x}})| \leq \frac{\beta}{2}\|\mathbf{x} - \bar{\mathbf{x}}\|^2$$

whenever $\mathbf{x} \in \mathcal{B}_\rho(\hat{\mathbf{x}})$. □

2.3.3 Local Convergence

Convergence analysis for exact minimization. We first analyze the proximal point method when the iterates are exact solutions of (2.105).

Theorem 9 *Assume the following conditions are satisfied:*

(E1) *f provides a local error bound at $\hat{\mathbf{x}} \in \mathbf{X}$ with positive scalars α and ρ satisfying (2.110).*

(E2) *$\mu_k \leq \beta$ for each k where $\beta < 2\alpha/3$.*

(E3) *$\mathbf{x}_0$ is close enough to $\hat{\mathbf{x}}$ that*

$$\|\mathbf{x}_0 - \hat{\mathbf{x}}\| \left(1 + \frac{1}{1-\gamma}\right) \leq \rho, \ \text{where } \gamma = \frac{2\beta}{2\alpha - \beta}. \tag{2.117}$$

Then the proximal iterates $\mathbf{x}_k$ are all contained in $\mathcal{B}_\rho(\hat{\mathbf{x}})$ and they approach a minimizer $\mathbf{x}^* \in \mathbf{X}$; for each k, we have

$$\begin{aligned} \|\mathbf{x}_k - \mathbf{x}^*\| &\leq \frac{\gamma^k}{1-\gamma}\|\bar{\mathbf{x}}_0 - \mathbf{x}_0\| \ \text{and} \\ \|\mathbf{x}_{k+1} - \bar{\mathbf{x}}_{k+1}\| &\leq \frac{2\mu_k}{2\alpha - \mu_k}\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|. \end{aligned} \tag{2.118}$$

Proof. For $j = 0$, we have

$$\|\mathbf{x}_j - \hat{\mathbf{x}}\| \leq \rho \quad \text{and} \quad \|\bar{\mathbf{x}}_j - \mathbf{x}_j\| \leq \gamma^j\|\bar{\mathbf{x}}_0 - \mathbf{x}_0\|. \tag{2.119}$$

Proceeding by induction, suppose that (2.119) holds for all $j \in [0, k]$ and for some $k \geq 0$. We show that (2.119) holds for all $j \in [0, k+1]$. By Proposition 1 and the induction hypothesis,

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \leq \|\bar{\mathbf{x}}_k - \mathbf{x}_k\| \leq \gamma^k \|\bar{\mathbf{x}}_0 - \mathbf{x}_0\|.$$

By (E2), we have $\beta < 2\alpha/3$ and hence,

$$\gamma = \frac{2\beta}{2\alpha - \beta} < 1.$$

By the triangle inequality, Proposition 1, and the induction hypothesis, it follows that

$$\begin{aligned} \|\mathbf{x}_{k+1} - \mathbf{x}_0\| &\leq \sum_{j=0}^{k} \|\mathbf{x}_{j+1} - \mathbf{x}_j\| \leq \sum_{j=0}^{k} \|\bar{\mathbf{x}}_j - \mathbf{x}_j\| \\ &\leq \sum_{j=0}^{k} \gamma^j \|\bar{\mathbf{x}}_0 - \mathbf{x}_0\| \leq \frac{1}{1-\gamma}\|\bar{\mathbf{x}}_0 - \mathbf{x}_0\| \leq \frac{1}{1-\gamma}\|\mathbf{x}_0 - \hat{\mathbf{x}}\|. \end{aligned}$$

Again, by the triangle inequality and (2.117),

$$\|\mathbf{x}_{k+1} - \hat{\mathbf{x}}\| \leq \|\mathbf{x}_{k+1} - \mathbf{x}_0\| + \|\mathbf{x}_0 - \hat{\mathbf{x}}\| \leq \left(1 + \frac{1}{1-\gamma}\right) \|\mathbf{x}_0 - \hat{\mathbf{x}}\| \leq \rho. \tag{2.120}$$

Observe that

$$\begin{aligned} &\|\bar{\mathbf{x}}_{k+1} - \mathbf{x}_k\|^2 - \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\ &\qquad = (\bar{\mathbf{x}}_{k+1} + \mathbf{x}_{k+1} - 2\mathbf{x}_k)(\bar{\mathbf{x}}_{k+1} - \mathbf{x}_{k+1}) \\ &\qquad \leq (\|\bar{\mathbf{x}}_{k+1} - \mathbf{x}_{k+1}\| + 2\|\mathbf{x}_{k+1} - \mathbf{x}_k\|)\|\bar{\mathbf{x}}_{k+1} - \mathbf{x}_{k+1}\|. \end{aligned} \tag{2.121}$$

Combining (2.121) with the relation $F_k(\mathbf{x}_{k+1}) \leq F_k(\bar{\mathbf{x}}_{k+1})$ gives

$$\begin{aligned} &f(\mathbf{x}_{k+1}) - f(\bar{\mathbf{x}}_{k+1}) \\ &\qquad = \frac{1}{2}\mu_k(\|\bar{\mathbf{x}}_{k+1} - \mathbf{x}_k\|^2 - \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2) \\ &\qquad \leq \frac{1}{2}\mu_k(\|\bar{\mathbf{x}}_{k+1} - \mathbf{x}_{k+1}\| + 2\|\mathbf{x}_{k+1} - \mathbf{x}_k\|)\|\bar{\mathbf{x}}_{k+1} - \mathbf{x}_{k+1}\|. \end{aligned} \tag{2.122}$$

By (2.120), $\mathbf{x}_{k+1} \in \mathcal{B}_\rho(\hat{\mathbf{x}})$. Since f provides a local error bound at $\hat{\mathbf{x}}$, we conclude that

$$\alpha \|\mathbf{x}_{k+1} - \bar{\mathbf{x}}_{k+1}\|^2 \le f(\mathbf{x}_{k+1}) - f(\bar{\mathbf{x}}_{k+1}). \tag{2.123}$$

Combining this with (2.122) gives

$$\left(\alpha - \frac{1}{2}\mu_k\right) \|\mathbf{x}_{k+1} - \bar{\mathbf{x}}_{k+1}\| \le \mu_k \|\mathbf{x}_{k+1} - \mathbf{x}_k\|. \tag{2.124}$$

Due to the assumption $\mu_k < 2\alpha/3$, the coefficient $(\alpha - \frac{1}{2}\mu_k)$ in (2.124) is positive and

$$2\mu_k/(2\alpha - \mu_k) \le 2\beta/(2\alpha - \beta) = \gamma < 1.$$

Hence, (2.124), Proposition 1, and (2.119), with $j = k$, give

$$\begin{aligned}
\|\mathbf{x}_{k+1} - \bar{\mathbf{x}}_{k+1}\| &\le \left(\frac{2\mu_k}{2\alpha - \mu_k}\right) \|\mathbf{x}_{k+1} - \mathbf{x}_k\| \le \left(\frac{2\mu_k}{2\alpha - \mu_k}\right) \|\bar{\mathbf{x}}_k - \mathbf{x}_k\| && (2.125)\\
&\le \gamma \|\bar{\mathbf{x}}_k - \mathbf{x}_k\| \le \gamma^{k+1} \|\bar{\mathbf{x}}_0 - \mathbf{x}_0\| && (2.126)
\end{aligned}$$

Relations (2.120) and (2.126) complete the proof of the induction step. Relation (2.125) gives estimate (2.118).

By (2.119) and Proposition 1, we have

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \le \|\bar{\mathbf{x}}_k - \mathbf{x}_k\| \le \gamma^k \|\bar{\mathbf{x}}_0 - \mathbf{x}_0\|.$$

Hence, the proximal iterates $\mathbf{x}_k$ form a Cauchy sequence, which has a limit denoted $\mathbf{x}^*$. Again, it follows from (2.119) and Proposition 1 that

$$\begin{aligned}
\|\mathbf{x}_k - \mathbf{x}^*\| &\le \sum_{j=k}^{\infty} \|\mathbf{x}_{j+1} - \mathbf{x}_j\| \le \sum_{j=k}^{\infty} \|\bar{\mathbf{x}}_j - \mathbf{x}_j\| \\
&\le \sum_{j=k}^{\infty} \gamma^j \|\bar{\mathbf{x}}_0 - \mathbf{x}_0\| = \frac{\gamma^k}{1-\gamma} \|\bar{\mathbf{x}}_0 - \mathbf{x}_0\|.
\end{aligned}$$

By (2.125)–(2.126), $\mathbf{x}^* \in \mathbf{X}$. □

Note that neither smoothness nor convexity assumptions enter into the convergence results of Theorem 9. In a further extension of these results, let us consider the case where the regularization sequence μ_k of Theorem 9 is expressed as a function of the current iterate. That is, we assume that $\mu_k = \mu(\mathbf{x}_k)$ where $\mu(\cdot)$ is defined on $\mathbb{R}^n$.

Corollary 1 *We make the following assumptions:*

(Q1) *f provides a local error bound at $\hat{\mathbf{x}} \in \mathbf{X}$ with positive scalars α and ρ satisfying (2.110).*

(Q2) *∇f is Lipschitz continuously differentiable in $\mathcal{B}_\rho(\hat{\mathbf{x}})$ with Lipschitz constant L.*

(Q3) *ρ is small enough that for some scalar β, we have*

$$\|\nabla f(\mathbf{x})\| \le \beta < \frac{2\alpha}{3} \textit{ for all } \mathbf{x} \in \mathcal{B}_r(\hat{\mathbf{x}}) \textit{ where } r = \rho/2.$$

(Q4) *$\mathbf{x}_0$ is close enough to $\hat{\mathbf{x}}$ that*

$$\|\mathbf{x}_0 - \hat{\mathbf{x}}\| \left(1 + \frac{1}{1-\gamma}\right) \le r, \textit{ where } \gamma = \frac{2\beta}{2\alpha - \beta}.$$

Then for the choice $\mu(\mathbf{x}) = \|\nabla f(\mathbf{x})\|$ and $\mu_k = \mu(\mathbf{x}_k)$, the proximal iterates (2.105) are all contained in $\mathcal{B}_r(\hat{\mathbf{x}})$ and they approach a minimizer of f. Moreover, we have

$$\|\mathbf{x}_{k+1} - \bar{\mathbf{x}}_{k+1}\| \le \left(\frac{3L}{2\alpha}\right) \|\mathbf{x}_k - \bar{\mathbf{x}}_k\|^2 \tag{2.127}$$

for each k.

Proof. The proof is identical to the inductive proof of Theorem 9 except that we append the condition $\mu_j \le \beta$ for each $j \in [0, k]$ to the induction hypothesis (2.119). That is, we assume that for all $j \in [0, k]$, we have

$$\|\mathbf{x}_j - \hat{\mathbf{x}}\| \le r, \quad \|\bar{\mathbf{x}}_j - \mathbf{x}_j\| \le \gamma^j \|\bar{\mathbf{x}}_0 - \mathbf{x}_0\|, \quad \text{and} \quad \mu_j \le \beta. \tag{2.128}$$

Since $\mathbf{x}_0 \in \mathcal{B}_r(\hat{\mathbf{x}})$, it follows by (Q3) that $\mu_0 = \|\nabla f(\mathbf{x}_0)\| \le \beta$. Hence, (2.128) is satisfied for $j = 0$. In the proof of Theorem 9, we show that if $\mu_j \le \beta$ for $j \in [0, k]$, then the first two conditions in (2.128) hold for $j = k + 1$. Also, as in (2.120), $\mathbf{x}_{k+1} \in \mathcal{B}_r(\hat{\mathbf{x}})$. Consequently, $\mu_{k+1} = \|\nabla f(\mathbf{x}_{k+1})\| \le \beta$, which implies that the last condition in (2.128) holds for $j = k + 1$. This completes the induction step; hence, (2.128) holds for all $j \geq 0$.

Since $\mathbf{x}_k \in \mathcal{B}_r(\hat{\mathbf{x}})$, it follows that $\bar{\mathbf{x}}_k \in \mathcal{B}_\rho(\hat{\mathbf{x}})$ (see (2.111)). Since $\mathbf{x}_k$ and $\bar{\mathbf{x}}_k \in \mathcal{B}_\rho(\hat{\mathbf{x}})$, we have

$$\mu_k = \|\nabla f(\mathbf{x}_k)\| = \|\nabla f(\mathbf{x}_k) - \nabla f(\bar{\mathbf{x}}_k)\| \leq L\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|, \tag{2.129}$$

where L is the Lipschitz constant for ∇f in $\mathcal{B}_\rho(\hat{\mathbf{x}})$. By estimate (2.118) in Theorem 9,

$$\|\mathbf{x}_{k+1} - \bar{\mathbf{x}}_{k+1}\| \leq \left(\frac{2\mu_k}{2\alpha - \mu_k}\right)\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|.$$

Using the bound (2.129) in the numerator and the bound $\mu_k \leq \beta < 2\alpha/3$ in the denominator, we obtain (2.127). □

Convergence analysis for approximate minimization. We now analyze the situation where the proximal point iteration (2.105) is implemented inexactly; the approximation to a solution of (2.105) need only satisfy (C1) or (C2). The following property of a convex function is used in the analysis.

Proposition 2 *If $\mathbf{x}^*$ is a local minimizer of F_k and f is convex and continuously differentiable in a convex neighborhood $\mathcal{N}$ of $\mathbf{x}^*$, then*

$$F_k(\mathbf{x}) \leq F_k(\mathbf{x}^*) + \frac{\|\nabla F_k(\mathbf{x})\|^2}{\mu_k}$$

for all $\mathbf{x} \in \mathcal{N}$.

Proof. The convexity of f in $\mathcal{N}$ implies that F_k is convex in $\mathcal{N}$ and

$$F_k(\mathbf{x}^*) \geq F_k(\mathbf{x}) + \nabla F_k(\mathbf{x})(\mathbf{x}^* - \mathbf{x}).$$

Since $\mathbf{x}^*$ is a local minimizer of F_k, we have

$$\begin{aligned}\nabla F_k(\mathbf{x})(\mathbf{x} - \mathbf{x}^*) &= (\nabla F_k(\mathbf{x}) - \nabla F_k(\mathbf{x}^*))(\mathbf{x} - \mathbf{x}^*) \\ &= (\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) + \mu_k\|\mathbf{x} - \mathbf{x}^*\|^2. \end{aligned} \tag{2.130}$$

Since f is convex in $\mathcal{N}$, the monotonicity condition

$$(\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}^*))(\mathbf{x} - \mathbf{x}^*) \geq 0$$

holds. Combining this with (2.130), we obtain

$$\nabla F_k(\mathbf{x})(\mathbf{x} - \mathbf{x}^*) \geq \mu_k \|\mathbf{x} - \mathbf{x}^*\|^2. \tag{2.131}$$

By the Schwarz inequality,

$$\|\mathbf{x} - \mathbf{x}^*\| \leq \frac{\|\nabla F_k(\mathbf{x})\|}{\mu_k}. \tag{2.132}$$

Combining (2.131) and (2.132), the proof is complete. □

Our convergence result for the inexact proximal point iterates is now established.

Theorem 10 *Assume that the following conditions are satisfied:*

(A1) *f provides a local error bound at $\hat{\mathbf{x}} \in \mathbf{X}$ with positive scalars α and ρ satisfying (2.110).*

(A2) *f is twice continuously differentiable and convex throughout $\mathcal{B}_\rho(\hat{\mathbf{x}})$; let L be a Lipschitz constant for ∇f in $\mathcal{B}_\rho(\hat{\mathbf{x}})$.*

(A3) *The parameter $\beta = \sup\{\mu(\mathbf{x}) : \mathbf{x} \in \mathcal{B}_\rho(\hat{\mathbf{x}})\}$ satisfies*

$$\beta < \alpha/\Lambda \tag{2.133}$$

where

$$\Lambda = L + \tau \quad \textit{and} \quad \tau^2 = 1 + 2L^2 \quad \text{if acceptance criterion (C1) is used,}$$

while

$$\Lambda = \tau(1 + \theta) \quad \textit{and} \quad \tau^2 = \frac{1}{1 - 2\theta^2} \quad \text{if acceptance criterion (C2) is used.}$$

(A4) *The parameter*

$$\epsilon = \tau \|\mathbf{x}_0 - \hat{\mathbf{x}}\| \left(1 + \frac{1}{1-\gamma}\right), \quad \textit{where} \quad \gamma = \frac{\beta\Lambda}{\alpha},$$

satisfies

$$\epsilon + \sup\left\{\sqrt{\frac{\lambda\|\mathbf{x} - \bar{\mathbf{x}}\|^2}{2\mu(\mathbf{x})}} : \mathbf{x} \in \mathcal{B}_\epsilon(\hat{\mathbf{x}}), \mathbf{x} \notin \mathbf{X}\right\} \le r, \quad \textit{where} \quad r = \rho/2,$$

and λ is any upper bound for the largest eigenvalue of $\nabla^2 f(\mathbf{x})$ over $\mathbf{x} \in \mathcal{B}_\rho(\hat{\mathbf{x}})$. If the approximate proximal iterates $\mathbf{x}_k$ satisfy either (C1) *or* (C2) *with $\mu_k = \mu(\mathbf{x}_k)$, then the iterates are all contained in $\mathcal{B}_\epsilon(\hat{\mathbf{x}})$, they approach a minimizer $\mathbf{x}^* \in \mathbf{X}$, and for each k, we have*

$$\begin{aligned} \|\mathbf{x}_k - \mathbf{x}^*\| &\le \frac{\tau\gamma^k}{1-\gamma}\|\bar{\mathbf{x}}_0 - \mathbf{x}_0\| \textit{ and} && (2.134) \\ \|\mathbf{x}_{k+1} - \bar{\mathbf{x}}_{k+1}\| &\le \left(\frac{\Lambda\mu_k}{\alpha}\right)\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|, && (2.135) \end{aligned}$$

where Λ is defined in (A3)*.*

Proof. The following relations hold trivially for $j = 0$ since the index range for the summation is vacuous and $\tau \ge 1$:

$$\|\mathbf{x}_j - \hat{\mathbf{x}}\| \le \tau\|\mathbf{x}_0 - \hat{\mathbf{x}}\| \left(1 + \sum_{l=0}^{j-1} \gamma^l\right) \quad \text{and} \quad \|\bar{\mathbf{x}}_j - \mathbf{x}_j\| \le \gamma^j \|\bar{\mathbf{x}}_0 - \mathbf{x}_0\|. \qquad (2.136)$$

Proceeding by induction, suppose that (2.136) holds for all $j \in [0, k]$ and for some $k \ge 0$. We show that (2.136) holds for all $j \in [0, k+1]$.

The condition $F_k(\mathbf{x}_{k+1}) \le f(\mathbf{x}_k)$ in (C1) or (C2) implies that

$$\mu_k\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \le f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \le f(\mathbf{x}_k) - f(\bar{\mathbf{x}}_k). \qquad (2.137)$$

Since $\gamma < 1$ by (2.133), the first half of (2.136) with $j = k$ implies that $\mathbf{x}_k \in \mathcal{B}_\epsilon(\hat{\mathbf{x}})$, where $\epsilon \le r = \rho/2$. Thus we have $\bar{\mathbf{x}}_k \in \mathcal{B}_\rho(\hat{\mathbf{x}})$ (see (2.111)). Expanding f in (2.137) in a Taylor series around $\bar{\mathbf{x}}_k$ and using the fact that $\nabla f(\bar{\mathbf{x}}_k) = \mathbf{0}$ gives

$$\mu_k \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \leq \frac{\lambda}{2}\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|^2, \tag{2.138}$$

where λ is the bound for the Hessian of f over $\mathcal{B}_\rho(\hat{\mathbf{x}})$. By the triangle inequality, we have

$$\|\mathbf{x}_{k+1} - \hat{\mathbf{x}}\| \leq \|\mathbf{x}_{k+1} - \mathbf{x}_k\| + \|\mathbf{x}_k - \hat{\mathbf{x}}\|. \tag{2.139}$$

Combining (2.139) with the condition $\mathbf{x}_k \in \mathcal{B}_\epsilon(\hat{\mathbf{x}})$, (2.138), and (A4) yields:

$$\|\mathbf{x}_{k+1} - \hat{\mathbf{x}}\| \leq \epsilon + \|\mathbf{x}_{k+1} - \mathbf{x}_k\| \leq \epsilon + \sqrt{\frac{\lambda}{2\mu_k}}\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| \leq r.$$

Hence, $\mathbf{x}_{k+1} \in \mathcal{B}_r(\hat{\mathbf{x}})$.

Let $\hat{\mathbf{x}}_{k+1}$ denote an exact proximal point iterate:

$$\hat{\mathbf{x}}_{k+1} \in \arg\ \min \{F_k(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\}.$$

By Proposition 1 and (2.136), we have

$$\|\hat{\mathbf{x}}_{k+1} - \mathbf{x}_k\| \leq \|\bar{\mathbf{x}}_k - \mathbf{x}_k\| \leq \gamma^k \|\bar{\mathbf{x}}_0 - \mathbf{x}_0\|. \tag{2.140}$$

By the triangle inequality, (2.140), the fact that $\tau \geq 1$, and (2.136), we obtain

$$\begin{aligned}
\|\hat{\mathbf{x}}_{k+1} - \hat{\mathbf{x}}\| &\leq \|\hat{\mathbf{x}}_{k+1} - \mathbf{x}_k\| + \|\mathbf{x}_k - \hat{\mathbf{x}}\| \\
&\leq \tau\gamma^k \|\bar{\mathbf{x}}_0 - \mathbf{x}_0\| + \|\mathbf{x}_k - \hat{\mathbf{x}}\| \\
&\leq \tau\gamma^k \|\bar{\mathbf{x}}_0 - \mathbf{x}_0\| + \tau\|\mathbf{x}_0 - \hat{\mathbf{x}}_0\| \left(1 + \sum_{l=0}^{k-1} \gamma^l\right) \\
&\leq \tau\|\mathbf{x}_0 - \hat{\mathbf{x}}\| \left(1 + \sum_{l=0}^{k} \gamma^l\right).
\end{aligned}$$

Referring to the definition of ϵ, we have $\hat{\mathbf{x}}_{k+1} \in \mathcal{B}_\epsilon(\hat{\mathbf{x}})$, where $\epsilon \leq r = \rho/2$.

By assumption (A1), f provides a local error bound with constants α and ρ. Hence, by Lemma 5, ∇f provides a local error bound with constants α and $r = \rho/2$. Since $\nabla F(\mathbf{x}_{k+1}) = \nabla f(\mathbf{x}_{k+1}) + \mu_k(\mathbf{x}_{k+1} - \mathbf{x}_k)$ and $\mathbf{x}_{k+1} \in \mathcal{B}_r(\hat{\mathbf{x}})$, the local error bound condition gives

$$\alpha\|\mathbf{x}_{k+1} - \bar{\mathbf{x}}_{k+1}\| \le \|\nabla f(\mathbf{x}_{k+1})\| \le \|\nabla F_k(\mathbf{x}_{k+1})\| + \mu_k\|\mathbf{x}_{k+1} - \mathbf{x}_k\|. \tag{2.141}$$

If (C1) is used, then $\|\nabla F_k(\mathbf{x}_{k+1})\| \le \mu_k\|\nabla f(\mathbf{x}_k)\|$, and (2.141) implies that

$$\begin{aligned} \alpha\|\mathbf{x}_{k+1} - \bar{\mathbf{x}}_{k+1}\| &\le \mu_k(\|\nabla f(\mathbf{x}_k)\| + \|\mathbf{x}_{k+1} - \mathbf{x}_k\|) \\ &\le \mu_k(L\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| + \|\mathbf{x}_{k+1} - \mathbf{x}_k\|). \end{aligned} \tag{2.142}$$

If (C2) is used, then $\|\nabla F_k(\mathbf{x}_{k+1})\| \le \theta\mu_k\|\mathbf{x}_{k+1} - \mathbf{x}_k\|$, and by (2.141), we have

$$\alpha\|\mathbf{x}_{k+1} - \bar{\mathbf{x}}_{k+1}\| \le \mu_k(1+\theta)\|\mathbf{x}_{k+1} - \mathbf{x}_k\|. \tag{2.143}$$

We now derive a bound for the $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|$ term in either (2.142) or (2.143). Since both $\mathbf{x}_{k+1}$ and $\hat{\mathbf{x}}_{k+1}$ lie in $\mathcal{B}_\rho(\hat{\mathbf{x}})$, we can apply Proposition 2 to obtain

$$\begin{aligned} F_k(\mathbf{x}_{k+1}) &= F_k(\hat{\mathbf{x}}_{k+1}) + (F_k(\mathbf{x}_{k+1}) - F_k(\hat{\mathbf{x}}_{k+1})) \\ &\le F_k(\bar{\mathbf{x}}_k) + \frac{1}{\mu_k}\|\nabla F_k(\mathbf{x}_{k+1})\|^2. \end{aligned} \tag{2.144}$$

Above we observed that $\mathbf{x}_k \in \mathcal{B}_\epsilon(\hat{\mathbf{x}})$ where $\epsilon \le r = \rho/2$. In (2.111) we show that $\bar{\mathbf{x}}_k \in \mathcal{B}_\rho(\hat{\mathbf{x}})$ when $\mathbf{x}_k \in \mathcal{B}_r(\hat{\mathbf{x}})$. Hence, by (A2) we have

$$\|\nabla f(\mathbf{x}_k)\| = \|\nabla f(\mathbf{x}_k) - \nabla f(\bar{\mathbf{x}}_k)\| \le L\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|. \tag{2.145}$$

If (C1) is used, then by (2.145),

$$\|\nabla F_k(\mathbf{x}_{k+1})\| \le \mu_k\|\nabla f(\mathbf{x}_k)\| \le \mu_k L\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|. \tag{2.146}$$

Using the relation $f(\bar{\mathbf{x}}_k) \le f(\mathbf{x}_{k+1})$ in (2.144) gives:

$$\frac{\mu_k}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \le \frac{\mu_k}{2}\|\bar{\mathbf{x}}_k - \mathbf{x}_k\|^2 + \frac{1}{\mu_k}\|\nabla F_k(\mathbf{x}_{k+1})\|^2. \tag{2.147}$$

Combining this with (2.146), we have

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \le (1 + 2L^2)\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|^2. \tag{2.148}$$

Similarly, if criterion (C2) is used, then $\|\nabla F_k(\mathbf{x}_{k+1})\| \le \theta\mu_k\|\mathbf{x}_{k+1} - \mathbf{x}_k\|$, and by (2.147), we have

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \le \frac{1}{1 - 2\theta^2}\|\mathbf{x}_k - \bar{\mathbf{x}}_k\|^2. \tag{2.149}$$

Combining (2.148) and (2.149) and referring to the definition of τ in (A3), we conclude that

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \le \tau\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| \tag{2.150}$$

if either acceptance criterion (C1) or (C2) is used.

Inserting the bound (2.150) in (2.142) or (2.143) yields (2.135). By (2.135) and the definition of β in (A3), we have

$$\|\mathbf{x}_{k+1} - \bar{\mathbf{x}}_{k+1}\| \le \left(\frac{\Lambda\mu_k}{\alpha}\right)\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| \le \gamma\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| \le \gamma^{k+1}\|\mathbf{x}_0 - \bar{\mathbf{x}}_0\|.$$

This establishes the second half of (2.136) for $j = k + 1$.

For the first half of (2.136), we use the triangle inequality, (2.150), and the induction hypothesis to obtain:

$$\begin{aligned}
\|\mathbf{x}_{k+1} - \hat{\mathbf{x}}\| &\le \|\mathbf{x}_{k+1} - \mathbf{x}_k\| + \|\mathbf{x}_k - \hat{\mathbf{x}}\| \\
&\le \tau\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| + \|\mathbf{x}_k - \hat{\mathbf{x}}\| \\
&\le \tau\gamma^k\|\mathbf{x}_0 - \bar{\mathbf{x}}_0\| + \|\mathbf{x}_k - \hat{\mathbf{x}}\| \\
&\le \tau\gamma^k\|\mathbf{x}_0 - \bar{\mathbf{x}}_0\| + \tau\|\mathbf{x}_0 - \hat{\mathbf{x}}\|\left(1 + \sum_{l=0}^{k-1}\gamma^l\right) \\
&\le \tau\|\mathbf{x}_0 - \hat{\mathbf{x}}\|\left(1 + \sum_{l=0}^{k}\gamma^l\right)
\end{aligned}$$

This completes the proof of the induction step, and in particular, it shows that $\mathbf{x}_{k+1} \in \mathcal{B}_\epsilon(\hat{\mathbf{x}})$, where ϵ is defined in (A4).

By (2.136) and (2.150), we have

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \le \tau \|\mathbf{x}_k - \bar{\mathbf{x}}_k\| \le \tau\gamma^k \|\mathbf{x}_0 - \bar{\mathbf{x}}_0\|. \tag{2.151}$$

Hence, the $\mathbf{x}_k$ form a Cauchy sequence, which has a limit denoted $\mathbf{x}^*$. By the triangle inequality and (2.151),

$$\begin{aligned} \|\mathbf{x}_k - \mathbf{x}^*\| &\le \sum_{j=k}^{\infty} \|\mathbf{x}_{j+1} - \mathbf{x}_j\| \le \tau \sum_{j=k}^{\infty} \|\bar{\mathbf{x}}_j - \mathbf{x}_j\| \\ &\le \tau \sum_{j=k}^{\infty} \gamma^j \|\bar{\mathbf{x}}_0 - \mathbf{x}_0\| = \frac{\tau\gamma^k}{1-\gamma} \|\bar{\mathbf{x}}_0 - \mathbf{x}_0\|. \end{aligned}$$

This establishes (2.134). □

Remark. In our analysis of the exact proximal point algorithm, our proof of Theorem 9 could rely on the bound provided by Proposition 1 for the step size $\mathbf{x}_{k+1} - \mathbf{x}_k$. In Theorem 10, we obtain a similar bound using either condition (C1) or (C2) along with the relationship between the regularization parameter μ_k and the current iterate $\mathbf{x}_k$. Condition (A4) is satisfied if

$$\lim_{\mathbf{x} \to \hat{\mathbf{x}}} \frac{\|\mathbf{x} - \bar{\mathbf{x}}\|^2}{\mu(\mathbf{x})} = 0 \tag{2.152}$$

and $\mathbf{x}_0$ is sufficiently close to $\hat{\mathbf{x}}$. For example, if $\mu(\mathbf{x}) = \beta\|\nabla f(\mathbf{x})\|^\eta$ where $\eta \in [0,2)$ and $\beta > 0$ is a constant, and if ∇f provides a local error bound at $\hat{\mathbf{x}}$, then

$$\frac{\|\mathbf{x} - \bar{\mathbf{x}}\|^2}{\mu(\mathbf{x})} = \frac{\|\mathbf{x} - \bar{\mathbf{x}}\|^2}{\beta\|\nabla f(\mathbf{x})\|^\eta} \le \left(\frac{1}{\beta\alpha^\eta}\right) \|\mathbf{x} - \bar{\mathbf{x}}\|^{2-\eta}.$$

Hence, (2.152) holds for $\eta \in [0,2)$. Also, (A3) holds if either $\eta \in (0,2)$ and ρ is sufficiently small, or $\eta = 0$ and β is sufficiently small. For this choice of $\mu(\cdot)$, it follows from Theorem 10, that the convergence rate to the set of minimizers $\mathbf{X}$ is linear for $\eta = 0$, superlinear for $\eta \in (0,1)$, and at least quadratic for $\eta \in [1,2)$.

2.3.4 Global Convergence

We now establish a global convergence result. It is assumed that the algorithm used to approximately solve (2.105) starts with a direction $\mathbf{d}_k$ which satisfies (L1) below, and which employs a line search in the direction $\mathbf{d}_k$ which satisfies (L2) below:

(L1) There exist positive constants c_1 and c_2 satisfying both the sufficient descent condition

$$\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k \le -c_1\|\mathbf{g}_k\|^2,$$

and the search direction bound

$$\|\mathbf{d}_k\| \le c_2\|\mathbf{g}_k\|,$$

where $\mathbf{g}_k = \nabla f(\mathbf{x}_k)^{\mathsf{T}} = \nabla F_k(\mathbf{x}_k)$.

(L2) The Wolfe conditions [122, 123] hold. That is, if α_k denotes the stepsize and $\mathbf{y}_k = \mathbf{x}_k + \alpha_k\mathbf{d}_k$, then

$$F_k(\mathbf{y}_k) \le F_k(\mathbf{x}_k) + \delta\alpha_k\nabla F_k(\mathbf{x}_k)\mathbf{d}_k = F_k(\mathbf{x}_k) + \delta\alpha_k\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k,$$

and

$$\nabla F_k(\mathbf{y}_k)\mathbf{d}_k \ge \sigma\nabla F_k(\mathbf{x}_k)\mathbf{d}_k = \sigma\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k.$$

Moreover, since $\mathbf{y}_k$ is an intermediate point in the move from $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$, we require that $F_k(\mathbf{y}_k) \ge F_k(\mathbf{x}_{k+1})$.

Our global convergence result is as follows:

Theorem 11 *Let $\mathbf{x}_k$, $k \ge 1$, denote a sequence of inexact proximal point iterates associated with (2.105). We assume that the algorithm used to approximately solve (2.105) satisfies* (L1) *and* (L2). *If the iterates are all contained in a convex set $\mathcal{B}$ where ∇f is Lipschitz continuous, and if $\mu_k \le \beta\|\nabla f(\mathbf{x}_k)\|^\eta$ for some $\eta \in [0, 2)$ and* *some constant β, then we have*

$$\lim_{k\to\infty} \mathbf{g}(\mathbf{x}_k) = \mathbf{0}.$$

Proof. Since $\mathbf{X}$ is nonempty, f is bounded from below. By (L2) and the definition of F_k,

$$f(\mathbf{x}_{k+1}) \le F_k(\mathbf{x}_{k+1}) \le F_k(\mathbf{y}_k) \le F_k(\mathbf{x}_k) = f(\mathbf{x}_k).$$

Hence, we have

$$\begin{aligned} \infty &> \sum_{k=0}^{\infty} f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) = \sum_{k=0}^{\infty} F_k(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) \\ &\ge \sum_{k=0}^{\infty} F_k(\mathbf{x}_k) - F_k(\mathbf{x}_{k+1}) \\ &\ge \sum_{k=0}^{\infty} F_k(\mathbf{x}_k) - F_k(\mathbf{y}_k). \end{aligned} \tag{2.153}$$

By (L1) and the first Wolfe condition in (L2), it follows that

$$F_k(\mathbf{y}_k) \le F_k(\mathbf{x}_k) - \delta c_1 \alpha_k \|\mathbf{g}_k\|^2.$$

Combining this with (2.153) gives

$$\sum_{k=0}^{\infty} \alpha_k \|\mathbf{g}(\mathbf{x}_k)\|^2 < \infty. \tag{2.154}$$

By the second Wolfe condition in (L2) and (L1), we have

$$\begin{aligned} (\nabla F_k(\mathbf{y}_k) - \nabla F_k(\mathbf{x}_k))\mathbf{d}_k &\ge -(1-\sigma)\nabla F_k(\mathbf{x}_k)\mathbf{d}_k \\ &= -(1-\sigma)\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k \\ &\ge (1-\sigma)c_1\|\mathbf{g}_k\|^2. \end{aligned} \tag{2.155}$$

If L is the Lipschitz constant for ∇f in $\mathcal{B}$, then we have

$$\begin{aligned}(\nabla F_k(\mathbf{y}_k) - \nabla F_k(\mathbf{x}_k))\mathbf{d}(\mathbf{x}_k) &= (\nabla f(\mathbf{y}_k) - \nabla f(\mathbf{x}_k) + \mu_k(\mathbf{y}_k - \mathbf{x}_k)^{\mathsf{T}})\mathbf{d}_k \\ &\leq (L + \mu_k)\|\mathbf{y}_k - \mathbf{x}_k\|\|\mathbf{d}_k\| = \alpha_k(L + \mu_k)\|\mathbf{d}_k\|^2 \\ &\leq c_2^2\alpha_k(L + \mu_k)\|\mathbf{g}_k\|^2.\end{aligned}$$

In the last inequality, we utilize (L1). Combining this with (2.155) yields:

$$\alpha_k \geq \frac{c_1(1-\sigma)}{c_2^2(L + \mu_k)}.$$

By (2.154) and the bound $\mu_k \leq \beta\|\nabla f(\mathbf{x}_k)\|^\eta$,

$$\begin{aligned}\infty &> \sum_{k=0}^{\infty} \frac{\|\mathbf{g}(\mathbf{x}_k)\|^2}{L + \mu_k} \\ &\geq \frac{1}{2}\sum_{k=0}^{\infty} \min\left(\frac{\|\mathbf{g}(\mathbf{x}_k)\|^2}{L}, \frac{\|\mathbf{g}(\mathbf{x}_k)\|^2}{\mu_k}\right) \\ &\geq \frac{1}{2}\sum_{k=0}^{\infty} \min\left(\frac{\|\mathbf{g}(\mathbf{x}_k)\|^2}{L}, \frac{\|\mathbf{g}(\mathbf{x}_k)\|^{2-\eta}}{\beta}\right).\end{aligned}$$

Since $\eta < 2$, we conclude that $\mathbf{g}(\mathbf{x}_k)$ approaches 0. □

2.3.5 Preliminary Numerical Results

We now present some preliminary numerical examples to illustrate the convergence theory. To illustrate the quadratic convergence when $\mu(\mathbf{x}) = \beta\|\nabla f(\mathbf{x})\|$, we consider the following two problems (the first is introduced in [88]):

$$\begin{aligned}&\text{(P1)} \qquad f(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{n-1}(x_i - x_{i+1})^2 + \frac{1}{12}\sum_{i=1}^{n-1}\alpha_i(x_i - x_{i+1})^4, \\ &\text{(P2)} \qquad f(\mathbf{x}) = \sum_{i=1}^{n} b_i(x_i - 1)^2 + \sum_{i=1}^{n}(x_i - 1)^4.\end{aligned}$$

In (P1) we take $n = 10$, $\alpha_i = 1$, and the starting guess $x_i = i$, $1 \leq i \leq n$. The set of minimizers are given by $x_1 = x_2 = \ldots = x_n$. In (P2) we take $n = 10$, $b_i = e^{-4i}$, and the starting guess $x_i = 1 + 1/i$, $1 \leq i \leq n$. For this problem,

Table 2–7: $\|\mathbf{g}(\mathbf{x}_k)\|$ versus iteration number k

	Problem 1		Problem 2	
Iteration	(C1)	(C2)	(C1)	(C2)
1	4.5e−01	5.4e−01	4.7e−01	1.3e−01
2	7.2e−02	1.0e−01	1.3e−02	3.2e−03
3	2.6e−03	7.1e−03	1.5e−04	2.5e−05
4	3.5e−06	2.6e−05	4.2e−07	4.5e−08
5	6.4e−12	4.3e−10	1.8e−10	1.2e−11

the minimizer is unique, however, the condition number of the Hessian at the solution is around .5e16. Table 2–7 gives the gradient norms corresponding to $\mu(\mathbf{x}) = .05\|\nabla f(\mathbf{x})\|$, and acceptance criterions (C1) and (C2). For (C2), we chose $\theta = .66$. The subproblems (2.105) were solved using our conjugate gradient routine CG_DESCENT [73, 72], stopping when either (C1) or (C2) is satisfied.

In the next series of numerical experiments, we solve some ill-conditioned problems from the CUTE library [13]. Experimentally, we find that it is more efficient to use the proximal strategy in a neighborhood $\mathcal{N}$ of an optimum; outside $\mathcal{N}$, we apply CD_DESCENT to the original problem (2.1). In our numerical experiments, our method for choosing $\mathcal{N}$ was the following: We applied the conjugate gradient method to the original problem (2.1) until the following condition was satisfied:

$$\|\mathbf{g}(\mathbf{x})\|_\infty \leq 10^{-2}(1 + |f(\mathbf{x})|). \tag{2.156}$$

When this condition holds, we continue to apply the conjugate gradient method until an estimate for the condition number exceeds 10^3; then we switch to the proximal point method (2.105), using CG_DESCENT to solve the subproblems. We estimate the condition number by first estimating the second derivative of the function along the normalized search direction. Our estimate for the condition number is the ratio between the maximum and minimum second derivative, during the iterations after (2.156) is satisfied.

Table 2–8 gives convergence results for ill-condition problems from CUTE [13]. The "exact condition numbers" are computed in the following way: We solve the problem and output the Hessian matrix at the solution. The extreme eigenvalues of the Hessian were computed using Matlab, and the eigenvalue ratio appears in the column labeled "Cond" of Table 2–8. For the proximal point iteration, we used acceptance criterion (C2). The iterations were continued until the stopping condition $\|\nabla f(\mathbf{x})\|_\infty \le 10^{-6}$ was satisfied. The number of iteration (It), number of function evaluations (NF), and number of gradient evaluations (NG) are given in the table. Observe that the reduction in the number of function and gradient evaluations varies from almost nothing in EIGENCLS to about 50% for EIGENBLS.

Table 2–8: Statistics for ill-condition CUTE problems and CG_DESCENT

			No Proximal Point			With Proximal Point		
Problem	Dim	Cond	It	NF	NG	It	NF	NG
SPARSINE	2000	2.6e17	12,528	25,057	12,529	10,307	20,615	10,308
SPARSINE	1000	3.4e14	4,657	9,315	4,658	3,760	7,521	3,761
NONDQUAR	1000	6.6e10	4,004	8,015	4,152	3,013	6,032	3,068
NONDQUAR	500	1.0e10	3,027	6,074	3,185	2,526	5,062	2,676
EIGENALS	420	1.2e06	1,792	3,591	1,811	1,464	2,935	1,482
EIGENBLS	420	3.0e05	5,087	10,185	5,099	2,453	4,910	2,458
EIGENCLS	420	8.2e04	1,733	3,484	1,754	1,774	3,566	1,795
NCB20	510	3.7e16	1,631	3,048	2,372	1,251	2,262	1,684

CHAPTER 3
BOX CONSTRAINED OPTIMIZATION

In this chapter, we will consider the problem (1.1) with the feasible set $\mathcal{S}$ to be a box set $\mathcal{B}$, i.e. the problem (1.1) turns out to be

$$\min \{f(\mathbf{x}) : \mathbf{x} \in \mathcal{B}\}, \tag{3.1}$$

where f is a real-valued, continuously differentiable function defined on the set

$$\mathcal{B} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}. \tag{3.2}$$

Here $\mathbf{l} < \mathbf{u}$ and possibly, $l_i = -\infty$ or $u_i = \infty$.

3.1 Introduction

The box constrained optimization problem appears in a wide range of applications including the obstacle problem [102], the elastic-plastic torsion problem [61], optimal design problems [8], journal bearing lubrication [20], inversion problems in elastic wave propagation [7], and molecular conformation analysis [62]. Problem (3.1) is often a subproblem of augmented Lagrangian or penalty schemes for general constrained optimization (see [24, 25, 46, 47, 52, 59, 70, 71, 98]). Thus the development of numerical algorithms to solve (3.1) efficiently, especially when the dimension is large, is important in both theory and applications.

We begin with an overview of the development of active set methods. A seminal paper is Polyak's 1969 paper [107] which considers a convex, quadratic cost function. The conjugate gradient method is used to explore a face of the feasible set, and the negative gradient is used to leave a face. Since Polyak's algorithm only added or dropped one constraint in each iteration, Dembo and Tulowitzki proposed [41] an algorithm CGP which could add and drop many constraints in an iteration. Later, Yang and Tolle [128] further developed this algorithm so as to obtain finite termination, even when the problem was degenerate at a local minimizer $\mathbf{x}^*$. That is, for some i, $x_i^* = l_i$ or $x_i^* = u_i$ and $\nabla f(\mathbf{x}^*)_i = 0$. Another variation of the CGP algorithm, for which there is a rigorous convergence theory, is developed by Wright [124]. Moré and Toraldo [102] point out that when the CGP scheme starts far from the solution, many iterations may be required to identify a suitable working face. Hence, they propose using the gradient projection method to identify a working face, followed by the conjugate gradient method to explore the face. Their algorithm, called GPCG, has finite termination for nondegenerate quadratic problems. Recently, adaptive conjugate gradient algorithms have been developed by Dostál *et al.* [44, 45, 47] which have finite termination for a strictly convex quadratic cost function, even when the problem is degenerate.

For general nonlinear functions, some of the earlier research [4, 19, 63, 87, 99, 113] focused on gradient projection methods. To accelerate the convergence, more recent research has developed Newton and trust region methods (see [26] for in-depth analysis). In [5, 17, 24, 51] superlinear and quadratic convergence is established for nondegenerate problems, while [53, 59, 86, 90] establish analogous convergence results, even for degenerate problems. Although computing a Newton step can be expensive computationally, approximation techniques, such as a sparse, incomplete Cholesky factorization [89], could be used to reduce the computational expense. Nonetheless, for large-dimensional problems or for problems where the initial guess is far from the solution, the Newton/trust region approach can be inefficient. In cases where the Newton step is unacceptable, a gradient projection step is preferred.

The affine-scaling interior point method of Coleman and Li [14, 21, 22, 23] is a different approach to (3.1), related to the trust region algorithm. More recent research on this strategy includes [42, 80, 83, 120, 133]. These methods are based on a reformulation of the necessary optimality conditions obtained by multiplication with a scaling matrix. The resulting system is often solved by Newton-type methods. Without assuming strict complementarity (i. e. for degenerate problems), the affine-scaling interior-point method converges superlinearly or quadratically, for a suitable choice of the scaling matrix, when the strong second-order sufficient optimality condition [110] holds. When the dimension is large, forming and solving the system of equations at each iteration can be time consuming, unless the problem has special structure. Recently, Zhang [133] proposes an interior-point gradient approach for solving the system at each iteration. Convergence results for other interior-point methods applied to more general constrained optimization appear in [48, 49, 125].

The method developed in this chapter is an active set algorithm (ASA) which consists of a nonmonotone gradient projection step, an unconstrained optimization step, and a set of rules for branching between the steps. A good survey of this active set method can be found in [76]. Global convergence to a stationary point is established. When the strong second-order sufficient optimality condition holds, we show that ASA eventually reduces to unconstrained optimization, without restarts. This property is obtained without assuming strict complementary slackness. If strict complementarity holds and all the constraints are active at a stationary point, then convergence occurs in a finite number of iterations. In general, our analysis does not show that the strictly active constraints are identified in a finite number of iterations; instead, when the strong second-order sufficient optimality condition holds, we show that ASA eventually branches to the unconstrained optimization step, and henceforth, the active set does not change. Thus in the limit, ASA reduces to unconstrained optimization without restarts.

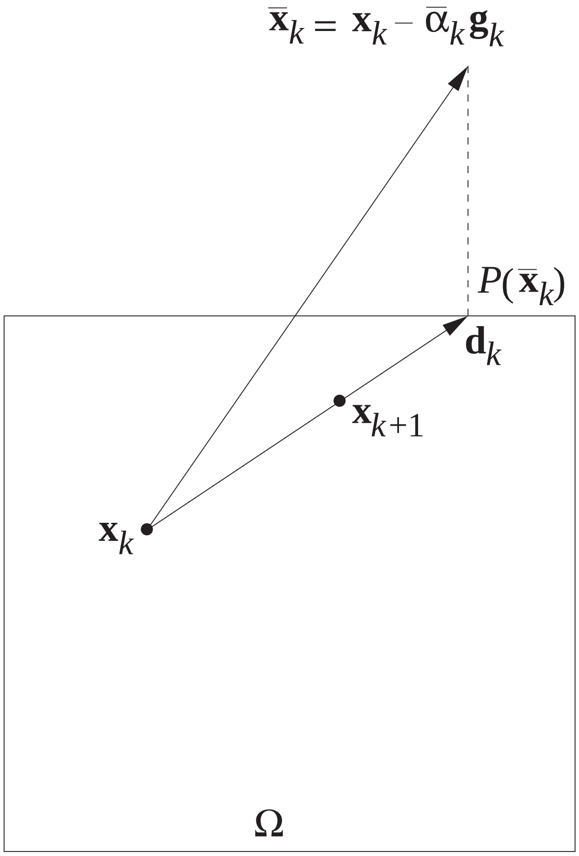

Figure 3–1: The gradient projection step.

3.2 Gradient Projection Methods

In this section, we consider a generalization of (3.1) in which the box $\mathcal{B}$ is replaced by a nonempty, closed convex set Ω:

$$\min \{f(\mathbf{x}) : \mathbf{x} \in \Omega\}. \tag{3.3}$$

We begin with an overview of our gradient projection algorithm. Step k in our algorithm is depicted in Figure 3–1. Here P denotes the projection onto Ω:

$$P(\mathbf{x}) = \arg \min_{\mathbf{y} \in \Omega} \|\mathbf{x} - \mathbf{y}\|. \tag{3.4}$$

Starting at the current iterate $\mathbf{x}_k$, we compute an initial iterate $\bar{\mathbf{x}}_k = \mathbf{x}_k - \overline{\alpha}_k \mathbf{g}_k$. The only constraint on the initial steplength $\overline{\alpha}_k$ is that $\overline{\alpha}_k \in [\alpha_{\min}, \alpha_{\max}]$, where

$\alpha_{\min}$ and $\alpha_{\max}$ are fixed, positive constants, independent of k. Since the nominal iterate may lie outside Ω, we compute its projection $P(\bar{\mathbf{x}}_k)$ onto Ω. The search direction is $\mathbf{d}_k = P(\bar{\mathbf{x}}_k) - \mathbf{x}_k$, similar to the choice made in SPG2 [11]. Using a nonmonotone line search along the line segment connecting $\mathbf{x}_k$ and $P(\bar{\mathbf{x}}_k)$, we arrive at the new iterate $\mathbf{x}_{k+1}$.

In the statement of the nonmonotone gradient projection algorithm (NGPA) given below, f_k^r denotes the "reference" function value. A monotone line search corresponds to the choice $f_k^r = f(\mathbf{x}_k)$. The nonmonotone GLL scheme takes $f_k^r = f_k^{\max}$ where

$$f_k^{\max} = \max\{f(\mathbf{x}_{k-i}) : 0 \le i \le \min(k, M-1)\}. \tag{3.5}$$

Here $M > 0$ is a fixed integer, the memory. For a specific procedure on how to choose the reference function value based on our cyclic BB scheme, one can refer the paper [39, 78, 132].

NGPA Parameters

$\epsilon \in [0, \infty)$, error tolerance

$\delta \in (0, 1)$, descent parameter used in Armijo line search

$\eta \in (0, 1)$, decay factor for stepsize in Armijo line search

$[\alpha_{\min}, \alpha_{\max}] \subset (0, \infty)$, interval containing initial stepsize

Nonmonotone Gradient Projection Algorithm (NGPA)

```
Initialize $k = 0$, $\mathbf{x}_0$ = starting guess, and $f_{-1}^r = f(\mathbf{x}_0)$.
While $\|P(\mathbf{x}_k - \mathbf{g}_k) - \mathbf{x}_k\| > \epsilon$
    1. Choose $\overline{\alpha}_k \in [\alpha_{\min}, \alpha_{\max}]$ and set $\mathbf{d}_k = P(\mathbf{x}_k - \overline{\alpha}_k \mathbf{g}_k) - \mathbf{x}_k$.
    2. Choose $f_k^r$ so that $f(\mathbf{x}_k) \le f_k^r \le \max\{f_{k-1}^r, f_k^{\max}\}$ and $f_k^r \le f_k^{\max}$
       infinitely often.
```

3. Let f_R be either f_k^r or $\min\{f_k^{\max}, f_k^r\}$. If $f(\mathbf{x}_k + \mathbf{d}_k) \leq f_R + \delta \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k$, then $\alpha_k = 1$.
4. If $f(\mathbf{x}_k + \mathbf{d}_k) > f_R + \delta \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k$, then $\alpha_k = \eta^j$ where $j > 0$ is the smallest integer such that

$$f(\mathbf{x}_k + \eta^j \mathbf{d}_k) \leq f_R + \eta^j \delta \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k. \tag{3.6}$$

5. Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$ and $k = k + 1$.

End

The condition $f(\mathbf{x}_k) \leq f_k^r$ guarantees that the Armijo line search in Step 4 can be satisfied. The requirement that "$f_k^r \leq f_k^{\max}$ infinitely often" in Step 2 is needed for the global convergence result Theorem 12. This is a rather weak requirement which can be satisfied by many strategies. For example, every L iteration, we could simply set $f_k^r = f_k^{\max}$. Another stragegy, closer in spirit to the one used in the numerical experiments, is to choose a decrease parameter $\Delta > 0$ and an integer $L > 0$ and set $f_k^r = f_k^{\max}$ if $f(\mathbf{x}_{k-L}) - f(\mathbf{x}_k) \leq \Delta$.

To begin the convergence analysis, recall that $\mathbf{x}^*$ is a stationary point for (3.3) if the first-order optimality condition holds:

$$\nabla f(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) \geq 0 \quad \text{for all } \mathbf{x} \in \Omega. \tag{3.7}$$

Let $\mathbf{d}^\alpha(\mathbf{x})$, $\alpha \in \mathbb{R}$, be defined in terms of the gradient $\mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x})^{\mathsf{T}}$ as follows:

$$\mathbf{d}^\alpha(\mathbf{x}) = P(\mathbf{x} - \alpha \mathbf{g}(\mathbf{x})) - \mathbf{x}.$$

In NGPA, the search direction is $\mathbf{d}_k = \mathbf{d}^{\overline{\alpha}_k}(\mathbf{x}_k)$. For unconstrained optimization, $\mathbf{d}^\alpha(\mathbf{x})$ points along the negative gradient at $\mathbf{x}$ when $\alpha > 0$. Some properties of P and $\mathbf{d}^\alpha$ are summarized below:

Proposition 3 P1. $(P(\mathbf{x}) - \mathbf{x})^{\mathsf{T}}(\mathbf{y} - P(\mathbf{x})) \geq 0$ *for all* $\mathbf{x} \in \mathbb{R}^n$ *and* $\mathbf{y} \in \Omega$.

P2. $(P(\mathbf{x}) - P(\mathbf{y}))^{\mathsf{T}}(\mathbf{x} - \mathbf{y}) \geq \|P(\mathbf{x}) - P(\mathbf{y})\|^2$ *for all* $\mathbf{x}$ *and* $\mathbf{y} \in \mathbb{R}^n$.

P3. $\|P(\mathbf{x}) - P(\mathbf{y})\| \leq \|\mathbf{x} - \mathbf{y}\|$ *for all* $\mathbf{x}$ *and* $\mathbf{y} \in \mathbb{R}^n$.

P4. $\|\mathbf{d}^\alpha(\mathbf{x})\|$ *is nondecreasing in* $\alpha > 0$ *for any* $\mathbf{x} \in \Omega$.

P5. $\|\mathbf{d}^\alpha(\mathbf{x})\|/\alpha$ *is nonincreasing in* $\alpha > 0$ *for any* $\mathbf{x} \in \Omega$.

P6. $\mathbf{g}(\mathbf{x})^\mathsf{T}\mathbf{d}^\alpha(\mathbf{x}) \le -\|\mathbf{d}^\alpha(\mathbf{x})\|^2/\alpha$ *for any* $\mathbf{x} \in \Omega$ *and* $\alpha > 0$.

P7. *For any* $\mathbf{x} \in \Omega$ *and* $\alpha > 0$, $\mathbf{d}^\alpha(\mathbf{x}) = \mathbf{0}$ *if and only if* $\mathbf{x}$ *is a stationary point for (3.3).*

P8. *Suppose* $\mathbf{x}^*$ *is a stationary point for (3.3). If for some* $\mathbf{x} \in \mathbb{R}^n$, *there exist positive scalars* λ *and* γ *such that*

$$(\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{x}^*))^\mathsf{T}(\mathbf{x} - \mathbf{x}^*) \ge \gamma\|\mathbf{x} - \mathbf{x}^*\|^2 \tag{3.8}$$

and

$$\|\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{x}^*)\| \le \lambda\|\mathbf{x} - \mathbf{x}^*\|, \tag{3.9}$$

then we have

$$\|\mathbf{x} - \mathbf{x}^*\| \le \left(\frac{1+\lambda}{\gamma}\right)\|\mathbf{d}^1(\mathbf{x})\|.$$

Proof. P1 is the first-order optimality condition associated with the solution of (3.4). Replacing $\mathbf{y}$ by $P(\mathbf{y})$ in P1 gives

$$(P(\mathbf{x}) - \mathbf{x})^\mathsf{T}(P(\mathbf{y}) - P(\mathbf{x})) \ge 0.$$

Adding this to the corresponding inequality obtained by interchanging $\mathbf{x}$ and $\mathbf{y}$ yields P2 (see [129]). P3 is the nonexpansive property of a projection (for example, see [6, Prop. 2.1.3]). P4 is given in [116]. For P5, see [6, Lem. 2.3.1]. P6 is obtained from P1 by replacing $\mathbf{x}$ with $\mathbf{x} - \alpha\mathbf{g}(\mathbf{x})$ and replacing $\mathbf{y}$ with $\mathbf{x}$. If $\mathbf{x}^*$ is a stationary point satisfying (3.7), then P6 with $\mathbf{x}$ replaced by $\mathbf{x}^*$ yields $\mathbf{d}^\alpha(\mathbf{x}^*) = \mathbf{0}$. Conversely, if $\mathbf{d}^\alpha(\mathbf{x}^*) = \mathbf{0}$, then by P1 with $\mathbf{x}$ replaced by $\mathbf{x}^* - \alpha\mathbf{g}(\mathbf{x}^*)$, we obtain

$$0 \le \alpha\mathbf{g}(\mathbf{x}^*)^\mathsf{T}(\mathbf{y} - P(\mathbf{x}^* - \alpha\mathbf{g}(\mathbf{x}^*)) = \alpha\mathbf{g}(\mathbf{x}^*)^\mathsf{T}(\mathbf{y} - \mathbf{x}^*),$$

which implies that $\mathbf{x}^*$ is a stationary point (see [6, Fig. 2.3.2]).

Finally, let us consider P8. Replacing $\mathbf{x}$ by $\mathbf{x} - \mathbf{g}(\mathbf{x})$ and replacing $\mathbf{y}$ by $\mathbf{x}^*$ in P1 gives

$$[P(\mathbf{x} - \mathbf{g}(\mathbf{x})) - \mathbf{x} + \mathbf{g}(\mathbf{x})]^{\mathsf{T}}[\mathbf{x}^* - P(\mathbf{x} - \mathbf{g}(\mathbf{x}))] \geq 0. \tag{3.10}$$

By the definition of $\mathbf{d}^\alpha(\mathbf{x})$, (3.10) is equivalent to

$$[\mathbf{d}^1(\mathbf{x}) + \mathbf{g}(\mathbf{x})]^{\mathsf{T}}[\mathbf{x}^* - \mathbf{x} - \mathbf{d}^1(\mathbf{x})] \geq 0.$$

Rearranging this and utilizing (3.8) gives

$$\begin{aligned} \mathbf{d}^1(\mathbf{x})^{\mathsf{T}}(\mathbf{x}^* - \mathbf{x}) - \mathbf{g}(\mathbf{x})^{\mathsf{T}}\mathbf{d}^1(\mathbf{x}) - \|\mathbf{d}^1(\mathbf{x})\|^2 &\geq \mathbf{g}(\mathbf{x})^{\mathsf{T}}(\mathbf{x} - \mathbf{x}^*) \qquad (3.11) \\ &\geq \gamma\|\mathbf{x} - \mathbf{x}^*\|^2 + \mathbf{g}(\mathbf{x}^*)^{\mathsf{T}}(\mathbf{x} - \mathbf{x}^*). \end{aligned}$$

Focusing on the terms involving $\mathbf{g}$ and utilizing (3.9), we have

$$\begin{aligned} \mathbf{g}(\mathbf{x}^*)^{\mathsf{T}}(\mathbf{x}^* - \mathbf{x}) - \mathbf{g}(\mathbf{x})^{\mathsf{T}}\mathbf{d}^1(\mathbf{x}) &\leq \lambda\|\mathbf{x} - \mathbf{x}^*\|\ \|\mathbf{d}^1(\mathbf{x})\| + \mathbf{g}(\mathbf{x}^*)^{\mathsf{T}}(\mathbf{x}^* - \mathbf{x} - \mathbf{d}^1(\mathbf{x})) \\ &= \lambda\|\mathbf{x} - \mathbf{x}^*\|\ \|\mathbf{d}^1(\mathbf{x})\| + \mathbf{g}(\mathbf{x}^*)^{\mathsf{T}}[\mathbf{x}^* - P(\mathbf{x} - \mathbf{g}(\mathbf{x}))] \\ &\leq \lambda\|\mathbf{x} - \mathbf{x}^*\|\ \|\mathbf{d}^1(\mathbf{x})\| \qquad (3.12) \end{aligned}$$

by (3.7) since $P(\mathbf{x} - \mathbf{g}(\mathbf{x})) \in \Omega$. Combining (3.11) and (3.12), the proof is complete. □

Next, we establish a convergence result for NGPA:

Theorem 12 *Let $\mathcal{L}$ be the level set defined by*

$$\mathcal{L} = \{\mathbf{x} \in \Omega : f(\mathbf{x}) \leq f(\mathbf{x}_0)\}. \tag{3.13}$$

We assume the following conditions hold:

G1. *f is bounded from below on $\mathcal{L}$ and $d_{\max} = \sup_k \|\mathbf{d}_k\| < \infty$.*

G2. *If $\bar{\mathcal{L}}$ is the collection of $\mathbf{x} \in \Omega$ whose distance to $\mathcal{L}$ is at most $d_{\max}$, then ∇f is Lipschitz continuous on $\bar{\mathcal{L}}$.*

Then NGPA with $\epsilon = 0$ *either terminates in a finite number of iterations at a stationary point, or we have*

$$\liminf_{k\to\infty} \|\mathbf{d}^1(\mathbf{x}_k)\| = 0.$$

Proof. By P6, the search direction $\mathbf{d}_k$ generated in Step 1 of NGPA is a descent direction. Since $f_k^r \geq f(\mathbf{x}_k)$ and $\delta < 1$, the Armijo line search condition (3.6) is satisfied for j sufficiently large. We now show that $\mathbf{x}_k \in \mathcal{L}$ for each k. Since $f_0^{\max} = f_{-1}^r = f(\mathbf{x}_0)$, Step 2 of NGPA implies that $f_0^r \leq f(\mathbf{x}_0)$. Proceeding by induction, suppose that for some $k \geq 0$, we have

$$f_j^r \leq f(\mathbf{x}_0) \quad \text{and} \quad f_j^{\max} \leq f(\mathbf{x}_0) \tag{3.14}$$

for all $j \in [0, k]$. Again, since the search direction $\mathbf{d}_k$ generated in Step 1 of NGPA is a descent direction, it follows from Steps 3 and 4 of NGPA and the induction hypothesis that

$$f(\mathbf{x}_{k+1}) \leq f_k^r \leq f(\mathbf{x}_0). \tag{3.15}$$

Hence, $f_{k+1}^{\max} \leq f(\mathbf{x}_0)$ and $f_{k+1}^r \leq \max\{f_k^r, f_{k+1}^{\max}\} \leq f(\mathbf{x}_0)$. This completes the induction. Thus (3.14) holds for all j. Consequently, we have $f_R \leq f(\mathbf{x}_0)$ in Steps 3 and 4 of NGPA. Again, since the search direction $\mathbf{d}_k$ generated in Step 1 of NGPA is a descent direction, it follows from Steps 3 and 4 that $f(\mathbf{x}_k) \leq f(\mathbf{x}_0)$, which implies that $\mathbf{x}_k \in \mathcal{L}$ for each k.

Let λ be the Lipschitz constant for ∇f on $\bar{\mathcal{L}}$. As in [131, Lem. 2.1], we have

$$\alpha_k \geq \min\left\{1,\ \left(\frac{2\eta(1-\delta)}{\lambda}\right)\frac{|\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k|}{\|\mathbf{d}_k\|^2}\right\} \tag{3.16}$$

for all k. By P6,

$$|\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k| \geq \frac{\|\mathbf{d}_k\|^2}{\overline{\alpha}_k} \geq \frac{\|\mathbf{d}_k\|^2}{\alpha_{\max}}.$$

It follows from (3.16) that

$$\alpha_k \geq \min\left\{1,\ \left(\frac{2\eta(1-\delta)}{\lambda\alpha_{\max}}\right)\right\} := c. \tag{3.17}$$

By Steps 3 and 4 of NGPA and P6, we conclude that

$$f(\mathbf{x}_{k+1}) \leq f_k^r + \delta c \mathbf{g}_k^\mathsf{T} \mathbf{d}_k \leq f_k^r - \delta c\|\mathbf{d}_k\|^2/\overline{\alpha}_k \leq f_k^r - \delta c\|\mathbf{d}_k\|^2/\alpha_{\max}. \tag{3.18}$$

We now prove that $\liminf\limits_{k\to\infty} \|\mathbf{d}_k\| = 0$. Suppose, to the contrary, that there exists a constant $\gamma > 0$ such that $\|\mathbf{d}_k\| \geq \gamma$ for all k. By (3.18), we have

$$f(\mathbf{x}_{k+1}) \leq f_k^r - \tau, \quad \text{where } \tau = \delta c\gamma^2/\alpha_{\max}. \tag{3.19}$$

Let k_i, $i = 0, 1, \ldots$, denote an increasing sequence of integers with the property that $f_j^r \leq f_j^{\max}$ for $j = k_i$ and $f_j^r \leq f_{j-1}^r$ when $k_i < j < k_{i+1}$. Such a sequence exists by the requirement on f_k^r given in Step 2 of NGPA. Hence, we have

$$f_j^r \leq f_{k_i}^r \leq f_{k_i}^{\max}, \quad \text{when } k_i \leq j < k_{i+1}. \tag{3.20}$$

By (3.19) it follows that

$$f(\mathbf{x}_j) \leq f_{j-1}^r - \tau \leq f_{k_i}^{\max} - \tau \quad \text{when } k_i < j \leq k_{i+1}. \tag{3.21}$$

It follows that

$$f_{k_{i+1}}^r \leq f_{k_{i+1}}^{\max} \leq f_{k_i}^{\max}. \tag{3.22}$$

Hence, if $a = k_{i_1}$ and $b = k_{i_2}$ where $i_1 > i_2$ and $a - b > M$, then by (3.20)–(3.22), we have

$$f_a^{\max} = \max_{0\leq j<M} f(\mathbf{x}_{a-j}) \leq \max_{1\leq j\leq M} f_{a-j}^r - \tau \leq f_b^{\max} - \tau.$$

Since the sequence k_i, $i = 0, 1, \ldots$, is infinite, this contradicts the fact that f is bounded from below. Consequently, $\liminf\limits_{k\to\infty} \|\mathbf{d}_k\| = 0$. By P4 and P5, it follows that

$$\|\mathbf{d}_k\| \geq \min\{\alpha_{\min}, 1\}\|\mathbf{d}^1(\mathbf{x}_k)\|.$$

Thus $\liminf_{k\to\infty} \|\mathbf{d}^1(\mathbf{x}_k)\| = 0.$ □

Recall that f is strongly convex on Ω if there exists a scalar $\gamma > 0$ such that

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \nabla f(\mathbf{y})(\mathbf{x}-\mathbf{y}) + \frac{\gamma}{2}\|\mathbf{x}-\mathbf{y}\|^2 \tag{3.23}$$

for all $\mathbf{x}$ and $\mathbf{y} \in \Omega$. Interchanging $\mathbf{x}$ and $\mathbf{y}$ in (3.23) and adding, we obtain the (usual) monotonicity condition

$$(\nabla f(\mathbf{y}) - \nabla f(\mathbf{x}))(\mathbf{y}-\mathbf{x}) \geq \gamma\|\mathbf{y}-\mathbf{x}\|^2. \tag{3.24}$$

For a strongly convex function, (3.3) has a unique minimizer $\mathbf{x}^*$ and the conclusion of Theorem 12 can be strengthened as follows:

Corollary 2 *Suppose f is strongly convex and twice continuously differentiable on Ω, and there is a positive integer L with the property that for each k, there exists $j \in [k, k+L)$ such that $f_j^r \leq f_j^{\max}$. Then the iterates $\mathbf{x}_k$ of NGPA with $\epsilon = 0$ converge to the global minimizer $\mathbf{x}^*$.*

Proof. As shown at the start of the proof of Theorem 12, $f(\mathbf{x}_k) \leq f(\mathbf{x}_0)$ for each k. Hence, $\mathbf{x}_k$ lies in the level set $\mathcal{L}$ defined in (3.13). Since f is strongly convex, $\mathcal{L}$ is a bounded set; since f is twice continuously differentiable, $\|\nabla f(\mathbf{x}_k)\|$ is bounded uniformly in k. For any $\mathbf{x} \in \Omega$, we have $P(\mathbf{x}) = \mathbf{x}$. By P3, it follows that

$$\|\mathbf{d}^\alpha\| = \|P(\mathbf{x}-\alpha\mathbf{g}(\mathbf{x})) - \mathbf{x}\| = \|P(\mathbf{x}-\alpha\mathbf{g}(\mathbf{x})) - P(\mathbf{x})\| \leq \alpha\|\mathbf{g}(\mathbf{x})\|.$$

Since $\bar{\alpha}_k \in [\alpha_{\min}, \alpha_{\max}]$, $d_{\max} = \sup_k \|\mathbf{d}_k\| < \infty$. Consequently, the set $\bar{\mathcal{L}}$ defined in G2 is bounded. Again, since f is twice continuously differentiable, ∇f is Lipschitz continuous on $\bar{\mathcal{L}}$. By assumption, $f_k^r \leq f_k^{\max}$ infinitely often. Consequently, the hypotheses of Theorem 12 are satisfied, and NGPA with $\epsilon = 0$ either terminates in a finite number of iterations at a stationary point, or we have

$$\liminf_{k\to\infty} \|\mathbf{d}^1(\mathbf{x}_k)\| = 0. \tag{3.25}$$

Since f is strongly convex on Ω, $\mathbf{x}^*$ is the unique stationary point for (3.3). Hence, when the iterates converge in a finite number of steps, they converge to $\mathbf{x}^*$. Otherwise, (3.25) holds, in which case, there exists an infinite sequence $l_1 < l_2 < \ldots$ such that $\|\mathbf{d}^1(\mathbf{x}_{l_j})\|$ approaches zero as j tends to ∞. Since (3.24) holds, it follows from P8 that $\mathbf{x}_{l_j}$ approaches $\mathbf{x}^*$ as j tends to ∞. By P4 and P5, we have

$$\|\mathbf{d}^\alpha(\mathbf{x})\| \le \max\{1, \alpha\}\|\mathbf{d}^1(\mathbf{x})\|.$$

Since $\bar{\alpha}_k \in [\alpha_{\min}, \alpha_{\max}]$, it follows that

$$\|\mathbf{d}_k\| \le \max\{1, \alpha_{\max}\}\|\mathbf{d}^1(\mathbf{x}_k)\|.$$

Since the stepsize $\alpha_k \in (0, 1]$, we deduce that

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\| = \alpha_k\|\mathbf{d}_k\| \le \|\mathbf{d}_k\| \le \max\{1, \alpha_{\max}\}\|\mathbf{d}^1(\mathbf{x}_k)\|. \tag{3.26}$$

By P3, P is continuous; consequently, $\mathbf{d}^\alpha(\mathbf{x})$ is a continuous function of $\mathbf{x}$. The continuity of $\mathbf{d}^\alpha(\cdot)$ and $f(\cdot)$ combined with (3.26) and the fact that $\mathbf{x}_{l_j}$ converges to $\mathbf{x}^*$ implies that for any $\delta > 0$ and for j sufficiently large, we have

$$f(\mathbf{x}_k) \le f(\mathbf{x}^*) + \delta \quad \text{for all } k \in [l_j, l_j + M + L].$$

By the definition of $f_k^{\max}$,

$$f_k^{\max} \le f(\mathbf{x}^*) + \delta \quad \text{for all } k \in [l_j + M, l_j + M + L]. \tag{3.27}$$

As in the proof of Theorem 12, let k_i, $i = 0, 1, \ldots$, denote an increasing sequence of integers with the property that $f_j^r \le f_j^{\max}$ for $j = k_i$ and $f_j^r \le f_{j-1}^r$ when $k_i < j < k_{i+1}$. As shown in (3.22),

$$f_{k_{i+1}}^{\max} \le f_{k_i}^{\max} \tag{3.28}$$

for each i. The assumption that for each k, there exists $j \in [k, k+L)$ such that $f_j^r \le f_j^{\max}$ implies that

$$k_{i+1} - k_i \le L. \tag{3.29}$$

Combining (3.27) and (3.29), for each l_j, there exists some $k_i \in [l_j + M, l_j + M + L]$ and

$$f_{k_i}^{\max} \le f(\mathbf{x}^*) + \delta. \tag{3.30}$$

Since δ was arbitrary, it follows from (3.28) and (3.30) that

$$\lim_{i\to\infty} f_{k_i}^{\max} = f(\mathbf{x}^*); \tag{3.31}$$

the convergence is monotone by (3.28). By the choice of k_i and by the inequality $f(\mathbf{x}_k) \le f_k^r$ in Step 2, we have

$$f(\mathbf{x}_k) \le f_k^r \le f_{k_i}^{\max} \quad \text{for all } k \ge k_i. \tag{3.32}$$

Combining (3.31) and (3.32),

$$\lim_{k\to\infty} f(\mathbf{x}_k) = f(\mathbf{x}^*). \tag{3.33}$$

Together, (3.7) and (3.23) yield

$$f(\mathbf{x}_k) \ge f(\mathbf{x}^*) + \frac{\gamma}{2}\|\mathbf{x}_k - \mathbf{x}^*\|^2. \tag{3.34}$$

Combining this with (3.33), the proof is complete. □

3.3 Active Set Algorithm (ASA)

Starting with this section, we focus on the box constrained problem (3.1). To simplify the exposition, we consider the special case $\mathbf{l} = \mathbf{0}$ and $\mathbf{u} = \boldsymbol{\infty}$:

$$\min \{f(\mathbf{x}) : \mathbf{x} \ge \mathbf{0}\}.$$

We emphasize that the analysis and algorithm apply to the general box constrained problem (3.1) with both upper and lower bounds.

Although the gradient projection scheme NGPA has an attractive global convergence theory, the convergence rate can be slow in a neighborhood of a local minimizer. In contrast, for unconstrained optimization, the conjugate gradient algorithm often exhibits superlinear convergence in a neighborhood of a local minimizer. We develop an active set algorithm which uses NGPA to identify active constraints, and which uses an unconstrained optimization algorithm, such as the CG_DESCENT scheme in [73, 74, 72, 75], to optimize f over a face identified by NGPA.

We begin with some notation. For any $\mathbf{x} \in \Omega$, let $\mathcal{A}(\mathbf{x})$ and $\mathcal{I}(\mathbf{x})$ denote the active and inactive indices respectively:

$$\begin{aligned} \mathcal{A}(\mathbf{x}) &= \{i \in [1,n] : x_i = 0\}, \\ \mathcal{I}(\mathbf{x}) &= \{i \in [1,n] : x_i > 0\}. \end{aligned}$$

The active indices are further subdivided into those indices satisfying strict complementarity and the degenerate indices:

$$\begin{aligned} \mathcal{A}_+(\mathbf{x}) &= \{i \in \mathcal{A}(\mathbf{x}) : g_i(\mathbf{x}) > 0\}, \\ \mathcal{A}_0(\mathbf{x}) &= \{i \in \mathcal{A}(\mathbf{x}) : g_i(\mathbf{x}) = 0\}. \end{aligned}$$

We let $\mathbf{g}_I(\mathbf{x})$ denote the vector whose components associated with the set $\mathcal{I}(\mathbf{x})$ are identical to those of $\mathbf{g}(\mathbf{x})$, while the components associated with $\mathcal{A}(\mathbf{x})$ are zero:

$$g_{Ii}(\mathbf{x}) = \begin{cases} 0 & \text{if } x_i = 0, \\ g_i(\mathbf{x}) & \text{if } x_i > 0. \end{cases}$$

An important feature of our algorithm is that we try to distinguish between active constraints satisfying strict complementarity, and active constraints that are degenerate using an identification strategy, which is related to the idea of an identification function introduced in [50]. Given fixed parameters $\alpha \in (0, 1)$ and

$\beta \in (1, 2)$, we define the (undecided index) set $\mathcal{U}$ at $\mathbf{x} \in \mathcal{B}$ as follows:

$$\mathcal{U}(\mathbf{x}) = \{i \in [1, n] : |g_i(\mathbf{x})| \geq \|\mathbf{d}^1(\mathbf{x})\|^\alpha \text{ and } x_i \geq \|\mathbf{d}^1(\mathbf{x})\|^\beta\}.$$

In the numerical experiments, we take $\alpha = 1/2$ and $\beta = 3/2$. In practice, $\mathcal{U}$ is almost always empty when we reach a neighborhood of a minimizer, and the specific choice of α and β does not have a significant effect on convergence. The introduction of the $\mathcal{U}$ set leads to a strong local convergence theory developed in the local convergence section.

The indices in $\mathcal{U}$ correspond to components of $\mathbf{x}$ for which the associated gradient component $g_i(\mathbf{x})$ is relatively large, while x_i is not close to 0 (in the sense that $x_i \geq \|\mathbf{d}^1(\mathbf{x})\|^\beta$). When the set $\mathcal{U}$ of uncertain indices is empty, we feel that the indices with large associated gradient components are almost identified. In this case we prefer the unconstrained optimization algorithm.

Although our numerical experiments are based on the conjugate gradient code CG_DESCENT, a broad class of unconstrained optimization algorithms (UA) can be applied. The following requirement for the unconstrained algorithm are sufficient for establishing the convergence results that follow. Conditions U1–U3 are sufficient for global convergence, while U1–U4 are sufficient for the local convergence analysis. U4 could be replaced by another descent condition for the initial line search, however, our local convergence analysis has been carried out under U4.

Unconstrained Algorithm (UA) Requirements

U1. $\mathbf{x}_k \geq \mathbf{0}$ and $f(\mathbf{x}_{k+1}) \leq f(\mathbf{x}_k)$ for each k.

U2. $\mathcal{A}(\mathbf{x}_k) \subset \mathcal{A}(\mathbf{x}_{k+1})$ for each k.

U3. If $\mathcal{A}(\mathbf{x}_{j+1}) = \mathcal{A}(\mathbf{x}_j)$ for $j \geq k$, then $\liminf\limits_{j \to \infty} \|\mathbf{g}_I(\mathbf{x}_j)\| = 0$.

U4. Whenever the unconstrained algorithm is started, $\mathbf{x}_{k+1} = P(\mathbf{x}_k - \alpha_k \mathbf{g}_I(\mathbf{x}_k))$ where α_k is obtained from a Wolfe line search. That is, α_k is chosen to satisfy

$$\phi(\alpha_k) \le \phi(0) + \delta \alpha_k \phi'(0) \quad \text{and} \quad \phi'(\alpha_k) \ge \sigma \phi'(0), \tag{3.35}$$

where

$$\phi(\alpha) = f(P(\mathbf{x}_k - \alpha \mathbf{g}_I(\mathbf{x}_k))), \quad 0 < \delta < \sigma < 1. \tag{3.36}$$

Condition U1 implies that the UA is a monotone algorithm, so that the cost function can only decrease in each iteration. Condition U2 concerns how the algorithm behaves when an infeasible iterate is generated. Condition U3 describes the global convergence of the UA when the active set does not change. In U4, $\phi'(\alpha)$ is the derivative from the right side of α; α_k exists since ϕ is piecewise smooth with a finite number of discontinuities in its derivative, and $\phi'(\alpha)$ is continuous at $\alpha = 0$.

We now present our Active Set Algorithm (ASA). In the first step of the algorithm, we execute NGPA until we feel that the active constraints satisfying strict complementarity have been identified. In Step 2, we execute the UA until a subproblem has been solved (Step 2a). When new constraints become active in Step 2b, we may decide to restart either NGPA or UA. By restarting NGPA, we mean that $\mathbf{x}_0$ in NGPA is identified with the current iterate $\mathbf{x}_k$. By restarting the UA, we mean that iterates are generated by the UA using the current iterate as the starting point.

ASA Parameters

$\epsilon \in [0, \infty)$, error tolerance, stop when $\|\mathbf{d}^1(\mathbf{x}_k)\| \le \epsilon$

$\mu \in (0, 1)$, $\|\mathbf{g}_I(\mathbf{x}_k)\| < \mu \|\mathbf{d}^1(\mathbf{x}_k)\|$ implies subproblem solved

$\rho \in (0, 1)$, decay factor for μ tolerance

$n_1 \in [1, n)$, number of repeated $\mathcal{A}(\mathbf{x}_k)$ before switch from NGPA to UA

$n_2 \in [1, n)$, used in switch from UA to NGPA

Active Set Algorithm (ASA)

```
1. While $\|\mathbf{d}^1(\mathbf{x}_k)\| > \epsilon$ execute NGPA and check the following:
    a. If $\mathcal{U}(\mathbf{x}_k) = \emptyset$, then
            If $\|\mathbf{g}_I(\mathbf{x}_k)\| < \mu\|\mathbf{d}^1(\mathbf{x}_k)\|$, then $\mu = \rho\mu$.
            Otherwise, goto Step 2.
    b. Else if $\mathcal{A}(\mathbf{x}_k) = \mathcal{A}(\mathbf{x}_{k-1}) = \ldots = \mathcal{A}(\mathbf{x}_{k-n_1})$, then
            If $\|\mathbf{g}_I(\mathbf{x}_k)\| \geq \mu\|\mathbf{d}^1(\mathbf{x}_k)\|$, then goto Step 2.
   End

2. While $\|\mathbf{d}^1(\mathbf{x}_k)\| > \epsilon$ execute UA and check the following:
    a. If $\|\mathbf{g}_I(\mathbf{x}_k)\| < \mu\|\mathbf{d}^1(\mathbf{x}_k)\|$, then restart NGPA (Step 1).
    b. If $|\mathcal{A}(\mathbf{x}_{k-1})| < |\mathcal{A}(\mathbf{x}_k)|$, then
            If $\mathcal{U}(\mathbf{x}_k) = \emptyset$ or $|\mathcal{A}(\mathbf{x}_k)| > |\mathcal{A}(\mathbf{x}_{k-1})| + n_2$, restart the UA at $\mathbf{x}_k$.
            Else restart NGPA.
       End
   End
```

3.3.1 Global Convergence

We begin with a global convergence result for ASA.

Theorem 13 *Let $\mathcal{L}$ be the level set defined by*

$$\mathcal{L} = \{\mathbf{x} \in \mathcal{B} : f(\mathbf{x}) \leq f(\mathbf{x}_0)\}.$$

Assume the following conditions hold:

A1. *f is bounded from below on $\mathcal{L}$ and $d_{\max} = \sup_k \|\mathbf{d}_k\| < \infty$.*

A2. *If $\bar{\mathcal{L}}$ is the collection of $\mathbf{x} \in \mathcal{B}$ whose distance to $\mathcal{L}$ is at most $d_{\max}$, then ∇f is Lipschitz continuous on $\bar{\mathcal{L}}$.*

A3. *The UA satisfies* U1–U3.

Then ASA with $\epsilon = 0$ either terminates in a finite number of iterations at a stationary point, or we have

$$\liminf_{k\to\infty} \|\mathbf{d}^1(\mathbf{x}_k)\| = 0. \tag{3.37}$$

Proof. If only the NGPA is performed for large k, then (3.37) follows from Theorem 12. If only the UA is performed for large k, then by U2, the active sets $\mathcal{A}(\mathbf{x}_k)$ must approach a limit. Since μ does not change in the UA, it follows from U3 and the condition $\|\mathbf{g}_I(\mathbf{x}_k)\| \geq \mu\|\mathbf{d}^1(\mathbf{x}_k)\|$ that (3.37) holds. Finally, suppose that NGPA is restarted an infinite number of times at $k_1 < k_2 < \ldots$ and it terminates at $k_1 + l_1 < k_2 + l_2 < \ldots$ respectively. Thus $k_i < k_i + l_i \leq k_{i+1}$ for each i. If (3.37) does not hold, then by (3.21) and (3.22), we have

$$f(\mathbf{x}_{k_i+l_i}) \leq f(\mathbf{x}_{k_i}) - \tau. \tag{3.38}$$

By U1,

$$f(\mathbf{x}_{k_{i+1}}) \leq f(\mathbf{x}_{k_i+l_i}). \tag{3.39}$$

Combining (3.38) and (3.39), we have $f(\mathbf{x}_{k_{i+1}}) \leq f(\mathbf{x}_{k_i}) - \tau$, which contradicts the assumption that f is bounded from below. □

When f is strongly convex, the entire sequence of iterates converges to the global minimizer $\mathbf{x}^*$, as stated in the following corollary. Since the proof of this result relies on the local convergence analysis, we will give the proof at the end of local convergence section.

Corollary 3 *If f is strongly convex and twice continuously differentiable on $\mathcal{B}$, and assumption* A3 *of Theorem* 13 *is satisfied, then the iterates $\mathbf{x}_k$ of ASA with $\epsilon = 0$ converge to the global minimizer $\mathbf{x}^*$.*

3.3.2 Local Convergence

In this section, we analyze local convergence properties of ASA. We begin by focusing on nondegenerate stationary points; that is, stationary points $\mathbf{x}^*$ with the property that $g_i(\mathbf{x}^*) > 0$ whenever $x_i^* = 0$.

Nondegenerate problems. In this case, it is relatively easy to show that ASA eventually performs only the UA without restarts. The analogous result for degenerate problems is established in the next section.

Theorem 14 *If f is continuously differentiable, $0 < \mu \le 1$, and the iterates $\mathbf{x}_k$ generated by ASA with $\epsilon = 0$ converge to a nondegenerate stationary point $\mathbf{x}^*$, then after a finite number of iterations, ASA performs only UA without restarts.*

Proof. Since $\mathbf{x}^*$ is a nondegenerate stationary point and f is continuously differentiable, there exists $\rho > 0$ with the property that for all $\mathbf{x} \in \mathcal{B}_\rho(\mathbf{x}^*)$, we have

$$g_i(\mathbf{x}) > 0 \text{ if } i \in \mathcal{A}(\mathbf{x}^*) \quad \text{and} \quad x_i > 0 \text{ if } i \in \mathcal{A}(\mathbf{x}^*)^c. \tag{3.40}$$

Let k_+ be chosen large enough that $\mathbf{x}_k \in \mathcal{B}_\rho(\mathbf{x}^*)$ for all $k \ge k_+$. If $k \ge k_+$ and $x_{ki} = 0$, then $d_{ki} = 0$ in Step 1 of NGPA. Hence, $x_{k+1,i} = 0$ if $\mathbf{x}_{k+1}$ is generated by NGPA. By U2, the UA cannot free a bound constraint. It follows that if $k \ge k_+$ and $x_{ki} = 0$, then $x_{ji} = 0$ for all $j \ge k$. Consequently, there exists an index $K \ge k_+$ with the property that $\mathcal{A}(\mathbf{x}_k) = \mathcal{A}(\mathbf{x}_j)$ for all $j \ge k \ge K$.

For any index i, $|d_i^1(\mathbf{x})| \le |g_i(\mathbf{x})|$. Suppose $\mathbf{x} \in \mathcal{B}_\rho(\mathbf{x}^*)$; by (3.40), $d_i^1(\mathbf{x}) = 0$ if $x_i = 0$. Hence,

$$\|\mathbf{d}^1(\mathbf{x})\| \le \|\mathbf{g}_I(\mathbf{x})\| \tag{3.41}$$

for all $\mathbf{x} \in \mathcal{B}_\rho(\mathbf{x}^*)$. If $k > K + n_1$, then in Step 1b of ASA, it follows from (3.41) and the assumption $\mu \in (0, 1]$ that NGPA will branch to Step 2 (UA). In Step 2, the condition "$\|\mathbf{g}_I(\mathbf{x}_k)\| < \mu\|\mathbf{d}^1(\mathbf{x}_k)\|$" of Step 2a is never satisfied by (3.41). Moreover, the condition "$|\mathcal{A}(\mathbf{x}_{k-1})| < |\mathcal{A}(\mathbf{x}_k)|$" of Step 2b is never satisfied since $k > K$. Hence, the iterates never branch from UA to NPGA and UA is never restarted. □

Degenerate problems. We now focus on degenerate problems, and show that a result analogous to Theorem 14 holds under the strong second-order sufficient optimality condition. We begin with a series of preliminary results.

Lemma 6 *If f is twice-continuously differentiable and there exists an infinite sequence of iterates $\mathbf{x}_k$ generated by ASA with $\epsilon = 0$ converging to a stationary point $\mathbf{x}^*$, $\mathbf{x}_k \neq \mathbf{x}^*$ for each k, then for each $i \in \mathcal{A}_+(\mathbf{x}^*)$, we have*

$$\limsup_{k \to \infty} \frac{x_{ki}}{\|\mathbf{x}_k - \mathbf{x}^*\|^2} < \infty. \tag{3.42}$$

Proof. Assume that $\mathcal{A}_+(\mathbf{x}^*)$ is nonempty, otherwise there is nothing to prove. Let k_+ be chosen large enough that $g_i(\mathbf{x}_k) > 0$ for all $i \in \mathcal{A}_+(\mathbf{x}^*)$ and $k \geq k_+$. Since f is twice-continuously differentiable, ∇f is Lipschitz continuous in a neighborhood of $\mathbf{x}^*$. Choose $\rho > 0$ and let λ be the Lipschitz constant for ∇f in the ball $B_\rho(\mathbf{x}^*)$ with center $\mathbf{x}^*$ and radius ρ. Since $\mathbf{d}^1(\mathbf{x}^*) = \mathbf{0}$, it follows from the continuity of $\mathbf{d}^1(\cdot)$ that $\mathbf{d}_k$ tends to $\mathbf{0}$ (see (3.26)). Choose k_+ large enough that the ball with center $\mathbf{x}_k$ and radius $\|\mathbf{d}_k\|$ is contained in $B_\rho(\mathbf{x}^*)$ for all $k \geq k_+$. If $x_{li} = 0$ for some $i \in \mathcal{A}_+(\mathbf{x}^*)$ and $l \geq k_+$, then by the definition of $\mathbf{d}_k$ in NGPA, we have $d_{ki} = 0$ for all $k \geq l$. Hence, $x_{ki} = 0$ for each $k \geq l$ in NGPA. Likewise, in the UA it follows from U2 that $x_{ji} = 0$ for $j \geq k$ when $x_{ki} = 0$; that is, the UA does not free an active constraint. In other words, when an index $i \in \mathcal{A}_+(\mathbf{x}^*)$ becomes active at iterate $\mathbf{x}_k$, $k \geq k_+$, it remains active for all the subsequent iterations. Thus (3.42) holds trivially for any $i \in \mathcal{A}_+(\mathbf{x}^*)$ with the property that $x_{ki} = 0$ for some $k \geq k_+$.

Now, let us focus on the nontrivial indices in $\mathcal{A}_+(\mathbf{x}^*)$. That is, suppose that there exists $l \in \mathcal{A}_+(\mathbf{x}^*)$ and $x_{kl} > 0$ for all $k \geq k_+$. By the analysis given in the previous paragraph, when k_+ is sufficiently large,

$$\text{either } x_{ki} > 0 \quad \text{or} \quad x_{ki} = 0 \tag{3.43}$$

for all $k \geq k_+$ and $i \in \mathcal{A}_+(\mathbf{x}^*)$ (since an index $i \in \mathcal{A}_+(\mathbf{x}^*)$ which becomes active at iterate $\mathbf{x}_k$ remains active for all the subsequent iterations). We consider the following possible cases:

Case 1. *For an infinite number of iterations k, $\mathbf{x}_k$ is generated by the UA, and the UA is restarted a finite number of times.* In this case, ASA eventually performs only the UA, without restarts. By U2 and U3, we have $\liminf_{k\to\infty} \|\mathbf{g}_I(\mathbf{x}_k)\| = 0$. On the other hand, by assumption, $l \in \mathcal{I}(\mathbf{x}_k)$ for $k \geq k_+$ and $g_l(\mathbf{x}^*) > 0$, which is a contradiction since $g_l(\mathbf{x}_k)$ converges to $g_l(\mathbf{x}^*)$.

Case 2. *For an infinite number of iterations k, $\mathbf{x}_k$ is generated by the UA, and the UA is restarted an infinite number of times.* In this case, we will show that after a finite number of iterations, $x_{ki} = 0$ for all $i \in \mathcal{A}_+(\mathbf{x}^*)$. Suppose, to the contrary, that there exists an $l \in \mathcal{A}_+(\mathbf{x}^*)$ such that $x_{kl} > 0$ for all $k \geq k_+$. By U3, each time the UA is restarted, we perform a Wolfe line search. By the second half of (3.35), we have

$$\phi'(\alpha_k) - \phi'(0) \geq (\sigma - 1)\phi'(0). \tag{3.44}$$

It follows from the definition (3.36) of $\phi(\alpha)$ that

$$\begin{aligned} \phi'(0) &= -\sum_{i\in\mathcal{I}(x_k)} g_{ki}^2 = -\|\mathbf{g}_I(\mathbf{x}_k)\|^2 \quad \text{and} && (3.45) \\ \phi'(\alpha_k) &= -\sum_{i\in\mathcal{I}(x_{k+1})} g_{ki}g_{k+1,i} \\ &= -\sum_{i\in\mathcal{I}(x_k)} g_{ki}g_{k+1,i} + \sum_{i\in\mathcal{A}(x_{k+1})\setminus\mathcal{A}(x_k)} g_{ki}g_{k+1,i}. && (3.46) \end{aligned}$$

By the Lipschitz continuity of ∇f and P3, we have

$$\begin{aligned} \|\mathbf{g}(\mathbf{x}_k) - \mathbf{g}(\mathbf{x}_{k+1})\| &= \|\mathbf{g}(P(\mathbf{x}_k)) - \mathbf{g}(P(\mathbf{x}_k - \alpha_k\mathbf{g}_I(\mathbf{x}_k)))\| \\ &\leq \lambda\alpha_k\|\mathbf{g}_I(\mathbf{x}_k)\|. \end{aligned}$$

Hence, by the Schwarz inequality,

$$\left| \sum_{i \in \mathcal{I}(\mathbf{x}_k)} g_{ki}(g_{ki} - g_{k+1,i}) \right| \le \lambda \alpha_k \|\mathbf{g}_I(\mathbf{x}_k)\|^2. \tag{3.47}$$

Since $\mathcal{A}(\mathbf{x}_{k+1}) \setminus \mathcal{A}(\mathbf{x}_k) \subset \mathcal{I}(\mathbf{x}_k)$, the Schwarz inequality also gives

$$\sum_{i \in \mathcal{A}(x_{k+1}) \setminus \mathcal{A}(x_k)} g_{ki} g_{k+1,i} \le \|\mathbf{g}_I(\mathbf{x}_k)\| \|\mathbf{g}_{k+1}\|_{\mathcal{N}}, \tag{3.48}$$

where

$$\|\mathbf{g}_{k+1}\|_{\mathcal{N}}^2 = \sum_{i \in \mathcal{A}(x_{k+1}) \setminus \mathcal{A}(x_k)} g_{k+1,i}^2.$$

Here $\mathcal{N} = \mathcal{A}(\mathbf{x}_{k+1}) \setminus \mathcal{A}(\mathbf{x}_k)$ corresponds to the set of constraints that are newly activated as we move from $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$. Combining (3.44)–(3.48),

$$\alpha_k \ge \frac{1-\sigma}{\lambda} - \frac{\|\mathbf{g}_{k+1}\|_{\mathcal{N}}}{\lambda \|\mathbf{g}_I(\mathbf{x}_k)\|}, \quad \text{where } \|\mathbf{g}_{k+1}\|_{\mathcal{N}}^2 = \sum_{i \in \mathcal{A}(x_{k+1}) \setminus \mathcal{A}(x_k)} g_{k+1,i}^2. \tag{3.49}$$

For k sufficiently large, (3.43) implies that the newly activated constraints $\mathcal{A}(\mathbf{x}_{k+1}) \setminus \mathcal{A}(\mathbf{x}_k)$ exclude all members of $\mathcal{A}_+(\mathbf{x}^*)$. Since the $\mathbf{x}_k$ converge to $\mathbf{x}^*$, $\|\mathbf{g}_{k+1}\|_{\mathcal{N}}$ tends to zero. On the other hand, $\|\mathbf{g}_I(\mathbf{x}_k)\|$ is bounded away from zero since the index l is contained in $\mathcal{I}(\mathbf{x}_k)$. Hence, the last term in (3.49) tends to 0 as k increases, and the lower bound for α_k approaches $(1-\sigma)/\lambda$. Since $x_l^* = 0$, it follows that x_{kl} approaches 0. Since the lower bound for α_k approaches $(1-\sigma)/\lambda$, $g_l(\mathbf{x}^*) > 0$, and $\mathbf{x}_k$ converges to $\mathbf{x}^*$, we conclude that

$$x_{k+1,l} = x_{kl} - \alpha_k g_{kl} < 0$$

for k sufficiently large. This contradicts the initial assumption that constraint l is inactive for k sufficiently large. Hence, in a finite number of iterations, $x_{ki} = 0$ for all $i \in \mathcal{A}_+(\mathbf{x}^*)$.

Case 3. *The UA is executed a finite number of iterations.* In this case, the iterates are generated by NGPA for k sufficiently large. Suppose that (3.42) is violated for some $l \in \mathcal{A}_+(\mathbf{x}^*)$. We show that this leads to a contradiction. By (3.43), $x_{kl} > 0$ for all $k \geq k_+$. Since $\mathbf{x}_k$ converges to $\mathbf{x}^*$, $\mathbf{x}_l^* = 0$, and $g_l(\mathbf{x}^*) > 0$, it is possible to choose k larger, if necessary, so that

$$x_{kl} - g_{kl}\alpha_{\min} < 0. \tag{3.50}$$

Since (3.42) is violated and $\mathbf{x}_k$ converges to $\mathbf{x}^*$, we can choose k larger, if necessary, so that

$$\frac{x_{kl}}{\|\mathbf{x}_k - \mathbf{x}^*\|^2} \geq \frac{\lambda(2+\lambda)^2 \max\{1, \alpha_{\max}\}^2}{2(1-\delta)g_{kl}}, \tag{3.51}$$

where $0 < \delta < 1$ is the parameter appearing in Step 3 of NGPA, and λ is the Lipschitz constant for ∇f. We will show that for this k, we have

$$f(\mathbf{x}_k + \mathbf{d}_k) \leq f_R + \delta \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k, \tag{3.52}$$

where f_R is specified in Step 3 of NGPA. According to Step 3 of NGPA, when (3.52) holds, $\alpha_k = 1$, which implies that

$$x_{k+1,l} = x_{kl} + d_{kl}. \tag{3.53}$$

Since (3.50) holds and $\bar{\alpha}_k \geq \alpha_{\min}$, we have since

$$d_{kl} = \max\{x_{kl} - \bar{\alpha}_k g_{kl}, 0\} - x_{kl} = -x_{kl}. \tag{3.54}$$

This substitution in (3.53) gives $x_{k+1,l} = 0$, which contradicts the fact that $x_{kl} > 0$ for all $k \geq k_+$. To complete the proof, we need to show that when (3.51) holds, (3.52) is satisfied. Expanding in a Taylor series around $\mathbf{x}_k$ and utilizing (3.54) gives

$$\begin{aligned}
f(\mathbf{x}_k + \mathbf{d}_k) &= f(\mathbf{x}_k) + \int_0^1 f'(\mathbf{x}_k + t\mathbf{d}_k)dt \\
&= f(\mathbf{x}_k) + \mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k + \int_0^1 (\nabla f(\mathbf{x}_k + t\mathbf{d}_k) - \mathbf{g}_k^{\mathsf{T}})\mathbf{d}_k dt \\
&\le f(\mathbf{x}_k) + \mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k + \frac{\lambda}{2}\|\mathbf{d}_k\|^2 \\
&= f(\mathbf{x}_k) + \delta\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k + (1-\delta)\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k + \frac{\lambda}{2}\|\mathbf{d}_k\|^2 \\
&\le f(\mathbf{x}_k) + \delta\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k + (1-\delta)g_{kl}d_{kl} + \frac{\lambda}{2}\|\mathbf{d}_k\|^2 \qquad (3.55) \\
&= f(\mathbf{x}_k) + \delta\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k - (1-\delta)g_{kl}x_{kl} + \frac{\lambda}{2}\|\mathbf{d}_k\|^2. \qquad (3.56)
\end{aligned}$$

The inequality (3.55) is due to the fact that $g_{ki}d_{ki} \le 0$ for each i. By P3, P4, P5, and P7, and by the Lipschitz continuity of ∇f, we have

$$\begin{aligned}
\|\mathbf{d}_k\| &\le \max\{1, \alpha_{\max}\}\|\mathbf{d}^1(\mathbf{x}_k)\| \\
&= \max\{1, \alpha_{\max}\}\|\mathbf{d}^1(\mathbf{x}_k) - \mathbf{d}^1(\mathbf{x}^*)\| \\
&= \max\{1, \alpha_{\max}\}\|P(\mathbf{x}_k - \mathbf{g}_k) - \mathbf{x}_k - P(\mathbf{x}^* - \mathbf{g}(\mathbf{x}^*)) + \mathbf{x}^*\| \\
&\le \max\{1, \alpha_{\max}\}(\|\mathbf{x}_k - \mathbf{x}^*\| + \|P(\mathbf{x}_k - \mathbf{g}_k) - P(\mathbf{x}^* - \mathbf{g}(\mathbf{x}^*))\|) \\
&\le \max\{1, \alpha_{\max}\}(\|\mathbf{x}_k - \mathbf{x}^*\| + \|\mathbf{x}_k - \mathbf{g}_k - (\mathbf{x}^* - \mathbf{g}(\mathbf{x}^*))\|) \\
&\le \max\{1, \alpha_{\max}\}(2\|\mathbf{x}_k - \mathbf{x}^*\| + \|\mathbf{g}_k - \mathbf{g}(\mathbf{x}^*)\|) \\
&\le \max\{1, \alpha_{\max}\}(2 + \lambda)\|\mathbf{x}_k - \mathbf{x}^*\|.
\end{aligned}$$

Combining this upper bound for $\|\mathbf{d}_k\|$ with the lower bound (3.51) for x_{kl}, we conclude that

$$\begin{aligned}
\frac{\lambda}{2}\|\mathbf{d}_k\| &\le \frac{\lambda}{2}\max\{1, \alpha_{\max}\}^2(2 + \lambda)^2\|\mathbf{x}_k - \mathbf{x}^*\|^2 \\
&\le \frac{1}{2}\left(\frac{2(1-\delta)x_{kl}g_{kl}}{\|\mathbf{x}_k - \mathbf{x}^*\|^2}\right)\|\mathbf{x}_k - \mathbf{x}^*\|^2 \\
&= (1-\delta)x_{kl}g_{kl}.
\end{aligned}$$

Hence, by (3.56) and by the choice for f_R specified in Step 3 of NGPA, we have

$$f(\mathbf{x}_k + \mathbf{d}_k) \leq f(\mathbf{x}_k) + \delta \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k \leq f_R + \delta \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k. \tag{3.57}$$

This completes the proof of (3.52). □

There is a fundamental difference between the gradient projection algorithm presented in this paper, and algorithms based on a "piecewise projected gradient" [15, 16, 17]. For our gradient projection algorithm, we perform a single projection, and then we backtrack towards the starting point. Thus we are unable to show that the active constraints are identified in a finite number of iterations; in contrast, with the piecewise project gradient approach, where a series of projections may be performed, the active constraints can be identified in a finite number of iterations. In Lemma 6 we show that even though we do not identify the active constraints, the components of $\mathbf{x}_k$ corresponding to the strictly active constraints are on the order of the error in $\mathbf{x}_k$ squared.

If all the constraints are active at a stationary point $\mathbf{x}^*$ and strict complementarity holds, then convergence is achieved in a finite number of iterations:

Corollary 4 *If f is twice-continuously differentiable, the iterates $\mathbf{x}_k$ generated by ASA with $\epsilon = 0$ converge to a stationary point $\mathbf{x}^*$, and $|\mathcal{A}_+(\mathbf{x}^*)| = n$, then $\mathbf{x}_k = \mathbf{x}^*$ after a finite number of iterations.*

Proof. Let $\mathbf{x}_{k,\max}$ denote the largest component of $\mathbf{x}_k$. Since $\|\mathbf{x}_k\|^2 \leq n\mathbf{x}_{k,\max}^2$, we have

$$\frac{\mathbf{x}_{k,\max}}{\|\mathbf{x}_k\|^2} \geq \frac{1}{n\mathbf{x}_{k,\max}}. \tag{3.58}$$

Since all the constraints are active at $\mathbf{x}^*$, $\mathbf{x}_{k,\max}$ tends to zero. By (3.58) the conclusion (3.42) of Lemma 6 does not hold. Hence, after a finite number of iterations, $\mathbf{x}_k = \mathbf{x}^*$. □

Recall [110] that for any stationary point $\mathbf{x}^*$ of (3.1), the strong second-order sufficient optimality condition holds if there exists $\gamma > 0$ such that

$$\mathbf{d}^{\mathsf{T}} \nabla^2 f(\mathbf{x}^*) \mathbf{d} \geq \gamma \|\mathbf{d}\|^2, \quad \text{whenever} \quad d_i = 0 \text{ for all } i \in \mathcal{A}_+(\mathbf{x}^*). \tag{3.59}$$

Using P8, we establish the following:

Lemma 7 *If f is twice-continuously differentiable near a stationary point $\mathbf{x}^*$ of (3.1) satisfying the strong second-order sufficient optimality condition, then there exists $\rho > 0$ with the following property:*

$$\|\mathbf{x} - \mathbf{x}^*\| \leq \sqrt{1 + \left(\frac{(1+\lambda)^2}{.5\gamma} \right)^2} \|\mathbf{d}^1(\mathbf{x})\| \tag{3.60}$$

for all $\mathbf{x} \in B_\rho(\mathbf{x}^*)$, where λ is any Lipschitz constant for ∇f over $B_\rho(\mathbf{x}^*)$.

Proof. By the continuity of the second derivative of f, it follows from (3.59) that for $\rho > 0$ sufficiently small,

$$(\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{x}^*))^{\mathsf{T}} (\mathbf{x} - \mathbf{x}^*) \geq .5\gamma \|\mathbf{x} - \mathbf{x}^*\|^2 \tag{3.61}$$

for all $\mathbf{x} \in B_\rho(\mathbf{x}^*)$ with $x_i = 0$ for all $i \in \mathcal{A}_+(\mathbf{x}^*)$. Choose ρ smaller if necessary so that

$$x_i - g_i(\mathbf{x}) \leq 0 \text{ for all } i \in \mathcal{A}_+(\mathbf{x}^*) \text{ and } \mathbf{x} \in B_\rho(\mathbf{x}^*). \tag{3.62}$$

Let $\bar{\mathbf{x}}$ be defined as follows:

$$\bar{x}_i = \begin{cases} 0 \text{ if } i \in \mathcal{A}_+(\mathbf{x}^*), \\ x_i \text{ otherwise.} \end{cases} \tag{3.63}$$

Since (3.62) holds, it follows that

$$\|\mathbf{x} - \bar{\mathbf{x}}\| \leq \|\mathbf{d}^1(\mathbf{x})\| \tag{3.64}$$

for all $\mathbf{x} \in B_\rho(\mathbf{x}^*)$. Also, by (3.62), we have

$$[P(\bar{\mathbf{x}} - \mathbf{g}(\mathbf{x})) - \bar{\mathbf{x}}]_i = 0 \quad \text{and} \quad \mathbf{d}^1(\mathbf{x})_i = [P(\mathbf{x} - \mathbf{g}(\mathbf{x})) - \mathbf{x}]_i = -x_i$$

for all $i \in \mathcal{A}_+(\mathbf{x}^*)$, while

$$[P(\bar{\mathbf{x}} - \mathbf{g}(\mathbf{x})) - \bar{\mathbf{x}}]_i = \mathbf{d}^1(\mathbf{x})_i = [P(\mathbf{x} - \mathbf{g}(\mathbf{x})) - \mathbf{x}]_i$$

for $i \notin \mathcal{A}_+(\mathbf{x}^*)$. Hence, we have

$$\|P(\bar{\mathbf{x}} - \mathbf{g}(\mathbf{x})) - \bar{\mathbf{x}}\| \le \|\mathbf{d}^1(\mathbf{x})\| \tag{3.65}$$

for all $\mathbf{x} \in B_\rho(\mathbf{x}^*)$. By the Lipschitz continuity of $\mathbf{g}$, (3.64), (3.65), and P3, it follows that

$$\begin{aligned} \|\mathbf{d}^1(\bar{\mathbf{x}})\| &= \|P(\bar{\mathbf{x}} - \mathbf{g}(\bar{\mathbf{x}})) - P(\bar{\mathbf{x}} - \mathbf{g}(\mathbf{x})) + P(\bar{\mathbf{x}} - \mathbf{g}(\mathbf{x})) - \bar{\mathbf{x}}\| \\ &\le \lambda\|\bar{\mathbf{x}} - \mathbf{x}\| + \|\mathbf{d}^1(\mathbf{x})\| \\ &\le (1+\lambda)\|\mathbf{d}^1(\mathbf{x})\| \end{aligned} \tag{3.66}$$

for all $\mathbf{x} \in B_\rho(\mathbf{x}^*)$. By P8, (3.61), and (3.66), we have

$$\|\bar{\mathbf{x}} - \mathbf{x}^*\| \le \left(\frac{1+\lambda}{.5\gamma}\right)\|\mathbf{d}^1(\bar{\mathbf{x}})\| \le \left(\frac{(1+\lambda)^2}{.5\gamma}\right)\|\mathbf{d}^1(\mathbf{x})\|. \tag{3.67}$$

Since $\|\mathbf{x} - \bar{\mathbf{x}}\|^2 + \|\bar{\mathbf{x}} - \mathbf{x}^*\|^2 = \|\mathbf{x} - \mathbf{x}^*\|^2$, the proof is completed by squaring and adding (3.67) and (3.64). □

We now show that the undecided index set $\mathcal{U}$ becomes empty as the iterates approach a stationary point where the strong second-order sufficient optimality condition holds.

Lemma 8 *Suppose f is twice-continuously differentiable, $\mathbf{x}^*$ is a stationary point of (3.1) satisfying the strong second-order sufficient optimality condition, and $\mathbf{x}_k$, $k = 0, 1, \ldots$, is an infinite sequence of feasible iterates for (3.1) converging to $\mathbf{x}^*$, $\mathbf{x}_k \ne \mathbf{x}^*$ for each k. If there exists a constant ξ such that*

$$\limsup_{k\to\infty} \frac{x_{ki}}{\|\mathbf{x}_k - \mathbf{x}^*\|^2} \le \xi < \infty \tag{3.68}$$

for all $i \in \mathcal{A}_+(\mathbf{x}^*)$, then $\mathcal{U}(\mathbf{x}_k)$ is empty for k sufficiently large.

Proof. To prove that $\mathcal{U}(\mathbf{x})$ is empty, we must show that for each $i \in [1, n]$, one of the following inequalities is violated:

$$\begin{aligned} |g_i(\mathbf{x})| &\geq \|\mathbf{d}^1(\mathbf{x})\|^\alpha \text{ or} && (3.69) \\ x_i &\geq \|\mathbf{d}^1(\mathbf{x})\|^\beta. && (3.70) \end{aligned}$$

By Lemma 7, there exists a constant c such that $\|\mathbf{x} - \mathbf{x}^*\| \leq c\|\mathbf{d}^1(\mathbf{x})\|$ for all $\mathbf{x}$ near $\mathbf{x}^*$. If $i \in \mathcal{A}_+(\mathbf{x}^*)$, then by (3.68), we have

$$\limsup_{k\to\infty} \frac{x_{ki}}{\|\mathbf{d}^1(\mathbf{x}_k)\|^\beta} \leq \limsup_{k\to\infty} \frac{\xi\|\mathbf{x}_k - \mathbf{x}^*\|^2}{\|\mathbf{d}^1(\mathbf{x}_k)\|^\beta} \leq \limsup_{k\to\infty} \ \xi c^2 \|\mathbf{d}^1(\mathbf{x}_k)\|^{2-\beta} = 0$$

since $\beta \in (1, 2)$. Hence, for each $i \in \mathcal{A}_+(\mathbf{x}^*)$, (3.70) is violated for k sufficiently large.

If $i \notin \mathcal{A}_+(\mathbf{x}^*)$, then $g_i(\mathbf{x}^*) = 0$. By Lemma 7, we have

$$\begin{aligned} \limsup_{k\to\infty} \frac{|g_i(\mathbf{x}_k)|}{\|\mathbf{d}^1(\mathbf{x}_k)\|^\alpha} &= \limsup_{k\to\infty} \frac{|g_i(\mathbf{x}_k) - g_i(\mathbf{x}^*)|}{\|\mathbf{d}^1(\mathbf{x}_k)\|^\alpha} \\ &\leq \limsup_{k\to\infty} \frac{\lambda\|\mathbf{x}_k - \mathbf{x}^*\|}{\|\mathbf{d}^1(\mathbf{x}_k)\|^\alpha} \\ &\leq \limsup_{k\to\infty} \ \lambda c \|\mathbf{d}^1(\mathbf{x}_k)\|^{1-\alpha} = 0, \end{aligned}$$

since $\alpha \in (0, 1)$. Here, λ is a Lipschitz constant for $\mathbf{g}$ in a neighborhood of $\mathbf{x}^*$. Hence, (3.69) is violated if $i \notin \mathcal{A}_+(\mathbf{x}^*)$. □

Remark. If $i \in \mathcal{A}_+(\mathbf{x}^*)$ and the iterates $\mathbf{x}_k$ converge to a stationary point $\mathbf{x}^*$, then $g_i(\mathbf{x}_k)$ is bounded away from 0 for k sufficiently large. Since $\mathbf{d}^1(\mathbf{x}_k)$ tends to zero, the inequality $|g_i(\mathbf{x}_k)| \geq \|\mathbf{d}^1(\mathbf{x}_k)\|^\alpha$ is satisfied for k sufficiently large. Hence, if $\mathcal{U}(\mathbf{x}_k)$ is empty and $i \in \mathcal{A}_+(\mathbf{x}^*)$, then $x_{ki} < \|\mathbf{d}^1(\mathbf{x}_k)\|^\beta$ where $\beta \in (1, 2)$. In other words, when $\mathcal{U}(\mathbf{x}_k)$ is empty, the components of $\mathbf{x}_k$ associated with strictly active indices $\mathcal{A}_+(\mathbf{x}^*)$ are going to zero faster than the error $\|\mathbf{d}^1(\mathbf{x}_k)\|$.

Lemma 9 *Suppose f is twice-continuously differentiable, $\mathbf{x}^*$ is a stationary point of (3.1) satisfying the strong second-order sufficient optimality condition, and $\mathbf{x}_k$,*

$k = 0, 1, \ldots$, is an infinite sequence of feasible iterates for (3.1) converging to $\mathbf{x}^*$, $\mathbf{x}_k \neq \mathbf{x}^*$ for each k. If there exists a constant ξ such that

$$\limsup_{k\to\infty} \frac{x_{ki}}{\|\mathbf{x}_k - \mathbf{x}^*\|^2} \le \xi < \infty \tag{3.71}$$

for all $i \in \mathcal{A}_+(\mathbf{x}^*)$, then there exist $\mu^* > 0$ such that

$$\|\mathbf{g}_I(\mathbf{x}_k)\| \ge \mu^* \|\mathbf{d}^1(\mathbf{x}_k)\| \tag{3.72}$$

for k sufficiently large.

Proof. Choose $\rho > 0$ and let λ be the Lipschitz constant for ∇f in $B_\rho(\mathbf{x}^*)$. As in (3.63), let $\bar{\mathbf{x}}$ be defined by $\bar{x}_i = 0$ if $i \in \mathcal{A}_+(\mathbf{x}^*)$ and $\bar{x}_i = x_i$ otherwise. If $\mathbf{x}_k \in B_\rho(\mathbf{x}^*)$, we have

$$\begin{aligned} \|\mathbf{d}^1(\mathbf{x}_k)\| &\le \|\mathbf{d}^1(\mathbf{x}_k) - \mathbf{d}^1(\mathbf{x}^*)\| \\ &\le \|\mathbf{d}^1(\mathbf{x}_k) - \mathbf{d}^1(\bar{\mathbf{x}}_k)\| + \|\mathbf{d}^1(\bar{\mathbf{x}}_k) - \mathbf{d}^1(\mathbf{x}^*)\| \\ &\le (2+\lambda)(\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| + \|\bar{\mathbf{x}}_k - \mathbf{x}^*\|) \end{aligned} \tag{3.73}$$

Utilizing (3.71) gives

$$\begin{aligned} \|\bar{\mathbf{x}}_k - \mathbf{x}_k\| &\le \sum_{i=1}^n |\bar{x}_{ki} - x_{ki}| \\ &= \sum_{i\in\mathcal{A}_+(\mathbf{x}^*)} x_{ki} \le n\xi\|\mathbf{x}_k - \mathbf{x}^*\|^2 \\ &\le n\xi\|\mathbf{x}_k - \mathbf{x}^*\|(\|\mathbf{x}_k - \bar{\mathbf{x}}_k\| + \|\bar{\mathbf{x}}_k - \mathbf{x}^*\|). \end{aligned}$$

Since $\mathbf{x}_k$ converges to $\mathbf{x}^*$, it follows that for any $\epsilon > 0$,

$$\|\bar{\mathbf{x}}_k - \mathbf{x}_k\| \le \epsilon\|\bar{\mathbf{x}}_k - \mathbf{x}^*\| \tag{3.74}$$

when k is sufficiently large. Combining (3.73) and (3.74), there exists a constant $c > 0$ such that

$$\|\mathbf{d}^1(\mathbf{x}_k)\| \le c\|\bar{\mathbf{x}}_k - \mathbf{x}^*\| \tag{3.75}$$

for k sufficiently large.

Let k be chosen large enough that

$$\|\mathbf{x}_k - \mathbf{x}^*\| < \min\{x_i^* : i \in \mathcal{I}(\mathbf{x}^*)\}. \tag{3.76}$$

Suppose, in this case, that $i \in \mathcal{A}(\mathbf{x}_k)$. If $x_i^* > 0$, then $\|\mathbf{x}_k - \mathbf{x}^*\| \geq x_i^*$, which contradicts (3.76). Hence, $\bar{x}_{ki} = x_i^* = 0$. Moreover, if $i \in \mathcal{A}_+(\mathbf{x}^*)$, then by the definition (3.63), $\bar{x}_{ki} = x_i^* = 0$. In summary,

$$\begin{cases} \bar{x}_{ki} = x_i^* = 0 & \text{for each } i \in \mathcal{A}(\mathbf{x}_k) \cup \mathcal{A}_+(\mathbf{x}^*), \\ g_i(\mathbf{x}^*) = 0 & \text{for each } i \in \mathcal{A}_+(\mathbf{x}^*)^c, \end{cases} \tag{3.77}$$

where $\mathcal{A}_+(\mathbf{x}^*)^c$ is the complement of $\mathcal{A}_+(\mathbf{x}^*)$. Define $\mathcal{Z} = \mathcal{A}(\mathbf{x}_k)^c \cap \mathbf{A}_+(\mathbf{x}^*)^c$.

By the strong second-order sufficient optimality condition and for $\mathbf{x}$ near $\mathbf{x}^*$, we have

$$\begin{aligned} \frac{\gamma}{2}\|\bar{\mathbf{x}} - \mathbf{x}^*\|^2 &\leq [\bar{\mathbf{x}} - \mathbf{x}^*]^{\mathsf{T}} \int_0^1 \nabla^2 f(\mathbf{x}^* + t(\bar{\mathbf{x}} - \mathbf{x}^*))dt \;\; [\bar{\mathbf{x}} - \mathbf{x}^*] \\ &= (\bar{\mathbf{x}} - \mathbf{x}^*)^{\mathsf{T}}(\mathbf{g}(\bar{\mathbf{x}}) - \mathbf{g}(\mathbf{x}^*)). \end{aligned} \tag{3.78}$$

We substitute $\mathbf{x} = \mathbf{x}_k$ in (3.78) and utilize (3.77) to obtain

$$\begin{aligned} (\bar{\mathbf{x}}_k - \mathbf{x}^*)^{\mathsf{T}}(\mathbf{g}(\bar{\mathbf{x}}_k) - \mathbf{g}(\mathbf{x}^*)) &= \sum_{i=1}^{n} (\bar{x}_{ki} - x_i^*)(g_i(\bar{\mathbf{x}}_k) - g_i(\mathbf{x}^*)) \\ &= \sum_{i \in \mathcal{Z}} (\bar{x}_{ki} - x_i^*) g_i(\bar{\mathbf{x}}_k) \\ &\leq \|\bar{\mathbf{x}}_k - \mathbf{x}^*\| \left(\sum_{i \in \mathcal{I}(\mathbf{x}_k)} g_i(\bar{\mathbf{x}}_k)^2 \right)^{1/2}, \end{aligned} \tag{3.79}$$

since $\mathcal{Z} \subset \mathcal{A}(\mathbf{x}_k)^c = \mathcal{I}(\mathbf{x}_k)$. Exploiting the Lipschitz continuity of ∇f, (3.79) gives

$$(\bar{\mathbf{x}}_k - \mathbf{x}^*)^{\mathsf{T}}(\mathbf{g}(\bar{\mathbf{x}}_k) - \mathbf{g}(\mathbf{x}^*)) \leq \|\bar{\mathbf{x}}_k - \mathbf{x}^*\|(\|\mathbf{g}_I(\mathbf{x}_k)\| + \lambda\|\bar{\mathbf{x}}_k - \mathbf{x}_k\|). \tag{3.80}$$

Combining (3.74), (3.78), and (3.80), we conclude that for k sufficiently large,

$$\frac{\gamma}{4}\|\bar{\mathbf{x}}_k - \mathbf{x}^*\| \le \|\mathbf{g}_I(\mathbf{x}_k)\|. \tag{3.81}$$

Combining (3.75) and (3.81), the proof is complete. □

Remark. If $\mathbf{x}_k$ is sequence converging to a nondegenerate stationary point $\mathbf{x}^*$, then (3.72) holds with $\mu^* = 1$, without assuming either the strong second-order sufficient optimality condition or (3.71) – see Theorem 14. In Lemma 9, the optimization problem could be degenerate.

We now show that after a finite number of iterations, ASA will perform only the unconstrained algorithm (UA) with a fixed active constraint set.

Theorem 15 *If f is twice-continuously differentiable and the iterates $\mathbf{x}_k$ generated by ASA with $\epsilon = 0$ converge to a stationary point $\mathbf{x}^*$ satisfying the strong second-order sufficient optimality condition, then after a finite number of iterations, ASA performs only the UA without restarts.*

Proof. By Lemma 6, the hypotheses (3.68) and (3.71) of Lemmas 8 and 9 are satisfied. Hence, for k sufficiently large, the undecided set $\mathcal{U}(\mathbf{x}_k)$ is empty and the lower bound (3.72) holds. In Step 1a, if $\|\mathbf{g}_I(\mathbf{x}_k)\| < \mu\|\mathbf{d}^1(\mathbf{x}_k)\|$, then μ is multiplied by the factor $\rho < 1$. When $\mu < \mu^*$, Lemma 9 implies that $\|\mathbf{g}_I(\mathbf{x}_k)\| \ge \mu\|\mathbf{d}^1(\mathbf{x}_k)\|$. Hence, Step 1a of ASA branches to Step 2, while Step 2 cannot branch to Step 1 since the condition $\|\mathbf{g}_I(\mathbf{x}_k)\| < \mu\|\mathbf{d}^1(\mathbf{x}_k)\|$ is never satisfied in Step 2a and $\mathcal{U}(\mathbf{x}_k)$ is empty in Step 2b for k sufficiently large. Since the UA only adds constraints, we conclude that after a finite number of iterations, the active set does not change. □

Remark. If f is a strongly convex quadratic function, then by Corollary 3, the iterates $\mathbf{x}_k$ converge to the global minimizer $\mathbf{x}^*$. If the UA is based on the conjugate gradient method for which there is finite convergence when applied to a convex quadratic, it follows from Theorem 15 that ASA converges in a finite number of iterations.

We now give the proof of Corollary 3; that is, when f is strongly convex and twice continuously differentiable on $\mathcal{B}$, and assumption A3 of Theorem 13 is satisfied, then the entire sequence of iterates generated by ASA converges to the global minimizer $\mathbf{x}^*$. Note that the assumptions of Corollary 3 are weaker than those of Corollary 2 (global convergence of NGPA) since Corollary 3 only requires that $f_k^r \leq f_k^{\max}$ infinitely often in NGPA.

Proof. For a strongly convex function, A1 and A2 always hold. Since all the assumptions of Theorem 13 are satisfied, there exists a subsequence $\mathbf{x}_{k_j}$, $j = 1, 2, \ldots$, of the iterates such that

$$\lim_{j \to \infty} \|\mathbf{d}^1(\mathbf{x}_{k_j})\| = 0.$$

Since the UA is monotone and since the NGPA satisfies (3.14) and (3.15), it follows from the strong convexity of f that the $\mathbf{x}_{k_j}$ are contained in a bounded set. Since $\mathbf{d}^1(\cdot)$ is continuous, there exists a subsequence, also denote $\mathbf{x}_{k_j}$, converging to a limit $\mathbf{x}^*$ with $\mathbf{d}^1(\mathbf{x}^*) = \mathbf{0}$. Since the unique stationary point of a strongly convex function is its global minimizer, $\mathbf{x}^*$ is the global solution of (3.1).

Case A. *There exists an infinite subsequence, also denoted* $\{\mathbf{x}_{k_j}\}$*, with the property that* $\mathbf{x}_{k_j+1}$ *is generated by the UA.* In this case, we are done since the UA is monotone and the inequality

$$f(\mathbf{x}_k) \leq f(\mathbf{x}_{k_j}) \tag{3.82}$$

holds for all $k \geq k_j$ (see (3.14) and (3.15)). Since $\mathbf{x}_{k_j}$ converges to $\mathbf{x}^*$, it follows that $f(\mathbf{x}_{k_j})$ converges to $f(\mathbf{x}^*)$, and hence, by (3.82) and (3.34), the entire sequence converges to $\mathbf{x}^*$.

Case B. *There exists an infinite subsequence, also denoted* $\{\mathbf{x}_{k_j}\}$*, with the property that* $\mathbf{x}_{k_j+1}$ *is generated by NGPA.* Either

$$\limsup_{j\to\infty} \frac{(\mathbf{x}_{k_j})_i}{\|\mathbf{x}_{k_j} - \mathbf{x}^*\|^2} < \infty \quad \text{for all } i \in \mathcal{A}_+(\mathbf{x}^*), \tag{3.83}$$

or (3.83) is violated. By the analysis given in Case 3 of Lemma 6, when (3.83) is violated, (3.52) holds, from which it follows that for j sufficiently large,

$$\mathbf{x}_{k_j+1,i} = 0 \quad \text{for all } i \in \mathcal{A}_+(\mathbf{x}^*). \tag{3.84}$$

Hence, either the sequence $\mathbf{x}_{k_j}$ satisfies (3.83) or the sequence $\mathbf{x}_{k_j+1}$ satisfies (3.84). In this latter case, it follows from (3.57) that

$$f(\mathbf{x}_{k_j+1}) \le f(\mathbf{x}_{k_j}).$$

Since $f(\mathbf{x}_{k_j})$ converges to $f(\mathbf{x}^*)$, we conclude that $f(\mathbf{x}_{k_j+1})$ converges to $f(\mathbf{x}^*)$, and $\mathbf{x}_{k_j+1}$ converges to $\mathbf{x}^*$.

In either case (3.83) or (3.84), there exists a sequence K_j (either $K_j = k_j$ or $K_j = k_j + 1$) with the property that $\mathbf{x}_{K_j}$ converges to $\mathbf{x}^*$ and

$$\limsup_{j\to\infty} \frac{(\mathbf{x}_{K_j})_i}{\|\mathbf{x}_{K_j} - \mathbf{x}^*\|^2} < \infty \quad \text{for all } i \in \mathcal{A}_+(\mathbf{x}^*).$$

By Lemma 8, $\mathcal{U}(\mathbf{x}_{K_j})$ is empty for j sufficiently large. By Lemma 9, there exists $\mu^* > 0$ such that

$$\|\mathbf{g}_I(\mathbf{x}_{K_j}) \ge \mu^* \|\mathbf{d}^1(\mathbf{x}_{K_j})\|$$

for j sufficiently large. As in the proof of Theorem 15, at iteration K_j for j sufficiently large, ASA jumps from Step 1 to the UA in Step 2. Hence, for j sufficiently large, $\mathbf{x}_{K_j+1}$ is generated by the UA, which implies that Case A holds.

□

3.3.3 Numerical Comparisons

This section compares the CPU time performance of ASA, implemented using the nonlinear conjugate gradient code CG_DESCENT for the unconstrained algorithm and the CBB method for the nonmonotone gradient projection algorithm, to the performance of the following codes:

- L-BFGS-B [18, 135]: The limited memory quasi-Newton method of Zhu, Byrd, Nocedal (ACM Algorithm 778).
- SPG2 Version 2.1 [10, 11]: The nonmonotone spectral projected gradient method of Birgin, Martínez, and Raydan (ACM Algorithm 813).
- GENCAN [9]: The monotone active set method with spectral projected gradients developed by Birgin and Martínez.
- TRON Version 1.2 [90]: A Newton trust region method with incomplete Cholesky preconditioning developed by Lin and Moré.

A detailed description of our implementation of ASA can be found in [78].

L-BFGS-B was downloaded from Jorge Nocedal's web page, TRON was downloaded from Jorge Moré's web page, SPG2 and GENCAN were downloaded on June 28, 2005, from the TANGO web page maintained by Ernesto Birgin. All codes are written in Fortran and compiled with f77 (default compiler settings) on a Sun workstation. The stopping condition was

$$\|P(\mathbf{x} - \mathbf{g}(\mathbf{x})) - \mathbf{x}\|_\infty \leq 10^{-6},$$

where $\| \cdot \|_\infty$ denotes the sup-norm of a vector. In running any of these codes, default values were used for all parameters. In NGPA, we chose the following parameter values:

$$\alpha_{\min} = 10^{-20}, \quad \alpha_{\max} = 10^{+20}, \quad \eta = .5, \quad \delta = 10^{-4}, \quad M = 8.$$

Here M is the memory used to evaluate $f_k^{\max}$ (see (3.5)). In ASA the parameter values were as follows:

$$\mu = .1, \quad \rho = .5, \quad n_1 = 2, \quad n_2 = 1.$$

In the CBB method, the parameter values were the following:

$$\theta = .975, \quad L = 3, \quad A = 40, \quad m = 4, \quad \gamma_1 = M/L, \quad \gamma_2 = A/M.$$

The separation parameter Δ in condition R4 was the natural separation between floating point numbers. That is, R4 was satisfied when the floating point version of f_{k+1} was strictly less than the floating point version of $f_k^{\min}$.

The test set consisted of all 50 box constrained problems in the CUTEr library [12] with dimensions between 50 and 15,625, and all 23 box constrained problems in the MINPACK-2 library [2] with dimension 2500. TRON is somewhat different from the other codes since it employs Hessian information and an incomplete Cholesky preconditioner, while the codes ASA, L-BFGS-B, SPG2, and GENCAN only utilize gradient information. When we compare our code to TRON, we use the same Lin/Moré preconditioner [89] used by TRON for our unconstrained algorithm. The preconditioned ASA code is called P-ASA. Since TRON is targeted to large-sparse problems, we compare with TRON using the 23 MINPACK-2 problems and the 42 sparsest CUTEr problems (the number of nonzeros in the Hessian was at most 1/5 the total number of entries). The codes L-BFGS-B, SPG2, and GENCAN were implemented for the CUTEr test problems, while ASA and TRON were implemented for both test sets, CUTEr and MINPACK-2.

The cpu time in seconds and the number of iterations, function evaluations, and gradient evaluations for each of the methods are posted at the author's web site. In running the numerical experiments, we checked whether different codes converged to different local minimizers; when comparing the codes, we

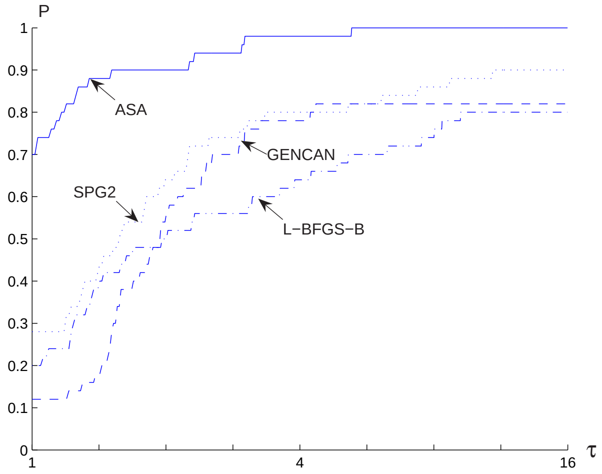

Figure 3–2: Performance profiles, 50 CUTEr test problems

restricted outselves to test problems where all codes converged to the same local minimizer, and where the running time of the fastest code exceeded .01 seconds. The numerical results are now analyzed.

The performance of the algorithms, relative to cpu time, was evaluated using the performance profiles of Dolan and Moré [43]. That is, for each method, we plot the fraction P of problems for which the method is within a factor τ of the best time. In Figure 3–2, we compare the performance of the 4 codes ASA, L-BFGS-B, SPG2, and GENCAN using the 50 CUTEr test problems. The left side of the figure gives the percentage of the test problems for which a method is the fastest; the right side gives the percentage of the test problems that were successfully solved by each of the methods. The top curve is the method that solved the most problems in a time that was within a factor τ of the best time. Since the top curve in Figure 3–2 corresponds to ASA, this algorithm is clearly fastest for this set of 50 test problems with dimensions ranging from 50 to 15,625. The relative difference in performance between ASA and the competing methods seen in Figure 3–2 is greater than the relative difference in performance between the CG_DESCENT

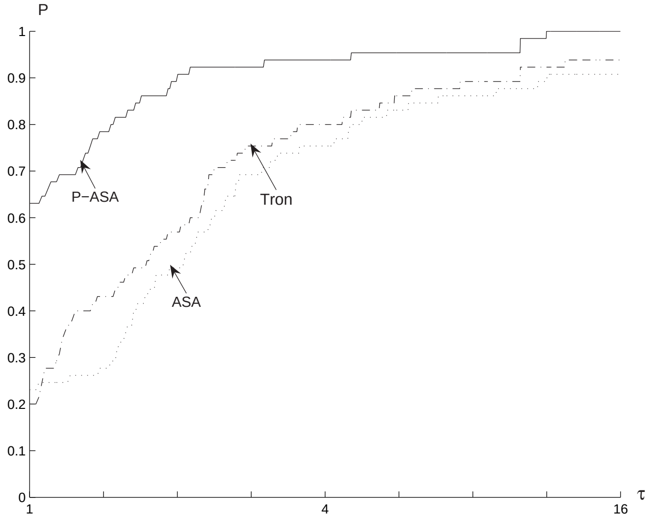

Figure 3–3: Performance profiles, 42 sparsest CUTEr problems, 23 MINPACK-2 problems, $\epsilon = 10^{-6}$

code and competing methods as seen in the figures in [73, 74]. Hence, both the gradient projection algorithm and the conjugate gradient algorithm are contributing to the better performance of ASA.

In Figure 3–3 we compare the performance of TRON with P-ASA and ASA for the 42 sparsest CUTEr test problems and the 23 MINPACK-2 problems. Observe that P-ASA has the top performance, and that ASA, which only utilizes the gradient, performs almost as well as the Hessian-based code TRON. The number of CG iterations performed by the P-ASA code is much less than the number of CG iterations performed by the ASA code. Finally, in Figure 3–4 we compare the performance of P-ASA with ASA for the relaxed convergence tolerance $\epsilon = 10^{-2}\|\mathbf{d}^1(\mathbf{x}_0)\|_\infty$. Based on Figures 3–3 and 3–4, the preconditioned ASA scheme is more efficient than unconditioned ASA for the more stringent stopping criterion, while the unconditioned and preconditioned schemes are equally effective for a more relaxed stopping criterion. Although the performance profile for ASA is beneath 1 in Figure 3–3, it reaches 1 as τ increases – there are some problems

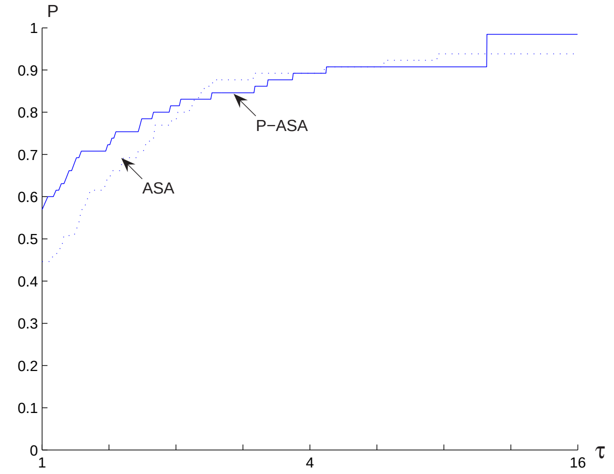

Figure 3–4: Performance profiles, $\epsilon = 10^{-2}\|\mathbf{d}^1(\mathbf{x}_0)\|_\infty$

where P-ASA is more than 16 times faster than ASA. Due to these difficult problems, the ASA profile is still beneath 1 for $\tau = 16$.

When we solve an optimization problem, the solution time consists of two parts:

T1. the time associated with the evaluation of the function or its gradient or its Hessian, and

T2. the remaining time, which is often dominated by the time used in the linear algebra.

The cpu time performance profile measures a mixture of T1 and T2 for a set of test problems. In some applications, T1 (the evaluation time) may dominate. In order to assess how the algorithms may perform in the limit, when T2 is negligible compared with T1, we could ignore T2 and compare the algorithms based on T1. In the next set of experiments, we explore how the algorithms perform in the limit, as T1 becomes infinitely large relative to T2.

Typically, the time to evaluate the gradient of a function is greater than the time to evaluate the function itself. And the time to evaluate the Hessian is greater

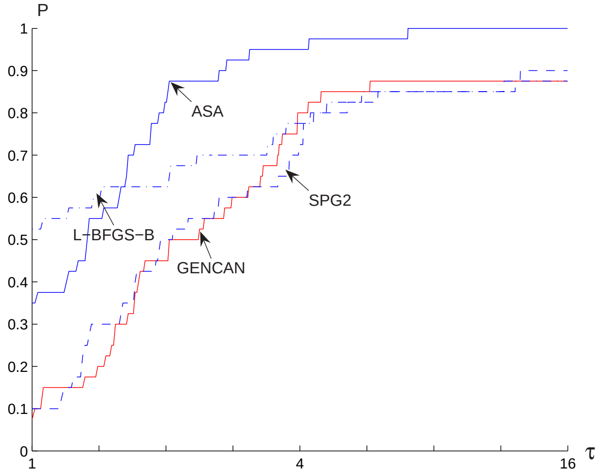

Figure 3–5: Performance profiles, evaluation metric, 50 CUTEr test problems, gradient-based methods

than the time to evaluate the gradient. If the time to evaluate the function is 1, then the average time to evaluate the gradient and Hessian for the CUTEr bound constrained test set is as follows:

$$\text{function} \ = 1, \quad \text{gradient} \ = 2.6, \quad \text{Hessian} \ = 21.0.$$

Similarly, for the MINPACK2 test set, the relative evaluation times are

$$\text{function} \ = 1, \quad \text{gradient} \ = 2.0, \quad \text{Hessian} \ = 40.5,$$

on average.

For each method and for each test problem, we compute an “evaluation time” where the time for a function evaluation is 1, the time for a gradient evaluation is either 2.6 (CUTEr) or 2.0 (MINPACK2), and the time for a Hessian evaluation is either 21.0 (CUTEr) or 40.5 (MINPACK2). In Figure 3–5 we compare the performance of gradient-based methods, and in Figure 3–6, we compare the performance of the gradient-based ASA and the methods which exploit the Hessian (P-ASA or TRON).

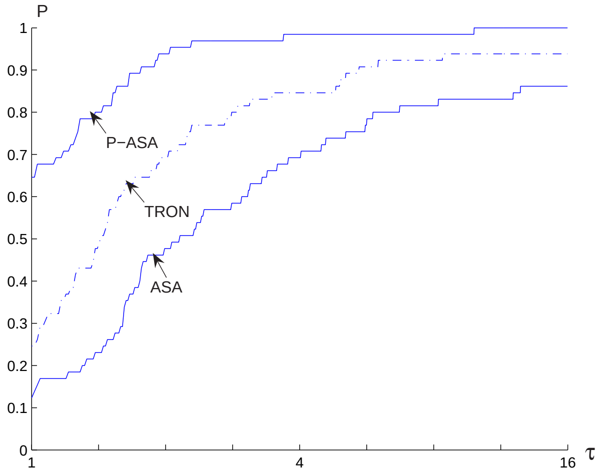

Figure 3–6: Performance profiles, evaluation metric, 42 sparsest CUTEr problems, 23 MINPACK-2 problems

In Figure 3–5 we see that for the evaluation metric and τ near 1, L-BFGS-B is performs better than ASA, but as τ increases, ASA dominates L-BFGS-B. In other words, in the evaluation metric, there are more problems where L-BFGS-B is faster than the other methods, however, ASA is not much slower than L-BFGS-B. When τ reaches 1.5, ASA starts to dominate L-BFGS-B.

In Figure 3–6 we see that P-ASA dominates TRON in the evaluation metric. Hence, even though TRON uses far fewer function evaluations, it uses many more Hessian evaluations. Since the time to evaluate the Hessian is much greater than the time to evaluate the function, P-ASA has better performance. In summary, by neglecting the time associated with the linear algebra, the relative gap between P-ASA and TRON decreases while the relative gap between TRON and ASA increases, as seen in Figure 3–6. Nonethelesss, in the evaluation metric, the performance profile for P-ASA is still above the profile for TRON.

CHAPTER 4
CONCLUSIONS AND FUTURE RESEARCH

We have presented some novel approaches by using gradient methods for solving large-scale general unconstrained and box constrained optimization. In the first part of this dissertation, we focus on the unconstrained optimization. We first introduced a nonlinear conjugate gradient method, CG_DESCENT, which guarantees to generate sufficient descent searching directions independent of linesearch. We proved global convergence of this method for strongly convex and general objective functions respectively. Extensive numerical results indicate CG_DESCENT is a very efficient and robust method which outperforms many state-of-the-art software for solving unconstrained optimization. We then introduced a recently developed cyclic Barzilai-Borwein (CBB) method. We proved for general nonlinear functions, CBB method will have at least linearly convergent at a stationary point with positive definite Hessian. In addition, our numerical experiments indicated if the cycle length is large, convergence rate could be superlinear. By combing the CBB method with an adaptive nonmonotone line search, we globalized the method to solve general unconstrained optimization. For degenerate optimization problems, where multiple minimizers may exist, or where the Hessian may be singular at a local minimizer, we prosed a class of self-adaptive proximal point methods. By particularly choosing the regularization parameter, we showed the distance between the iterates and the solution set will be at least superlinearly convergent. In the second part of this dissertation, we proposed an completely new active set method for box constrained optimization. Our active set algorithm consists of a nonmonotone gradient projection step, an unconstrained optimization step, and a set of rules for branching between the two steps. Global convergence to a stationary point is established. This algorithm eventually reduces to unconstrained optimization even without assuming the strict complementarity condition. A specific implementation of ASA is given which exploits the cyclic Barzilai-Borwein algorithm for the gradient projection step and the CG_DESCENT for unconstrained optimization. Extensive numerical results are also provided.

This project is far from finished yet. Right now, I am continuing the line of research outlined above. First, I am working on extending ASA [78] to solve nonlinear optimization problems with box and linear constraints. Second, an efficient preconditioner which takes the advantage of the structure of problem need to be developed. Both of the above projects are crucial to the development of highly efficient software for general nonlinear optimization. This work also exploits many techniques in numerical linear algebra, sparse matrix computations and parallel computing. At the same time that I work on the development of general, efficient optimization techniques, I would like apply these techniques to signal processing, medical image processing, both on MRI an PET, molecular model and related fields.

REFERENCES

[1] M. Al-Baali and R. Fletcher. An efficient line search for nonlinear least squares. *J. Optim. Theory Appl.*, 48:359–377, 1984.

[2] B. M. Averick, R. G. Carter, J. J. Moré, and G. L. Xue. The MINPACK-2 test problem collection. Technical report, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1992.

[3] J. Barzilai and J. M. Borwein. Two point step size gradient methods. *IMA J. Numer. Anal.*, 8:141–148, 1988.

[4] D. P. Bertsekas. On the Goldstein-Levitin-Polyak gradient projection method. *IEEE Trans. Automatic Control*, 21:174–184, 1976.

[5] D. P. Bertsekas. Projected Newton methods for optimization problems with simple constraints. *SIAM J. Control Optim.*, 20:221–246, 1982.

[6] D. P. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, MA, 1999.

[7] E. G. Birgin, R. Biloti, M. Tygel, and L. T. Santos. Restricted optimization: a clue to a fast and accurate implementation of the common reflection surface stack method. *J. Appl. Geophysics*, 42:143–155, 1999.

[8] E. G. Birgin, I. Chambouleyron, and J. M. Martínez. Estimation of the optical constants and the thickness of thin films using unconstrained optimization. *J. Comput. Phys.*, 151:862–880, 1999.

[9] E. G. Birgin and J. M. Martínez. Large-scale active-set box-constrained optimization method with spectral projected gradients. *Comput. Optim. Appl.*, 23:101–125, 2002.

[10] E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods for convex sets. *SIAM J. Optim.*, 10:1196–1211, 2000.

[11] E. G. Birgin, J. M. Martínez, and M. Raydan. Algorithm 813: SPG - software for convex-constrained optimization. *ACM Trans. Math. Software*, 27:340–349, 2001.

[12] I. Bongartz, A. R. Conn, N. I. M. Gould, and P. L. Toint. CUTE: constrained and unconstrained testing environments. *ACM Trans. Math. Software*, 21:123–160, 1995.

[13] I. Bongartz, A. R. Conn, N. I. M. Gould, and P. L. Toint. CUTE: constrained and unconstrained testing environments. *ACM Trans. Math. Software*, 21:123–160, 1995.

[14] M.A. Branch, T.F. Coleman, and Y. Li. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM J. Sci. Comput.*, 21:1–23, 1999.

[15] J. V. Burke and J. J. Moré. On the identification of active constraints. *SIAM J. Numer. Anal.*, 25:1197–1211, 1988.

[16] J. V. Burke and J. J. Moré. Exposing constraints. *SIAM J. Optim.*, 25:573–595, 1994.

[17] J. V. Burke, J. J. Moré, and G. Toraldo. Convergence properties of trust region methods for linear and convex constraints. *Math. Prog.*, 47:305–336, 1990.

[18] R. H. Byrd, P. Lu, and J. Nocedal. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Statist. Comput.*, 16:1190–1208, 1995.

[19] P. Calamai and J. Moré. Projected gradient for linearly constrained problems. *Math. Prog.*, 39:93–116, 1987.

[20] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems.* Amsterdam, North-Holland, 1978.

[21] T. F. Coleman and Y. Li. On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds. *Math. Prog.*, 67:189–224, 1994.

[22] T. F. Coleman and Y. Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM J. Optim.*, 6:418–445, 1996.

[23] T. F. Coleman and Y. Li. A trust region and affine scaling interior point method for nonconvex minimization with linear inequality constraints. Technical report, Cornell University, Ithaca, NY, 1997.

[24] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM J. Numer. Anal.*, 25:433–460, 1988.

[25] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM J. Numer. Anal.*, 28:545–572, 1991.

[26] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods.* SIAM, Philadelphia, 2000.

[27] Y. H. Dai. A nonmonotone conjugate gradient algorithm for unconstrained optimization. *J. Syst. Sci. Complex.*, 15:139–145, 2002.

[28] Y. H. Dai. On the nonmonotone line search. *J. Optim. Theory Appl.*, 112:315–330, 2002.

[29] Y. H. Dai. Alternate stepsize gradient method. *Optimization*, 52:395–415, 2003.

[30] Y. H. Dai and R. Fletcher. Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. *Numer. Math.*, 100:21–47, 2005.

[31] Y. H. Dai and R. Fletcher. New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds. *Math. Prog.*, to appear 2006.

[32] Y. H. Dai, W. W. Hager, K. Schittkowski, and H. Zhang. The cyclic Barzilai-Borwein method for unconstrained optimization. *IMA J. Numer. Anal.*, to appear.

[33] Y. H. Dai and K. Schittkowski. A sequential quadratic programming algorithm with non-monotone line search. Technical report, Dept. Math., Univ. Bayreuth, submitted, 2005.

[34] Y. H. Dai and X. Q. Yang. A new gradient method with an optimal stepsize property. Technical report, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, 2001.

[35] Y. H. Dai and Y. Yuan. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.*, 10:177–182, 1999.

[36] Y. H. Dai and Y. Yuan. *Nonlinear Conjugate Gradient Methods.* Shang Hai Science and Technology Publisher, Beijing, 2000.

[37] Y. H. Dai and Y. Yuan. An efficient hybrid conjugate gradient method for unconstrained optimization. *Ann. Oper. Res.*, 103:33–47, 2001.

[38] Y. H. Dai and Y. Yuan. Alternate minimization gradient method. *IMA J. Numer. Anal.*, 23:377–393, 2003.

[39] Y. H. Dai and H. Zhang. An adaptive two-point stepsize gradient algorithm. *Numer. Algorithms*, 27:377–385, 2001.

[40] J. W. Daniel. The conjugate gradient method for linear and nonlinear operator equations. *SIAM J. Numer. Anal.*, 4:10–26, 1967.

[41] R. S. Dembo and U. Tulowitzki. On the minimization of quadratic functions subject to box constraints. Technical report, School of Organization and Management, Yale University, New Haven, CT, 1983.

[42] J. E. Dennis, M. Heinkenschloss, and L. N. Vicente. Trust-region interior-point algorithms for a class of nonlinear programming problems. *SIAM J. Control Optim.*, 36:1750–1794, 1998.

[43] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91:201–213, 2002.

[44] Z. Dostál. Box constrained quadratic programming with proportioning and projections. *SIAM J. Optim.*, 7:871–887, 1997.

[45] Z. Dostál. A proportioning based algorithm for bound constrained quadratic programming with the rate of convergence. *Numer. Algorithms*, 34:293–302, 2003.

[46] Z. Dostál, A. Friedlander, and S. A. Santos. Solution of coercive and semicoercive contact problems by FETI domain decomposition. *Contemp. Math.*, 218:82–93, 1998.

[47] Z. Dostál, A. Friedlander, and S. A. Santos. Augmented Lagrangians with adaptive precision control for quadratic programming with simple bounds and equality constraints. *SIAM J. Optim.*, 13:1120–1140, 2003.

[48] A. S. El-Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang. On the formulation and theory of the primal-dual Newton interior-point method for nonlinear programming. *J. Optim. Theory Appl.*, 89:507–541, 1996.

[49] A. S. El-Bakry, R. A. Tapia, and Y. Zhang. On the convergence rate of Newton interior-point methods in the absence of strict complementarity. *Comp. Optim. Appl.*, 6:157–167, 1996.

[50] F. Facchinei, A. Fischer, and C. Kanzow. On the accurate identification of active constraints. *SIAM J. Optim.*, 9:14–32, 1998.

[51] F. Facchinei, J. Júdice, and J. Soares. An active set Newton's algorithm for large-scale nonlinear programs with box constraints. *SIAM J. Optim.*, 8:158–186, 1998.

[52] F. Facchinei and S. Lucidi. A class of penalty functions for optimization problems with bound constraints. *Optimization*, 26:239–259, 1992.

[53] F. Facchinei, S. Lucidi, and L. Palagi. A truncated Newton algorithm for large-scale box constrained optimization. *SIAM J. Optim.*, 4:1100–1125, 2002.

[54] J. Fan and Y. Yuan. On the convergence of the Levenberg-Marquardt method without nonsingularity assumption. *Computing*, 74:23–39, 2005.

[55] R. Fletcher. *Practical Methods of Optimization Vol. 1: Unconstrained Optimization.* John Wiley & Sons, New York, 1987.

[56] R. Fletcher. *Practical Methods of Optimization Vol. 2: Constrained Optimization.* John Wiley & Sons, New York, 1987.

[57] R. Fletcher and C. Reeves. Function minimization by conjugate gradients. *Comput. J.*, 7:149–154, 1964.

[58] A. Friedlander, J. M. Martínez, B. Molina, and M. Raydan. Gradient method with retards and generalizations. *SIAM J. Numer. Anal.*, 36:275–289, 1999.

[59] A. Friedlander, J. M. Martínez, and S. A. Santos. A new trust region algorithm for bound constrained minimization. *Appl. Math. Optim.*, 30:235–266, 1994.

[60] J. C. Gilbert and J. Nocedal. Global convergence properties of conjugate gradient methods for optimization. *SIAM J. Optim.*, 2:21–42, 1992.

[61] R. Glowinski. *Numerical Methods for Nonlinear Variational Problems.* Springer-Verlag, Berlin, New York, 1984.

[62] W. Glunt, T. L. Hayden, and M. Raydan. Molecular conformations from distance matrices. *J. Comput. Chem.*, 14:114–120, 1993.

[63] A .A. Goldstein. Convex programming in Hilbert space. *Bull. Amer. Math. Soc.*, 70:709–710, 1964.

[64] A. A. Goldstein. On steepest descent. *SIAM J. Control*, 3:147–151, 1965.

[65] G. H. Golub and D. P. O'leary. Some history of the conjugate gradient and Lanczos algorithms: 1948–1976. *SIAM Rev.*, 31:50–100, 1989.

[66] L. Grippo, F. Lampariello, and S. Lucidi. A nonmonotone line search technique for Newton's method. *SIAM J. Numer. Anal.*, 23:707–716, 1986.

[67] L. Grippo, F. Lampariello, and S. Lucidi. A truncated Newton method with nonmonotone line search for unconstrained optimization. *J. Optim. Theory Appl.*, 60:401–419, 1989.

[68] L. Grippo and M. Sciandrone. Nonmonotone globalization techniques for the Barzilai-Borwein gradient method. *Comput. Optim. Appl.*, 23:143–169, 2002.

[69] C. D. Ha. A generalization of the proximal point algorithm. *SIAM J. Control*, 28:503–512, 1990.

[70] W. W. Hager. Dual techniques for constrained optimization. *J. Optim. Theory Appl.*, 55:37–71, 1987.

[71] W. W. Hager. Analysis and implementation of a dual algorithm for constrained optimization. *J. Optim. Theory Appl.*, 79:427–462, 1993.

[72] W. W. Hager and H. Zhang. CG_DESCENT user's guide. Technical report, Dept. Math., Univ. Florida, 2004.

[73] W. W. Hager and H. Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM J. Optim.*, 16:170–192, 2005.

[74] W. W. Hager and H. Zhang. Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent. *ACM Trans. Math. Software*, 32, 2006.

[75] W. W. Hager and H. Zhang. A survey of nonlinear conjugate gradient methods. *Pacific J. Optim.*, 2:35–58, 2006.

[76] W. W. Hager and H. Zhang. Recent advances in bound constrained optimization. In F. Ceragioli, A. Dontchev, H. Furuta, K. Marti, and L. Pandolfi, editors, *System Modeling and Optimization, Proceedings of the 22nd IFIP TC7 Conference held in July 18–22, 2005, Turin, Italy*. Springer, 2006, to appear.

[77] W. W. Hager and H. Zhang. Self-adaptive inexact proximal point methods. *Comput. Optim. Appl.*, submitted, 2005.

[78] W. W. Hager and H. Zhang. A new active set algorithm for box constrained optimization. *SIAM J. Optim.*, to appear.

[79] J. Y. Han, G. H. Liu, and H. X. Yin. Convergence of Perry and Shanno's memoryless quasi-Newton method for nonconvex optimization problems. *OR Trans.*, 1:22–28, 1997.

[80] M. Heinkenschloss, M. Ulbrich, and S. Ulbrich. Superlinear and quadratic convergence of affine-scaling interior-point Newton methods for problems with simple bounds without strict complementarity assumption. *Math. Prog.*, 86:615–635, 1999.

[81] M. R. Hestenes and E. L. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Research Nat. Bur. Standards*, 49:409–436, 1952.

[82] C. Humes and P. Silva. Inexact proximal point algorithms and descent methods in optimization. *Optim. Eng.*, 6:257–271, 2005.

[83] C. Kanzow and A. Klug. On affine-scaling interior-point Newton methods for nonlinear minimization with bound constraints. *Comput. Optim. Appl.*, to appear 2006.

[84] A. Kaplan and R. Tichatschke. Proximal point methods and nonconvex optimization. *J. Global Optim.*, 13:389–406, 1998.

[85] C. Lemaréchal. A view of line-searches. In *Optimization and Optimal Control*, volume 30, pages 59–79, Heidelberg, 1981. Springer-Verlag.

[86] M. Lescrenier. Convergence of trust region algorithms for optimization with bounds when strict complementarity does not hold. *SIAM J. Numer. Anal.*, 28:476–495, 1991.

[87] E. S. Levitin and B. T. Polyak. Constrained minimization problems. *USSR Comput. Math. Math. Physics*, 6:1–50, 1966.

[88] D. Li, M. Fukushima, L. Qi, and N. Yamashita. Regularized Newton methods for convex minimization problems with singular solutions. *Comput. Optim. Appl.*, 28:131–147, 2004.

[89] C. J. Lin and J. J. Moré. Incomplete cholesky factorizations with limited memory. *SIAM J. Sci. Comput.*, 21:24–45, 1999.

[90] C. J. Lin and J. J. Moré. Newton's method for large bound-constrained optimization problems. *SIAM J. Optim.*, 9:1100–1127, 1999.

[91] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Program.*, 45:503–528, 1989.

[92] W. B. Liu and Y. H. Dai. Minimization algorithms based on supervisor and searcher cooperation. *J. Optim. Theory Appl.*, 111:359–379, 2001.

[93] Y. Liu and C. Storey. Efficient generalized conjugate gradient algorithms, part 1: Theory. *J. Optim. Theory Appl.*, 69:129–137, 1991.

[94] S. Lucidi, F. Rochetich, and M. Roma. Curvilinear stabilization techniques for truncated Newton methods in large-scale unconstrained optimization. *SIAM J. Optim.*, 8:916–939, 1998.

[95] F. J. Luque. Asymptotic convergence analysis of the proximal point algorithm. *SIAM J. Control*, 22:277–293, 1984.

[96] B. Martinet. Régularisation d'inéquations variationnelles par approximations successives. *Rev. Francaise Inform. Rech. Oper. Ser. R-3*, 4:154–158, 1970.

[97] B. Martinet. Determination approachée d'un point fixe d'une application pseudo-contractante. *Comptes Rendus des Séances de l'Académie des Sciences*, 274:163–165, 1972.

[98] J. M. Martínez. BOX-QUACAN and the implementation of augmented Lagrangian algorithms for minimization with inequality constraints. *J. Comput. Appl. Math.*, 19:31–56, 2000.

[99] G. P. McCormick and R. A. Tapia. The gradient projection method under mild differentiability conditions. *SIAM J. Control*, 10:93–98, 1972.

[100] J. J. Moré and D. C. Sorensen. Newton's method. In G. H. Golub, editor, *Studies in Numerical Analysis*, pages 29–82, Washington, D.C., 1984. Mathematical Association of America.

[101] J. J. Moré and D. J. Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Trans. Math. Software*, 20:286–307, 1994.

[102] J. J. Moré and G. Toraldo. On the solution of large quadratic programming problems with bound constraints. *SIAM J. Optim.*, 1:93–113, 1991.

[103] J. Nocedal. Updating quasi-Newton matrices with limited storage. *Math. Comp.*, 35:773–782, 1980.

[104] E. R. Panier and A. L. Tits. Avoiding the maratos effect by means of a nonmonotone line search. *SIAM J. Numer. Anal.*, 28:1183–1195, 1991.

[105] J. M. Perry. A class of conjugate gradient algorithms with a two step variable metric memory. Technical Report 269, Center for Mathematical Studies in Economics and Management Science, Northwestern University, 1977.

[106] E. Polak and G. Ribière. Note sur la convergence de méthodes de directions conjuguées. *Rev. Française Informat. Recherche Opérationnelle*, 3:35–43, 1969.

[107] B. T. Polyak. The conjugate gradient method in extremal problems. *USSR Comp. Math. Math. Phys.*, 9:94–112, 1969.

[108] M. Raydan. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM J. Optim.*, 7:26–33, 1997.

[109] M. Raydan and B. F. Svaiter. Relaxed steepest descent and Cauchy-Barzilai-Borwein method. *Comput. Optim. Appl.*, 21:155–167, 2002.

[110] S. M. Robinson. Strongly regular generalized equations. *Math. Oper. Res.*, 5:43–62, 1980.

[111] R. T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Math. Oper. Res.*, 2:97–116, 1976.

[112] R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control*, 14:877–898, 1976.

[113] A. Schwartz and E. Polak. Family of projected descent methods for optimization problems with simple bounds. *J. Optim. Theory Appl.*, 92:1–31, 1997.

[114] T. Serafini, G. Zanghirati, and L. Zanni. Gradient projection methods for quadratic programs and applications in training support vector machines. *Optim. Methods Softw.*, 20:353–378, 2005.

[115] D. F. Shanno. On the convergence of a new conjugate gradient algorithm. *SIAM J. Numer. Anal.*, 15:1247–1257, 1978.

[116] P. L. Toint. Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space. *IMA J. Numer. Anal.*, 8:231–252, 1988.

[117] P. L. Toint. An assessment of non-monotone line search techniques for unconstrained optimization. *SIAM J. Sci. Comput.*, 17:725–739, 1996.

[118] P. L. Toint. A non-monotone trust region algorithm for nonlinear optimization subject to convex constraints. *Math. Prog.*, 77:69–94, 1997.

[119] P. Tseng. Error bounds and superlinear convergence analysis of some Newton-type methods in optimization. In G. Di Pillo and F. Giannessi, editors, *Nonlinear Optimization and Related Topics*, pages 445–462. Kluwer, 2000.

[120] M. Ulbrich, S. Ulbrich, and M. Heinkenschloss. Global convergence of affine-scaling interior-point Newton methods for infinite-dimensional nonlinear problems with pointwise bounds. *SIAM J. Control Optim.*, 37:731–764, 1999.

[121] C. Wang, J. Han, and L. Wang. Global convergence of the Polak-Ribière and Hestenes-Stiefel conjugate gradient methods for the unconstrained nonlinear optimization. *OR Transactions*, 4:1–7, 2000.

[122] P. Wolfe. Convergence conditions for ascent methods. *SIAM Rev.*, 11:226–235, 1969.

[123] P. Wolfe. Convergence conditions for ascent methods II: some corrections. *SIAM Rev.*, 13:185–188, 1971.

[124] S. J. Wright. Implementing proximal point methods for linear programming. *J. Optim. Theory Appl.*, 65:531–554, 1990.

[125] H. Yamashita and H. Yabe. Superlinear and quadratic convergence of some primal-dual interior-point methods for constrained optimization. *Math. Prog.*, 75:377–397, 1996.

[126] N. Yamashita and M. Fukushima. The proximal point algorithm with genuine superlinear convergence for the monotone complementarity problem. *SIAM J. Optim.*, 11:364–379, 2000.

[127] N. Yamashita and M. Fukushima. On the rate of convergence of the Levenberg-Marquardt method. In *Topics in numerical analysis*, volume 15 of *Comput. Suppl.*, pages 239–249. Springer, 2001.

[128] E. K. Yang and J. W. Tolle. A class of methods for solving large convex quadratic programs subject to box constraints. *Math. Prog.*, 51:223–228, 1991.

[129] E. H. Zarantonello. Projections on convex sets in Hilbert space and spectral theory. In E. H. Zarantonello, editor, *Contributions to Nonlinear Functional Analysis*, pages 237–424, New York, 1971. Academic Press.

[130] H. Zhang. A nonmonotone trust region algorithm for nonlinear optimization subject to general constrains. *Jornal of Computational Mathematics*, 2:237–276, 2003.

[131] H. Zhang and W. W. Hager. A nonmonotone line search technique and its application to unconstrained optimization. *SIAM J. Optim.*, 14:1043–1056, 2004.

[132] H. Zhang and W. W. Hager. PACBB: A projected adaptive cyclic Barzilai-Borwein method for box constrained optimization. In William W. Hager, Shu-Jen Huang, Panos M. Pardalos, and Oleg A. Prokopyev, editors, *Multiscale Optimization Methods and Applications*, pages 387–392, New York, 2005. Springer.

[133] Y. Zhang. Interior-point gradient methods with diagonal-scalings for simple-bound constrained optimization. Technical Report TR04-06, Department of Computational and Applied Mathematics, Rice University, Houston, Texas, 2004.

[134] J. L. Zhou and A. L. Tits. Nonmonotone line search for minimax problem. *J. Optim. Theory Appl.*, 76:455–476, 1993.

[135] C. Zhu, R. H. Byrd, and J. Nocedal. Algorithm 778: L-BFGS-B, Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Software*, 23:550–560, 1997.

BIOGRAPHICAL SKETCH

Hongchao Zhang was born in the Shandong Province, People's Republic of China, in 1976. He received his Bachelor of Science degree in computing mathematics in 1998 from Shangdong University, P.R. China. He obtained a Master of Science degree from the Institute of Computational Mathematics and Scientific/Engineering Computing of Chinese Academy of Sciences in 2001. In fall 2001, Hongchao started his graduate study in mathematics at the University of Florida, from which he received his Ph.D. in mathematics in 2006.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

William W. Hager, Chair

Professor of Mathematics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Panos M. Pardalos

Professor of Industrial and Systems Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Shari Moskow

Associate Professor of Mathematics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Sergei S. Pilyugin

Associate Professor of Mathematics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Jay Gopalakrishnan

Assistant Professor of Mathematics

This dissertation was submitted to the Graduate Faculty of the Department of Mathematics in the College of Liberal Arts and Sciences and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August 2006 ______________________________

Dean, College of Libral Arts and Sciences

GRADIENT METHODS ON LARGE SCALE NONLINEAR OPTIMIZATION

Hongchao Zhang
(352) 392–0281 ext. 329
Department of Mathematics
Chair: William W. Hager
Degree: Doctor of Philosophy
Graduation Date: August 2006

Optimization might be defined as the science of determining the “best strategies” by solving certain type of mathematical problems. Now the applicability of optimization methods is widely spread into almost every activity in which numerical optimal solution need to be found and in which the huge number of variables need to be determined. This dissertation generally considers, from mathematical kind of view, how to solve those large-scale problems efficiently. All the methods discussed in this dissertation uses least memory and proved by extensive numerical experiments to be very efficient for optimization problems without constraints or only bound constraints. These recently developed methods should be interesting not only to those mathematicians who work on numerical analysis, computational optimization, but also to many scientists and engineers who are dealing with real optimization problems coming from industry, economics, commerce, etc.