STUDIES ON ALTERNATING DIRECTION METHOD OF MULTIPLIERS WITH ADAPTIVE PROXIMAL TERMS FOR CONVEX OPTIMIZATION PROBLEMS

YAN GU

STUDIES ON ALTERNATING DIRECTION METHOD OF MULTIPLIERS WITH ADAPTIVE PROXIMAL TERMS FOR CONVEX OPTIMIZATION PROBLEMS

by

YAN GU

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

KYOTO UNIVERSITY
KYOTO 606–8501, JAPAN
NOVEMBER 2020

Preface

In this thesis, we study two proximal type splitting methods for the structured convex optimization problems: the alternating direction methods of multipliers (ADMM) and the Peaceman-Rachford splitting methods (PRSM). These two first-order algorithms are well studied for the structured convex optimization problems since applying the separable properties of the objective functions. Besides, these algorithms are widely used in real practical problems, particularly the large-scale problems arising in statistics, machine learning, and related areas.

Due to the large scales, the subproblems in the classical ADMM and PRSM may be difficult to be solved exactly in many applications. Thus proximal terms with a positive semidefinite matrix had been added to the subproblems to make them easier. In the proximal ADMM, it always solves subproblems with an approximate solution. Although such classical and proximal version algorithms are efficient methods to solve the separated convex program and have been widely studied, there still exist a lot of issues which should be considered. For example, the two main problems as following: one is that the classical splitting method can get a global solution within fewer iterative steps, but sometimes it needs a quite long time to solve the subproblems exactly or cannot; another one is that the proximal versions can reach approximate solutions with a faster time, but the solutions may be infected by the proximal matrix and it uses more iterations. Recently, some modifications also have been studied, such as replacing the semidefinite proximal term with an indefinite term for the proximal ADMM, allowing two different stepsizes for the PRSM. From the mentioned issues, we will directly have the question that: Do the proximal methods can reach nearly the same optimal solutions as the classical ADMM, or more efficient on the iteration steps? Hence, proposing efficient and practical solution methods to solve them is a worth studying topic.

The main contribution of this thesis is to propose efficient proximal splitting methods for solving large scale structured nonlinear programming problems.

We propose two classes of the splitting methods. One is the proximal ADMM, where we propose to generate a variable positive semidefinite matrices sequence for an unconstraint structured convex quadratic problem. We first construct these proximal matrices via the

BFGS update at every iteration step which can satisfy the convergence conditions. Then we extend this algorithm for two general convex optimization problems and also extend the constructions of the proximal term by the Broyden family update. At last, we further propose a variable metric indefinite proximal ADMM by replacing the semidefinite terms to allow for a larger stepsize. We even show the sufficient conditions on the indefinite proximal matrices for the global convergence. The other method is proximal PRSM with an indefinite proximal term and two different constants step sizes in the dual updates. We establish its global convergence and also the $o(1/t)$ convergence rate in the nonergodic sense. Moreover, for all of the methods, some numerical experiments have also been carried out, which demonstrate the excellent performances of the proposed methods.

The author hopes that the results in this thesis will contribute to further studies on the ADMM and PRSM for the structured convex optimization problems and their related problems.

Yan Gu

November 2020

Acknowledgments

Upon the end of my PhD period, I would like to express my gratitude to many people who have been with me throughout my struggles and sufferings for these years. First and foremost, my deepest thanks and appreciation go to my supervisor, Professor Nobuo Yamashita of Kyoto University, for his providing me the opportunity to be a member of his excellent optimization laboratory, guiding my research, enhancing my manuscripts, and inspiring me thereafter. His invaluable encouragements and advice have been a dramatic help for me, let alone the sufficient freedom and trust he granted me to do my research. I am enlightened by each and every discussion with him, admiring his acute and intuitive judgement on the nature of the problem. I have learned from him not only knowledge but also the rigorous attitude of researchers which will undoubtedly be of a great benefit to me all my life. Meanwhile, I would also like to thank Professor Yamashita for precious opportunities he kindly offered to participate in academic conferences, which opened up my horizons to a great extent.

Besides, I owe a lot to many professors in Kyoto University during my doctoral years. Associate Professor Ellen Hidemi Fukuda encouraged me greatly whenever I was depressed and frustrated in my study. She always invited me to a meal and provided much care and assistance especially when I was new here. Associate Professor Hiroyuki Sato offered me insightful suggestions and comments when I faced difficulties. I am also thankful to my advisor Professor Ohta for his help. He provided encouragement to me and gave me useful advice. I would like to tender my acknowledgement to Professor Hisashi Kashima of Kyoto University for his supporting of my research. Professor Hisashi Kashima provided me a research position in his laboratory and led me into the world of machine learning where I can apply my knowledge to real application. I feel really grateful to him for his help and kindness.

I have to extent my heartfelt thanks to Professor Jonathan Eckstein of Rutgers University, who accepted me as a visiting student in his laboratory. He gave me a lot of advice on research approaches. Plus, I am also greatly indebted to Professor Deren Han of Beihang University, Associate Professor Xingju Cai and Bo Jiang of Nanjing Normal University for their full support for my research.

At the same time, I will never forget all the members of Yamashita Laboratory and

Kashima Laboratory, Associate Professor Makoto Yamada, Ms. Xiaoqin Hua, Ms. Siti Nor Habibah Binti Hassan, Mr. Hardik Tankaria, Mr. Rafael Pedro Longhi and others for their direct and indirect help to me. Especially, I thank Ms. Yuiko Igura, Ms. Fumie Yagura and Ms. Katsumi Nishi, who have rendered me help on my work and life which made my stay here easier and happier.

What's more, I must give my sincere gratitude to Ms. Wakeda and Ms. Tagawa. They assisted me with my Japanese and life in Japan and guided me in experiencing the unique and exotic culture of Kyoto. I also would like to express my thanks to Ichikawa International Scholarship Foundation for the generous financial support, so that I can have more time to concentrate on my academic research.

Not only that, my dissertation would be far from completion without my parents, my brother and my sister-in-law for their considerations and great confidence in me. It is their everlasting love that put me forward these years. Apart from my family, my friends, Ms. Ziting Xu, Ms. Yifan Zhang, Ms. Xiaoying Sang, Mr. Fei Xia and many others are a solid backup for me as well. Their continued support and encouragement comforted me in my hard times. In addition, my favorite singers Mayday and JJ Lin encouraged me by their songs to shed off cowardly ideas of giving up and refresh myself to go on.

Lastly and above all, I see this thesis not an end but a beginning of the new period of life to me. I hope that I can never stop in scientific research and be a better person to help more people.

Contents

List of Figures

List of Tables

List of Notations

$\lvert x \rvert$	the absolute value of number x
x_i	the i-th coordinate of vector x
$\|x\|_1$	the 1-norm of vector x
$\|x\|_2$	the 2-norm of vector x
$\|x\|_G$	the G-norm of vector x
$\langle \cdot, \cdot \rangle$	the Euclidean inner product
$\mathrm{dom} f$	the effective domain of function f
∇f	the gradient of f
$\nabla^2 f$	the Hessian matrix of f
$\nabla_i f$	the i-th coordinate of gradient ∇f
∂f	the subdifferential of function f
I	the identity matrix
I_n	the $n \times n$ identity matrix
A^T	the transpose of matrix A
A_j	the j-th column matrix A
A_{ij}	the element at the i-th row and j-th column of matrix A
$A \succ 0$	matrix A is positive definite
$A \succeq 0$	matrix A is positive semidefinite
$A \succ B$	matrix $A - B$ is positive definite
$A \succeq B$	matrix $A - B$ is positive semidefinite
$\lambda_i(A)$	the eigenvalue of a real symmetric matrix A
$\lambda_{\min}(A)$	the minimum eigenvalue of a real symmetric matrix A
$\lambda_{\max}(A)$	the maximum eigenvalue of a real symmetric matrix A

Chapter 1

Introduction

The *mathematical optimization* is a branch of applied mathematics which was introduced by professor Robert Dorfman in 1940s. Optimization is to maximize or minimize a specific function or variable under a defined domain. *Convex optimization* is a subfield of the optimization that studies the problem of minimizing convex functions over convex sets and is very important because of the rapid development of the intersection between various disciplines. In some applications such as machine learning, signal processing and statistics problems, a general convex model fitting problem very often can be written in the form of minimizing the sum of convex functions called structured convex optimization problem.

In this chapter, we give an overview of the structured convex optimization problem and then outline the contents of the thesis.

1.1 Overview of problems

1.1.1 Structured convex optimization

It has long been recognized that many convex optimization problems can be put into the following form:

$$\underset{x\in C}{\text{minimize}} \;\; f(x)+g(x), \tag{1.1.1}$$

where C is a finite-dimensional Euclidean space, $f\colon C\to\mathbb{R}$ and $g\colon C\to\mathbb{R}$ are closed convex functions.

An equivalent formulation of Problem (1.1.1) is as follows:

$$\begin{aligned} \text{minimize} \quad & f(x)+g(y) \\ \text{subject to} \quad & x-y=0, \\ & x\in C, y\in C. \end{aligned} \tag{1.1.2}$$

In this thesis, we consider a more general *structured convex optimization problems* related to Problem (1.1.2):

$$\begin{aligned} \text{minimize} \quad & f(x) + g(y) \\ \text{subject to} \quad & Ax + By = b, \\ & x \in \mathcal{X}, y \in \mathcal{Y}, \end{aligned} \tag{1.1.3}$$

where $f\colon \mathbb{R}^{n_1} \to \mathbb{R}$ and $g\colon \mathbb{R}^{n_2} \to \mathbb{R}$ are closed convex (not necessarily smooth) functions; $A \in \mathbb{R}^{m\times n_1}$ and $B \in \mathbb{R}^{m\times n_2}$ are given matrices; $b \in \mathbb{R}^m$ is a given vector; $\mathcal{X}$ and $\mathcal{Y}$ are nonempty closed convex subsets of $\mathbb{R}^{n_1}$ and $\mathbb{R}^{n_2}$ (e.g., positive orthant, spheroidal or box areas), respectively.

The above problems consist of two functions. Next, we describe another optimization problem which is *multi-block structured convex optimization problem.* The objective function is the sum of functions without coupled variables.

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{N} f_i(x_i) \\ \text{subject to} \quad & \sum_{i=1}^{N} A_i x_i = b, \\ & x_i \in \mathcal{X}_i, i = 1, \cdots, N, \end{aligned} \tag{1.1.4}$$

where $f_i\colon \mathbb{R}^{n_i} \to \mathbb{R}(i = 1, \cdots, N)$ are closed convex functions; $A_i \in \mathbb{R}^{m\times n_i}$, $b \in \mathbb{R}^m$ and $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}(i = 1, \cdots, N)$ are closed convex sets. It is obvious that the problem (1.1.1) or (1.1.3) is a special case when the $N = 2$ in the multi-block problem (1.1.4).

Various practical problems of science and engineering, such as machine learning [73, 115], total variation denoising [102] and statistics [59] can be formulated as Problem (1.1.1). In the following, we give some of such applications.

1.1.2 Applications

The problems (1.1.1) and (1.1.3) have been widely used in many applications. Generally, f is a loss function and g is a structured regularization term. Regularization is a technique often used in practice, either because in some cases the observation matrix can be ill-conditioned, or to impose additional information on the model like sparsity to improve the conditioning of the problem. The most common regularizers used are the squared l_2 norm (Tikhonov regularization), and the l_1 norm. We list some common variants of function $g(x)$ as follows.

(a) l_1-regularization [73, 105, 106, 112] is a good technique for obtaining relatively sparse solution that many elements of the variable x are 0, i.e.,

$$g(x) = \|x\|_1;$$

(b) l_2-regularization [91, 107] is introduced to prevent overfitting and to make the values relatively dense and uniformly concentrated near zero, i.e.,

$$g(x) = \|x\|_2;$$

(c) Block l_2-regularization [84, 116] is an extension of the l_1-regularization, i.e.,

$$g(x) = \sum_{i=1}^{N} \|x_{J^i}\|_2,$$

where $\{x_{J^i}\}$, $i = 1, 2, \ldots, N$, denote the disjoint subvectors of vector x. Its sparsity is obtained at the group level, that is, a group is picked or dropped. But within each group, sparsity can not be guaranteed;

(d) The mixed norm penalty [71, 74] yields solutions that are sparse at both group and individual elements, i.e.,

$$g(x) = \|x\|_1 + \sum_{i=1}^{N} \|x_{J^i}\|_2;$$

(e) Indicator function with respect to closed separable convex set C, i.e.,

$$g(x) = \begin{cases} 0 & \text{if } x \in C, \\ \infty & \text{otherwise.} \end{cases} \tag{1.1.5}$$

Apparently, the above regularization terms have different forms. However, from the optimization point of view, they are special forms of models (1.1.1) and (1.1.3). Next, some examples of functions f and g are described as follows.

(1) Quadratic Programming: The standard form of quadratic program (QP) is

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} x^\top Q x + p^\top x \\ \text{subject to} \quad & Ax = b, x \geq 0, \end{aligned}$$

where $x \in \mathbb{R}^n$ and Q is a positive definite matrix. We can rewrite it as the (1.1.3) type that

$$\begin{aligned} \text{minimize} \quad & f(x) + g(y) \\ \text{subject to} \quad & x - y = 0, \end{aligned}$$

where the function $f(x) = \frac{1}{2} x^\top Q x + p^\top x$ with $\mathrm{dom} f = \{x \mid Ax = b\}$ and $g(y)$ is the indicator function of the nonnegative orthant. Note that when the matrix $Q = 0$, this problem reduces to a standard form linear program (LP). More generally, the constraint $x \geq 0$ can be replaced by any conic constraint that $x \in \mathcal{K}$ when the problem becomes a quadratic conic program.

(2) l_1**-norm Problems**: Here we show some problems that involve l_1 norm, which are important in statistics, signal processing and machine learning.

The **Basis Pursuit** problem [24], which plays a significant role in many applications like the compressed sensing [15, 31, 115], is the equality constrained l_1 minimization problem

$$\begin{aligned} \text{minimize} \quad & \|x\|_1 \\ \text{subject to} \quad & Ax = b, \end{aligned}$$

with $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, and $b \in \mathbb{R}^n$. It is a basic technique to determine a sparse solution of a underdetermined linear system that $m < n$ (in some cases that $m \ll n$). Then we make the function $f(x)$ be an indicator function of $\{x \in \mathbb{R}^n \mid Ax = b\}$, the problem can be reformulated by

$$\begin{aligned} \text{minimize} \quad & f(x) + \|y\|_1 \\ \text{subject to} \quad & x - y = 0. \end{aligned}$$

Lasso [106, 59] is an important l_1 regularized linear regression case.

$$\text{minimize } \frac{1}{2}\|Ax - b\|_2^2 + \mu\|x\|_1,$$

where μ is a scalar regularization parameter that is normally chosen by cross-validation. The Lasso can be written as the structured form as

$$\begin{aligned} \text{minimize} \quad & f(x) + g(y) \\ \text{subject to} \quad & x - y = 0, \end{aligned}$$

with $f(x) = \frac{1}{2}\|Ax - b\|_2^2$ and $g(y) = \mu\|y\|_1$.

An another similar extension example which is called the **Group Lasso** [116], consider replacing the regularization term $\|x\|_1$ with $\Sigma_{i=1}^N \|x_i\|_2$, where $x = (x_1, \ldots, x_N)$, with $x_i \in \mathbb{R}^{n_i}$. Group Lasso arises in applications (i.e. bioinformatics), where correlated features can be put into groups.

(3) Model Fitting: Some large scale problems arising in model fitting like regression, classification, and signal processing also can be written in the Problem (1.1.1) form. A general convex model fitting problem can be written as

$$\text{minimize } l(Ax - b) + r(x),$$

where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ is the feature matrix, $b \in \mathbb{R}^m$ is the output vector, $l\colon \mathbb{R}^m \to \mathbb{R}$ is a convex loss function, and r is a convex regularization function. The common examples for r are $r(x) = \mu\|x\|_2^2$ and $r(x) = \mu\|x\|_1$ with a positive regularization parameter μ. In some cases, one or more parameters are not regularized such as the offset parameter in a classification model. Next we give some examples that have the general form above.

Regression (see, e.g.[14, S7.1.1]) is a special case of the model fitting. First we assume that l is additive, i.e.

$$l(Ax - b) = \sum_{i=1}^{m} l_i(a_i^\top x - b_i),$$

where $l_i \colon \mathbb{R} \to \mathbb{R}$ is the loss function for the ith example, $a_i \in \mathbb{R}^n$ is the feature vector and b_i is the response for example i. Consider a linear model fitting problem with

$$b_i = a_i^\top x + v_i,$$

where a_i is the ith feature vector and v_i are independent measurement noises with log-concave densities p_i. Then the negative log-likelihood function is $l(Ax - b)$ with $l_i(w) = -\log p_i(-w)$.

Classification [4, 73] problems can also be written in the general form above. Let $p_i \in \mathbb{R}^{n-1}$ be the feature vector of the ith example and $q_i \in \{-1, 1\}$ be the binary outcome or class label for $i = 1, \ldots, m$. Function $l(Ax - b)$ is given by

$$l(Ax - b) = \frac{1}{m} \sum_{i=1}^{m} l_i\big(q_i(p_i^\top w + v)\big),$$

where $x = (w, v) \in \mathbb{R}^n$ with a weight vector $w \in \mathbb{R}^{n-1}$ and offset $v \in \mathbb{R}$, $a_i = (q_i p_i, -q_i)$, and $b_i = 0$ for $i = 1, \ldots, m$. (Here we need to scale l_i by $1/m$.) Some common loss functions are *hinge loss* $\big(1 - q_i(p_i^\top w + v)\big)_+$, *exponential loss* $\exp\big(-q_i(p_i^\top w + v)\big)$, and *logistic loss* $\log\Big(1 + \exp\big(-q_i(p_i^\top w + v)\big)\Big)$.

1.2 Solution methods

In many applied fields, particularly the data analysis, the problems are often large datasets in high dimensions and contain huge number of training examples which have been referred to as "Big Data". It is challenging to directly solve the optimization problem like Problem (1.1.1). Thus it is natural to look to parallel optimization algorithms as a mechanism for solving large-scale statistical tasks.

In recent years, a number of efficient first-order algorithms have been developed for problems (1.1.1) and (1.1.3) including operator splitting methods [1, 5, 26, 32, 35, 80], gradient methods [88, 97, 108, 109], primal dual methods [18, 23, 39], Bregman iterative methods [16, 51], etc (see, e.g.[6, 111]). The dual version of the operator splitting method, which is simple but powerful, known as *Alternating Direction Method of Multipliers (ADMM)* [46, 48, 49] is well suited for such convex optimization problems.

1.2.1 Proximal Gradient Method

For simplicity, consider the problem (1.1.1) with $C = \mathbb{R}^n$. The proximal gradient method can solve the problem with assumption that g is differentiable.

The proximal operator $\mathrm{prox}_f \colon \mathbb{R}^n \to \mathbb{R}^n$ of f is defined by

$$\mathrm{prox}_f(x) = \arg\min_u \left\{ f(u) + \frac{1}{2}\|u - x\|_2^2 \right\}.$$

The proximal operator of f with parameter $\lambda > 0$ can be expressed by

$$\mathrm{prox}_{\lambda f}(x) = \arg\min_u \left\{ f(u) + \frac{1}{2\lambda}\|u - x\|_2^2 \right\},$$

which means that for a given point x, finding the optimal point $u = \mathrm{prox}_{\lambda f}(x)$ to minimize $f(u) + \frac{1}{2\lambda}\|u - x\|_2^2$.

The proximal gradient method for solving problem (1.1.1) is

$$x^{k+1} = \mathrm{prox}_{\lambda f}\left(x^k - \lambda \nabla g(x^k)\right), \tag{1.2.1}$$

where $\lambda > 0$ can be viewed as step size.

There are two general ways to decide the step size λ. One is to take a fixed step size $\lambda \in (0, \frac{1}{L}]$ when the gradient $\nabla g(x)$ is Lipschitz continuous with a constant L. This method converges with rate $O(1/k)$[1] [6, 27]. Another one allows variable step size λ^k by a line search when L is unknown [7].

However, it is always difficult to compute the proximal operation for structured sparsity functions to capture complex structures of data. Examples of that include overlapped group lasso, low rank tensor estimation, and graph lasso.

1.2.2 Augmented Lagrangian Method

One may solve problem (1.1.3) is the augmented Lagrangian method (ALM), which was originally known as the *method of multipliers*. These methods were first discussed by Hestenes [69] and Powell [99] in the late 1960s, and were also studied by Bertsekas [8].

[1] As the work [87, 88] and many others, a worst-case $O(1/k)$ convergence rate means the accuracy to a solution under certain criteria is of the order $O(1/k)$ after k iterations of an iterative scheme; or equivalently, it requires at most $O(1/\epsilon)$ iterations to achieve an approximate solution with an accuracy of ϵ.

The Lagrangian function for problem (1.1.3) is

$$L(x, y, \lambda) = f(x) + g(y) - \langle \lambda, Ax + By - b \rangle, \tag{1.2.2}$$

where $\lambda \in \mathbb{R}^m$ is the Lagrangian multiplier for the linear constraints $Ax + By = b$ in (1.1.3).

The augmented Lagrangian function of (1.1.3) is

$$L_\beta(x, y, \lambda) = f(x) + g(y) - \langle \lambda, Ax + By - b \rangle + \frac{\beta}{2}\|Ax + By - b\|^2, \tag{1.2.3}$$

where $\beta > 0$ is called the penalty parameter. It also can be viewed as the Lagrangian function for the following problem:

$$\begin{aligned} \text{minimize} \quad & f(x) + g(y) + \frac{\beta}{2}\|Ax + By - b\|^2 \\ \text{subject to} \quad & Ax + By = b, \\ & x \in \mathcal{X}, y \in \mathcal{Y}. \end{aligned} \tag{1.2.4}$$

Applying the dual ascent method to the problem (1.2.4) above, it generates the updates

$$\begin{cases} (x^{k+1}, y^{k+1}) = \arg\min\limits_{x,y} \ \mathcal{L}_\beta(x, y, \lambda^k), \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} - b). \end{cases} \tag{1.2.5}$$

This method of multipliers converges under very general conditions including the case that f and g are not strictly convex. However, in this case, the vectors x^{k+1} and y^{k+1} should be updated at the same time ignoring the separability of the original functions. Generally, the joint minimization problem (1.1.3) is a challenge to be solved exactly or approximately with a high accuracy. To exploit the separable property of (1.1.3), the classical ADMM is developed to efficiently solve the x- and y-subproblems.

1.2.3 Alternating Direction Method of Multipliers and extensions

Our main problem is the structured convex optimization problem (1.1.3). Because of the separability of the objective functions, we can effectively apply some properties of f and g in algorithm design, respectively. We first introduce the ADMM method. ADMM was first proposed by by Gabay and Mercier [46], Glowinski and Marrocco [49] in the mid-1970s. It derived from the augmented Lagrangian method and also can be viewed as an application of the Douglas-Rachford algorithm to the dual of (1.1.1).

The classical ADMM for solving (1.1.3) takes the iterative sequence via the following recursions:

$$\begin{cases} x^{k+1} = \arg\min\limits_{x \in \mathcal{X}} \mathcal{L}_\beta(x, y^k, \lambda^k), & \text{(1.2.6a)} \\ y^{k+1} = \arg\min\limits_{y \in \mathcal{Y}} \mathcal{L}_\beta(x^{k+1}, y, \lambda^k), & \text{(1.2.6b)} \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} - b). & \text{(1.2.6c)} \end{cases}$$

ADMM is similar with dual ascent method and the method of multipliers. It consists of an x-minimization step, a y-minimization step and a dual variable update step. The step-size of dual variable λ update is equal to augmented Lagrangian parameter β. In ADMM, the solutions of subproblem (1.2.6a) and (1.2.6b) are used to find the global solution of the big problem. At the k iteration, for fixed y^k and multiplier λ^k, the new point x^{k+1} is obtained from the exact minimizer of the augmented Lagrangian with respect to x. The y^{k+1} is updated with the new point x^{k+1} in a similar way. That is, the variables x and y are updated in a *Gauss-Seidel* pass [52] that x is updated while y is fixed then the new value of x is used to find a new y.

The convergence of ADMM has established in some literatures [13, 35, 44, 45]. One approach [44] was to split the Lagrangian function of problem (1.1.3) into a sum of two convex-concave functions by using the monotonicity of the subgradient of the Lagrangian. Another approach [45] was based on Douglas-Rachford operator splitting theory, and yields considerable insight into the convergence of the ADMM. An $O(1/k)$ convergence rate had been shown for two parts convex problems [65, 67, 85] with the k on behalf of the number of iterations. A rate of $O(1/k^2)$ was given for the accelerated ADMM when the problems are strongly convex [50]. Under the assumptions of strongly convex and Lipschitz continuous gradient on one of the two functions, it turns out to have a global linear convergence rate [30].

By noting the fact that the x-, y-subproblems in (1.2.6a)-(1.2.6c) may be difficult to solve exactly in many applications, Eckstein [34, 36] have considered the proximal ADMM by adding proximal terms to the subproblems which takes the following scheme:

$$
\begin{cases}
x^{k+1} = \arg\min_{x\in\mathcal{X}} \mathcal{L}_\beta(x, y^k, \lambda^k) + \dfrac{1}{2}\|x - x^k\|_S^2, & \text{(1.2.7a)}\\
y^{k+1} = \arg\min_{y\in\mathcal{Y}} \mathcal{L}_\beta(x^{k+1}, y, \lambda^k) + \dfrac{1}{2}\|y - y^k\|_T^2, & \text{(1.2.7b)}\\
\lambda^{k+1} = \lambda^k - \gamma\beta(Ax^{k+1} + By^{k+1} - b), & \text{(1.2.7c)}
\end{cases}
$$

where $\beta > 0$, and $\|z\|_G = \sqrt{z^\top Gz}$ for $z \in \mathbb{R}^n$ and $G \in \mathbb{R}^{n\times n}$. They proposed the proximal matrices S and T to be positive definite, and the step size $\gamma = 1$ in this classical proximal ADMM (1.2.7). The proximal ADMM covers the classical ADMM when $S = T = 0$.

Fazel et al. [40] proposed a semi-proximal ADMM with the semidefinite matrices S and T, and the step size γ here to be in a range of $\gamma \in (0, (1+\sqrt{5})/2)$. Fazel et al. [40] showed its global convergence when S and T are positive semidefinite, in contrast to the positive definite requirements in the classical proximal ADMM [34, 60], which makes the algorithm more flexible. The proximal ADMM has an advantage that its subproblems are easy to solve, and it also can efficiently handle the multi-block convex optimization problem which is known as block-wise ADMM [66]. See [30, 40, 65, 114] for a brief history of the developments of the semi-proximal ADMM and the corresponding convergence results.

The global convergence of the semi-proximal ADMM is easy to prove. However, it is not satisfactory in numerical performance since it always takes a lot of iterations. The paper [30] mentioned that the proximal matrix T in (1.2.7b) could be indefinite if $\alpha \in (0,1)$ though it provided no further discussions on theoretical properties. Li et al. [77] proved the global convergence. He et al. [63] proposed a linearized version of ADMM with an indefinite proximal term. They considered the case that the matrix $S=0$ and $\alpha=1$ in (1.2.7), and generated the proximal matrix T as

$$T = \tau r I - \beta B^\top B \quad \text{with} \quad r > \beta \|B^\top B\|, \quad \tau \in (0.75, 1). \tag{1.2.8}$$

The proximal matrix T is not necessarily positive semidefinite. A smaller value $\tau \in (0.75,1)$ can ensure the convergence and also give better numerical performance.

Among the family of alternating direction algorithms, inexact versions are allowed the subproblems in (1.2.6) and (1.2.7) to be solved approximately with certain implementable criteria. For the classical ADMM, the approximate ADMM was first developed in [35]. They showed an approximate solution replacing the exact minimizations of (1.2.6a) and (1.2.6b) with absolute summable error criteria and some research papers are proposed such as [21, 60, 117, 92]. For the proximal ADMM (1.2.7), let x_{exact}^{k+1} and y_{exact}^{k+1} be the exact solution of the corresponding subproblems in (1.2.7). Inexact ADMM aims to find approximate solution x^{k+1} and y^{k+1} of x_{exact}^{k+1} and y_{exact}^{k+1}, respectively. Based on (1.2.7), He et al. [60] proposed an inexact proximal ADMM where the parameters β, S and T are replaced by some bounded sequences of positive definite matrices $\{H_k\}$, $\{S_k\}$ and $\{T_k\}$, respectively. The approximate solution x^{k+1} and y^{k+1} are obtained by absolute summable error criteria

$$\|x^{k+1} - x_{exact}^{k+1}\| \le \nu_k, \quad \|y^{k+1} - y_{exact}^{k+1}\| \le \nu_k, \quad \text{and} \quad \sum_0^\infty \nu_k < +\infty.$$

A relaxed error criteria (relative error criterion) was further given in [117] as

$$\|x^{k+1} - x_{exact}^{k+1}\| \le \nu_k \|x^k - x^{k+1}\|, \ \ \|y^{k+1} - y_{exact}^{k+1}\| \le \nu_k \|y^k - y^{k+1}\|, \ \ \text{and} \ \ \sum_0^\infty \nu_k^2 < +\infty.$$

The convergence properties of such algorithms have been established in [3, 53, 82].

Recently, many researchers are interested in extending two block ADMM to multi-block ADMM for solving the multi-block convex optimization problems (1.1.4). For the general case, i.e., $N \ge 3$, a straightway idea is to extend the classical ADMM as (1.2.9) and it indeed works well for some applications, see e.g. [104].

$$\begin{cases} x_i^{k+1} = \arg\min\limits_{x_i \in \mathcal{X}_i} \mathcal{L}_\beta(x_1^{k+1}, ..., x_{i-1}^{k+1}, x_i, x_{i+1}^k, ..., x_N^k, \lambda^k), \quad i = 1, \cdots, N, \\ \lambda^{k+1} = \lambda^k - \beta \left(\sum_{i=1}^N A_i x_i^{k+1} - b \right). \end{cases} \tag{1.2.9}$$

However, according to [19], the classical ADMM for N-block ($N > 2$) convex optimization problems is not necessarily convergent. In order to guarantee the convergence of the extended ADMM, additional assumptions on the objective functions such as smooth, or at least $N-2$ functions are strongly convex are needed [17, 20, 56, 76, 79]. Some works imposed to slightly change the order of the iterative scheme or add some restrictions to the Lagrangian multiplier update step [57, 64, 70]. Moreover, some researchers considered to group the N functions in the objective of (1.1.4) and all the variables, accordingly as two groups [66]. Problem (1.1.4) can be written as

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{N_1} \Phi_i(x_i) + \sum_{j=1}^{N_2} \Psi_j(y_j) \\ &\text{subject to} \quad \sum_{i=1}^{N_1} A_i x_i + \sum_{j=1}^{N_2} B_j y_j = b,\ x_i \in \mathcal{X}_i,\ y_j \in \mathcal{Y}_j, \end{aligned} \tag{1.2.10}$$

where $N_1 \geq 1$, $N_2 \geq 1$, $N = N_1 + N_2$.

Then the original ADMM (1.2.6) becomes applicable in a block wise form and the proximal techniques also can be applied to the block-wise ADMM (1.2.10) [66].

1.2.4 Peaceman-Rachford Splitting method

The solution for problem (1.1.1) is x^* such that

$$0 \in \partial f(x^*) + \partial g(x^*).$$

We consider these two subdifferentials as two maximum monotone operator J_1 and J_2 such that

$$0 \in (J_1 + J_2)(x^*). \tag{1.2.11}$$

The operator splitting methods are the methods for finding a zero of the sum of two maximum monotone operators (1.2.11).

As known, the ADMM is a special case of a method called the Douglas-Rachford splitting method (DRSM) for monotone operators [32, 80]. The variant of ADMM that performs an extra λ-update between the x- and y-updates is equivalent to the Peaceman-Rachford splitting method (PRSM) [80, 98] instead, as shown by [45, 48], one can derive the following iterative scheme for (1.1.1) and (1.2.6):

$$\begin{cases} x^{k+1} = \arg\min_{x \in \mathcal{X}} \mathcal{L}_\beta(x, y^k, \lambda^k), & \text{(1.2.12a)} \\ \lambda^{k+\frac{1}{2}} = \lambda^k - \beta(Ax^{k+1} + By^k - b), & \text{(1.2.12b)} \\ y^{k+1} = \arg\min_{y \in \mathcal{Y}} \mathcal{L}_\beta(x^{k+1}, y, \lambda^{k+\frac{1}{2}}), & \text{(1.2.12c)} \\ \lambda^{k+1} = \lambda^{k+\frac{1}{2}} - \beta(Ax^{k+1} + By^{k+1} - b), & \text{(1.2.12d)} \end{cases}$$

where $\lambda \in \mathbb{R}^m$, β have the same meaning as (1.2.6). Different from the global convergence of the ADMM (1.2.6), which can be established under very mild conditions [13], the convergence of the Peaceman-Rachford-based method (1.2.12) cannot be guaranteed without further conditions [28]. As analyzed in [45], the PRSM has the addition of the intermediate update of the multiplier $\lambda^{k+\frac{1}{2}}$, and it thus offers the same set of advantages. However, the PRSM scheme is less robust and thus it converges under more restrictive assumptions than ADMM. Also as remarked in [45], under the Lipschitz continuity and coercivity of ∂g^* (g^* denotes the conjugate function of g) assumptions, the PRSM (1.2.12) with optimal parameters converges on the linear rate. Some numerical experiments for the efficiency of PRSM had been verified [8, 48].

He et al. [61] proposed a modification of (1.2.12) by introducing a parameter α to the update scheme of the dual variable λ in (1.2.12b) and (1.2.12d). Note that when $\alpha = 1$, it is the same as (1.2.12). They explained the non-convergence behavior of (1.2.12) from the contract perspective, i.e., the distance from the iterative point to the solution set is merely nonexpansive, but not contractive. Under the condition that $\alpha \in (0, 1)$, they proved the same sublinear convergence rate as that for ADMM [65]. Particularly, they showed that it achieves an approximate solution of (1.1.1) with the accuracy of $O(1/t)$ after t iterations, both in the ergodic and nonergodic sense. Besides, Gu [54] and He et al. [62] took two different constants α and γ to different step sizes in (1.2.12b) and (1.2.12d). The convergence results, including global convergence and the worst-case $O(1/t)$ convergence rate in the ergodic and nonergodic sense, have been established in [54]. Chen et al. [25] proposed a new Peaceman-Rachford splitting method in a prediction-correction framework. For some recent advances of the Peaceman-Rachford splitting method, one can refer to [2, 47, 68, 72, 75, 113], to name a few.

1.3 Motivations and contributions

As mentioned above, the ADMM and PRSM are parallel methods that are verified to be very efficient for large-scale optimization problems. Moreover, although these methods have been widely studied, there still exist a lot of issues that should be considered. For example, does the proximal ADMM reach nearly the same optimal solutions as the exact ADMM or more efficient on the iteration steps? Therefore, such many applications and problems motivate us to join the research on the proximal alternating direction methods.

In this thesis, we carry out our study from two aspects; one is the research for the ADMM and the other is on another splitting method PRSM. In the following, the contributions of this thesis are itemized.

(1) To propose a proximal ADMM with the BFGS update for structured convex

quadratic problem

We present a new proximal ADMM with variable positive semidefinite matrices sequence $\{T_k\}$ for an unconstrained structured convex quadratic problem. Under some sufficient conditions on T_k, the sequence generated by the proposed algorithm globally converges to an optimal solution. Since the subproblems of normal proximal ADMM do not include any second-order information on the objective function, the convergence of it might be slower (actually it is fast on the computation time but takes more iteration steps). We construct these T_k via the BFGS update at every iteration step which also satisfy the above convergence conditions. The subproblems are easily solved and have some information on the Hessian matrix.

(2) To extend the ADMM with the Broyden family update for two general convex optimization problems

Inspired by the proposed ADMM with the BFGS update, we consider extending it to more general problems. According to the results in the above algorithm. The BFGS update for the positive semidefinite matrix of the proximal term only can be applied when the Hessian matrix of the augmented Lagrangian function is constant. We describe how to extend our algorithm for two general convex optimization problems and the constructions of T_k via the Broyden family update. For the generic variable metric semi-proximal ADMM, we establish the convergence under certain flexible conditions on the proximal matrices sequence.

(3) To propose a variable metric indefinite proximal ADMM to allow for a larger stepsize

The above two papers reported numerical results for LASSO and l_1 regularized logistic regression. The results show that the algorithms can get a solution faster than the general indefinite proximal ADMM whose proximal term is fixed. Another interesting numerical result is that a variable indefinite sequence via the BFGS update also shows a good performance. Inspired by the interesting results and the indefinite proximal ADMM, it is worth considering ADMM with a sequence of indefinite proximal matrices. We proposed a variable metric indefinite proximal ADMM (VMIP-ADMM), which can unify the several existing ADMMs. We present sufficient conditions on the indefinite proximal matrices for the global convergence of VMIP-ADMM. The proof is followed by separating the constant indefinite term into two semidefinite parts. Moreover, we provide a construction of the indefinite term via the BFGS update. We also show that this construction of the proximal term satisfies the above conditions for the global convergence property.

(4) To propose an indefinite proximal PRSM to allow for a larger stepsize

We first extend the so-called strictly contractive Peaceman-Rachford splitting method by using two different relaxation factors α and γ in (1.2.12b) and (1.2.12d). As already mentioned in Section 1.2, the semi-proximal ADMM is flexible and easy to prove the convergence. The ADMM with a positive indefinite proximal term also had been studied for purpose of improved numerical performance. Li et al. [77] proved the convergence under the assumption that f and g are smooth convex functions with Lipschitz continuous gradient. Recently, He et al. [63] obtained a linearized version of ADMM with a positive-indefinite proximal term by using a linearization technique to the subproblem and showed the convergence without further condition. Motivated by the recent advances on the ADMM type method with indefinite proximal terms, we employ the indefinite proximal term in the strictly contractive Peaceman-Rachford splitting method. Moreover, we show how to choose T under different α and γ. The results of the proposed algorithm can unify that of several existing splitting methods.

1.4 Outline of the thesis

This thesis is organized as follows.

In Chapter 2, we first introduce some preliminaries, including notations, basic definitions, and some properties which are necessary for the later discussion.

In Chapter 3, we propose a variable metric semi-proximal ADMM whose regularized matrix in the proximal term is generated by the BFGS update (or limited memory BFGS) at every iteration for a structured convex quadratic problem. These types of matrices use the second-order information of the objective function. We establish the global convergence of the proposed method under certain assumptions. Finally, numerical results are given to show the effectiveness of the proposed proximal ADMM.

In Chapter 4, we consider the extension of the variable metric semi-proximal ADMM for more general convex optimization problems. We apply the ADMM with the Broyden family update when the x-subproblems of these convex problems can be rewritten as unconstrained quadratic programming problems, as shown in Chapter 3. Moreover, the global convergence of such methods for general cases has also been established under some standard conditions. The numerical results for the l_1 regularized logistic regression problem are given to show the feasibility and effectiveness of the proposed algorithms.

In Chapter 5, we consider a variable metric indefinite proximal ADMM and give sufficient conditions on the proximal terms for the global convergence. Moreover, based on the BFGS update, we propose a new indefinite proximal term which can satisfy the conditions for the global convergence. Experiments on several datasets demonstrated that our proposed variable metric indefinite proximal ADMM outperforms most of the compared proximal

ADMMs.

In Chapter 6, we propose an indefinite-proximal strictly contractive Peaceman-Rachford splitting method. We generalize the proximal matrix from positive definite to indefinite. We show that the proposed indefinite-proximal strictly contractive Peaceman-Rachford splitting method is convergent and also prove the $o(1/t)$ convergence rate in the nonergodic sense. The numerical tests on the l_1 regularized least square problem demonstrate the efficiency of the proposed method.

Finally, in Chapter 7, we give some concluding remarks and mention some issues for future work.

Chapter 2

Preliminaries

In this chapter, we introduce some mathematical notations, basic definitions and properties which will be used in the subsequent chapters.

2.1 Notations

Let $\mathbb{R}^n$ denote the n-dimensional real Euclidean space. All of the vectors in $\mathbb{R}^n$ are column vectors and $^\top$ means the transpose operation. For any vectors $x, y \in \mathbb{R}^n$, the Euclidean inner product $\langle x, y\rangle$ and $x^\top y$ are defined by

$$\langle x, y\rangle = x^\top y = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n.$$

For any vector $x \in \mathbb{R}^n$, we let

$$(x)_+ = \max(x, 0).$$

For a vector $x \in \mathbb{R}^n$ and a matrix $G \in \mathbb{R}^{n\times n}$, $G \succeq 0$, the norms $\|x\|_1$, $\|x\|_2$, $\|x\|_\infty$ and $\|x\|_G$ are defined as follows:

$$\begin{aligned}
\|x\|_1 :&= |x_1| + \cdots + |x_n|,\\
\|x\|_2 :&= \sqrt{\langle x, x\rangle} = \sqrt{x_1^2 + \cdots + x_n^2},\\
\|x\|_\infty :&= \max\left\{|x_1|, \cdots, |x_n|\right\},\\
\|x\|_G :&= \sqrt{\langle x, Gx\rangle}.
\end{aligned}$$

Particularly, we let $\|\cdot\|$ denote the 2-norm $\|\cdot\|_2$. For a matrix $A \in \mathbb{R}^{n\times n}$, $\|A\|$ denotes the operator norm defined by

$$\|A\| = \max_{x\neq 0,\, x\in\mathbb{R}^n} \frac{\|Ax\|}{\|x\|}.$$

For a nonsingular matrix $A \in \mathbb{R}^{n\times n}$, we define $A^{-\top}$ by

$$A^{-\top} = (A^{-1})^{\top} = (A^{\top})^{-1}.$$

Let $S_1, \cdots, S_n$ be some sets. We define the Cartesian product of $S_1, \cdots, S_n$ as

$$S_1 \times \cdots \times S_n \colon = \{(s_1, \cdots, s_n) \mid s_1 \in S_1, \cdots, s_n \in S_n\}.$$

For a differentiable function $f\colon \mathbb{R}^n \to \mathbb{R}$, the gradient of f at x, $\nabla f(x) \in \mathbb{R}^n$ is defined by

$$\nabla f(x) \colon = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix} \in \mathbb{R}^n,$$

where $\frac{\partial f(x)}{\partial x_i}$ denotes a partial derivative of f at x with respect to its i-th component. Moreover, if f is twice differentiable, the Hessian matrix of f at x, $\nabla^2 f(x) \in \mathbb{R}^{n\times n}$ is defined by

$$\nabla^2 f(x) \colon = \begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{pmatrix} \in \mathbb{R}^{n\times n}.$$

For a differentiable vector-valued function $F\colon \mathbb{R}^n \to \mathbb{R}^m$, $\nabla F(x)$ denotes the Jacobian of F at x, that is,

$$\nabla F(x) \colon = (\nabla F_1(x), \cdots, \nabla F_m(x)) \colon = \begin{pmatrix} \frac{\partial F_1(x)}{\partial x_1} & \cdots & \frac{\partial F_m(x)}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_1(x)}{\partial x_n} & \cdots & \frac{\partial F_m(x)}{\partial x_n} \end{pmatrix} \in \mathbb{R}^{n\times m}.$$

2.2 Basic properties

In this section, we introduce some preliminaries that will be useful in this thesis. We give some basic definitions and properties of convex functions, variational inequality, and the optimal conditions.

2.2.1 Convexity, monotonicity and Lipschitz continuity

To begin with, we give the definitions and relevant properties related to the convexity.

Definition 2.2.1. *A set $C \subseteq \mathbb{R}^n$ is said to be convex if*

$$(1-\theta)x + \theta y \in C, \ \forall x, y \in C, \ \forall \theta \in [0, 1].$$

Definition 2.2.2. *For a given function $f\colon \mathbb{R}^m \to (-\infty, \infty]$, we define the effective domain of f by*

$$\mathrm{dom} f\colon = \{x \in \mathbb{R}^n \mid f(x) < \infty\}.$$

Then, we say that

1. *the function f is proper if* $\mathrm{dom} f \neq \emptyset$*;*

2. *the function f is closed if* $\mathrm{dom} f$ *is closed and f is lower-semi-continuous.*

Definition 2.2.3. *Let $C \subseteq \mathrm{dom}\, f$ be a convex set and $f\colon C \to \mathbb{R}$ be a scalar function. Then, f is said to be*

(1) *convex if it holds that*

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y),\ \forall x, y \in C, \theta \in [0,1].$$

(2) *strictly convex if it holds that*

$$f(\theta x + (1-\theta)y) < \theta f(x) + (1-\theta) f(y),\ \forall x, y \in C, \theta \in [0,1].$$

(3) *μ_f-strongly convex on C, $\mu_f > 0$, if it holds that*

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y) - \frac{1}{2}\mu_f \theta(1-\theta)\|x-y\|_2^2,\ \forall x, y \in C, \theta \in [0,1].$$

Besides, the function f is called (strictly) concave on C if $-f$ is (strictly) convex on C. Additionally, if the function f is differentiable, we have the following necessary and sufficient condition for convexity.

Proposition 2.2.1 ([10, 14, 100])**.** *Let $C \subseteq \mathrm{dom}\, f$ be a convex set and $f\colon C \to \mathbb{R}$ be a differentiable function. Then, the function f is convex over C if and only if*

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x),\ \forall x, y \in C.$$

Moreover, the function f is strictly convex on C if and only if the above inequality is strict whenever $x \neq y$.

Convexity plays an important role in the field of optimization. For example, the nonlinear programming problem, if the objective function f and constraint functions are all convex, then any local minimum of such problem is a global minimum.

Now we give the definition of monotonicity for a mapping $f(\cdot)$, both singe-valued and set-valued.

Definition 2.2.4. *A mapping* $f\colon \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ *is called*

(1) *monotone if it holds that*

$$(x-y)^\top(u-v) \geq 0,\ \forall\, x,y \in \mathbb{R}^n, \text{ and } \ u \in f(x), v \in f(y),$$

and strictly monotone if the inequality is strict when $x \neq y$*.*

(2) *strongly monotone with modulus* $\mu > 0$ *if it holds that*

$$(x-y)^\top(u-v) \geq \mu\|x-y\|^2,\ \forall\, x,y \in \mathbb{R}^n, \text{ and } \ u \in f(x), v \in f(y).$$

(3) *maximal monotone if no monotone mapping* Ψ *exists such that* $\operatorname{gph} f \subset \operatorname{gph}\Psi$*. The graph of the mapping* f *is defined by*

$$\operatorname{gph} f = \{(x,y) \in \mathbb{R}^n \times \mathbb{R}^n \colon y \in f(x)\}.$$

When the f is single-valued, the monotonicity takes the form

$$(x-y)^\top(f(x)-f(y)) \geq 0,\ \forall\, x,y \in \mathbb{R}^n.$$

Definition 2.2.5. *A single-valued function* $f\colon \mathbb{R}^n \to \mathbb{R}$ *is said to be Lipschitz continuous with the Lipschitz constant* $L > 0$*, if it holds that*

$$\|f(x)-f(y)\| \leq L\|x-y\|,\ \forall x,y \in \operatorname{dom} f.$$

More generally, for a convex (not necessary smooth) function f, it has a similar character as Proposition 2.2.1 by subgradient.

Definition 2.2.6. *Let a function* $f\colon \mathbb{R}^n \to \mathbb{R}$ *be convex. A vector* $\xi \in \mathbb{R}^n$ *is a subgradient of the function* f *at a point* $x \in \operatorname{dom} f$ *if*

$$f(y) \geq f(x) + \xi^\top(y-x),\ \forall y \in \operatorname{dom} f.$$

The set of all subgradients of the function f *at the point* $x \in \operatorname{dom} f$*, denoted by* $\partial f(x)$*, is called the subdifferential of function* f *at the point* $x \in \operatorname{dom} f$*, i.e.,*

$$\partial f(x)\colon = \left\{\xi \mid \xi^\top(y-x) \leq f(y)-f(x),\ \forall y \in \operatorname{dom} f\right\}.$$

In particular, there is a relation between monotonicity of subdifferential mappings and convexity of functions. We define the subdifferential function $\partial f\colon \mathbb{R}^n \to \mathbb{R}^n$ of the convex function f.

Proposition 2.2.2 ([40, Appendix B], [101, Theorem 12.17])**.** *Let* $f\colon \mathbb{R}^n \to \mathbb{R}$ *be a proper convex function. Then, its corresponding subdifferential mapping* $\partial f\colon \mathbb{R}^n \to \mathbb{R}^n$ *is monotone. Moreover, consider a proper closed convex function* f*, its subdifferential mapping* ∂f *is maximal monotone, and there exist a positive semidefinite matrices* Σ_f *such that for all* $x, \hat{x} \in \mathbb{R}^n$*,*

$$f(x) \geq f(\hat{x}) + \partial f(\hat{x})^\top(x-\hat{x}) + \frac{1}{2}\|x-\hat{x}\|^2_{\Sigma_f}, \text{ and } (x-\hat{x})^\top(\partial f(x)-\partial f(\hat{x})) \geq \|x-\hat{x}\|^2_{\Sigma_f}.$$

2.2.2 Optimal solution and optimal conditions

In this subsection, we introduce the definitions of optimal solutions, as well as the first order optimality condition of problem (1.1.3). For more details, see [9, 10, 14, 89, 100] and references therein.

At first, we should give the definitions of cones that will be used.

Definition 2.2.7. *A set C is called a cone if*

$$\alpha > 0,\ x \in C \ \Rightarrow\ \alpha x \in C.$$

Definition 2.2.8. *Given a nonempty set C, the polar cone of C is given by*

$$C^* = \{y \mid y^\top x \leq 0,\ \forall\, x \in C\}.$$

Definition 2.2.9. *Given a subset $C \subseteq \mathbb{R}^n$ and a vector $x \in C$, a vector $y \in \mathbb{R}^n$ is said to be a tangent of C at x if either $y = 0$ or there exits a sequence $\{x_k\} \subset C$ such that*

$$x_k \to x,\quad \frac{x_k - x}{\|x_k - x\|} \to \frac{y}{\|y\|},\quad \forall\, k, x_k \neq x.$$

Then the set of all tangents of C at x is called the tangent cone of C at x and denoted by $T_C(x)$.

Definition 2.2.10. *Given a set C and a vector $x \in C$, the normal cone of C at x is defined by*

$$\mathcal{N}_C(x) = \{d \in \mathbb{R}^n \colon d^\top(y - x) \leq 0,\ \forall y \in C\}.$$

Vectors in this set are called normal vectors of the set C at x.

Then we give the relation between the tangent cone and the norm cone for a convex set.

Proposition 2.2.3. *Let C be a nonempty convex subset of $\mathbb{R}^n$ and $x \in C$.*

1. $d \in T_C(x)^* \ \Leftrightarrow\ \forall y \in C \quad d^\top(y - x) \leq 0.$

2. *C is regular for all $x \in C$: $T_C(x)^* = \mathcal{N}_C(x)$.*

3. $T_C(x) = \mathcal{N}_C(x)^*$.

Next, we give the definitions of optimal solutions. Let $F(x)$ be denoted by $F(x) :=$ $f(x) + g(x)$ in problem (1.1.1).

Definition 2.2.11. *A vector x^* is said to be*

(1) *a locally optimal solution of problem* (1.1.1) *if there exists a scalar* $\delta > 0$ *such that*

$$F(x^*) \leq F(x), \ \forall x \in \{x \mid \|x - x^*\| \leq \delta, \ x \in \operatorname{dom} F\}.$$

The value $F(x^*)$ *is called the local minimum of problem* (1.1.1).

(2) *a globally optimal solution of problem* (1.1.1) *if it holds that*

$$F(x^*) \leq F(x), \ \forall x \in \operatorname{dom} F.$$

The value $F(x^*)$ *is called the global minimum of problem* (1.1.1).

When F is a smooth function and C is a convex set, we can have the following basic necessary condition for the local optimality.

Theorem 2.2.1 ([9, 10])**.** *Let* $F(x)\colon \mathbb{R}^n \to \mathbb{R}$ *be a smooth function, and let* x^* *be a local minimum of* F *over a convex subset* C *of* $\mathbb{R}^n$*. Then*

$$\nabla F(x^*)^\top (x - x^*) \geq 0, \ \ \forall\, x \in C.$$

In the case where $C = \mathbb{R}^n$*, it reduces to* $\nabla F(x^*) = 0$.

Moreover, if the function F is convex, then the above condition $\nabla F(x^*) = 0$ is a necessary and sufficient condition for a vector x^* to be a globally optimal solution. Next we give a general necessary and sufficient condition when the problem is convex but not necessarily smooth.

Theorem 2.2.2 ([10, 12, 89])**.** *Let* $F(x)\colon \mathbb{R}^n \to \mathbb{R}$ *be a convex function. A vector* x^* *minimizes* F *over a convex set* $C \subseteq \mathbb{R}^n$ *if and only if one of the following equivalently statements holds :*

(1) $0 \in \partial F(x^*) + \mathcal{N}_C(x^*)$.

(2) *there exists a subgradient* $d \in \partial F(x^*)$ *such that*

$$d^\top (x - x^*) \geq 0, \ \ \forall\, x \in C.$$

The optimality condition in Theorem 2.2.2 (2) of the constrained nonsmooth convex problem is always used in the convergence analysis. Specially when $C = \mathbb{R}^n$, we obtain a necessary and sufficient condition for the unconstrained optimality of x^* :

$$0 \in \partial F(x^*).$$

Next we consider the structured convex problem (1.1.3) with the linear constraint, and the set constraints X, Y to be $\mathbb{R}^n$. We have the first-order necessary conditions for a solution to be optimal, which is named Karush-Kuhn-Tucker (KKT) conditions, also known as the Kuhn-Tucker conditions.

Definition 2.2.12. *Let the objective functions f and g are continuously differentiable and convex. A pair of point (x^*, y^*) is called a KKT point, along with a Lagrange multiplier λ^* (KKT multiplier) if they satisfy*

$$\begin{aligned} \nabla f(x^*) - A^\top \lambda^* &= 0, \qquad \text{(Stationarity)} \\ \nabla g(y^*) - B^\top \lambda^* &= 0, \qquad \text{(Stationarity)} \\ Ax^* + By^* - b &= 0, \qquad \text{(Primal feasiblility)} \end{aligned} \tag{2.2.1}$$

where

$$\mathcal{L}(x, y, \lambda) \colon = f(x) + g(y) - \lambda^\top (Ax + By - b) \tag{2.2.2}$$

is the Lagrangian function of the problem (1.1.3).

If some of the objective functions are non-differentiable, the subdifferential version of the KKT conditions are available via replacing the gradients in (2.2.1) by subgradients. When a triple (x^*, y^*, λ^*) satisfies the KKT conditions (2.2.1), then (x^*, y^*) is the optimal solution of problem (1.1.3). Conversely, in order for a optimal solution (x^*, y^*) to satisfy the above KKT conditions, the problem should satisfy a suitable constraint qualification (CQ) such as linearity constraint qualification (LCQ), linear independence constraint qualification (LICQ), Mangasarian-Fromovitz constraint qualification (MFCQ), etc.

For example, we consider the problem (1.1.3) with certain set constraints X and Y. Let $(\bar{x}, \bar{y}) \in \operatorname{dom} f \times \operatorname{dom} g$ be an optimal solution, and $(\bar{x}, \bar{y}.\bar{\lambda})$ be the KKT point for problem (1.1.3). The existence of such KKT points can be guaranteed if a certain constraint qualification such as the Slater condition holds:

$$\exists\, (x', y') \in \operatorname{ri}(\operatorname{dom} f \times \operatorname{dom} g) \cap \{(x, y) \in \mathcal{X} \times \mathcal{Y} : Ax + By = b\},$$

where $\operatorname{ri}(S)$ denotes the relative interior of a given convex set S.

2.2.3 Variational Inequality

The variational inequality (VI) problem is another class of problems but closely related with convex optimization problems. The KKT conditions of convex optimization problems can be expressed as forms of these VI problems although the VI problem is not the optimization problem for minimizing a specific objective function. In a sense, a differentiable convex optimization problem is a kind of VI problem with special characters. First we focus on some basic concepts of the VI and describe the problem (1.1.3) to the VI.

Definition 2.2.13. *Given a nonempty closed subset K of $\mathbb{R}^n$ and a continuous mapping $H \colon K \to \mathbb{R}^n$, the variational inequality (VI), denoted by VI(K,H) is a problem to find a vector $x^* \in K$ such that*

$$(x - x^*)^\top H(x^*) \geq 0, \quad \forall x \in K. \tag{2.2.3}$$

The set of solutions to this problem is denoted as SOL(K,H).

From the definition of the VI, it is clear that a vector $x \in K$ solves the VI(K, H) if and only if $-H(x)$ is a norm vector to K at x, equivalently,

$$0 \in H(x) + \mathcal{N}_K(x).$$

A number of results on the existence and uniqueness of a solution in the VI(K, H) problem have been studied [96]. One of the most basic results relies on the compactness and convexity of the set K. Instead of the compactness of K, some other existence results can be obtained by imposing another condition such as the coerciveness of the mapping H. On the other hand, we have the following results on the uniqueness of the solution under the monotonicity assumptions on H.

Proposition 2.2.4 ([96, Proposition 3.2])**.** *The VI(K, H) has at least one solution if H is strictly monotone.*

Proposition 2.2.5. *If H is strictly monotone, then there exists a unique solution to the problem VI(K, H).*

Now we relate the VI(K, H) problem to some projection equations.

Definition 2.2.14. *The projection of a vector $x \in \mathbb{R}^n$ onto a set K under the Euclidean norm is denoted by $P_K(x)$, i.e.,*

$$P_K(x) = \text{argmin}\{\|x - y\| \mid y \in K\}.$$

The mapping $P_K : \mathbb{R}^n \to K$ is called the projection operator.

Basic properties of the projection are obtained.

Proposition 2.2.6 ([96])**.** *Let K be a nonempty closed convex subset of $\mathbb{R}^n$.*

(1) *For each $x \in \mathbb{R}^n$, $P_K(x)$ exists and is unique.*

(2) *$P_K(x)$ is nonexpansive, that is for any two vectors $u, v \in \mathbb{R}^n$,*

$$\|P_K(u) - P_K(v)\| \leq \|u - v\|.$$

The following lemma describes the solution of VI(K, H) and the projection.

Lemma 2.2.1 ([11, 96])**.** *Let VI(K, H) be defined in* (2.2.3) *and γ be any positive constant. Then $x^* \in K$ is a solution of VI(K, H) if and only if*

$$x^* = P_K[x^* - \gamma H(x^*)].$$

Therefor, it is obvious to see that solving VI(K,H) is equivalent to finding a zero point of the following equation,

$$e(x,K,\gamma H):=x-P_K[x-\gamma H(x)]. \tag{2.2.4}$$

Many applications of the VI can be found in various areas [96, 41, 86], such as the economics, transportation systems, mechanics, etc. Next we explain some relations between the convex problems with the VI problems. First we consider the convex optimization problem (1.1.1), with the feasible set C defined by

$$C=\{x\in\mathbb{R}^n \mid Ax=b,\ x\in C\}.$$

By introducing a Lagrangian multiplier $\lambda\in\mathbb{R}^m$ for the linear constraint $Ax=b$, we can obtain the following Lagrangian function which is defined on $C\times\mathbb{R}^m$

$$\mathcal{L}(x,\lambda)=f(x)+g(x)-\lambda^\top(Ax-b).$$

(x^*,λ^*) is called the saddle point if it satisfies

$$\mathcal{L}_{\lambda\in\mathbb{R}^m}(x^*,\lambda)\le\mathcal{L}(x^*,\lambda^*)\le\mathcal{L}_{x\in C}(x,\lambda^*).$$

It is equivalent to find $(x^*,\lambda^*)\in C\times\mathbb{R}^m$, $f'(x^*)\in\partial f(x^*)$ and $g'(x^*)\in\partial g(x^*)$, such that

$$\begin{cases}(x-x^*)^\top(f'(x^*)+g'(x^*)-A^\top\lambda^*)\ge 0, & \forall x\in C,\\ (\lambda-\lambda^*)^\top(Ax^*-b)\ge 0, & \forall\lambda\in\mathbb{R}^m.\end{cases} \tag{2.2.5}$$

Similarly, we consider the structured convex optimization problem (1.1.3). The Lagrangian function of this problem is written as (2.2.2), where λ is the Lagrangian multiplier for the linear constraint $Ax+By=b$ in (1.1.3). Let Ω be the set defined by $\Omega=\mathcal{X}\times\mathcal{Y}\times\mathbb{R}^m$, (x^*,y^*,λ^*) be an saddle point of the Lagrangian function, then $(x^*,y^*,\lambda^*)\in\Omega$ and it satisfies

$$\begin{cases}(x-x^*)^\top(f'(x^*)-A^\top\lambda^*)\ge 0,\\ (y-y^*)^\top(g'(y^*)-B^\top\lambda^*)\ge 0, & \forall(x,y,\lambda)\in\Omega,\\ (\lambda-\lambda^*)^\top(Ax^*+By^*-b)\ge 0,\end{cases} \tag{2.2.6}$$

where $f'(x^*)\in\partial f(x^*)$ and $g'(y^*)\in\partial g(y^*)$. Furthermore, by denoting

$$u=\begin{pmatrix}x\\ y\end{pmatrix},\ w=\begin{pmatrix}x\\ y\\ \lambda\end{pmatrix},\ F(w)=\begin{pmatrix}f'(x)-A^\top\lambda\\ g'(y)-B^\top\lambda\\ Ax+By-b\end{pmatrix},$$

where the subgradients $f'(x)\in\partial f(x)$ and $g'(y)\in\partial g(y)$, the first-order optimal condition (2.2.6) can be rewritten in the VI(Ω,F) form as

$$(w-w^*)^\top F(w^*)\ge 0,\quad \forall w\in\Omega. \tag{2.2.7}$$

As mentioned in subsection 2.2.2, $\partial f(x)$ and $\partial g(y)$ are maximal monotone operators. Therefor, according to [96], the VI(Ω, F) (2.2.7) is solvable. We use Ω^* for the solution set of VI(Ω, F), and it is nonempty when under the assumption that the solution set of (1.1.3) is not empty.

From the above Lemma 2.2.1 and (2.2.4), we have the following relations between the VI(Ω, F) with projection.

Lemma 2.2.2. *Let VI(Ω, F) be defined in* (2.2.7) *and γ be any positive constant. Then solving VI(Ω, F) is equivalent to finding a zero point of*

$$e(x, \Omega, \gamma F) :\, = x - P_{\Omega}[x - \gamma F(x)] = \begin{pmatrix} x - P_{\mathcal{X}}\left[x - \gamma\left(f'(x) - A^\top \lambda\right)\right] \\ y - P_{\mathcal{Y}}\left[y - \gamma\left(g'(y) - B^\top \lambda\right)\right] \\ \gamma(Ax + By - b) \end{pmatrix}.$$

Normally, we can choose the constant $\gamma = 1$.

Chapter 3

An Alternating Direction Method of Multipliers with the BFGS update for Structured Convex Quadratic Optimization

3.1 Introduction

In this Chapter, we first consider the following convex optimization problem:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|Ax-b\|^2 + g(x) \\ \text{subject to} & x \in \mathbb{R}^n, \end{array} \tag{3.1.1}$$

where $g : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is a proper convex function, $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. For example, "g" here can be an indicator function on a convex set or the l_1 penalty function defined as $\|x\|_1 := \sum_{i=1}^m |x_i|$. Problem (3.1.1) includes many important statistical learning problems such as the LASSO problem [106]. The number n of variables in these learning problems is usually large.

Let $f(x) = \frac{1}{2}\|Ax-b\|^2$. Then problem (3.1.1) can be written as

$$\begin{array}{ll} \text{minimize} & f(x) + g(y) \\ \text{subject to} & x - y = 0 \\ & x, y \in \mathbb{R}^n. \end{array} \tag{3.1.2}$$

As mentioned in Introduction 1.2.3, the semi-proximal ADMM scheme are given as:

$$\begin{cases} x^{k+1} = \arg\min\limits_x \ \mathcal{L}_\beta(x, y^k, \lambda^k) + \dfrac{1}{2}\|x - x^k\|_T^2, & (3.1.3a) \\ y^{k+1} = \arg\min\limits_y \ \mathcal{L}_\beta(x^{k+1}, y, \lambda^k) + \dfrac{1}{2}\|y - y^k\|_S^2, & (3.1.3b) \\ \lambda^{k+1} = \lambda^k - \alpha\beta(x^{k+1} - y^{k+1}), & (3.1.3c) \end{cases}$$

where $\alpha \in (0, (1+\sqrt{5})/2)$, $\|z\|_G = \sqrt{z^\top G z}$ for $z \in \mathbb{R}^n$ and $G \in \mathbb{R}^{n\times n}$.

In this chapter, we assume that y^{k+1} in (3.1.3b) can be easily obtained. For example, if $g(y) = \tau\|y\|_1$ with $\tau > 0$ and $S = \mathrm{diag}(s_1, \cdots, s_n)$ is diagonal positive semidefinite, then y^{k+1} is calculated by

$$y_i^{k+1} = \mathcal{S}_{\tau/(\beta+s_i)}\left(\frac{1}{\beta+s_i}\left(\beta x_i^{k+1} - \lambda_i^k + s_i y_i^k\right)\right), \quad i = 1, \cdots, n,$$

where $\mathcal{S}_\nu$ is the shrinkage operator defined by $\mathcal{S}_\nu(z) \coloneqq \mathrm{sgn}(z) \odot \max\{|z| - \nu, 0\}$. We may also consider a problem where $g(x) = \tau\|Bx\|_1$, B is a certain matrix, and $B^\top B$ is positive definite. Note that some existing first-order methods are difficult to solve such a problem. For the ADMM, we may set $S = \rho B^\top B - \beta I$, where ρ is a parameter such that S is positive semidefinite. The subproblem of y is written as

$$\begin{aligned} y^{k+1} &= \arg\min_y \left\{\tau\|By\|_1 - \langle\lambda^k, x^{k+1} - y\rangle + \frac{\beta}{2}\|x^{k+1} - y\|^2 + \frac{1}{2}\|y - y^k\|_S^2\right\} \\ &= \arg\min_y \left\{\tau\|By\|_1 + \frac{\rho}{2}(y-a)^\top B^\top B(y-a)\right\}, \end{aligned}$$

where $a = y^k + \frac{\beta}{\rho}(B^\top B)^{-1}\left(x^{k+1} - y^k - (1/\beta)\lambda^k\right)$. This subproblem is equivalent to

$$z^{k+1} = \arg\min_z \left\{\tau|z|_1 + \frac{\rho}{2}\|z - z^k\|^2\right\},$$

where $z = By$, $z^k = Ba$. Then we can easily get z^{k+1} by using the shrinkage operator, and set $y^{k+1} = (B^\top B)^{-1}B^\top z^{k+1}$. We note that for some applications (e.g., Total Variation regularization), $(B^\top B)^{-1}$ is easily calculated.

Therefore, our main focus is how to solve (3.1.3a) when n is large. We may select a reasonable positive semidefinite matrix T such that x^{k+1} can be obtained quickly.

One of such examples of T is $T = \xi I - A^\top A$ with $\xi > \lambda_{\max}(A^\top A)$, where $\lambda_{\max}(A^\top A)$ denotes the maximum eigenvalue of $A^\top A$. Then (3.1.3a) is written as

$$\begin{aligned} x^{k+1} &= \arg\min_x \left\{f(x) - \langle\lambda^k, x - y^k\rangle + \frac{\beta}{2}\|x - y^k\|^2 + \frac{1}{2}\|x - x^k\|_T^2\right\} \\ &= \arg\min_x \left\{\langle Ax^k - b, Ax\rangle - \langle\lambda^k, x\rangle + \frac{\beta}{2}\|x - y^k\|^2 + \frac{\xi}{2}\|x - x^k\|^2\right\} \\ &= (\lambda^k + \beta y^k + \xi x^k - A^\top A x^k + A^\top b)/(\beta + \xi). \end{aligned}$$

The other example is $T = \xi I - \beta I - A^\top A$ with $\xi > \lambda_{\max}(\beta I + A^\top A)$. Then (3.1.3a) is written as

$$x^{k+1} = x^k - \xi^{-1}(A^\top A x^k - A^\top b - \lambda^k + \beta x^k - \beta y^k).$$

In both cases, x^{k+1} is calculated within $O(mn)$ time complexity. However, since these subproblems do not include second-order information on f, the convergence of ADMM with such T might be slow.

We desire a matrix T that is positive semidefinite, enables subproblem (3.1.3a) to be easily solved, and contains some second-order information on f. Let M be the Hessian matrix of the augmented Lagrangian function $\mathcal{L}_\beta$, that is, $M\colon = \nabla^2_{xx}\mathcal{L}_\beta(x, y, \lambda) = A^\top A + \beta I$. Note that $M \succ 0$ whenever $\beta > 0$. Subsequently, we consider a matrix B that has the following three properties:

(i) $T = B - M$;

(ii) $B \succeq M$;

(iii) B retains some second-order information from M.

Properties (i) and (ii) imply that T is positive semidefinite. Moreover, subproblem (3.1.3a) is written as

$$\begin{aligned} x^{k+1} &= \arg\min_x \left\{ f(x) - \langle \lambda^k, x - y^k \rangle + \frac{\beta}{2}\|x - y^k\|^2 + \frac{1}{2}\|x - x^k\|_T^2 \right\} \\ &= \arg\min_x \left\{ \langle A^\top(Ax^k - b) + \beta(x^k - y^k) - \lambda^k, x \rangle + \frac{1}{2}\|x - x^k\|_B^2 \right\} \\ &= x^k - B^{-1}\left(A^\top A x^k - A^\top b - \lambda^k + \beta(x^k - y^k)\right). \end{aligned}$$

In this chapter, we propose to construct B^{-1} via the BFGS update at every iteration. Then subproblem (3.1.3a) can be solved easily. Note that matrices B and T at every step depend on k, that is, they become B_k and T_k, respectively, and the resulting ADMM is a variable metric semi-proximal ADMM (abbreviated as VMSP-ADMM) expressed as follows:

$$\begin{cases} x^{k+1} = \arg\min_x \ \mathcal{L}_\beta(x, y^k, \lambda^k) + \frac{1}{2}\|x - x^k\|_{T_k}^2, & (3.1.4a) \\ y^{k+1} = \arg\min_y \ \mathcal{L}_\beta(x^{k+1}, y, \lambda^k) + \frac{1}{2}\|y - y^k\|_S^2, & (3.1.4b) \\ \lambda^{k+1} = \lambda^k - \beta(x^{k+1} - y^{k+1}). & (3.1.4c) \end{cases}$$

The VMSP-ADMM was studied in [60] where the T_k was assumed to be positive definite. Additionally, the convergence and complexity results have been studied in [3, 53, 82]. Moreover, the VMSP-ADMM is closely related to the inexact ADMM, where the subproblems in (3.1.3) are to be solved approximately with certain implementable criteria [21, 35, 37, 38, 60, 117]. In this chapter, we suppose that subproblems (3.1.4a)-(3.1.4b) are solved exactly.

The main contributions of the chapter are as follows:

1. An update formula on positive semidefinite matrices T_k and B_k is proposed via the BFGS update that satisfies the three properties (i)-(iii) above.

2. Numerical results for the proposed methods are reported, demonstrating that they outperform the existing ADMM when n and m are large.

The rest of this chapter is organized as follows. In Section 3.2, we propose a new ADMM with the BFGS update, and show its global convergence. In Section 3.4, we present some numerical experiment results for the ADMM with the BFGS and limited memory BFGS update. Finally, we make some concluding remarks in Section 3.5.

3.2 Construction of the proximal matrix

In this section, we first propose the updating rule of T_k via the BFGS update for VMSP-ADMM, and show a key property on T_k for the convergence.

3.2.1 Construction of the regularized matrix T_k via the BFGS update

As discussed in Introduction, we propose to construct T_k as $T_k = B_k - M$, where $M = \nabla^2_{xx}\mathcal{L}_\beta(x, y, \lambda)$. We want T_k to be positive semidefinite for global convergence as a usual semi-proximal ADMM. Moreover we want B_k to be as close to M as possible for rapid convergence. To this end, we propose to generate B_k by the BFGS update with respect to M. Then we may consider the BFGS update with a given $s \in R^n$ and $l = Ms$. Note that $s^\top l > 0$ when $s \neq 0$. Since BFGS usually constructs the inverse of B_k, we let $H_k = B_k^{-1}$. Using H_k, we can easily solve subproblem (3.1.4a).

Now we briefly sketch the BFGS update and the limited memory BFGS (L-BFGS) [93, 55]. Let $s_k = x^{k+1} - x^k, l_k = Ms_k$. Then the BFGS updates for B_{k+1} and H_{k+1} are given as

$$B_{k+1}^{BFGS} = B_k + \frac{l_k l_k^\top}{l_k^\top s_k} - \frac{B_k s_k s_k^\top B_k^\top}{s_k^\top B_k s_k}, \tag{3.2.1}$$

$$H_{k+1}^{BFGS} = \left(I - \frac{s_k l_k^\top}{s_k^\top l_k}\right) H_k \left(I - \frac{l_k s_k^\top}{s_k^\top l_k}\right) + \frac{s_k s_k^\top}{s_k^\top l_k}. \tag{3.2.2}$$

Since $s_k^\top l_k > 0$, B_{k+1}^{BFGS} and H_{k+1}^{BFGS} are positive definite whenever $B_k, H_k \succ 0$. Moreover

$$l_k = B_{k+1}^{BFGS} s_k \quad \text{and} \quad s_k = H_{k+1}^{BFGS} l_k.$$

The BFGS update requires only matrix-vector multiplications, which results in the computational cost at each iteration being $O(n^2)$ time complexity. If the number of variables is very large, even $O(n^2)$ per iteration is too expensive in terms of both CPU time and memory usage.

A less computationally intensive method is the limited memory BFGS method [55, 93]. Instead of updating and storing the entire approximated inverse Hessian matrix, the L-BFGS method uses the vectors (s_i, l_i) in the last h iterations and constructs H_{k+1} by using these vectors. The updating in L-BFGS reduces the computational cost to $O(hn)$ time complexity per iteration.

3.2.2 Property of the regularized matrix T_k via the BFGS update

For the global convergence, we need $T_k = B_k - M \succeq 0$, that is $B_k \succeq M$. Note that $B_k \succeq M$ is equivalent to $H_k \preceq M^{-1}$, where $H_k = (B_k)^{-1}$. We will show that $H_k \preceq M^{-1}$ for all k when the initial matrix H_0 satisfies

$$H_0 \preceq M^{-1}.$$

We first show a technical lemma on s and l.

Lemma 3.2.1. *Let $s \in R^n$ such that $s \neq 0$. Moreover let $l = Ms$ and $\Phi = \{z \in R^n \mid \langle s, z \rangle = 0\}$. Then for any $v \in R^n$, there exist $c \in R$ and $z \in \Phi$ such that $v = cl + z$.*

Proof. Let $v \in R^n$. Then there exist $c_1, c_2 \in R$ and $z^1, z^2 \in \Phi$ such that $v = c_1 s + z^1$ and $l = c_2 s + z^2$. Since $s^\top l > 0$, we have $c_2 \neq 0$. Thus $s = \frac{1}{c_2} l - \frac{1}{c_2} z^2$. Substituting it into $v = c_1 s + z^1$ yields

$$v = c_1 \left(\frac{1}{c_2} l - \frac{1}{c_2} z^2 \right) + z^1 = \frac{c_1}{c_2} l + z^1 - \frac{c_1}{c_2} z^2.$$

Let $c = \frac{c_1}{c_2}$ and $z = z^1 - \frac{c_1}{c_2} z^2$. Then $z \in \Phi$ and $v = cl + z$. ■

Note that the BFGS update (3.2.2) can be

$$H_{\text{next}} = H - \frac{Hls^\top + sl^\top H}{s^\top l} + \left(1 + \frac{l^\top H l}{s^\top l} \right) \frac{ss^\top}{s^\top l}, \tag{3.2.3}$$

where H is the proximal matrix for the current step, $s = x_{\text{next}} - x$ and H_{next} is the new matrix generated via BFGS update. Moreover we have

$$H_{\text{next}} l = s = M^{-1} l. \tag{3.2.4}$$

The following theorem will play a key role for the global convergence of the proposed method.

Theorem 3.2.1. *Let $s \in R^n$ such that $s \neq 0$, and let $l = Ms$. If $H \preceq M^{-1}$, then $H_{next} \preceq M^{-1}$.*

Proof. Let v be an arbitrary nonzero vector in R^n. Let $\Phi = \{z \in R^n \mid \langle s, z\rangle = 0\}$. From Lemma 3.2.1 there exist $c \in R$ and $z \in \Phi$ such that $v = cl + z$. It then follows from (3.2.4) and the definition of z that

$$\begin{aligned} v^\top H_{\text{next}} v &= (cl+z)^\top H_{\text{next}}(cl+z) \\ &= c^2 l^\top s + 2cs^\top z + z^\top H_{\text{next}} z \\ &= c^2 l^\top s + z^\top H_{\text{next}} z \\ &= c^2 l^\top M^{-1} l + z^\top H_{\text{next}} z. \end{aligned}$$

We now consider the last term of the right-hand side of the last equation. Since $z \in \Phi$, we have

$$z^\top \left(\frac{sl^\top}{s^\top l} H \frac{ls^\top}{s^\top l}\right) z = 0,$$

$$z^\top \left(\frac{sl^\top}{s^\top l} H\right) z = 0$$

and

$$z^\top \left(\frac{ss^\top}{s^\top l}\right) z = 0.$$

It then follows from (3.2.3) that

$$z^\top H_{\text{next}} z = z^\top H z - 2z^\top \left(\frac{sl^\top}{s^\top l} H\right) z + z^\top \left(\frac{sl^\top}{s^\top l} H \frac{ls^\top}{s^\top l}\right) z + \frac{z^\top ss^\top z}{s^\top l} = z^\top H z.$$

Moreover equation (3.2.4) implies

$$l^\top M^{-1} z = s^\top z = 0.$$

Consequently we have

$$\begin{aligned} v^\top H_{\text{next}} v &= c^2 l^\top M^{-1} l + z^\top H z \\ &\le c^2 l^\top M^{-1} l + z^\top M^{-1} z \\ &= (cl+z)^\top M^{-1}(cl+z) - 2cl^\top M^{-1} z \\ &= v^\top M^{-1} v, \end{aligned}$$

where the inequality follows from the assumption. Since v is arbitrary, we have $H_{\text{next}} \preceq M^{-1}$. ∎

This theorem shows that if $H_0 \preceq M^{-1}$, then $H_k \preceq M^{-1}$, and hence $T_k \succeq 0$.

3.3 Global convergence

We propose the following variable metric semi-proximal ADMM with the BFGS update algorithm (ADM-BFGS).

Variable metric semi-proximal ADMM with the BFGS update (ADM-BFGS)

Input : data matrix A, initial point (x^0, y^0, λ^0), penalty parameter β, maxIter;
initial matrix $H_0 \preceq M^{-1}$, constant $\bar{k} \in [1, \infty]$, stopping criterion ϵ.

Output:
approximative solution (x^k, y^k, λ^k)

initialization;
while $k <$ maxIter *or not converged* **do**
  if $k \leq \bar{k}$ and $x_k - x_{k-1} \neq 0$ **then**
    update H_k via BFGS (or L-BFGS) with the initial matrix H_0;
  else
    $H_k = H_{k-1}$;
  end
  update x^{k+1} by solving the x-subproblem:
    $x^{k+1} = x^k + H_k\left(\lambda^k + \beta y^k + A^\top b - Mx^k\right);$
  update y^{k+1} by solving the $y-$subproblem:
    $y^{k+1} = \arg\min_y \left\{ g(y) - \langle \lambda^k, x^{k+1} - y\rangle + \frac{\beta}{2}\|x^{k+1} - y\|^2 + \frac{1}{2}\|y - y^k\|_S^2 \right\};$
  update Lagrange multipliers: $\lambda^{k+1} = \lambda^k - \beta(x^{k+1} - y^{k+1})$.
end

3.3.1 Convergence of variable metric semi-proximal ADMM

We now develop a general convergence result for variable metric semi-proximal ADMM (3.1.4) for problem (3.1.2) with a general convex function f. We first give some notations and properties which will be frequently used in the analysis.

Let (x^*, y^*) be an optimal solution of problem (3.1.2), and let λ^* be a Lagrange multiplier that satisfies the following KKT conditions of problem (3.1.2):

$$\begin{cases} \eta_f^* - \lambda^* = 0, & (3.3.1a) \\ \eta_g^* + \lambda^* = 0, & (3.3.1b) \\ x^* - y^* = 0, & (3.3.1c) \end{cases}$$

where $\eta_f^* \in \partial f(x^*)$ and $\eta_g^* \in \partial g(y^*)$.

Now we rewrite the iteration schemes (3.1.4a)-(3.1.4b). Let $\Omega = \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n$. Using the first-order optimality conditions for subproblems (3.1.4a)-(3.1.4b), we see that the new iterate (x^{k+1}, y^{k+1}) is generated by the following procedure.

- step 1: Find $x^{k+1} \in \mathbb{R}^n$ such that $\eta_f^{k+1} \in \partial f(x^{k+1})$ and

$$\eta_f^{k+1} - \lambda^k + \beta(x^{k+1} - y^k) + T_k(x^{k+1} - x^k) = 0,$$

- step 2: Find $y^{k+1} \in \mathbb{R}^n$ such that $\eta_g^{k+1} \in \partial g(y^{k+1})$ and

$$\eta_g^{k+1} + \lambda^k - \beta(x^{k+1} - y^{k+1}) + S(y^{k+1} - y^k) = 0.$$

For $k = 0, 1, 2, ...$, we use the following notation:

$$u^* = \begin{pmatrix} x^* \\ y^* \end{pmatrix}, \; u^k = \begin{pmatrix} x^k \\ y^k \end{pmatrix}, \; w^k = \begin{pmatrix} x^k \\ y^k \\ \lambda^k \end{pmatrix}, \; D_k = \begin{pmatrix} T_k & 0 \\ 0 & S \end{pmatrix}, \text{ and } G_k = \begin{pmatrix} T_k & 0 & 0 \\ 0 & S + \beta I & 0 \\ 0 & 0 & \frac{1}{\beta} I \end{pmatrix}. \tag{3.3.2}$$

Moreover, for simplicity, we denote

$$F^k = \begin{pmatrix} \eta_f^k - \lambda^k \\ \eta_g^k + \lambda^k \\ x^k - y^k \end{pmatrix}, \tag{3.3.3}$$

where η_f^k and η_g^k are obtained in steps 1 and 2 in VMSP-ADMM.

For the sequences $\{w^k\}$ and $\{F^k\}$, we have the following lemma, which is a direct consequence of [60, Theorem 1] and [60, Lemma 3].

Lemma 3.3.1. *Let $w^* = (x^*, y^*, \lambda^*)$, and $\{w^k\}$ be generated by the scheme* (3.1.4)*. Then we have the following two statements.*

(i) $\|w^{k+1} - w^*\|_{G_k}^2 \le \|w^k - w^*\|_{G_k}^2 - (\|u^{k+1} - u^k\|_{D_k}^2 + \beta\|x^{k+1} - y^k\|^2)$.

(ii) *Suppose that sequence $\{T_k\}$ is bounded. Then, there exists a constant $\mu > 0$ such that for all $k \ge 0$, we have*

$$\|F^{k+1}\| \le \mu \left(\|u^{k+1} - u^k\|_{D_k}^2 + \|x^{k+1} - y^k\|^2\right).$$

Proof. The proofs of (i) and (ii) can be found in [60, Theorem 1] and [60, Lemma 3], respectively. ∎

We give some conditions for sequence $\{T_k\}$ that should be obeyed to guarantee the global convergence.

Condition 3.3.1. *For a sequence $\{T_k\}$ in framework (3.1.4), there exist $T \succeq 0$ and a sequence $\{\gamma_k\}$ such that*

(i) $T \preceq T_{k+1} \preceq (1+\gamma_k)T_k$ *for all* k,

(ii) $\sum_0^\infty \gamma_k < \infty$ *and* $\gamma_k \geq 0$ *for all* k.

From the definitions of $\{D_k\}$ and $\{G_k\}$ in (3.3.2), together with $T_k \succeq T \succeq 0, S \succeq 0$ and $\beta > 0$, it follows that the sequences $\{D_k\}$ and $\{G_k\}$ also satisfy $0 \preceq D \preceq D_{k+1} \preceq (1+\gamma_k)D_k$, and $0 \preceq \bar{G} \preceq G_{k+1} \preceq (1+\gamma_k)G_k$ for all k, where $D = \begin{pmatrix} T & 0 \\ 0 & S \end{pmatrix}$ and $\bar{G} = \begin{pmatrix} T & 0 & 0 \\ 0 & S+\beta I & 0 \\ 0 & 0 & \frac{1}{\beta}I \end{pmatrix}$, respectively.

We define two constants C_s and C_p as follows:

$$C_s: = \sum_{k=0}^{\infty} \gamma_k \quad \text{and} \quad C_p: = \prod_{k=0}^{\infty} (1+\gamma_k). \tag{3.3.4}$$

Condition 3.3.1 (ii) implies that $0 \leq C_s < \infty$ and $1 \leq C_p < \infty$. Moreover, we have $T \preceq T_k \preceq C_p T_0$ for all k, which means that the sequences $\{T_k\}$ and $\{D_k\}$ are bounded.

Under the conditions, we have the following convergence result.

Theorem 3.3.1. *Let $\{(x^k, y^k, \lambda^k)\}$ be generated by (3.1.4), and let $\{T_k\}$ be a sequence satisfying Condition 3.3.1. Then sequence $\{(x^k, y^k, \lambda^k)\}$ converges to a point $(x^*, y^*, \lambda^*) \in \Omega^*$.*

Proof. First we show that the sequence $\{w^k\}$ is bounded. Since $\bar{G} \preceq G_{k+1} \preceq (1+\gamma_k)G_k$, we have

$$\|w^{k+1} - w^*\|^2_{G_{k+1}} \leq (1+\gamma_k)\|w^{k+1} - w^*\|^2_{G_k}. \tag{3.3.5}$$

Combining the inequality (3.3.5) with Lemma 3.3.1 (i), we have

$$\begin{aligned} \|w^{k+1} - w^*\|^2_{G_{k+1}} &\leq (1+\gamma_k)\|w^k - w^*\|^2_{G_k} - (1+\gamma_k)\left(\|u^{k+1} - u^k\|^2_{D_k} + \beta\|x^{k+1} - y^k\|^2\right) \\ &\leq (1+\gamma_k)\|w^k - w^*\|^2_{G_k} - c_1\left(\|u^{k+1} - u^k\|^2_{D_k} + \|x^{k+1} - y^k\|^2\right), \end{aligned} \tag{3.3.6}$$

where $c_1 = \min\{1, \beta\}$. It then follows that we have for all k,

$$\|w^{k+1} - w^*\|^2_{G_{k+1}} \leq (1+\gamma_k)\|w^k - w^*\|^2_{G_k} \leq \cdots \leq \left(\prod_{i=0}^{k}(1+\gamma_i)\right)\|w^0 - w^*\|^2_{G_0} \leq C_p\|w^0 - w^*\|^2_{G_0}. \tag{3.3.7}$$

Note that $\|w^{k+1} - w^*\|^2_{G_{k+1}} = \|x^{k+1} - x^*\|^2_{T_{k+1}} + \|y^{k+1} - y^*\|^2_{S+\beta I} + \frac{1}{\beta}\|\lambda^{k+1} - \lambda^*\|^2$, $S + \beta I$ is positive definite, and $C_p\|w^0 - w^*\|^2_{G_0}$ is a constant. It then follows from (3.3.7) that $\{y^k\}$ and $\{\lambda^k\}$ are bounded. We now show that $\{x^k\}$ is also bounded.

From (3.3.6) and (3.3.7), we have

$$c_1\left(\|u^{k+1}-u^k\|_{D_k}^2+\|x^{k+1}-y^k\|^2\right)\le \|w^k-w^*\|_{G_k}^2-\|w^{k+1}-w^*\|_{G_{k+1}}^2+\gamma_k C_p\|w^0-w^*\|_{G_0}^2.$$

Summing up the inequalities, we obtain

$$\begin{aligned}
&\sum_{k=0}^{\infty} c_1\left(\|u^{k+1}-u^k\|_{D_k}^2+\|x^{k+1}-y^k\|^2\right)\\
&\le \|w^0-w^*\|_{G_0}^2-\|w^{k+1}-w^*\|_{G_{k+1}}^2+\left(\sum_{k=0}^{\infty}\gamma_k\right)C_p\|w^0-w^*\|_{G_0}^2\\
&\le (1+C_sC_p)\|w^0-w^*\|_{G_0}^2.
\end{aligned}$$

Since $(1+C_sC_p)\|w^0-w^*\|_{G_0}^2$ is a finite constant, we have

$$\lim_{k\to\infty}\left(\|u^{k+1}-u^k\|_{D_k}^2+\|x^{k+1}-y^k\|^2\right)=0, \tag{3.3.8}$$

which indicates that

$$\lim_{k\to\infty}\|x^{k+1}-y^k\|=0. \tag{3.3.9}$$

Note that $x^*=y^*$ and $\|x^{k+1}-x^*\|=\|x^{k+1}-y^k+y^k-y^*\|\le\|x^{k+1}-y^k\|+\|y^k-y^*\|$. It then follows from (3.3.9) that $\{x^k\}$ is bounded. Consequently, the sequence $\{w^k\}$ is bounded.

Next we show that any cluster point of the sequence $\{w^k\}$ is a KKT point of (3.1.2). Since the sequence $\{w^k\}$ is bounded, it has at least one cluster point in Ω. Let $w^\infty=(x^\infty,y^\infty,\lambda^\infty)\in\Omega$ be a cluster point of $\{w^k\}$, and let $\{w^{k_j}\}$ be a subsequence of $\{w^k\}$ that converges to point w^∞.

From (3.3.8) and Lemma 3.3.1 (ii), we have $\lim_{j\to\infty}\|F^{k_j}\|=0$. It then follows from the definition of F^k that $x^\infty=y^\infty$. Moreover, since ∂f and ∂g are upper semi-continuous, there exists η_f^∞ and η_g^∞ such that $\eta_f^\infty\in\partial f(x^\infty)$, $\eta_g^\infty\in\partial g(y^\infty)$, $\eta_f^{k_j}\to\eta_f^\infty$ and $\eta_g^{k_j}\to\eta_g^\infty$, taking a subsequence if necessary. It then follows from $\lim_{j\to\infty}\|F^{k_j}\|=0$ that $\eta_f^\infty-\lambda^\infty=0$ and $\eta_g^\infty+\lambda^\infty=0$. Consequently w^∞ satisfies the KKT conditions of problem (3.1.2).

Finally, we show that the whole sequence $\{w^k\}$ converges to w^∞.

Since $\{w^{k_j}\}$ converges to w^∞, for any positive scalar ϵ, there exists positive integer q such that

$$\|w^{k_q}-w^\infty\|_{G_{k_q}}<\frac{\epsilon}{C_p^{\frac{1}{2}}}, \tag{3.3.10}$$

Note that (3.3.7) holds for an arbitrary KKT point w^* of problem (3.1.2). It then follows from (3.3.7) with $w^*=w^\infty$ that for any $k\ge k_q$, we have

$$\|w^k-w^\infty\|_{G_k}\le\left(\prod_{i=k_q}^{k-1}(1+\gamma_i)\right)^{1/2}\|w^{k_q}-w^\infty\|_{G_{k_q}}\le\left(\prod_{i=0}^{\infty}(1+\gamma_i)\right)^{1/2}\|w^{k_q}-w^\infty\|_{G_{k_q}}<\epsilon,$$

where the second inequality follows from Condition 3.3.1 (ii), and the last inequality follows from (3.3.10) and the definition of C_p. Since ϵ is an arbitrary positive scalar, this shows that $\{w^k\}$ converges to w^∞. ■

3.3.2 Convergence of variable metric semi-proximal ADMM with the BFGS update

Now we give the global convergence of ADM-BFGS as a consequence of Theorem 3.3.1.

Theorem 3.3.2. *Suppose that sequence $\{T_k\}$ is generated by the BFGS (or L-BFGS) update with M. Suppose also that $\{T_k\}$ satisfies Condition 3.3.1. Then sequence $\{(x^k, y^k, \lambda^k)\}$ generated by ADM-BFGS converges to a point $(x^*, y^*, \lambda^*) \in \Omega^*$.*

Proof. The theorem directly follows from Theorem 3.3.1. ■

Currently, we cannot show that $\{T_k\}$ satisfies Condition 3.3.1 when $\{H_k\}$ is updated by a pure BFGS (or L-BFGS) update and $\bar{k} = \infty$. Hence we give the following two remedies for T_k to be satisfied Condition 3.3.1.

Remedy 1: Let $\bar{k}$ be finite, and the updating of B_k to be stopped at $\bar{k}$, that is

$$B_k = B_{\bar{k}},\ T_k = T_{\bar{k}} \quad \text{for all} \quad k \geq \bar{k},$$

i.e., $\gamma_k = 0$ in Condition 3.3.1 when $k \geq \bar{k}$. Thus, it is reasonable to say that the sequence $\{T_k\}$ generated by ADM-BFGS and some existing $\{\gamma_k\}$ satisfy the Condition 3.3.1. Note that the resulting ADM-BFGS becomes ADMM (3.1.3) with $T = T_{\bar{k}}$ for large k.

Remedy 2: Suppose that $\bar{k} = \infty$. We generate $\{B_k\}$ as follows:

$$B_{k+1} = B_k + c_k \left(\frac{\tilde{l}_k \tilde{l}_k^\top}{\tilde{l}_k^\top s_k} - \frac{B_k s_k s_k^\top B_k^\top}{s_k^\top B_k s_k} \right), \tag{3.3.11}$$

where $\tilde{l}_k = M s_k + \delta s_k$ with $\delta > 0$, and $\{c_k\}$ is a sequence such that $c_k \in [0, 1]$, and $\sum_{k=0}^{\infty} c_k < \infty$.

Now we show that Condition 3.3.1 holds when $B_0 \succeq M + \delta I$. Suppose that $B_0 \succeq M + \delta I$. Note that $B_{k+1} = B_k + c_k(\bar{B}_{k+1} - B_k)$, where $\bar{B}_{k+1}$ is updated by the pure BFGS update (3.2.1) with s_k and $\tilde{l}_k$. From Theorem 3.2.1, $\bar{B}_{k+1} \succeq M + \delta I$ when $B_k \succeq M + \delta I$. Since $B_{k+1} = c_k \bar{B}_{k+1} + (1 - c_k) B_k$, we have $B_{k+1} \succeq M + \delta I$, and hence $T_{k+1} = B_{k+1} - M \succeq \delta I \succ 0$. Therefore the first matrix inequality in Condition 3.3.1 (i) holds.

Next we show the second inequality in Condition 3.3.1 (i) holds. Note that $s_k^\top B_k s_k \geq \delta\|s_k\|^2$, $\tilde{l}_k^\top s_k = s_k^\top M s_k + \delta\|s_k\|^2 \geq \delta\|s_k\|^2$, and M is the constant matrix. Thus we have

$$\|\bar{B}_{k+1} - B_k\| \leq \left\|\frac{\tilde{l}_k\tilde{l}_k^\top}{\tilde{l}_k^\top s_k}\right\| + \left\|\frac{B_k s_k s_k^\top B_k}{s_k^\top B_k s_k}\right\| \leq \frac{\|(M+\delta I)s_k s_k^\top (M+\delta I)\|}{\delta\|s_k\|^2} + \frac{\|B_k s_k s_k^\top B_k\|}{\delta\|s_k\|^2}.$$

The first term in the right-hand side is bounded. The second term is also bounded since B_k is bounded which could be seen in [94, Theorem 6.3]. Therefore, $\|\bar{B}_{k+1} - B_k\|$ is bounded above by some $Q > 0$, that is, $\|\bar{B}_{k+1} - B_k\| \leq Q$. Then we have

$$c_k(\bar{B}_{k+1} - B_k) \preceq c_k\|\bar{B}_{k+1} - B_k\| \cdot I \preceq c_k\frac{Q}{\delta} \cdot \delta I \preceq c_k\frac{Q}{\delta}T_k.$$

Therefore,

$$\begin{aligned} T_{k+1} &= B_{k+1} - M \\ &= B_k + c_k(\bar{B}_{k+1} - B_k) - M \\ &= T_k + c_k(\bar{B}_{k+1} - B_k) \\ &\preceq T_k + \frac{c_k Q}{\delta}T_k \\ &= (1 + \frac{c_k Q}{\delta})T_k. \end{aligned}$$

Let $\gamma_k = \frac{Q}{\delta}c_k$. Then $T_{k+1} \preceq (1+\gamma_k)T_k$.

Finally we show that Condition 3.3.1 (ii) holds. From the definition of γ_k, we have $\sum_{k=0}^{\infty}\gamma_k = \frac{Q}{\delta}\sum_{k=0}^{\infty}c_k < \infty$.

We will present numerical results for the BFGS with Remedy 2 and L-BFGS with Remedy 1 in the next section.

3.4 Numerical results

In this section, we demonstrate the potential efficiency of our method by some numerical experiments. All the experiments are implemented by Matlab R2018b on Windows 10 pro with a 2.10 GHz Intel Xeon E5-2620 v4 processor and 128 GB of RAM.

3.4.1 Detail settings in the numerical experiments

In this subsection, we give the settings used in the numerical experiments.

Test problems

We consider to solve the following Lasso problem:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|Ax-b\|_2^2 + \tau\|x\|_1, \tag{3.4.1}$$

where

- $A \in \mathbb{R}^{m\times n}$ is a given data matrix;
- $x \in \mathbb{R}^n$ is a vector of feature coefficients to be estimated;
- $b \in \mathbb{R}^m$ is an observation vector and $\tau \in \mathbb{R}$ is a positive regularization parameter;
- m is the number of data points, and n is the number of features.

By introducing an auxiliary variable $y \in \mathbb{R}^n$, we reformulate problem (3.4.1) as follows:

$$\min_{x\in\mathbb{R}^n,\ y\in\mathbb{R}^n} \frac{1}{2}\|Ax-b\|_2^2 + \tau\|y\|_1 \quad \text{s.t.} \quad x-y=0. \tag{3.4.2}$$

We randomly generate A and b as follows. We first randomly select $\bar{x} \in \mathbb{R}^n$ with the sparsity s, i.e., the number of nonzero elements in $\bar{x}$ over n is s. The Matlab code is given as

```
xbar = sprandn(n,1,s).
```

We generate A by the standard normal $\mathcal{N}(0,1)$ distribution whose sparsity density is p:

```
A = sprandn(m,n,p). % N(0,1) with the density p
```

Then we calculate $b = A\bar{x} + \varrho$, where ϱ is a noise under an $\mathcal{N}(0,10^{-3})$ distribution. The Matlab code is

```
b = A*xbar + sqrt(0.001)*randn(m,1).
```

The regularization parameter is set to $\tau = 0.1\tau_{max}$, where $\tau_{max} = \|A^\top b\|_\infty$:

```
tau = 0.1* norm(A'*b, 'inf').
```

Test ADMMs

In the numerical experiments, we test the following seven different versions of the ADMM. The differences of the versions of ADMM are the choices of the proximal term T_k in x-subproblems.

ADM-OPT: the classical ADMM [46, 49]. ADM-OPT solves the original subproblem exactly, that is, $T_k = 0$ for all k;

ADM-SPRO: the semi-proximal ADMM in [40]. A positive semidefinite matrix T_k is chosen as

$$T_k = \xi I - \beta I - A^\top A \text{ with } \xi = \kappa_1 * \lambda_{\max}\left(\beta I + A^\top A\right), \quad \kappa_1 > 1, \quad \text{for all } \ k; \tag{3.4.3}$$

ADM-IPRO: the indefinite proximal ADMM based on [77] and [63]. An indefinite proximal matrix T_k is chosen as

$$T_k = \xi I - A^\top A \text{ with } \xi = \kappa_2 * \lambda_{\max}\left(A^\top A\right), \quad \kappa_2 > 0.75, \quad \text{for all } \ k; \tag{3.4.4}$$

ADM-BFGS: the proximal ADMM with the BFGS update with $\bar{k} = \infty$, which is not guaranteed to converge theoretically. An initial matrix of B_k (or H_k) is given as

$$B_0 = \xi I, \quad \xi = \kappa_3 * \lambda_{\max}(\beta I + A^\top A), \quad \kappa_3 > 0.75; \tag{3.4.5}$$

ADM-LBFGS: the proximal ADMM with the L-BFGS update with $\bar{k} = \infty$. The initial semidefinite proximal matrix for the limited memory BFGS is the same as (3.4.5). Note that $H_0 = \frac{1}{\xi} I$. We fix $H_0^k = H_0$ in each updating step for the L-BFGS matrix;

ADM-BFGS-R: the ADMM-BFGS with Remedy 2 given in Subsection 3.3.2;

ADM-LBFGS-R: the proximal ADMM with the L-BFGS update with Remedy 1, that is ADM-LBFGS with $\bar{k} < \infty$.

These ADMMs except for ADM-OPT require the maximum eigenvalues $\lambda_{\max}\left(\beta I + A^\top A\right)$ and $\lambda_{\max}\left(A^\top A\right)$. We adopt the following Matlab codes to compute these eigenvalues:

```
eig_max = svds(A,1)^2 + beta;
eig_max = svds(A,1)^2.
```

Whereas ADM-OPT must solve the unconstrained quadratic optimization. We use a Cholesky factorization to solve it. When $m = n$, we use "chol" in Matlab for $(A^\top A + \beta I)$. When A is fat (i.e., $m < n$), we apply the Sherman-Morrison formula to $(\beta I + A^\top A)^{-1}$ as

$$(\beta I + A^\top A)^{-1} = \frac{1}{\beta} I - \frac{1}{\beta^2} \cdot A^\top \cdot \left(I + \frac{1}{\beta} A A^\top\right)^{-1} \cdot A,$$

and compute the factorization $LL^\top$ of a smaller matrix $(I + (1/\beta)AA^\top)$ by the "chol" function. Then the x-subproblems are solved as

```
q = A'*b + lambda + beta*y;
x = q/beta - (A'*(L' \ ( L \ (A*q) )))/beta^2.
```

Note that the Cholesky factorization of $(A^\top A + \beta I)$ or $(I + (1/\beta)AA^\top)$ is calculated only once for each test problem.

Other setting and notations

Stopping criterion: We adopt the same stopping criterion as in [13] for all the numerical experiments, that is, if the primal and dual residuals r^k and σ^k satisfy

$$\|r^k\|_2 \le \epsilon_k^{\mathrm{pri}} \quad \text{and} \quad \|\sigma^k\|_2 \le \epsilon_k^{\mathrm{dual}}, \tag{3.4.6}$$

then we stop the algorithms, where $r^k = x^k - y^k$, $\sigma^k = -\beta(y^k - y^{k-1})$, and $\epsilon^{\mathrm{pri}} > 0$ and $\epsilon^{\mathrm{dual}} > 0$ are feasibility tolerances for the primal and dual feasibility conditions, respectively. These tolerances can be selected using an absolute and relative criterion from the suggestion in [13], such as

$$\begin{aligned}
\epsilon_k^{\mathrm{pri}} &= \sqrt{n}\epsilon^{\mathrm{abs}} + \epsilon^{\mathrm{rel}}\max\{\|x^k\|_2, \|-y^k\|_2\}, \\
\epsilon_k^{\mathrm{dual}} &= \sqrt{n}\epsilon^{\mathrm{abs}} + \epsilon^{\mathrm{rel}}\|\lambda^k\|_2,
\end{aligned}$$

where $\epsilon^{\mathrm{abs}} > 0$ is an absolute tolerance and $\epsilon^{\mathrm{rel}} > 0$ is a relative tolerance.

The stopping criteria are set to $\epsilon^{\mathrm{abs}} = 10^{-4}$ and $\epsilon^{\mathrm{rel}} = 10^{-3}$ in all the experiments.

Other setting: We always select $S = 0$ for (3.1.4b). We set the initial points as $x^0 = y^0 = 0$ and $\lambda^0 = 0$. The maximum iterations are set to be 20000 in all randomly generated experiments.

Notations in tables for numerical results:

- Iter.: the iteration steps for each algorithm;
- Time: the total CPU time for each algorithm;
- T-L: the CPU time for the Cholesky factorization and the calculation of $AA^\top$ or $A^\top A$;
- T-ME: the CPU time for computing for the maximum eigenvalue;
- T-A: the CPU time for the algorithm proceed without T-L or T-ME;
- T-QN: the CPU time for BFGS update (matrix H_k) of ADM-BFGS.

All of the CPU times are recorded in seconds.

3.4.2 Test I: ADMM with the BFGS update

In the subsection, we first compare four different methods: ADM-OPT, ADM-SPRO with $\kappa_1 = 1.01$, ADM-IPRO with $\kappa_2 = 0.8$, and ADM-BFGS with $\kappa_3 = 1.01$. We also present numerical results for the ADMM with Remedy 2 given in Subsection 3.3.2 for the global convergence.

We solve problem (3.4.2) with $n = 2000$, $m = 1000$, $s = 0.1$ and $p \in \{0.1, 0.5\}$. All of the other settings and calculations are based on Subsection 3.4.1. We solve 10 problems in each test, and Table 3.1 shows the average of iterative steps and CPU time.

Table 3.1: Comparison on iteration steps and CPU time among the methods

Problem				β	ADM-OPT		ADM-SPRO		ADM-IPRO		ADM-BFGS		
n	m	s	p		Iter.	Time(s)	Iter.	Time(s)	Iter.	Time(s)	Iter.	Time(s)	T-QN(s)
2000	1000	0.1	0.1	100	20.5	0.21	64.3	0.15	54.3	0.14	38.4	3.64	2.89
2000	1000	0.1	0.5	100	63.1	0.67	197.9	0.45	160.0	0.41	71.4	7.35	5.49
2000	1000	0.1	0.5	500	20.9	0.55	68.5	0.32	58.4	0.31	37.3	4.50	2.84

From the table, it is obvious to see that the classical ADMM finds solutions within the least iterative steps, while the indefinite proximal ADMM admits the faster one at the CPU time. The ADMM with BFGS can get solutions with relatively fewer iterations. However, it spends much time to compute the H_k as indicated in the T-QN column. When data matrix A is ill-condition or it is impossible to compute the inverse of the Hessian matrix of augmented Lagrangian function, it is meaningful to use the matrix H_k since it can yield a solution with fewer iterative steps.

Next, we present numerical results for the iteration steps of ADM-BFGS-R in Table 3.2. We update c_k by $c_k = \zeta^k$ with $\zeta \in [0, 1]$, and chose a positive $\delta \in \{100,\ 1\text{e-}5\}$. We solve problem (3.4.2) using the same settings as those shown in Table 3.1, that is, $n = 2000$, $m = 1000$, $s = 0.1$ and $p \in \{0.1, 0.5\}$. The results are compared with those of the ADM-BFGS.

Table 3.2: Results for Remedy 2 with different δ and c_k ($c_k = \zeta^k$)

Problem				β	ADM-BFGS	ADM-BFGS-R $\delta = 100$			ADM-BFGS-R $\delta = 1\text{e-}5$		
						$\zeta = 0.1$	$\zeta = 0.5$	$\zeta = 0.99$	$\zeta = 0.1$	$\zeta = 0.5$	$\zeta = 0.99$
n	m	s	p		Iter.	Iter.	Iter.	Iter.	Iter.	Iter.	Iter.
2000	1000	0.1	0.1	100	38.4	75.7	71.3	43.9	67.6	63.0	38.8
2000	1000	0.1	0.5	100	71.4	204.4	198.6	74.7	197.0	190.4	71.8
2000	1000	0.1	0.5	500	37.3	74.0	68.7	39.4	71.9	66.6	37.8

Table 3.2 shows that for each $\delta > 0$ and $\zeta \in [0, 1]$, ADM-BFGS-R can find a solution. When δ is approximately 0 and ζ is close to 1, the iterative steps of ADM-BFGS-R approach those of ADM-BFGS .

3.4.3 Test II: ADMM with limited memory BFGS update

In this subsection, we test how the ADMM with limited memory BFGS (ADM-LBFGS) works.

We set the number h of vectors stored in L-BFGS to be 10. The comparisons are among the ADM-OPT, ADM-IPRO, and the proposed ADM-LBFGS. We consider large scale problems with $n = 10000$ and $s = 0.1$.

Behaviors of ADMMs for different β

First, we analyzed the behaviors of ADMMs for different β. We solve problem (3.4.2) with $m \in \{1000, 5000, 10000\}$ and $p \in \{0.1, 0.5, 1\}$. We take $\kappa_2 = 0.8$ for ADM-IPRO (3.4.4) and $\kappa_3 = 1.01$ for ADM-LBFGS (3.4.5). The other settings used in the test problems are given in Subsection 3.4.

The results of the iteration steps and CPU time (seconds) averaged over 10 random trials are shown in Table 3.3.

From Table 3.3, we can observe that the ADMM with L-BFGS performs well for different β. In each case, ADM-LBFGS can obtain solutions within the same amount of CPU time for the algorithm proceed (T-A) as that of the classical ADMM (ADM-OPT). The ADM-OPT appears to be the best method to find a solution with the least iterations and CPU time when the size $m = 1000$ and sparsity $p = 0.1$. However, it becomes slower due to the CPU time for Cholesky factorization (T-L) when $p = 0.5$ and 1. Note that the T-L requires more time compared with the computations of the maximum eigenvalue (T-ME) for ADM-LBFGS and ADM-IPRO when the size m of matrix A is larger than 1000, especially when A is less sparse with $p = 0.5$ and 1. Compared with ADM-OPT when $m = 5000$, ADM-LBFGS can reduce the CPU time to approximately 50% for the sparse case of $p = 0.1$, and about 80% for the hard cases, where $p = 0.5$ and 1. Besides, for a large and dense matrix A with $m = 10000$ and $p = 1$, ADM-LBFGS can reduce the CPU time by 93% as compared to ADM-OPT. On the other hand, ADM-LBFGS is a bit faster than ADM-IPRO when ξ is selected appropriately with the maximum eigenvalue.

Behaviors of ADM-IPRO and ADM-LBFGS for some different ξ

In the above experiments, we have chosen $\xi = 0.8 * \lambda_{\max}\left(A^\top A\right)$ for the indefinite proximal term and $\xi = 1.01 * \lambda_{\max}(\beta I + A^\top A)$ for the semidefinite proximal term. This is unrealistic for some large scale applications where the calculation of the maximum eigenvalue is expensive. Next we test the behaviours of ADM-LBFGS and proximal ADMM (ADM-IPRO) with different κ_2 and κ_3.

We solve problem (3.4.2) with $m = 5000$ and $p \in \{0.5, 1\}$. Since the results in Table 3.3

Table 3.3: Comparison among ADMMs for different β

Size	β	ADM-OPT				ADM-IPRO				ADM-LBFGS			
		Iter.	Time(s)	T-A(s)	T-L(s)	Iter.	Time(s)	T-A(s)	T-ME(s)	Iter.	Time(s)	T-A(s)	T-ME(s)
m=1000	50	90.9	0.66	0.34	0.32	280.1	0.92	0.46	0.46	247.3	1.27	0.81	0.46
p=0.1	100	59.1	0.56	0.22	0.34	175.7	0.74	0.29	0.45	182.2	1.04	0.59	0.45
	150	67.7	0.57	0.25	0.32	197.9	0.77	0.33	0.44	157.3	0.95	0.51	0.44
	200	88.4	0.67	0.33	0.34	240.2	0.82	0.38	0.44	147.4	0.90	0.46	0.44
m=1000	200	93.3	4.63	0.99	3.64	288.6	4.40	2.46	1.94	247.5	4.57	2.63	1.94
p=0.5	300	69.9	4.38	0.73	3.65	218.8	3.79	1.83	1.96	209.0	4.15	2.19	1.96
	500	60.3	4.37	0.62	3.75	179.2	3.39	1.49	1.90	174.2	3.72	1.82	1.90
	800	85.0	4.29	0.87	3.42	240.6	3.91	1.98	1.93	143.4	3.42	1.49	1.93
m=1000	200	131.7	9.73	2.01	7.72	428.2	9.02	5.84	3.18	325.9	8.29	5.11	3.18
p=1	300	97.3	9.13	1.53	7.60	305.4	7.43	4.24	3.19	273.5	7.56	4.37	3.19
	500	67.8	8.97	1.05	7.92	213.0	6.05	2.85	3.20	220.9	6.66	3.46	3.20
	800	67.5	9.17	1.05	8.12	195.9	5.85	2.65	3.20	183.9	6.08	2.88	3.20
m=5000	100	80.4	15.67	4.48	11.19	176.6	7.24	2.29	4.95	86.6	6.25	1.30	4.95
p=0.1	200	41.2	13.90	2.34	11.56	103.1	6.24	1.32	4.92	55.9	5.75	0.83	4.92
	500	20.4	12.38	1.14	11.24	51.2	5.62	0.64	4.98	38.0	5.55	0.57	4.98
	800	22.0	12.87	1.27	11.60	61.0	5.81	0.78	5.03	37.2	5.58	0.55	5.03
m=5000	500	67.1	95.46	6.36	89.10	151.3	24.45	7.54	16.91	75.6	20.84	3.93	16.91
p=0.5	1000	34.3	92.83	3.02	89.81	86.7	21.00	4.09	16.91	50.2	19.44	2.53	16.91
	2000	20.4	93.28	1.79	91.49	51.4	19.91	2.36	17.55	38.1	19.45	1.90	17.55
	2500	20.6	91.92	1.88	90.04	53.6	19.94	2.61	17.33	36.0	19.13	1.80	17.33
m=5000	1000	52.9	201.69	6.16	195.53	130.8	33.67	10.00	23.67	65.6	28.82	5.15	23.67
p=1	2000	27.2	200.39	3.07	197.32	73.0	29.23	5.39	23.84	44.9	27.27	3.43	23.84
	3000	20.3	210.74	2.45	208.29	55.0	31.01	4.16	26.85	39.3	29.98	3.13	26.85
	3200	20.7	208.26	2.54	205.72	53.2	30.78	4.08	26.70	38.8	29.81	3.11	26.70
m=10000	200	59.2	59.57	10.72	48.85	100.3	15.90	3.02	12.88	60.2	14.85	1.97	12.88
p=0.1	500	24.5	51.97	4.75	47.22	48.0	14.39	1.50	12.89	30.6	13.93	1.04	12.89
	1000	15.9	51.13	3.06	48.07	30.0	13.82	0.95	12.87	23.7	13.66	0.79	12.87
	1500	16.9	50.61	3.19	47.42	34.4	13.88	1.08	12.80	24.4	13.62	0.82	12.80
m=10000	1000	49.8	426.04	9.69	416.35	88.2	52.69	9.42	43.27	50.2	48.77	5.50	43.27
p=0.5	2000	25.7	413.19	5.20	407.99	49.0	48.17	5.40	42.77	31.2	46.26	3.49	42.77
	3000	17.6	432.36	3.28	429.08	38.6	47.64	4.26	43.38	26.0	46.29	2.91	43.38
	3500	15.9	408.71	2.86	405.85	34.4	45.14	3.31	41.83	25.0	44.37	2.54	41.83
m=10000	2000	40.8	983.12	6.91	976.21	72.0	67.15	10.49	56.66	42.6	62.94	6.28	56.66
p=1	3000	27.6	948.20	4.68	943.52	52.2	64.34	7.60	56.74	34.0	61.74	5.00	56.74
	5000	17.4	947.22	3.06	944.16	36.4	62.45	5.34	57.11	26.0	60.97	3.86	57.11
	5500	16.4	931.24	2.87	928.37	33.8	61.88	4.96	56.92	25.0	60.62	3.70	56.92

for $m = 5000$ indicate that a reasonable β is around 2000, we take $\beta \in \{1000, 2000, 3000\}$ in this experiments. We also take $\kappa_2, \kappa_3 \in \{0.75, 0.8, 1.01, 5.0, 10.0, 100\}$ in (3.4.4) and (3.4.5).

The other settings and notations used are shown in Subsection 3.4. Table 3.4 shows the results of iteration steps and CPU time (seconds) averaged over 10 random trials for every κ_2 and κ_3.

Table 3.4: Different κ_2 and κ_3 for proximal ADMM

Setting	κ_2, κ_3	ADM-IPRO			ADM-LBFGS		
		Iter.	T-A(s)	Time(s)	Iter.	T-A(s)	Time(s)
$p = 0.5$	0.75	74.6	3.40	20.20	46.4	2.21	19.01
$\beta = 1000$	0.80	80.3	3.67	20.47	47.3	2.26	19.06
T-ME = 16.80s	1.01	99.6	4.56	21.36	49.7	2.42	19.22
	5.0	400.6	18.44	35.24	102.9	5.01	21.81
	10.0	734.2	33.57	50.37	147.7	7.02	23.82
	100.0	4626.3	211.53	228.33	330.7	15.91	32.71
$p = 0.5$	0.75	42.0	1.95	18.30	33.6	1.62	17.97
$\beta = 2000$	0.80	44.3	2.06	18.41	34.7	1.68	18.03
T-ME = 16.35s	1.01	53.8	2.51	18.86	38.2	1.85	18.20
	5.0	213.5	9.85	26.20	100.4	4.89	21.24
	10.0	376.6	17.39	33.74	137.2	6.63	22.98
	100.0	2324.4	107.59	123.94	327.7	16.02	32.37
$p = 1$	0.75	62.0	4.38	30.38	41.0	3.00	29.00
$\beta = 2000$	0.80	64.9	4.58	30.58	41.5	3.02	29.02
T-ME = 26.00s	1.01	81.8	5.78	31.78	44.6	3.22	29.22
	5.0	334.7	23.59	49.59	105.3	7.71	33.71
	10.0	593.8	41.76	67.76	145.1	10.67	36.67
	100.0	3623.6	254.90	280.90	344.2	25.06	51.06
$p = 1$	0.75	43.8	3.13	27.29	34.5	2.49	26.65
$\beta = 3000$	0.80	46.0	3.28	27.44	35.5	2.59	26.75
T-ME = 24.16s	1.01	56.5	4.02	28.18	39.1	2.85	27.01
	5.0	222.6	15.61	39.77	104.7	7.58	31.74
	10.0	405.5	28.40	52.56	139.8	10.11	34.27
	100.0	2467.4	172.53	196.69	359.7	26.17	50.33

As shown in Table 3.4, the ADM-LBFGS always works well and remains stable. Note that ADM-LBFGS is slightly faster than ADM-IPRO when ξ is chosen nearly around the maximum eigenvalue, $\kappa_2, \kappa_3 = 0.75, 0.80, 1.01$ for instance. On average, this can lead to a 30% reduction in the number of iterations. There are no much differences in the CPU time because T-ME counts for a lot. When $\kappa_2, \kappa_3 = 100$ which are chosen far away from maximum eigenvalue, ADM-LBFGS can always bring out a 85-90% improvement in the number of iterations and a 75-85% improvement in the CPU time compared with the ADM-IPRO. Moreover, we find that ADM-LBFGS also works well even when the proximal term is a slightly indefinite matrix, i.e., $\kappa_3 < 1$.

Remedy 1: ADM-LBFGS stops updating of H_k for some finite $\bar{k}$

Finally, we investigate the behavior of ADM-LBFGS-R with various $\bar{k}$ when the updating of H_k stops.

We solve problem (3.4.2) with $m = 5000$, $p \in \{0.5, 1\}$, $\beta \in \{1000, 2000\}$, and set $\bar{k} = \{5, 10, 20, 40, 50, 100\}$ and $\kappa_3 = 1.01$ in (3.4.5). All the other settings are the same as those used in the above-mentioned experiments. The results of CPU time and iterations of different stopping $\bar{k}$ averaged over 10 random trials are provided in Table 3.5.

Table 3.5: Results for stopping at different $\bar{k}$

	$\bar{k}$	ADM-LBFGS-R		
		Iter.	T-A(s)	Time(s)
$p = 0.5$	5	106.0	4.89	19.25
$\beta = 1000$	10	93.2	4.35	18.51
T-ME = 14.16s	20	76.8	3.63	17.79
	40	54.2	2.56	16.72
	50	50.2	2.43	16.59
	100	50.1	2.42	16.58
$p = 1$	5	85.7	6.06	32.76
$\beta = 2000$	10	76.1	5.37	32.07
T-ME = 26.70s	20	62.9	4.49	31.19
	40	43.8	3.15	29.85
	50	44.3	3.19	29.89
	100	44.3	3.22	29.92

The results above indicate that for all $\bar{k}$, the ADM-LBFGS-R can successfully obtain a solution within the maximum iteration. In particular, the results for $\bar{k} = 50$ and 100 are similar, indicating that $\bar{k}_{50}$ is a well-tuned proximal matrix for the test problems.

3.4.4 Test III: Numerical experiments for real-world datasets

In this subsection, we investigate the behaviour of the proposed method for three real-world models "dbworld-bodies", "Madelon" and "sido0" which are selected from the literature [33, 81, 110], respectively. The main purpose of the experiments is to justify the feasibility of the proposed proximal ADMM with the BFGS update. We thus choose the existing proximal ADMMs (ADM-SPRO and ADM-IPRO) as the benchmark for a numerical comparison in this subsection.

We solve the Lasso problem (3.4.1) with "dbworld-bodies", "Madelon" and "sido0". Since the reasonable parameters in the methods are different for each real-world data, we should predefine them. We choose parameter β based on the sizes of the problems and considering the results in the previous subsections. The concrete values of β are shown in Table 3.7. Moreover, we set $\kappa_1 = 1.01$ for ADM-SPRO, $\kappa_2 = 0.8$ for ADM-IPRO, and $\kappa_3 = 1.01$ for ADM-BFGS and ADM-LBFGS.

Table 3.6 lists the sizes m and n, sources of the dataset, and some parameters used in the methods. Note that "dbworld-bodies" contains 64 instances and 4702 features, which means that the matrix A has more columns than rows. This dataset is useful for illustrating the behaviors of the various algorithms used for the sparse optimization, such as the Lasso problem (3.4.1). We exploit only the stopping criterion (3.4.6) for the dataset since all the algorithms successfully solve it. Dataset 'Madelon" consists of 2000 samples and 500 attributes, which implies that the matrix A has more rows than columns. The first-order method such as the ADMM usually takes a lot of iterations for such data. In our experiments, all the algorithms did not find a solution that satisfied the stopping criterion (3.4.6). Thus we set the maximum iteration to 10000 and observe the behaviors of the algorithms from Figure 2. Dataset "sido0" is a larger one with 12678 examples and 4932 variables. For the data, we set ε^{abs} and ε^{rel} to larger values in the stopping criterion.

Table 3.6: Summary of datasets and some parameters used in the experiments

Datasets	sizes		source	stopping criterion		maximum iterations
	m	n				
dbworld-bodies	64	4702	[33]	$\epsilon^{\text{abs}} = 10^{-4}$	$\epsilon^{\text{rel}} = 10^{-3}$	-
Madelon	2000	500	[33]	-	-	10000
sido0	12678	4932	[110]	$\epsilon^{\text{abs}} = 10^{-3}$	$\epsilon^{\text{rel}} = 10^{-2}$	20000

Table 3.7 shows the results averaged over 10 trials. Since datasets "dbworld-bodies" and "Madelon" are small, we omit T-L and T-M and present the total CPU time, "Time". Moreover, we omit to give the results of the ADM-BFGS for datasets "Madelon" and "sido0". The reason is that the ADM-BFGS takes much longer time due to the size of n.

Table 3.7: Results for real dataset

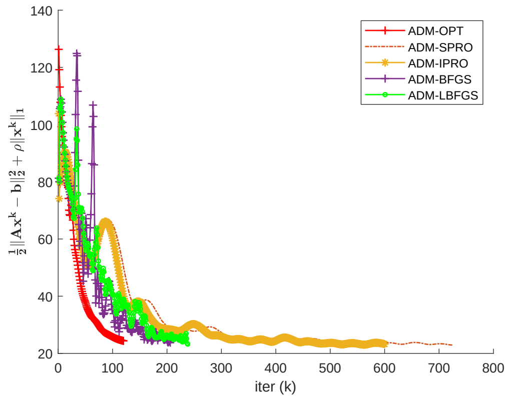
Datasets	β	ADM-OPT		ADM-SPRO		ADM-IPRO		ADM-BFGS		ADM-LBFGS	
		Iter.	Time(s)	Iter.	Time(s)	Iter.	Time(s)	Iter.	Time(s)	Iter.	Time(s)
dbworld-bodies	10	120.0	0.03	725.0	0.13	600.0	0.11	206.0	64.92	238.0	0.13
$m = 64$	50	51.0	0.01	191.0	0.05	167.0	0.05	143.0	45.21	51.0	0.04
$n = 4702$	95	92.0	0.02	167.0	0.05	169.0	0.05	174.0	55.01	99.0	0.05
	100	97.0	0.02	172.0	0.05	186.0	0.05	119.0	37.58	153.0	0.08

	β	Iter.	Time(s)	Iter.	Time(s)	Iter.	Time(s)	Iter.	Time(s)	Iter.	Time(s)
Madelon	100	10000	4.88	10000	3.03	10000	3.05	-	-	10000	3.80
$m = 2000$	150	10000	4.70	10000	3.02	10000	3.01	-	-	10000	3.50
$n = 500$	200	10000	4.67	10000	3.38	10000	3.26	-	-	10000	3.71
	500	10000	4.80	10000	3.28	10000	3.25	-	-	10000	3.75

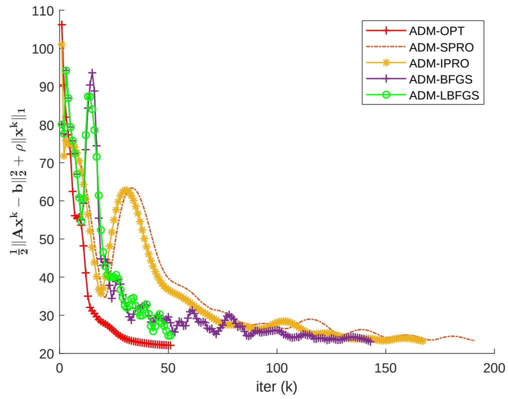
	β	Iter.	T-A(s)	Iter.	T-A(s)	Iter.	T-A(s)	Iter.	T-A(s)	Iter.	T-A(s)
sido0	100	1430.0	54.23	1638.0	26.44	1636.0	23.73	-	-	1435.0	23.32
$m = 12678$	1000	145.0	5.52	169.0	3.07	181.0	3.36	-	-	162.0	2.99
$n = 4932$	3000	50.0	1.62	59.0	0.75	56.0	0.70	-	-	55.0	0.72
T-L = 14.46s	5000	31.0	1.12	38.0	0.53	37.0	0.52	-	-	34.0	0.53
T-ME = 0.46s	6500	24.0	0.82	30.0	0.39	29.0	0.38	-	-	26.0	0.37

We also plot the objective function values with respect to the iterations for dataset "dbworld-bodies" based on five methods: ADM-OPT, ADM-SPRO, ADM-IPRO, ADM-BFGS, and ADM-LBFGS in Figure 3.1. Figure 3.2 shows the same graph for the "Madelon" dataset among four methods: ADM-OPT, ADM-SPRO, ADM-IPRO, and ADM-LBFGS.

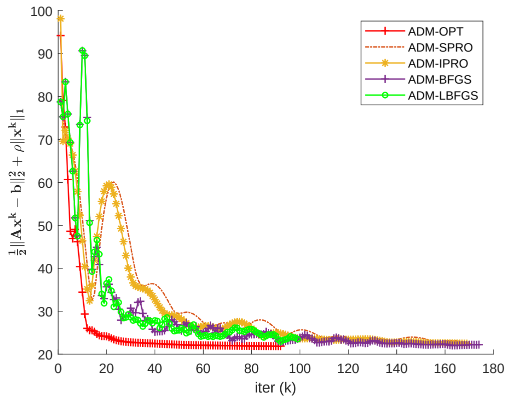
Overall, we see from Table 3.7 and Figure 3.1 that the ADM-OPT is the best method to find a solution with the least iterations and CPU time for the "dbworld-bodies" datasets. The proposed ADM-LBFGS outperforms the proximal ADMMs in terms of the iterations but does not have an improvement in computational time. For such a small dataset, the classical ADMM is the best choice. For the "Madelon" dataset, as we mentioned above, all of the methods cannot stop by the maximum iteration. Thus we compare the methods from Figure 3.2. The ADM-IPRO solves this problem with the least CPU time. The figures show that the graphs of ADM-LBFGS are close to those of ADM-OPT, and are better than those of the proximal ADMMs. This indicates that the proposed ADM-LBFGS can obtain a more accurate solution than the proximal ADMM. We also conclude from the results for "sido0" that the CPU times of the proximal methods are much shorter than 14.46 seconds of the computation of Cholesky factorization. The computational times of the proximal ADMMs are almost the same. At the same time, the numbers of iterative steps of ADM-LBFGS are close to those of the classical ADMM.



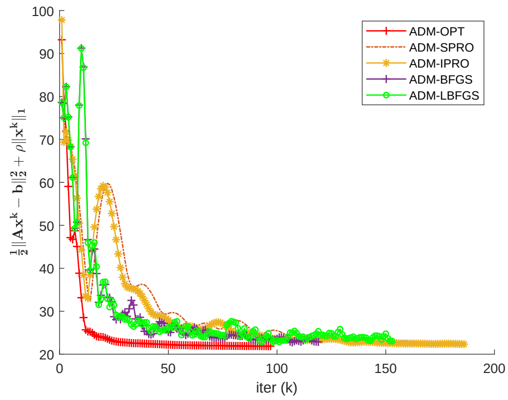
(a) $m = 64, n = 4702, \beta = 10$



(b) $m = 64, n = 4702, \beta = 50$

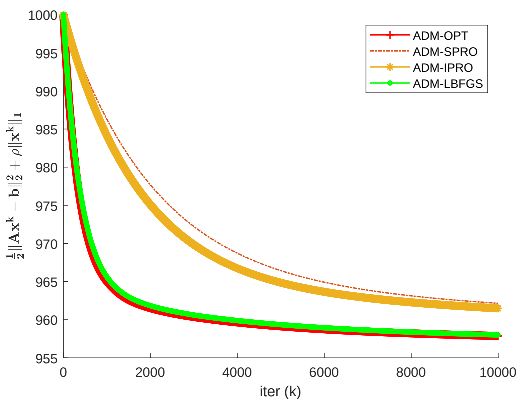


(c) $m = 64, n = 4702, \beta = 95$

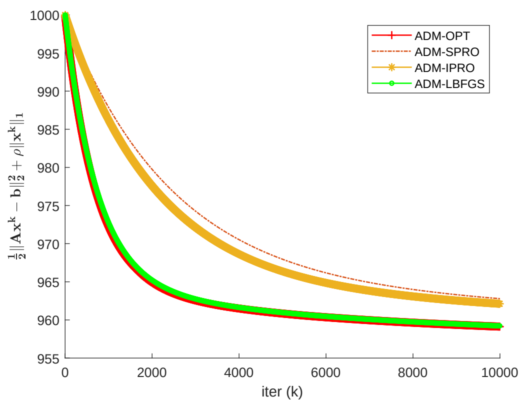


(d) $m = 64, n = 4702, \beta = 100$

Figure 3.1: Evolution of the objective function values with respect to iterations for "dbworld-bodies"



(a) $m = 2000, n = 500, \beta = 100$



(b) $m = 2000, n = 500, \beta = 200$

Figure 3.2: Evolution of the objective function values with respect to iterations for "Madelon"

3.4.5 Conclusions of the numerical experiments

Based on the numerical results above, we conclude the following:

1. For the problems with a well-condition or small size matrix A, all of the ADMMs can solve them efficiently. The classical ADMM (ADM-OPT) always performs the best;

2. For the problems whose maximum eigenvalues $\lambda_{\max}\left(A^\top A\right)$ are large, the classical ADMM can directly solve them while the proximal ADMM could be applied after normalizing the columns of matrix A;

3. Compared with the classical ADMM (ADM-OPT), the ADM-LBFGS is more suitable for dense large scale problems because the calculation of the inverse of $A^\top A$ is not necessary for ADM-LBFGS;

4. ADM-LBFGS can outperform the general proximal ADMM (ADM-SPRO or ADM-IPRO) in terms of iteration count, especially when the accurate estimation of the maximum eigenvalues is difficult. ADM-LBFGS could obtain a more precise solution than that of the proximal ADMM.

3.5 Conclusions

In this chapter, we have proposed a special proximal ADMM where the proximal matrix is derived from the BFGS update or limited memory BFGS method. We have given two remedies for the proximal matrix with the BFGS update to ensure the global convergence of the proposed method. Numerical results of several random problems with large scale data and several real-world datasets have been provided to illustrate the effectiveness of the proposed method.

Recall that Theorem 3.2.1 holds only when the Hessian matrix of the augmented Lagrangian function, that is, $M = \beta I + A^\top A$ is a constant matrix. We will consider more general problems using the ADMM with the BFGS update whose x-subproblems become an unconstrained quadratic programming problem, as presented in this chapter. Then we may apply Theorem 3.2.1 for global convergence. On the other hand, as shown in the numerical results, the ADMM with the L-BFGS also performs well with a slightly indefinite proximal matrix. This will facilitate the exploration for an indefinite proximal ADMM with the BFGS update.

Chapter 4

A Proximal Alternating Direction Method of Multipliers with the Broyden family for Convex Optimization Problems

4.1 Introduction

In this Chapter, we consider the general convex optimization problem:

$$\begin{aligned} &\text{minimize} && f(x) + g(y) \\ &\text{subject to} && Ax + By = b, \\ & && x \in \mathbb{R}^{n_1}, \quad y \in \mathbb{R}^{n_2}, \end{aligned} \tag{4.1.1}$$

where $f\colon \mathbb{R}^{n_1} \to \mathbb{R} \cup \{\infty\}$ and $g\colon \mathbb{R}^{n_2} \to \mathbb{R} \cup \{\infty\}$ are proper convex functions, $A \in \mathbb{R}^{m \times n_1}, B \in \mathbb{R}^{m \times n_2}$ and $b \in \mathbb{R}^m$. We define the augmented Lagrangian function for (4.1.1) as $\mathcal{L}_\beta\colon \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^m \to \mathbb{R}$:

$$\mathcal{L}_\beta(x, y, \lambda) := f(x) + g(y) - \langle \lambda, Ax + By - b \rangle + \frac{\beta}{2}\|Ax + By - b\|^2. \tag{4.1.2}$$

Here $\lambda \in \mathbb{R}^m$ are multipliers associated to the equality constraints and $\beta > 0$ is a penalty parameter.

When we apply ADMM for solving some concrete applications, we usually assume one of the subproblems is relatively easy to solve. Therefore, we suppose that y-subproblems in the proximal ADMM (3.1.3) are easily solved throughout this chapter. We allow the positive semidefinite matrix T in the proximal term (3.1.3a) to be changed at every step as that in the inexact ADMM [60]. Then T depends on k, and thus we denote it by T_k. The resulting ADMM is a variable metric semi-proximal ADMM. The iterative scheme of the variable metric semi-proximal ADMM (VMSP-

ADMM) algorithm is given as

$$\begin{cases} x^{k+1} = \arg\min_x \ \mathcal{L}_\beta(x, y^k, \lambda^k) + \dfrac{1}{2}\|x - x^k\|^2_{T_k}, & (4.1.3a) \\ y^{k+1} = \arg\min_y \ \mathcal{L}_\beta(x^{k+1}, y, \lambda^k) + \dfrac{1}{2}\|y - y^k\|^2_S, & (4.1.3b) \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} - b). & (4.1.3c) \end{cases}$$

In this chapter, we assume that (4.1.3b) is solved exactly, and focus on how to construct T_k.

When $f(x) = \frac{1}{2}\|x\|^2$, $B = I$ and $b = 0$, a new construction of the proximal matrix T_k has been proposed in Chapter 3 to let the T_k be $T_k = B_k - M$, where $M = \nabla^2_{xx}\mathcal{L}_\beta(x, y, \lambda) = \beta A^\top A + I$ and B_k was a certain positive definite matrix. Note that $M \succ 0$, where $V \succeq 0$ ($V \succ 0$) means that V is symmetric and positive semidefinite (positive definite). This chapter proposed to generate B_k via the BFGS update with respect to M at every iteration, and then showed that $B_{k+1} \succeq M$ for all k whenever $B_k \succeq M$. Therefore $T_k \succeq 0$ if the initial matrix B_0 satisfies $B_0 \succeq M$. The x-subproblem of the variable metric semi-proximal ADMM with BFGS update (ADM-BFGS) for this special problem was given as

$$x^{k+1} = x^k + H_k \left(A^\top \lambda^k - \beta A^\top y^k - M x^k \right), \tag{4.1.4}$$

where $H_k = B_k^{-1}$. The numerical experiments in Chapter 3 reported that the numbers of iterations were almost same as those by the exact ADMM.

In this chapter, we extend this method to more general convex problems. In particular, we consider the following two problems.

One is formulated as

$$\begin{aligned} \text{Problem 1:} \quad & \text{minimize} \ \sum_{i=1}^N f_i(A_i x) \\ & \text{subject to} \ x \in \mathbb{R}^n, \end{aligned} \tag{4.1.5}$$

where $f_i : \mathbb{R}^{m_i} \to \mathbb{R} \cup \{\infty\}$, $i = 1, 2, ..., N$ are proper convex functions, and $A_i \in \mathbb{R}^{m_i \times n}$, $i = 1, 2, ..., N$.

The other problem is

$$\begin{aligned} \text{Problem 2:} \quad & \text{minimize} \ \sum_{i=1}^N f_i(x_i) \\ & \text{subject to} \ \sum_{i=1}^N A_i x_i = b, \\ & \qquad\qquad x_i \in \mathbb{R}^{n_i}, \quad i = 1, 2, ..., N, \end{aligned} \tag{4.1.6}$$

where $n = \sum_{i=1}^N n_i$, $f_i : \mathbb{R}^{n_i} \to \mathbb{R} \cup \{\infty\}$, $i = 1, 2, ..., N$ are proper convex functions, $A_i \in \mathbb{R}^{m \times n_i}$, $i = 1, 2, ..., N$ and $b \in \mathbb{R}^m$.

Now, we apply the ADMM (4.1.3a)-(4.1.3b) to the above two convex problems. To this end, we reformulate the problems as (4.1.1). By introducing some auxiliary variables $y_i \in \mathbb{R}^{m_i}(i =$

$1, 2, ..., N$), problem (4.1.5) can be reformulated as

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{N} f_i(y_i) \\ \text{subject to} & y_i = A_i x, \quad i = 1, 2, ..., N \\ & x \in \mathbb{R}^n, y_i \in \mathbb{R}^{m_i}, \quad i = 1, 2, ..., N. \end{array} \tag{4.1.7}$$

Letting $y = (y_1^\top, y_2^\top, ..., y_N^\top)^\top$, $f(x) \equiv 0$, $g(y) = \sum_{i=1}^{N} f_i(y_i)$, $A = [A_1^\top, A_2^\top, ..., A_N^\top]^\top$, $B = -I$, $b = 0$, problem (4.1.7) is reduced to (4.1.1).

Similarly, by introducing some auxiliary variables $y_i \in \mathbb{R}^{n_i} (i = 1, 2, ..., N)$, problem (4.1.6) can be reformulated as

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{N} f_i(y_i) \\ \text{subject to} & \sum_{i=1}^{N} A_i x_i = b, \\ & x_i = y_i, \quad i = 1, 2, ..., N \\ & x_i, y_i \in \mathbb{R}^{n_i}, \quad i = 1, 2, ..., N. \end{array} \tag{4.1.8}$$

Letting $x = (x_1^\top, x_2^\top, ..., x_N^\top)^\top$, $y = (y_1^\top, y_2^\top, ..., y_N^\top)^\top$, $f(x) \equiv 0$, $g(y) = \sum_{i=1}^{N} f_i(y_i)$, $\tilde{A} = [A_1, A_2, ..., A_N]$, and $A = \begin{bmatrix} \tilde{A} \\ I \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ -I \end{bmatrix}$, $b = \begin{bmatrix} b \\ 0 \end{bmatrix}$, problem (4.1.8) is also reduced to (4.1.1).

The main contributions of this chapter are as follows. At first, inspired by the work in Chapter 3, which considered the proximal ADMM with the BFGS update for structured quadratic optimization problems, we further extend the proximal ADMM with the Broyden family update for two general convex problems (4.1.7) and (4.1.8). Moreover, we show the global convergence of the proposed method. We also present some numerical results of the proposed method for solving the l_1 regularized logistic regression problem.

The rest of the chapter is organized as follows. We describe the construction of T_k via the Broyden family update and show the details on applying the ADMM for the above two convex problems (4.1.5) and (4.1.6) in Section 4.2. Section 4.3 discusses the global convergence of the proposed method under certain flexible conditions on the proximal matrices sequence. In Section 4.4, we test the l_1 regularized logistic regression problem to illustrate the efficiency of the proposed method. Finally, we make some concluding remarks in Section 4.5.

4.2 Proximal ADMMs with Broyden family update for two convex optimization problems

In this section, we first explain how to construct T_k via the Broyden family update, and then present concrete algorithms for two convex optimization problems (4.1.5) and (4.1.6).

4.2.1 Construction of the regularized matrix T_k via the Broyden family

Throughout this chapter, we assume that x-subproblems (4.1.3a) are unconstrained quadratic programming problem, and that the Hessian matrix of the augmented Lagrangian function (4.1.2) is a constant matrix defined as

$$M: \ = \nabla^2_{xx}\mathcal{L}_\beta(x, y, \lambda). \tag{4.2.1}$$

Note that M is always positive semidefinite. Note also that x-subproblems for (4.1.7) and (4.1.8) become unconstrained quadratic programming problems as seen later.

The Hessian of the objective function of x-subproblem (4.1.3a) is $B_k = T_k + M$. Note that if $T_k = 0$, that is, we consider the standard ADMM, then $B_k = M$. In order to avoid computing the inverse of M but still has some information on M, we consider a matrix B_k that has the following two properties:

Property (i) $B_k \succeq M$;

Property (ii) B_k has some second order information on M.

Property (i) implies that $T_k = B_k - M$ is positive semidefinite, which is reguired for global convergence. Property (ii) is needed for rapid convergence, since $T_k \approx 0$ as $B_k \approx M$. A new construction of B_k was proposed in Chapter 3 to construct B_k via the BFGS update with respect to M at every iteration. Since the BFGS update usually constructs the inverse of B_k, let $H_k = B_k^{-1}$. Using H_k, we can easily solve the x-subproblems as in (4.1.4).

When $M \succ 0$, we consider the normal BFGS update with a given $s \in \mathbb{R}^n$ and $l = Ms$. Note that $s^\top l > 0$ when $s \neq 0$. Let $s_k = x^{k+1} - x^k, l_k = Ms_k$. Then BFGS recursions for B^{BFGS}_{k+1} and H^{BFGS}_{k+1} are given as

$$B^{\mathrm{BFGS}}_{k+1} = B_k + \frac{l_k l_k^\top}{l_k^\top s_k} - \frac{B_k s_k s_k^\top B_k^\top}{s_k^\top B_k s_k}, \tag{4.2.2}$$

$$H^{\mathrm{BFGS}}_{k+1} = \left(I - \frac{s_k l_k^\top}{s_k^\top l_k}\right) H_k \left(I - \frac{l_k s_k^\top}{s_k^\top l_k}\right) + \frac{s_k s_k^\top}{s_k^\top l_k}. \tag{4.2.3}$$

Besides, the DFP updates for B^{DFP}_{k+1} and H^{DFP}_{k+1} are given as

$$B^{\mathrm{DFP}}_{k+1} = \left(I - \frac{l_k s_k^\top}{l_k^\top s_k}\right) B_k \left(I - \frac{s_k l_k^\top}{l_k^\top s_k}\right) + \frac{l_k l_k^\top}{l_k^\top s_k} \tag{4.2.4}$$

$$H^{\mathrm{DFP}}_{k+1} = H_k + \frac{s_k s_k^\top}{s_k^\top l_k} - \frac{H_k l_k l_k^\top H_k^\top}{l_k^\top H_k l_k}. \tag{4.2.5}$$

Since $s_k^\top l_k > 0$, all matrices B^{BFGS}_{k+1}, H^{BFGS}_{k+1}, B^{DFP}_{k+1} and H^{DFP}_{k+1} are positive definite whenever $B_k, H_k \succ 0$.

Motivated by the proximal ADMM with the BFGS update, we propose an extension of the proximal ADMM with the Broyden family update [83, 43, 94, 42], that is, a linear combination of the BFGS and DFP updates for B_{k+1} as follows:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^\top B_k}{s_k^\top B_k s_k} + \frac{l_k l_k^\top}{l_k^\top s_k} + t(s_k^\top B_k s_k) v_k v_k^\top, \quad v_k = \frac{l_k}{l_k^\top s_k} - \frac{B_k s_k}{s_k^\top B_k s_k}, \quad t \in [0,1], \tag{4.2.6}$$

which is equivalent to

$$B_{k+1} = (1-t) B_{k+1}^{\mathrm{BFGS}} + t B_{k+1}^{\mathrm{DFP}}, \quad \text{with} \quad t \in [0,1]. \tag{4.2.7}$$

Similarly, we have the Broyden family update for H_{k+1} with a $t \in [0,1]$ as follows:

$$H_{k+1} = H_k - \frac{H_k l_k l_k^\top H_k}{l_k^\top H_k l_k} + \frac{s_k s_k^\top}{l_k^\top s_k} + (1-t)(l_k^\top H_k l_k) v_k v_k^\top, \quad v_k = \frac{s_k}{s_k^\top l_k} - \frac{H_k l_k}{l_k^\top H_k l_k}, \tag{4.2.8}$$

which yields

$$H_{k+1} = (1-t) H_{k+1}^{\mathrm{BFGS}} + t H_{k+1}^{\mathrm{DFP}}, \quad \text{with} \quad t \in [0,1]. \tag{4.2.9}$$

The update scheme (4.2.7) becomes the pure BFGS update (4.2.2) when $t = 0$, and becomes the pure DFP update (4.2.4) when $t = 1$.

Now we consider when Property (i) holds. For the pure BFGS update, the following useful property has been shown in Chapter 3.

Lemma 4.2.1. *Let $s \in \mathbb{R}^n$ such that $s \neq 0$, and let $l = Ms$. If $H \preceq M^{-1}$, then $H^{\mathrm{BFGS}} \preceq M^{-1}$.*

This lemma is based on Lemma 3.2.1 in Chapter 3. We extend Lemma 4.2.1 to the Broyden family.

Lemma 4.2.2. *Let $l \in \mathbb{R}^n$ such that $l \neq 0$. Moreover let $s = M^{-1} l$ and $\Phi = \{z \in \mathbb{R}^n \mid \langle l, z \rangle = 0\}$. Then for any $v \in \mathbb{R}^n$, there exist $c \in \mathbb{R}$ and $z \in \Phi$ such that $v = cs + z$.*

Proof. We can easily show the lemma in a way similar to the proof of Lemma 3.2.1. ∎

We can rewrite the DFP update (4.2.4) as

$$B^{\mathrm{DFP}} = B - \frac{Bsl^\top + ls^\top B}{l^\top s} + \left(1 + \frac{s^\top B s}{l^\top s}\right) \frac{ll^\top}{l^\top s}. \tag{4.2.10}$$

Moreover we have

$$B^{\mathrm{DFP}} s = l = Ms. \tag{4.2.11}$$

Then we can obtain the following lemma for B^{DFP}.

Lemma 4.2.3. *Let $l \in \mathbb{R}^n$ such that $l \neq 0$, and let $s = M^{-1} l$. If $B \succeq M$, then $B^{\mathrm{DFP}} \succeq M$.*

Proof. Let v be an arbitrary nonzero vector in $\mathbb{R}^n$. Let $\Phi = \{z \in \mathbb{R}^n \mid \langle l, z\rangle = 0\}$. From Lemma 4.2.2 there exist $c \in \mathbb{R}$ and $z \in \Phi$ such that $v = cs + z$. It then follows from (4.2.11) and the definition of z that

$$\begin{aligned} v^\top B^{\mathrm{DFP}} v &= (cs+z)^\top B^{\mathrm{DFP}}(cs+z) \\ &= c^2 s^\top l + 2cz^\top l + z^\top B^{\mathrm{DFP}} z \\ &= c^2 s^\top l + z^\top B^{\mathrm{DFP}} z \\ &= c^2 s^\top M s + z^\top B^{\mathrm{DFP}} z. \end{aligned} \tag{4.2.12}$$

Since $z \in \Phi$, we have

$$z^\top \left(\frac{ls^\top}{l^\top s} B \frac{sl^\top}{l^\top s}\right) z = 0, \quad z^\top \left(\frac{ls^\top}{l^\top s} B\right) z = 0 \text{ and } \frac{z^\top l l^\top z}{l^\top s} = 0. \tag{4.2.13}$$

It then follows from (4.2.10) that

$$z^\top B^{\mathrm{DFP}} z = z^\top B z - 2z^\top \left(\frac{ls^\top}{l^\top s} B\right) z + z^\top \left(\frac{ls^\top}{l^\top s} B \frac{sl^\top}{l^\top s}\right) z + \frac{z^\top l l^\top z}{l^\top s} = z^\top B z. \tag{4.2.14}$$

Furthermore, equation (4.2.11) implies

$$s^\top M z = l^\top z = 0. \tag{4.2.15}$$

Combining (4.2.12) and (4.2.14), we have

$$\begin{aligned} v^\top B^{\mathrm{DFP}} v &= c^2 s^\top M s + z^\top B z \\ &\geq c^2 s^\top M s + z^\top M z \\ &= (cs+z)^\top M (cs+z) - 2cs^\top M z \\ &= v^\top M v, \end{aligned}$$

where the inequality follows from the assumption. Since v is arbitrary, we have $B^{\mathrm{DFP}} \succeq M$. ■

Using the above lemmas, we can show the following desired property for the Broyden family update.

Theorem 4.2.1. *Let $s, l \in R^n$ such that $s, l \neq 0$, and let $l = Ms$. Suppose that B_k is updated by the Broyden family* (4.2.7) *for all $k > 0$. If $B_0 \succeq M$, then $B_k \succeq M$ for any $t \in [0,1]$ and $k > 0$.*

Proof. Due to Lemmas 4.2.1 and 4.2.3 we show that if $(H_0)^{-1} = B_0 \succeq M$, then $B_k^{\mathrm{BFGS}} \succeq M$ and $B_k^{\mathrm{DFP}} \succeq M$ for all $k > 0$. It follows from (4.2.7) that for any $t \in [0,1]$ and $k > 0$, $B_k \succeq M$. ■

Theorem 4.2.1 implies that $T_k = B_k - M \succeq 0$ when the initial matrix B_0 satisfies $B_0 \succeq M$.

When M is merely positive semidefinite, we cannot directly use the BFGS and DFP updates. In this case we may consider $M^\delta :\, = M + \delta I$ with sufficiently small $\delta > 0$, and construct B_k by the BFGS and DFP updates for M^δ. That is, B^{BFGS} and B^{DFP} are updated by the BFGS and DFP with respect to M^δ at every iteration, respectively, and $l_\delta = M_\delta s = Ms + \delta s$. Then

$T_k = B_k - M \succeq \delta I \succ 0$ is a positive definite matrix. Note that $s^\top l_\delta > 0$ when $s \neq 0$, and recursions (4.2.3) and (4.2.4) still hold. When δ is small enough, B_k is also close to M.

In the subsequent subsections, we describe the details of applying the VMSP-ADMM with the Broyden family update for two convex problems (4.1.5) and (4.1.6) given in Introduction. Note that the y-subproblems also can be applied with the Broyden family update when they do not have closed-form solutions. For simplicity, we suppose the y-subproblems are easily solved here.

4.2.2 VMSP-ADMM for convex problem 1

First, we consider problem 1. Let $\mathcal{L}^1_\beta(x, y_1, ..., y_N, \lambda_1, ..., \lambda_N)$ be the augmented Lagrangian function for (4.1.7) defined by

$$\mathcal{L}^1_\beta(x, y_1, ..., y_N, \lambda_1, ..., \lambda_N) := \sum_{i=1}^N f_i(y_i) - \sum_{i=1}^N \langle \lambda_i, A_i x - y_i \rangle + \sum_{i=1}^N \frac{\beta}{2} \|A_i x - y_i\|^2, \tag{4.2.16}$$

where $\lambda_i \in \mathbb{R}^{m_i} (i = 1, 2, ..., N)$ are multipliers associated to the linear constraints and $\beta > 0$ is a penalty parameter. As shown in Introduction, we further define $y = (y_1^\top, y_2^\top, ..., y_N^\top)^\top$, $\mathcal{A}_1 = [A_1^\top, A_2^\top, ..., A_N^\top]^\top$, $\lambda = (\lambda_1^\top, \lambda_2^\top, ..., \lambda_N^\top)^\top$, $g(y) = \sum_{i=1}^N f_i(y_i)$, and $m = \sum_{i=1}^N m_i$. Then $y \in \mathbb{R}^m$, $\lambda \in \mathbb{R}^m$ and the augmented Lagrangian function is rewritten as

$$\mathcal{L}^1_\beta(x, y, \lambda) := g(y) - \langle \lambda, \mathcal{A}_1 x - y \rangle + \frac{\beta}{2} \|\mathcal{A}_1 x - y\|^2. \tag{4.2.17}$$

By using a proximal matrix $T^1_k \in \mathbb{R}^{n \times n}$, the x-subproblem (4.1.3a) for (4.1.7) is written as

$$\begin{aligned} x^{k+1} &= \arg\min_x \left\{ -\langle \lambda^k, \mathcal{A}_1 x - y^k \rangle + \frac{\beta}{2} \|\mathcal{A}_1 x - y^k\|^2 + \frac{1}{2} \|x - x^k\|^2_{T^1_k} \right\} \\ &= \left(\beta \mathcal{A}_1^\top \mathcal{A}_1 + T^1_k \right)^{-1} \left(\mathcal{A}_1^\top \lambda^k + \beta \mathcal{A}_1^\top y^k + T^1_k x^k \right). \end{aligned} \tag{4.2.18}$$

Note that $\beta \mathcal{A}_1^\top \mathcal{A}_1 = \sum\limits_{i=1}^N \beta A_i^\top A_i = \nabla^2_{xx} \mathcal{L}^1_\beta(x, y, \lambda)$. Let M^1 be defined as

$$M^1 : = \nabla^2_{xx} \mathcal{L}^1_\beta(x, y, \lambda) = \beta \mathcal{A}_1^\top \mathcal{A}_1 \succeq 0. \tag{4.2.19}$$

Note that M^1 is not necessarily positive definite. Then, as written in the previous subsection, we may not apply the BFGS and DFP updates for M. Therefore we use the following perturbed matrix M^{δ_1} instead of M^1.

$$M^{\delta_1} : = M^1 + \delta_1 I \succ 0 \quad \text{with} \quad \delta_1 > 0. \tag{4.2.20}$$

Then, we propose to construct a matrix B^1_k by the Broyden family with $s_k = x_1^{k+1} - x_1^k$ and $l_k = M^{\delta_1} s_k$, and set

$$T^1_k = B^1_k - M^1. \tag{4.2.21}$$

Note that Theorem 4.2.1 implies that $T^1_k \succ 0$ if $B^1_0 \succeq M^{\delta_1}$.

Using B_k^1, x-subproblem (4.2.18) is written as

$$x^{k+1} = x^k + (B_k^1)^{-1}\left(\mathcal{A}_1^\top \lambda^k + \beta \mathcal{A}_1^\top y^k - M^1 x^k\right). \tag{4.2.22}$$

Based on the descriptions of the BFGS (or limited memory BFGS, abbreviated to L-BFGS) and DFP updates, we first give our algorithm for problem (4.1.5).

Algorithm 2: VMSP-ADMM with the Broyden family update (ADM-BD1) for convex Problem 1 (4.1.5)

Input : data matrix $\mathcal{A}_1$, initial point (x^0, y^0, λ^0), penalty parameter β, δ_1; maxIter, initial matrix $H_0^1 \preceq (M^{\delta_1})^{-1}$, coefficient $t_1 \in [0,1]$; constant $\bar{k}^1 \in [1,\infty]$, stopping criterion ϵ.

Output: approximative solution (x^k, y^k, λ^k)

initialization;
while $k <$ maxIter *or not convergence* **do**
    if $k \le \bar{k}^1$ and $x_k - x_{k-1} \neq 0$ **then**
        update H_k^1 via (4.2.8) or L-BFGS;
    else
        $H_k^1 = H_{k-1}^1$;
    end
    update x^{k+1} by solving the x-subproblem:
        $x^{k+1} = x^k + H_k^1\left(\mathcal{A}_1^\top \lambda^k + \beta \mathcal{A}_1^\top y^k - M^1 x^k\right)$;
    update y^{k+1} by solving the y-subproblem:
        $y^{k+1} = \arg\min\limits_y \left\{ g(y) - \langle \lambda^k, \mathcal{A}_1 x^{k+1} - y\rangle + \frac{\beta}{2}\|\mathcal{A}_1 x^{k+1} - y\|^2 + \frac{1}{2}\|y - y^k\|_S^2 \right\}$;
    update Lagrangian multipliers: $\lambda^{k+1} = \lambda^k - \beta(\mathcal{A}_1 x^{k+1} - y^{k+1})$.
end

Remark 4.2.1. *Note that constant $\bar{k}^1 \in [1,\infty]$ in the algorithm means that the B_k^1 (H_k^1) updated by the BFGS (or L-BFGS) and DFP procedures will be stopped at $\bar{k}^1$, that is, $B_k^1 = B_{\bar{k}^1}^1$ ($H_k^1 = H_{\bar{k}^1}^1$) for $k \ge \bar{k}^1$. Specially,*

- $\bar{k}^1 < \infty$ *we can show the global convergence since it is reduced to the usual proximal ADMM after $\bar{k}^1$ steps;*
- $\bar{k}^1 = \infty$ *the B_k^1 (H_k^1) are updated for all k, and the numerical experiments in Chapter 3 show it works;*
- $\bar{k}^1$ **is small** *it is not efficient, since $B_{\bar{k}^1}^1$ is not close to M^{δ_1}.*

4.2.3 VMSP-ADMM for convex problem 2

For problem 2, let the augmented Lagrangian function for (4.1.8) $\mathcal{L}_\beta^2(x_1, ..., x_N, y_1, ..., y_N, \lambda, \mu_1, ..., \mu_N)$ be defined by

$$\mathcal{L}_\beta^2(x_1, ..., x_N, y_1, ..., y_N, \lambda, \mu_1, ..., \mu_N)$$

$$
\begin{aligned}
&= \sum_{i=1}^{N} f_i(y_i) - \langle \lambda, \sum_{i=1}^{N} A_i x_i - b\rangle - \sum_{i=1}^{N}\langle \mu_i, x_i - y_i\rangle \\
&\quad + \frac{\beta_1}{2}\|\sum_{i=1}^{N} A_i x_i - b\|^2 + \sum_{i=1}^{N}\frac{\beta_2}{2}\|x_i - y_i\|^2, \qquad (4.2.23)
\end{aligned}
$$

where $\lambda \in \mathbb{R}^m, \mu_i \in \mathbb{R}^{n_1}(i = 1, 2, ..., N)$ are multipliers associated to the linear constraints and $\beta_1, \beta_2 > 0$ are the penalty parameters, respectively.

We also define $x = (x_1^\top, x_2^\top, ..., x_N^\top)^\top$, $y = (y_1^\top, y_2^\top, ..., y_N^\top)^\top$, $g(y) = \sum_{i=1}^{N} f_i(y_i)$, $\mathcal{A}_2 = [A_1, A_2, ..., A_N]$, and $n = \sum_{i=1}^{N} n_i$. Then $x, y \in \mathbb{R}^n$ and $\mathcal{A}_2 \in \mathbb{R}^{m\times n}$. Moreover let $\mu = (\mu_1^\top, \mu_2^\top, ..., \mu_N^\top)^\top \in \mathbb{R}^n$. The augmented Lagrangian function (4.2.23) can be written as

$$
\mathcal{L}_\beta^2(x, y, \lambda, \mu) = g(y) - \langle \lambda, \mathcal{A}_2 x - b\rangle - \langle \mu, x - y\rangle + \frac{\beta_1}{2}\|\mathcal{A}_2 x - b\|^2 + \frac{\beta_2}{2}\|x - y\|^2. \qquad (4.2.24)
$$

Similar to problem 1 in Subsection 4.2.2, by using a proximal matrix $T_k^2 \in \mathbb{R}^{n\times n}$, the x-subproblem (4.1.3a) for (4.1.8) is written as

$$
\begin{aligned}
x^{k+1} &= \arg\min_x \Big\{ -\langle \lambda^k, \mathcal{A}_2 x - b\rangle - \langle \mu^k, x - y^k\rangle \\
&\quad + \frac{\beta_1}{2}\|\mathcal{A}_2 x - b\|^2 + \frac{\beta_2}{2}\|x - y^k\|^2 + \frac{1}{2}\|x - x^k\|_{T_k^2}^2 \Big\} \\
&= \left(\beta_1 \mathcal{A}_2^\top \mathcal{A}_2 + \beta_2 I + T_k^2\right)^{-1}\left(\mathcal{A}_2^\top \lambda^k + \mu^k + \beta_1 \mathcal{A}_2^\top b + \beta_2 y^k + T_k^2 x^k\right). \qquad (4.2.25)
\end{aligned}
$$

Let M^2 be defined as

$$
M^2 := \nabla_{xx}^2 \mathcal{L}_\beta^2(x, y, \lambda, \mu) = \beta_1 \mathcal{A}_2^\top \mathcal{A}_2 + \beta_2 I. \qquad (4.2.26)
$$

Note that $M^2 \succ 0$ whenever $\beta_2 > 0$. Then, we construct a matrix B_k^2 via the Broyden family for M^2 and set:

$$
T_k^2 = B_k^2 - M^2. \qquad (4.2.27)
$$

Theorem 4.2.1 implies that $T_k^2 \succeq 0$ if $B_0^2 \succeq M^2$. By using B_k^2, x-subproblem (4.2.25) is written as

$$
x^{k+1} = x^k + (B_k^2)^{-1}\left(\mathcal{A}_2^\top \lambda^k + \mu^k + \beta_1 \mathcal{A}_2^\top b + \beta_2 y^k - M^2 x^k\right). \qquad (4.2.28)
$$

The algorithm for problem (4.1.6) is as follows:

Algorithm 3: VMSP-ADMM with the Broyden family update (ADM-BD2) for convex Problem 2 (4.1.6)

Input : data matrix $\mathcal{A}_2$, initial point $(x^0, y^0, \lambda^0, \mu^0)$, penalty parameters β_1, β_2; initial matrix $H_0^2 \preceq (M^2)^{-1}$, coefficient $t_2 \in [0, 1]$, constant $\bar{k}^2 \in [1, \infty]$; maxIter, stopping criterion ϵ.

Output: approximative solution $(x^k, y^k, \lambda^k, \mu^k)$

initialization;
while $k <$ maxIter *or not convergence* **do**
  if $k \leq \bar{k}^2$ and $x_k - x_{k-1} \neq 0$ **then**
    update H_k^2 via (4.2.8) or L-BFGS;
  else
    $H_k^2 = H_{k-1}^2$;
  end
  update x^{k+1} by solving the x-subproblem:
    $x^{k+1} = x^k + H_k^2 \left(\mathcal{A}_2^\top \lambda^k + \mu^k + \beta_1 \mathcal{A}_2^\top b + \beta_2 y^k - M^2 x^k \right);$
  update y^{k+1} by solving the y-subproblem:
    $y^{k+1} = \arg\min_y \left\{ g(y) - \langle \mu^k, x^{k+1} - y \rangle + \frac{\beta_2}{2} \|x^{k+1} - y\|^2 + \frac{1}{2} \|y - y^k\|_S^2 \right\};$
  update Lagrangian multipliers: $\lambda^{k+1} = \lambda^k - \beta_1 (\mathcal{A}_2 x^{k+1} - b)$;
  update Lagrangian multipliers: $\mu^{k+1} = \mu^k - \beta_2 (x^{k+1} - y^{k+1})$.
end

The constant $\bar{k}^2$ in the Algorithm 2 plays the same role as $\bar{k}^1$ in Algorithm 1.

4.3 Convergence analysis

In this section, we first consider general convergence properties for the variable metric semi-proximal ADMM (4.1.3a)-(4.1.3c) (VMSP-ADMM). Then we discuss the convergence of VMSP-ADMM with the Broyden family update (ADM-BD1 and ADM-BD2) for problem 1 and problem 2.

4.3.1 Global convergence of the variable metric semi-proximal ADMM

Let Ω^* be a set of (x^*, y^*, λ^*) satisfying the KKT condition of problem (4.1.1). Since the subdifferential mapping of the closed proper convex functions are maximal monotone [101], there exist two positive semidefinite matrices Σ_f and Σ_g such that for all $x, \hat{x} \in \mathbb{R}^{n_1}$, $f'(x) \in \partial f(x)$, and $f'(\hat{x}) \in \partial f(\hat{x})$,

$$(x - \hat{x})^\top (f'(x) - f'(\hat{x})) \geq \|x - \hat{x}\|_{\Sigma_f}^2, \tag{4.3.1}$$

and for all $y, \hat{y} \in \mathbb{R}^{n_2}$, $g'(y) \in \partial g(y)$, and $g'(\hat{y}) \in \partial g(\hat{y})$,

$$(y - \hat{y})^\top (g'(y) - g'(\hat{y})) \geq \|y - \hat{y}\|_{\Sigma_g}^2. \tag{4.3.2}$$

Throughout this chapter, we make the following assumptions.

Assumption 4.3.1. *The set Ω^* of KKT points is non-empty.*

Assumption 4.3.2. *For all $k > 0$, $T_k + \Sigma_f + \beta A^\top A$ and $S + \Sigma_g + \beta B^\top B$ are positive definite.*

Now, we begin to investigate the convergence of VMSP-ADMM. First, we assume some conditions for the sequence $\{T_k\}$ to guarantee the convergence.

Condition 4.3.1. *For the sequence $\{T_k\}$ generated by the framework (4.1.3), there exist $T \succeq 0$ and a non-negative sequence $\{\gamma_k\}$ such that*

1 $\quad T \preceq T_{k+1} \preceq (1+\gamma_k)T_k$, *for all* k,

2 $\quad T + \Sigma_f + \beta A^\top A$ *is positive definite,*

3 $\quad \sum_0^\infty \gamma_k < \infty$.

Now we give the main theorem of this subsection.

Theorem 4.3.1. *Suppose that Assumptions 4.3.1 and 4.3.2 hold. Suppose also that $\{T_k\}$ is a sequence satisfying Condition 4.3.1. Let $\{(x^k, y^k, \lambda^k)\}$ be generated by (4.1.3). Then the following statements hold:*

(a) *we have for $k \geq 1$ that*

$$
\begin{aligned}
&\|x^k - x^*\|_{T_k}^2 + \|y^k - y^*\|_S^2 + \beta\|B(y^k - y^*)\|^2 + \frac{1}{\beta}\|\lambda^k - \lambda^*\|^2 + \|y^k - y^{k-1}\|_S^2 \\
&\quad - \Bigg(\|x^{k+1} - x^*\|_{T_k}^2 + \|y^{k+1} - y^*\|_S^2 + \beta\|B(y^{k+1} - y^*)\|^2 + \frac{1}{\beta}\|\lambda^{k+1} - \lambda^*\|^2 \\
&\qquad + \|y^{k+1} - y^k\|_S^2\Bigg) \\
&\geq \|x^{k+1} - x^k\|_{T_k}^2 + \|y^{k+1} - y^k\|_S^2 + \beta\|B(y^{k+1} - y^k)\|^2 \\
&\quad + \beta\|Ax^{k+1} + By^{k+1} - b\|^2 + 2\|x^{k+1} - x^*\|_{\Sigma_f}^2 + 2\|y^{k+1} - y^*\|_{\Sigma_g}^2.
\end{aligned}
$$

(b) *the sequence $\{(x^k, y^k, \lambda^k)\}$ converges to a point $(x^*, y^*, \lambda^*) \in \Omega^*$.*

Proof. The proof can be done in a way similar to the proofs in 3.3.1 and [40, Theorem B.1]. ∎

4.3.2 Global convergences of Algorithms 2 and 3

We establish the global convergences of Algorithms 2 and 3 as corollaries of the convergence results in Subsection 4.3.1.

For the global convergence, we require condition 4.3.1 holds. In particular, we need that $0 \preceq T \preceq T_k$ for all k, that is, T_k should be positive semidefinite for all k. To see this, we first note that initial matrices H_0^1 and H_0^2 satisfy $H_0^1 \preceq (M^{\delta_1})^{-1}$ and $H_0^2 \preceq (M^2)^{-1}$. Then Theorem 4.2.1 implies $H_k^1 \preceq (M^{\delta_1})^{-1}$ and $H_k^2 \preceq (M^2)^{-1}$ for all k, and hence $T_k^1, T_k^2 \succeq 0$ for all k.

Therefore, Algorithms 2 and 3 are well-defined, that is, they can generate a sequence $\{x^k, y^k, \lambda^k\}$. Moreover, we get the following convergence properties based on Theorem 4.3.1.

Theorem 4.3.2. *Suppose that the sequence $\{T_k^1\}$ satisfies Condition 4.3.1. Let a sequence $\{(x^k, y^k, \lambda^k)\}$ be generated by Algorithm 2. Then the sequence $\{(x^k, y^k, \lambda^k)\}$ converges to a KKT point of* (4.1.7).

Theorem 4.3.3. *Suppose that the sequence $\{T_k^2\}$ satisfies Condition 4.3.1. Let a sequence $\{(x^k, y^k, \lambda^k, \mu^k)\}$ be generated by Algorithm 3. Then the sequence $\{(x^k, y^k, \lambda^k, \mu^k)\}$ converges to a KKT point of* (4.1.8).

Until now, we cannot give sufficient conditions for Condition 4.3.1 to hold when $\{B_k\}$ is updated by a pure Broyden family. As those remedies shown in Chapter 3, now we provide two amendments for the sequences $\{T_k^1\}$ and $\{T_k^2\}$ with the Broyden family (4.2.7) to guarantee Condition 4.3.1.

Amendment I Let $\bar{k}^1$ be finite in Algorithm 2. Then the updating of B_k^1 stops at $\bar{k}^1$, and thus the sequence $\{T_k^1\}$ satisfies Condition 4.3.1. Similarly, let $\bar{k}^2$ be finite in Algorithm 3. Then the sequence $\{T_k^2\}$ satisfies Condition 4.3.1 as well.

Amendment II Suppose that $\bar{k}^1 = \infty$ in Algorithm 2. We generate $\{B_k^1\}$ as follows:

$$B_{k+1}^1 = B_k^1 + c_k^1 \left(\bar{B}_{k+1}^1 - B_k^1\right), \tag{4.3.3}$$

where $\bar{B}_{k+1}^1$ is updated by the pure Broyden family (4.2.6) for $M^{\delta_1} \succ 0$ with $\delta_1 > 0$, and $\{c_k^1\}$ is a sequence such that $c_k^1 \in [0,1]$, and $\sum_{k=0}^{\infty} c_k^1 < \infty$. We can easily obtain that Condition 4.3.1 holds for the sequence $\{T_k^1\}$ as shown in Chapter 3.

Suppose that $\bar{k}^2 = \infty$ in Algorithm 3. We generate $\{B_k^2\}$ as follows:

$$B_{k+1}^2 = B_k^2 + c_k^2 \left(\frac{\tilde{l}_k^2 (\tilde{l}_k^2)^\top}{(\tilde{l}_k^2)^\top s_k^2} - \frac{B_k^2 s_k^2 (s_k^2)^\top (B_k^2)^\top}{(s_k^2)^\top B_k^2 s_k^2} + t_2 \left((s_k^2)^\top B_k^2 s_k^2 \right) \tilde{v}_k^2 (\tilde{v}_k^2)^\top \right), \tag{4.3.4}$$

where $\tilde{l}_k^2 = M^{\delta_2} s_k^2 = M^2 s_k^2 + \delta_2 s_k^2$ with $\delta_2 > 0$, $t_2 \in [0,1]$ is a scalar parameter,

$$\tilde{v}_k^2 = \left(\frac{\tilde{l}_k^2}{(\tilde{l}_k^2)^\top s_k^2} - \frac{B_k^2 s_k^2}{(s_k^2)^\top B_k^2 s_k^2} \right), \tag{4.3.5}$$

and $\{c_k^2\}$ is a sequence such that $c_k^2 \in [0,1]$, and $\sum_{k=0}^{\infty} c_k^2 < \infty$.

We can rewrite (4.3.4) as

$$B_{k+1}^2 = B_k^2 + c_k^2 \left(\bar{B}_{k+1}^2 - B_k^2 \right), \tag{4.3.6}$$

$$\bar{B}_{k+1}^2 = (1-t_2) B_{k+1}^{\mathrm{B2}} + t_2 B_{k+1}^{\mathrm{D2}}, \quad \text{with} \quad t_2 \in [0,1], \tag{4.3.7}$$

where B_{k+1}^{B2} and B_{k+1}^{D2} are updated by the pure BFGS and DFP updates with respect to M^{δ_2}, respectively.

As shown in Chapter 3, Condition 4.3.1 holds for the sequence $\{T_k^2\}$.

4.4 Numerical Experiments

In this section, we test the proposed proximal ADMM by solving a popular sparse learning problem, l_1 regularized logistic regression model. All the experiments are implemented by Matlab R2018b on Windows 10 pro with a 2.10 GHz Intel Xeon E5-2620 v4 processor and 128 GB of RAM. The l_1 regularized logistic regression model is given as

$$\min \left\{ \frac{1}{m} \sum_{i=1}^{m} \log\Big(1 + \exp\big(-r_i(A_i x + \sigma)\big)\Big) + \rho \|x\|_1 \;\Big|\; x \in \mathbb{R}^n \right\}, \tag{4.4.1}$$

where $A \in \mathbb{R}^{m \times n}$ is a feature matrix, $A_i \in \mathbb{R}^{1 \times n}$ is the row vector of matrix A, and $r \in \mathbb{R}^m$ is a response vector. The scalar m is the number of data points, and n is the dimension of data. Moreover, $\sigma \in \mathbb{R}$ is a decided intercept scalar, and $\rho > 0$ is a regularization parameter. The decision variable of (4.4.1) is $x \in \mathbb{R}^n$.

4.4.1 Reformulation and algorithms

Our purpose is to justify the advantages of the proximal ADMM with the Broyden family update (Algorithm 2). We choose some classical ADMMs as the benchmark for numerical comparison in this subsection.

By introducing some auxiliary variables $y_i \in \mathbb{R}$ $(i = 1, 2, ..., m)$ and $z \in \mathbb{R}^n$, the l_1 regularized logistic regression model (4.4.1) can be reformulated as

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{m} \sum_{i=1}^{m} \log\Big(1 + \exp\big(-r_i(y_i + \sigma)\big)\Big) + \rho \|z\|_1 \\ \text{subject to} \quad & y_i = A_i x, \quad i = 1, 2, ..., m \\ & z = x, \\ & x, z \in \mathbb{R}^n, y_i \in \mathbb{R}. \end{aligned} \tag{4.4.2}$$

Note that, letting $\tilde{y} = \begin{pmatrix} y \\ z \end{pmatrix}$, $g(\tilde{y}) = \frac{1}{m} \sum_{i=1}^{m} \log\Big(1 + \exp\big(-r_i(y_i + \sigma)\big)\Big) + \rho \|z\|_1$, $\tilde{A} = \begin{bmatrix} A \\ I_n \end{bmatrix}$, $B = -I_{m+n}$, and $b = 0$, problem (4.4.2) is reduced to (4.1.1).

The augmented Lagrangian function of (4.4.2) can be written as

$$\mathcal{L}_\beta(x, y, z, \lambda, \mu) = g(\tilde{y}) - \langle \lambda, Ax - y \rangle - \langle \mu, x - z \rangle + \frac{\beta}{2} \|Ax - y\|^2 + \frac{\beta}{2} \|x - z\|^2, \tag{4.4.3}$$

where $\lambda \in \mathbb{R}^m, \mu \in \mathbb{R}^n$ are multipliers associated to the linear constraints and $\beta > 0$ is the penalty parameter. We further define $\tilde{\lambda} = \begin{pmatrix} \lambda \\ \mu \end{pmatrix}$. Let M be the Hessian matrix of the augmented Lagrangian function (4.4.3), that is, $M\colon = \nabla^2_{xx}\mathcal{L}_\beta(x,y,z,\lambda,\mu) = \beta\tilde{A}^\top\tilde{A} = \beta A^\top A + \beta I$. Note that $M \succ 0$ whenever $\beta > 0$. Therefore, we set $\delta_1 = 0$ in Algorithm 2.

The maximum eigenvalue $\lambda_{\max}\left(\beta I + \beta A^\top A\right)$ is needed in some algorithms. We adopt the the following codes in Matlab to compute the maximum eigenvalue $\lambda_{\max}\left(A^\top A\right)$:

```
AAfun = @(x)A'*(A*x);    eig_max = eigs(AAfun,n,1).
```

The inverse of matrix M will also be used later for some x-subproblems. We use a Cholesky factorization to solve it. When $m \geq n$, we use "chol" in Matlab for $(A^\top A + \beta I)$. When $m < n$, we apply Sherman-Morrison formula to $(\beta I + A^\top A)^{-1}$:

$$(\beta I + A^\top A)^{-1} = \frac{1}{\beta}I - \frac{1}{\beta^2}\cdot A^\top \cdot \left(I + \frac{1}{\beta}AA^\top\right)^{-1}\cdot A, \tag{4.4.4}$$

and compute the factorization $LL^\top$ of a smaller matrix $(I + (1/\beta)AA^\top)$ by the "chol" function. Note that the maximum eigenvalue and the Cholesky factorization are calculated only once for each test problem.

Besides, a soft-thresholding operator $S_\kappa\colon \mathbb{R}^n \to \mathbb{R}^n$ will be used in y-subproblems, which is defined as $(S_\kappa(a))_i = (1-\kappa/|a_i|)_+ \cdot a_i$, for all $i = 1,...,m$, $\kappa > 0$ and $a \in \mathbb{R}^m$.

We test the following methods:

ADMM-1 the classical ADMM [46, 49] which is applied for the original problem (4.4.1):

$$\begin{cases} x^{k+1} = \arg\min\limits_x \dfrac{1}{m}\sum\limits_{i=1}^m \log\Big(1+\exp\big(-r_i(A_ix+\sigma)\big)\Big) + \dfrac{\beta}{2}\|x - y^k - \dfrac{\lambda^k}{\beta}\|_2^2, \\ y^{k+1} = S_{\rho/\beta}(x^{k+1} - \lambda^k/\beta), \\ \lambda^{k+1} = \lambda^k - \beta(x^{k+1} - y^{k+1}); \end{cases} \tag{4.4.5}$$

ADMM-2 the classical ADMM [46, 49] applied for problem (4.4.2);

ADM-PRO the proximal ADMM for problem (4.4.2) with an indefinite proximal matrix T as [63, 77]:

$$T = \xi I - \beta I - \beta A^\top A, \text{ with } \xi = 0.8 * \lambda_{\max}\left(\beta I + \beta A^\top A\right); \tag{4.4.6}$$

ADM-Broyden the proximal ADMM with the Broyden family update for problem (4.4.2) with a semidefinite proximal matrix sequence $\{T_k\}$:

$$\begin{cases} x^{k+1} = x^k + (B_k)^{-1}\left(A^\top\lambda^k + \mu^k + \beta A^\top y^k + \beta z^k - Mx^k\right), \\ y^{k+1} = \arg\min\limits_y \dfrac{1}{m}\sum\limits_{i=1}^m \log\Big(1+\exp\big(-r_i(y_i+\sigma)\big)\Big) + \dfrac{\beta}{2}\|Ax^{k+1} - y - \dfrac{\lambda^k}{\beta}\|_2^2, \\ z^{k+1} = S_{\rho/\beta}(x^{k+1} - \mu^k/\beta), \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} - y^{k+1}), \\ \mu^{k+1} = \mu^k - \beta(x^{k+1} - z^{k+1}), \end{cases} \tag{4.4.7}$$

where B_k is generated by Broyden family (4.2.6), and the initial matrix B_0 is chosen as

$$B_0 = \xi I, \text{ with } \xi = 1.01 * \lambda_{\max}(\beta I + \beta A^\top A). \tag{4.4.8}$$

It becomes the pure BFGS update when $t = 0$ (ADM-BFGS) and becomes the pure DFP update when $t = 1$;

ADM-BFGS (or ADM-LBFGS) the proximal ADMM with the BFGS (or L-BFGS) update for problem (4.4.2) with a semidefinite proximal matrix sequence $\{T_k\}$. The matrix B_k is generated by BFGS (4.2.10), and the initial matrix B_0 is chosen as (4.4.8);

ADM-ILBFGS the proximal ADMM with the L-BFGS update for problem (4.4.2) with an indefinite proximal matrix sequence $\{T_k\}$ where the initial matrix B_0 is chosen as

$$B_0 = \xi I, \xi = 0.8 * \lambda_{\max}(\beta I + \beta A^\top A).$$

Since the x-subproblems in (4.4.5) and y-subproblems in (4.4.7) have no closed-form solution, we adopt a custom Newton solver for these subproblems with the tolerance of 10^{-6}. We set a maximum iteration number of the Newton solver to 50. The codes for the original ADMM (4.4.5) are referred to the paper [13].

4.4.2 Problem data and algorithm settings

Now we specify the data for the l_1 regularized logistic regression model (4.4.1) to be tested. We generate data by the codes of [13] as follows. We first generate $D \in \mathbb{R}^{m\times n}$ as a sparse matrix normally distributed with p sparsity nonzero entries, where $p \in (0, 1]$. Then we generate σ, r, A and ρ as follows:

- the intercept σ is chosen from $\mathcal{N}(0, 1)$;
- the vector r is generated by $r = \text{sign}(Dw + \sigma + \epsilon)$, where ϵ is the noise drawn from $\mathcal{N}(0, 0.1)$, and $w \in \mathbb{R}^n$ is a random and sparse vector with approximately 10% normally distributed nonzero entries;
- the matrix A can be written as $A = \text{spdiags}(r, 0, m, m) * D$ by MATLAB, which is the product of a banded sparse matrix with D;
- $\rho = 0.1\rho_{\max}$, where $\rho_{\max}$ is the maximum regularization parameter when the solution is $x^* = 0$. The concrete definition of $\rho_{\max}$ can be found in [73, Subsection 2.1].

Then we give the settings of initial points, maximum iterations and stopping criterions for the above algorithms. We set the initial points as $x^0 = y^0 = z^0 = 0$ and $\lambda^0 = \mu^0 = 0$. The maximum outer iteration step is 5000.

We adopt the stopping criterion as in [13] that the primal residual and dual residual are small at the iteration k:

$$\|\tilde{A}x^k + B\tilde{y}^k\|_2 \le \epsilon_k^{\text{pri}} \quad \text{and} \quad \|\beta\tilde{A}^\top B(\tilde{y}^k - \tilde{y}^{k-1})\|_2 \le \epsilon_k^{\text{dual}}, \tag{4.4.9}$$

where $\epsilon^{\text{pri}} > 0$ and $\epsilon^{\text{dual}} > 0$ are feasibility tolerances for the primal and dual feasibility conditions, respectively. These tolerances can be chosen using absolute and relative criteria from the suggestion in [13], such as

$$\begin{aligned}\epsilon_k^{\text{pri}} &= \sqrt{n}\epsilon^{\text{abs}} + \epsilon^{\text{rel}}\max\{\|\tilde{A}x^k\|_2, \|B\tilde{y}^k\|_2\},\\ \epsilon_k^{\text{dual}} &= \sqrt{n}\epsilon^{\text{abs}} + \epsilon^{\text{rel}}\|\tilde{A}^\top\tilde{\lambda}^k\|_2,\end{aligned}$$

where $\epsilon^{\text{abs}} > 0$ is an absolute tolerance and $\epsilon^{\text{rel}} > 0$ is a relative tolerance. We set $\epsilon^{\text{abs}} = 10^{-4}, \epsilon^{\text{rel}} = 10^{-3}$.

We give the notations that will be used in the following tables for numerical results.

- Iter.: the outer iteration steps for each algorithm;
- Int.It.: the total internal iterations of Newton method for each algorithm;
- Time: the total CPU time for each algorithm;
- T-M: the CPU time of computing for the Cholesky factorization of ADMM-2; or the CPU time of computing for the maximum eigenvalue of ADM-PRO, ADM-LBFGS, and ADM-ILBFGS;
- T-A: the CPU time for the algorithm proceed without T-M.

All of the CPU times are recorded in seconds.

4.4.3 Numerical results

It is well known that the BFGS update is more effective than the DFP update from computational experience. We first choose $t \in \{-0.1, 0, 0.1\}$ in (4.2.6). Note that $t = -0.1$ cannot guarantee the positive definiteness in theory but work in practice. At first, we set the sizes and sparsity of the matrix D be $m \in \{500, 1000\}$, $n \in \{200, 500\}$, and $p = 0.1$. We test different $\beta \in (0, 500]$ and chose the best one. The results on iterations and CPU time (in seconds) for ADM-Broyden with different t are provided in Table 4.1. All the results are averaged over 10 trials.

Table 4.1: Comparison on iteration steps and CPU time for different t of ADM-Broyden

Setting			ADM-Broyden $t=-0.1$				ADM-Broyden $t=0$				ADM-Broyden $t=0.1$			
m	n	p	β	Iter.	Int.It.	Time(s)	β	Iter.	Int.It.	Time(s)	β	Iter.	Int.It.	Time(s)
500	200	0.1	2.4	72.0	216.0	0.16	2.3	74.0	222.0	0.16	2.4	72.0	216.0	0.16
500	500	0.1	3.3	81.0	243.0	0.46	3.5	83.0	249.0	0.49	2.2	82.0	246.0	0.44
1000	500	0.1	2.3	158.0	474.0	1.14	2.2	155.0	465.0	1.11	2.2	157.0	471.0	1.12

Table 4.1 shows that ADM-Broyden with $t = -0.1$ and 0.1 some times work well from the viewpoint of the number of iteration.

Table 4.2: Comparison on iteration steps and CPU time among the four methods

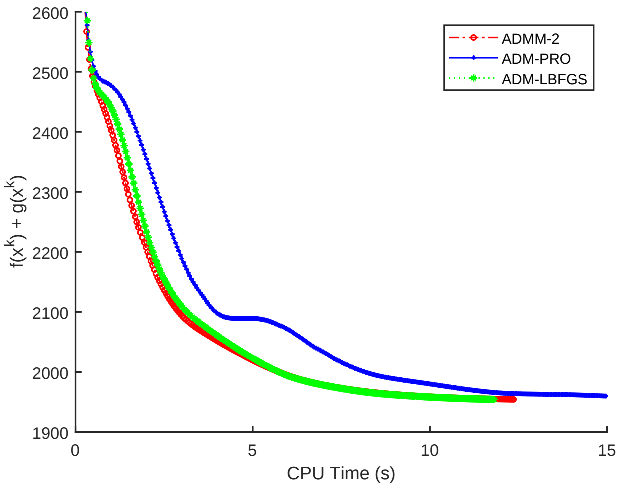
Setting			ADMM-1				ADMM-2				ADM-PRO				ADM-BFGS			
m	n	p	β	Iter.	Int.It.	Time(s)	β	Iter.	Int.It.	Time(s)	β	Iter.	Int.It.	Time(s)	β	Iter.	Int.It.	Time(s)
500	200	0.1	4.0	15.0	60.0	0.35	0.6	77.0	306.0	0.14	0.3	131.0	524.0	0.23	2.3	74.0	222.0	0.16
500	500	0.1	4.0	20.0	100.0	2.50	0.6	105.0	418.0	0.26	0.3	390.0	1559.0	0.75	3.5	83.0	249.0	0.49
1000	500	0.1	9.0	19.0	85.0	4.59	0.8	128.0	510.0	0.51	0.3	233.0	931.0	0.88	2.2	155.0	465.0	1.11

Next we test the same problems as those in Table 4.1 among the different classical ADMMs, proximal ADMM and ADMM with the BFGS update. Table 4.2 shows the iterative steps and the CPU time (in seconds) results.

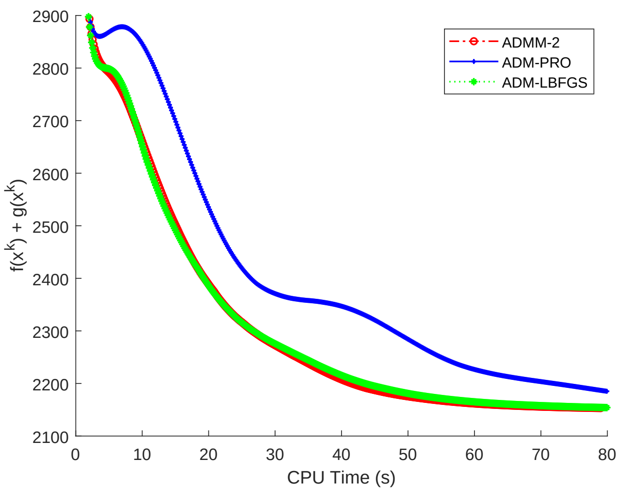
It is obvious to see from Table 4.2 that the classical ADMM-1 always gets a solution within least iterative steps but takes a lot of time computing the x-subproblems by the Newton method. We also know that classical ADMM-2 spends less CPU time to get the solution. The ADMM with the BFGS update (ADM-BFGS) is better than the proximal ADMM (ADM-PRO) on the number of iterations. When the data matrix A is ill-conditioned, or the inverse of the Hessian matrix of augmented Lagrangian function is impossible to be computed, it is meaningful to use the proximal matrix H_k. ADM-BFGS needs more memories to save the H_k, and thus we consider to use the L-BFGS update to construct the H_k for some larger cases.

We take the memory as 40 for the ADMM with the L-BFGS update. The upper results in Table 4.3 show the iteration steps and CPU time (in seconds) for a smaller and sparse matrix D when $m = 1000$, $n \in \{500, 1000, 2000\}$ and $p = 0.1$. The middle part in Table 4.3 shows the iteration steps and CPU time results when $m = 5000$, $n = 1000$ and $p \in \{0.1, 0.5, 1\}$. The results are compared among ADMM-1, ADMM-2, ADM-PRO, ADM-LBFGS, and ADM-ILBFGS. We also plot the objective function values with respect to the CPU time of matrix D (same as matrix A) with different sparsities among three methods, ADMM-2, ADM-PRO and ADM-LBFGS in Figure 4.1. In the lower part of Table 4.3, we test the behaviours of ADMMs for a larger matrix D when $m = 10000$, $n = 5000$ and $p \in \{0.1, 0.5, 1\}$. The best CPU time is highlighted with red color of each item, while the best iteration colors with blue except for ADMM-1. The iterations of ADM-LBFGS are highlighted with purple color in order to compare clearly with the classical ADMM-2.

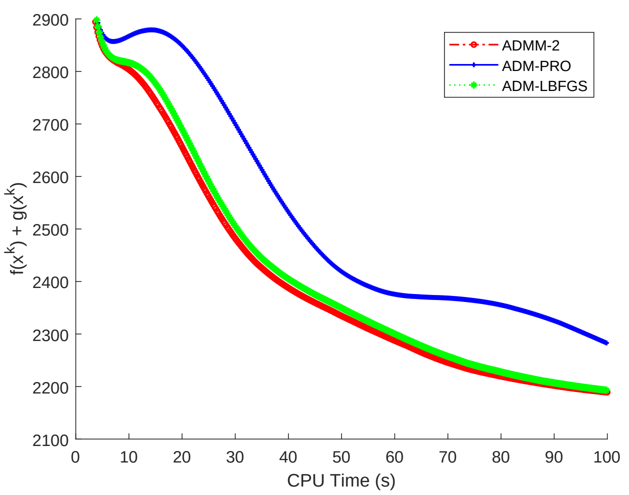
From Table 4.3 and Figure 4.1, we conclude that ADM-LBFGS performs well. When matrix A is larger and has more non-zero elements, ADMM-1 spends too much time. The usual proximal ADMM (ADM-PRO) takes much more iteration steps. ADM-LBFGS and ADM-ILBFGS algorithms can reach the solutions within the same levels both at iteration and CPU time for the algorithm (T-A) as the ADMM-2 while the indefinite ADM-ILBFGS is a slight better than the semidefinite ADM-LBFGS. If matrix A is large enough, non-sparse or ill-conditioned, i.e., it is difficult or impossible to compute the inverse of the matrix $\left(\beta I + \beta A^\top A\right)$, the CPU time of computing for the Cholesky factorization of ADMM-2 (T-M) is longer than the CPU time of computing for the maximum eigenvalue of ADM-LBFGS, and thus it is useful to choose ADMM with the L-BFGS update methods.



(a) $m = 5000, n = 1000, p = 0.1$



(b) $m = 5000, n = 1000, p = 0.5$



(c) $m = 5000, n = 1000, p = 1$

Figure 4.1: Evolution of the objective function values with respect to CPU time for small problems

Table 4.3: Comparison on iteration steps and CPU time among the five methods

Algorithm	$m=1000,\ n=500,\ p=0.1$						$m=1000,\ n=1000,\ p=0.1$						$m=1000,\ n=2000,\ p=0.1$					
	β	Iter.	Int.It.	Time(s)	T-A(s)	T-M(s)	β	Iter.	Int.It.	Time(s)	T-A(s)	T-M(s)	β	Iter.	Int.It.	Time(s)	T-A(s)	T-M(s)
ADMM-1	9.0	17.0	85.0	4.59	–	–	10.0	19.0	95.0	13.46	–	–	9.0	23.0	115.0	53.47	–	–
ADMM-2	0.8	128.0	510.0	0.51	0.46	0.05	0.8	159.0	633.0	1.18	0.99	0.19	0.6	223.0	891.0	3.13	2.93	0.20
ADM-PRO	0.3	233.0	931.0	0.86	0.83	0.03	0.4	550.0	2199.0	2.64	2.59	0.05	0.2	941.0	3764.0	10.68	10.61	0.07
ADM-LBFGS	0.7	143.0	570.0	0.55	0.52	0.03	0.8	184.0	733.0	0.94	0.89	0.05	0.8	305.0	1217.0	3.75	3.68	0.07
ADM-ILBFGS	0.7	139.0	554.0	0.55	0.52	0.03	1.0	185.0	736.0	0.95	0.90	0.05	0.9	294.0	1173.0	3.58	3.51	0.07

	$m=5000,\ n=1000,\ p=0.1$						$m=5000,\ n=1000,\ p=0.5$						$m=5000,\ n=1000,\ p=1.0$					
	β	Iter.	Int.It.	Time(s)	T-A(s)	T-M(s)	β	Iter.	Int.It.	Time(s)	T-A(s)	T-M(s)	β	Iter.	Int.It.	Time(s)	T-A(s)	T-M(s)
ADMM-1	40	16.0	80.0	153.79	–	–	161	15.0	74.0	612.32	–	–	250	15.0	74.0	984.23	–	–
ADMM-2	1.7	266.0	1057.0	11.67	11.41	0.26	3.9	526.0	1578.0	81.18	79.56	1.62	4.6	665.0	1995.0	142.19	138.47	3.72
ADM-PRO	1.1	344.0	1373.0	14.40	14.19	0.12	2.1	719.0	2859.0	111.96	111.41	0.55	2.6	920.0	3625.0	196.81	195.91	0.90
ADM-LBFGS	1.7	268.0	1064.0	11.22	11.10	0.12	3.7	531.0	1593.0	80.30	79.75	0.55	4.6	669.0	2007.0	142.71	141.81	0.90
ADM-ILBFGS	1.7	267.0	1061.0	11.02	10.90	0.12	3.8	528.0	1584.0	79.80	79.25	0.55	4.6	667.0	2001.0	141.63	140.73	0.90

	$m=10000,\ n=5000,\ p=0.1$						$m=10000,\ n=5000,\ p=0.5$						$m=10000,\ n=5000,\ p=1.0$					
	β	Iter.	Int.It.	Time(s)	T-A(s)	T-M(s)	β	Iter.	Int.It.	Time(s)	T-A(s)	T-M(s)	β	Iter.	Int.It.	Time(s)	T-A(s)	T-M(s)
ADMM-2	2.2	475.0	1892.0	193.10	181.67	11.43	4.3	1005.0	3015.0	1407.36	1316.69	90.67	5.3	1258.0	3774.0	2769.45	2554.13	215.32
ADM-PRO	0.6	736.0	2944.0	262.77	259.87	2.90	0.9	1405.0	5618.0	1843.21	1834.80	8.41	1.0	1780.0	7118.0	3664.39	3653.84	10.55
ADM-LBFGS	2.2	486.0	1935.0	178.95	176.05	2.90	4.0	1038.0	3114.0	1370.20	1361.79	8.41	4.5	1319.0	3957.0	2712.25	2701.70	10.55
ADM-ILBFGS	2.3	477.0	1898.0	175.98	173.08	2.90	4.2	1024.0	3072.0	1359.35	1350.94	8.41	5.0	1278.0	3834.0	2629.89	2619.34	10.55

'–' means that no such result in this item

4.5 Conclusions

In this chapter, we proposed a proximal ADMM where the proximal matrix derived from the Broyden family update for the general convex optimization problems (4.1.5) and (4.1.6). The x-subproblems of these convex problems can be rewritten as unconstrained quadratic programming problems in Subsections 4.2.2 and 4.2.3 as that in the Chapter 3, and hence the Hessian matrix of the augmented Lagrangian function is a constant matrix. The global convergences of such methods have also been established under some standard conditions. The numerical results for the l_1 regularized logistic regression problem are given to show the feasibility and effectiveness of the proposed algorithms.

Chapter 5

Alternating Direction Method of Multipliers with Variable Metric Indefinite Proximal Terms for Convex Optimization

5.1 Introduction

We consider the following convex composite optimization problem in this Chapter:

$$\min\left\{f(x)+g(y)\mid Ax+By=b,\ x\in\mathbb{R}^l,\ y\in\mathbb{R}^n\right\}, \tag{5.1.1}$$

where $f\colon\mathbb{R}^l\to\mathbb{R}\cup\{\infty\}$ and $g\colon\mathbb{R}^n\to\mathbb{R}\cup\{\infty\}$ are proper convex functions, $A\in\mathbb{R}^{m\times l}, B\in\mathbb{R}^{m\times n}$ and $b\in\mathbb{R}^m$. Various practical problems of science and engineering, such as machine learning [115], total variation denoising [102] and statistics [59] can be formulated as problem (5.1.1).

The augmented Lagrangian function $\mathcal{L}_\beta\colon\mathbb{R}^l\times\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}\cup\{\infty\}$ of (5.1.1) is defined as

$$\mathcal{L}_\beta(x,y,\lambda):=f(x)+g(y)-\langle\lambda,Ax+By-b\rangle+\frac{\beta}{2}\|Ax+By-b\|^2, \tag{5.1.2}$$

where $\lambda\in\mathbb{R}^m$ is the Lagrange multiplier for the linear constraints $Ax+By=b$ in (5.1.1), and β is a positive scalar. For a vector $z\in\mathbb{R}^n$ and a positive semidefinite matrix G, the norm $\|\cdot\|_G$ is defined by $\|z\|_G=\sqrt{z^\top Gz}$. In this chapter, even if $G\in\mathbb{R}^{n\times n}$ is not positive semidefinite, we denote $\|z\|_G^2=z^\top Gz$ for simplicity.

As discussed in the previous Chapters, how to choose the proximal term is also one of the important research topics for ADMM. The popular proximal term is always chosen as a constant matrix. He et al. [60] extended the work to allow the parameters β, proximal terms T and S to be replaced by some bounded sequences of positive definite matrices $\{T_k\}$ and $\{S_k\}$. The resulting ADMM is a variable metric proximal ADMM, which is also closely related to the inexact ADMM [38]. The convergences of such methods have been studied in [53, 82] but a better selection of the sequence $\{T_k\}$ has not been provided.

In Chapter 3, we constructed a variable positive semidefinite sequence $\{T_k\}$ with $T_k = B_k - \nabla^2_{xx}\mathcal{L}_\beta(x, y, \lambda)$ when f is quadratic. Note that $M = \nabla^2_{xx}\mathcal{L}_\beta(x, y, \lambda)$ is a constant matrix. They generated B_k via the BFGS update with respect to M at every iteration. In Chapter 4, we further extended such a proximal ADMM for more general convex optimization problems with the proximal term generated by the Broyden family update. In these ADMMs, the proximal terms T_k contain some second order information on the augmented Lagrangian function. Chapters 3 and 4 report some numerical results for LASSO and l_1 regularized logistic regression. The results show that the algorithms can get a solution faster than the general indefinite proximal ADMM whose proximal matrix T is fixed. Another interesting numerical result in Chapters 3 and 4 is that a variable indefinite sequence via the BFGS update also shows a good performance.

Inspired by the variable metric semi-proximal ADMM in Chapters 3, 4 and the indefinite proximal ADMM [63], it is worth considering ADMM with a sequence of indefinite proximal matrices. We call the resulting ADMM a variable metric indefinite proximal ADMM (VMIP-ADMM). Throughout our discussion, we always choose the stepsize α for the dual update to be 1 as that in [63], which is good enough for such methods in practice and simple for the convergence analysis.

We now introduce the whole update scheme of the VMIP-ADMM:

$$\begin{cases} x^{k+1} = \arg\min\limits_x \ \mathcal{L}_\beta(x, y^k, \lambda^k) + \dfrac{1}{2}\|x - x^k\|^2_S, & (5.1.3a) \\ y^{k+1} = \arg\min\limits_y \ \mathcal{L}_\beta(x^{k+1}, y, \lambda^k) + \dfrac{1}{2}\|y - y^k\|^2_{T_k}, & (5.1.3b) \\ \lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} - b), & (5.1.3c) \end{cases}$$

where S is a fixed positive semi-definite and T_k is possibly indefinite. Note that the proximal matrix S in x-subproblems could also be variable indefinite sequence $\{S_k\}$ with further conditions as $\{T_k\}$, and penalty parameter β could also be a positive sequence $\{\beta_k\}$ which shown in previously research [60]. However, for the sake of simplicity, we only consider the fixed matrix S and parameter β. Note also that the VMIP-ADMM can unify the several existing ADMMs.

- Let $S = 0$, $T_k \equiv 0$, VMIP-ADMM reduces to the classical ADMM;
- Let S and $T_k \equiv T$ be positive semidefinite matrices, VMIP-ADMM turns to be the semi-proximal ADMM;
- Let $\{T_k\}$ be a positive semidefinite sequence, that is, $T_k \succeq 0$ for all k. VMIP-ADMM becomes the variable semi-proximal ADMM;
- Let $S = 0$, $T_k \equiv T$ be an indefinite matrix, VMIP-ADMM covers the indefinite-proximal ADMM proposed in [63].

We present sufficient conditions on $\{T_k\}$ for the global convergence of VMIP-ADMM. The analysis technique of the proof is to split the indefinite matrix "T_k" into two positive semidefinite parts as $T_k = T^k_+ - T_-$. Moreover, we provide a construction of the indefinite term T_k via the BFGS update. We extend a useful theorem in Chapter 3 for a special case when y-subproblems (5.1.3b) are unconstrained quadratic programming problems. We construct the T_k with $T_k = B_k - M$, where M is the Hessian matrix of the augmented Lagrangian function (5.1.2) and B_k is generated by the BFGS update with respect to τM, $\tau < 1$. We also show that this construction of T_k satisfies the above conditions for the global convergence property when $\tau \in (0.75, 1)$. We use Table 5.1 to compare the conditions for global convergence with some existing methods.

Table 5.1: Comparisons among existing methods

method	S, T	α	τ
Classical ADMM [46, 49]	$S = T = 0$	$(0, (1+\sqrt{5})/2)$	-
Semi-proximal ADMM [40]	fixed positive semidefinite matrices	$(0, (1+\sqrt{5})/2)$	$(1, +\infty)$
Variable semi-proximal ADMM	variable positive semidefinite matrices	1	$(1, +\infty)$
Indefinite proximal ADMM [63]	fixed indefinite matrices	1	$(0.75, 1)$
Proposed method	variable indefinite matrices	1	$(0.75, 1)$

The remaining parts of the chapter are organized as follows. We first give notations and some preliminaries that will be useful for subsequent analysis in Section 5.2. Then we present sufficient conditions on the proximal matrices $\{T_k\}$ for the global convergence. In Section 5.3, we discuss the choices of proximal matrix T_k that guarantees the global convergence. We also show how to determine the value of τ. We conduct experiments on several real-world datasets and synthetic datasets for Lasso problem to validate our proposed algorithm in Section 5.4. Some conclusions and future works are given in Section 5.5.

5.2 Global convergence of the variable metric indefinite proximal ADMM

In this section, we show the global convergence of the variable metric indefinite proximal ADMM (5.1.3) (VMIP-ADMM) for problem (5.1.1). To this end, we first present optimality conditions of problem (5.1.1) and some useful properties which will be frequently used in our analysis. Then we give sufficient conditions on $\{T_k\}$ under which VMIP-ADMM converges globally.

5.2.1 Optimality conditions for problem (5.1.1)

Let $\Omega = \mathbb{R}^l \times \mathbb{R}^n \times \mathbb{R}^m$. The KKT conditions of problem (5.1.1) are written as

$$\begin{cases} \xi_x^* - A^\top \lambda^* = 0, & (5.2.1a) \\ \xi_y^* - B^\top \lambda^* = 0, & (5.2.1b) \\ Ax^* + By^* - b = 0, & (5.2.1c) \\ \xi_x^* \in \partial f(x^*), \;\; \xi_y^* \in \partial g(y^*). & (5.2.1d) \end{cases}$$

Let Ω^* be a set of (x^*, y^*, λ^*) satisfying the KKT conditions (5.2.1).

Throughout this chapter, we make the following assumption.

Assumption 5.2.1. *The set Ω^* of KKT points is non-empty.*

The optimality conditions of subproblems (5.1.3a) and (5.1.3b) can be obtained respectively that for all $x \in \mathbb{R}^l$,

$$(x - x^{k+1})^\top \left(\xi_x^{k+1} - A^\top\lambda^k + \beta A^\top(Ax^{k+1} + By^k - b) + S(x^{k+1} - x^k)\right) \geq 0,$$

and for all $\forall y \in \mathbb{R}^n$,

$$(y - y^{k+1})^\top \left(\xi_y^{k+1} - B^\top\lambda^k + \beta B^\top(Ax^{k+1} + By^{k+1} - b) + T_k(y^{k+1} - y^k)\right) \geq 0,$$

where $\xi_x^{k+1} \in \partial f(x^{k+1})$ and $\xi_y^{k+1} \in \partial g(y^{k+1})$.

Since $\lambda^{k+1} = \lambda^k - \beta(Ax^{k+1} + By^{k+1} - b)$, we have

$$-A^\top\lambda^k + \beta A^\top(Ax^{k+1} - b) = -A^\top\lambda^{k+1} - \beta A^\top By^{k+1}$$

and

$$-B^\top\lambda^k + \beta B^\top(Ax^{k+1} + By^{k+1} - b) = -B^\top\lambda^{k+1}.$$

Then the above optimality conditions of (5.1.3a) and (5.1.3b) can be written as, for all $x \in \mathbb{R}^l$,

$$(x - x^{k+1})^\top \left(\xi_x^{k+1} - A^\top\lambda^{k+1} + \beta A^\top B(y^k - y^{k+1}) + S(x^{k+1} - x^k)\right) \geq 0, \tag{5.2.2}$$

and for all $\forall y \in \mathbb{R}^n$,

$$(y - y^{k+1})^\top \left(\xi_y^{k+1} - B^\top\lambda^{k+1} + T_k(y^{k+1} - y^k)\right) \geq 0. \tag{5.2.3}$$

5.2.2 Notations and conditions on $\{T_k\}$

We use the following notations throughout this chapter:

$$u = \begin{pmatrix} x \\ y \end{pmatrix}, \; w = \begin{pmatrix} x \\ y \\ \lambda \end{pmatrix}.$$

Since the subdifferential mappings of the closed proper convex functions f and g are maximal monotone, there exist two positive semidefinite matrices Σ_f and Σ_g such that

$$(x - \hat{x})^\top(\xi_x - \hat{\xi}_x) \geq \|x - \hat{x}\|^2_{\Sigma_f}, \; \forall x, \hat{x} \in \mathbb{R}^l, \; \xi_x \in \partial f(x), \text{ and } \hat{\xi}_x \in \partial f(\hat{x}), \tag{5.2.4}$$

and

$$(y - \hat{y})^\top(\xi_y - \hat{\xi}_y) \geq \|y - \hat{y}\|^2_{\Sigma_g}, \; \forall y, \hat{y} \in \mathbb{R}^n, \; \xi_y \in \partial g(y), \text{ and } \hat{\xi}_y \in \partial g(\hat{y}). \tag{5.2.5}$$

Let $\Sigma \in \mathbb{R}^{(l+n)\times(l+n)}$ denote

$$\Sigma = \begin{pmatrix} \Sigma_f & 0 \\ 0 & \Sigma_g \end{pmatrix}.$$

We first give the conditions for S and the indefinite proximal sequence $\{T_k\}$ to guarantee the global convergence.

Condition 5.2.1. *The matrix S in* (5.1.3a) *satisfies*

(a) $S + \frac{1}{2}\Sigma_f \succeq 0$;

(b) $S + \Sigma_f + \beta A^\top A \succ 0$.

Moreover, for sequence $\{T_k\}$ generated in (5.1.3), *there exists a non-negative sequence $\{\gamma_k\}$ and positive semidefinite sequences $\{T_+^k\}$ and $\{T_-\}$ such that*

(c) $T_k = T_+^k - T_-$ *for all k;*

(d) $T_k + \Sigma_g + \beta B^\top B \succ 0$ *for all k;*

(e) $\frac{1}{1+\gamma_k} T_+^k \preceq T_+^{k+1} \preceq (1+\gamma_k) T_+^k$ *for all k,* $\sum\limits_{k=0}^{\infty} \gamma_k < \infty$;

(f) $T_{k+1} + \Sigma_g + \beta B^\top B \preceq (1+\gamma_k)(T_k + \Sigma_g + \beta B^\top B)$ *for all k;*

(g) $\exists\, c \in (0, 0.5)$, $T_k + \frac{3}{2}\Sigma_g - \frac{\gamma_{k-1}}{2} T_+^k - 2T_- + (\frac{3}{4} - \frac{1}{2}c)\beta B^\top B \succeq 0$ *for all k.*

Conditions (a) and (b) indicate that the proximal matrix S is allowed to be slight indefinite but no less than $-\frac{1}{2}\Sigma_f$. Condition (c) decomposes the indefinite matrix T_k to two positive semidefinite parts. For any fixed T, it can be written as $T = T_+ - T_-$. When T is positive semidefinite, we can set $T_+ = T$ and $T_- = 0$. If T is indefinite, for instance, a possible choice is $T_+ = 0, T_- = -T$. For the indefinite sequence $\{T_k\}$, it could be written as $T_k = T_+^k - T_-^k$ generally. The variable positive semidefinite sequence $\{T_-^k\}$ may relax the Condition 5.2.1 (g), which is under our future consideration. Note that we require the second part T_- be fixed. An example will be given in next section. This condition will play an important role in the main analysis. Condition (d) allows T_k to be indefinite. Conditions (e) and (f) are the boundness for positive semi-definite part T_+^k and indefinite T_k, respectively. Condition (g) is a requirement for global convergence and also an important condition for us to discuss the range of the indefiniteness.

For simplicity, we further define the following matrices. For all k,

$$P_k = \begin{pmatrix} S & 0 \\ 0 & T_k \end{pmatrix}, D_k = \begin{pmatrix} S & 0 & 0 \\ 0 & T_k & 0 \\ 0 & 0 & \frac{1}{\beta} I \end{pmatrix},$$

and

$$G_k = \begin{pmatrix} S + \Sigma_f & 0 & 0 \\ 0 & T_k + \Sigma_g + \beta B^\top B & 0 \\ 0 & 0 & \frac{1}{\beta} I \end{pmatrix}, \tag{5.2.6}$$

where S, T_k and β are given in (5.1.3).

Moreover, we also define the following matrices

$$\Gamma_k = T_+^k + T_- \quad \text{for all} \quad k, \tag{5.2.7a}$$

$$\Lambda_k = -\frac{\gamma_{k-1}}{2} T_+^k - 2T_- + \Sigma_g \quad \text{for all} \quad k, \tag{5.2.7b}$$

$$\Delta_k = T_k + \frac{3}{2}\Sigma_g - \frac{\gamma_{k-1}}{2} T_+^k - 2T_- + (\frac{3}{4} - \frac{1}{2}c)\beta B^\top B \quad \text{for all} \quad k, \tag{5.2.7c}$$

where $\{\gamma_k\}$ is a sequence satisfying Condition 5.2.1. Note that $\Gamma_k \succeq 0$ for all k.

5.2.3 Technical lemmas for convergence analysis of the VMIP-ADMM

In order to show that the sequence generated by VMIP-ADMM converges to a solution of (5.1.1) globally, we first give some properties for the sequence $\{w_k\} = \{(x^k, y^k, \lambda^k)\}$ generated by (5.1.3).

Lemma 5.2.1. *Let $\{w^k\}$ be generated by* (5.1.3). *Then, for given $w^* = (x^*, y^*, \lambda^*) \in \Omega^*$, we have*

$$\begin{aligned}&(w^{k+1} - w^*)^\top D_k (w^{k+1} - w^k) + \|u^{k+1} - u^*\|_\Sigma^2 \\ &\leq \beta(Ax^{k+1} - Ax^*)^\top (By^{k+1} - By^k).\end{aligned} \tag{5.2.8}$$

Proof. By taking $x = x^*$ and $y = y^*$ in the optimality conditions (5.2.2) and (5.2.3), respectively, we have

$$(x^{k+1} - x^*)^\top (\xi_x^{k+1} - A^\top \lambda^{k+1} + \beta A^\top B(y^k - y^{k+1}) + S(x^{k+1} - x^k)) \leq 0,$$

and

$$(y^{k+1} - y^*)^\top (\xi_y^{k+1} - B^\top \lambda^{k+1} + T_k(y^{k+1} - y^k)) \leq 0,$$

where $\xi_x^{k+1} \in \partial f(x^{k+1})$ and $\xi_y^{k+1} \in \partial g(y^{k+1})$.

The inequalities are further rearranged as

$$\begin{aligned}&(x^{k+1} - x^*)^\top S(x^{k+1} - x^k) + (x^{k+1} - x^*)^\top (\xi_x^{k+1} - A^\top \lambda^{k+1}) \\ &\leq \beta(Ax^{k+1} - Ax^*)^\top (By^{k+1} - By^k)\end{aligned} \tag{5.2.9}$$

and

$$(y^{k+1} - y^*)^\top T_k (y^{k+1} - y^k) + (y^{k+1} - y^*)^\top (\xi_y^{k+1} - B^\top \lambda^{k+1}) \leq 0. \tag{5.2.10}$$

Moreover, from (5.2.4)-(5.2.5) with $x = x^{k+1}$, $y = y^{k+1}$, $\hat{x} = x^*$ and $\hat{y} = y^*$, we have

$$(x^{k+1} - x^*)^\top (\xi_x^{k+1} - \xi_x^*) \geq \|x^{k+1} - x^*\|_{\Sigma_f}^2, \tag{5.2.11}$$

and

$$(y^{k+1} - y^*)^\top (\xi_y^{k+1} - \xi_y^*) \geq \|y^{k+1} - y^*\|_{\Sigma_g}^2, \tag{5.2.12}$$

where $\xi_x^* \in \partial f(x^*)$ and $\xi_y^* \in \partial g(y^*)$ satisfy the KKT conditions (5.2.1a) and (5.2.1b), respectively.

It then follows from (5.2.1a) and (5.2.11) that

$$\begin{aligned}&(x^{k+1}-x^*)^\top(\xi_x^{k+1}-A^\top\lambda^{k+1})\\&=(x^{k+1}-x^*)^\top(\xi_x^{k+1}-\xi_x^*)+(x^{k+1}-x^*)^\top(\xi_x^*-A^\top\lambda^{k+1})\\&\ge\|x^{k+1}-x^*\|_{\Sigma_f}^2+(Ax^{k+1}-Ax^*)^\top(\lambda^*-\lambda^{k+1}).\end{aligned}$$

Combining this inequality and (5.2.9), we have

$$\begin{aligned}&(x^{k+1}-x^*)^\top S(x^{k+1}-x^k)+(Ax^{k+1}-Ax^*)^\top(\lambda^*-\lambda^{k+1})+\|x^{k+1}-x^*\|_{\Sigma_f}^2\\&\le\beta(Ax^{k+1}-Ax^*)^\top(By^{k+1}-By^k).\end{aligned}\tag{5.2.13}$$

In a similar way, we have from (5.2.1b), (5.2.10) and (5.2.12) that

$$(y^{k+1}-y^*)^\top T_k(y^{k+1}-y^k)+(By^{k+1}-By^*)^\top(\lambda^*-\lambda^{k+1})+\|y^{k+1}-y^*\|_{\Sigma_g}^2\le 0.\tag{5.2.14}$$

Rearranging (5.1.3c), we have $Ax^{k+1}+By^{k+1}-b=\frac{1}{\beta}\left(\lambda^k-\lambda^{k+1}\right)$. It then follows from (5.2.1c) that

$$Ax^{k+1}+By^{k+1}-Ax^*-By^*=\frac{1}{\beta}(\lambda^k-\lambda^{k+1}).$$

Adding (5.2.13) and (5.2.14), and recalling the definition of D_k and Σ, it holds that

$$\begin{aligned}&(w^{k+1}-w^*)^\top D_k(w^{k+1}-w^k)+\|u^{k+1}-u^*\|_\Sigma^2\\&=(x^{k+1}-x^*)^\top S(x^{k+1}-x^k)+(y^{k+1}-y^*)^\top T_k(y^{k+1}-y^k)\\&\quad+\frac{1}{\beta}(\lambda^{k+1}-\lambda^k)^\top(\lambda^{k+1}-\lambda^*)+\|u^{k+1}-u^*\|_\Sigma^2\\&=(x^{k+1}-x^*)^\top S(x^{k+1}-x^k)+(y^{k+1}-y^*)^\top T_k(y^{k+1}-y^k)\\&\quad+(Ax^{k+1}+By^{k+1}-Ax^*-By^*)^\top(\lambda^*-\lambda^{k+1})+\|u^{k+1}-u^*\|_\Sigma^2\\&\le\beta(Ax^{k+1}-Ax^*)^\top(By^{k+1}-By^k),\end{aligned}$$

which completes the proof. ∎

The inequality (5.2.8) in Lemma 5.2.1 is further rearranged as follows.

Lemma 5.2.2. *Let $\{w^k\}$ be generated by (5.1.3). Then, for given $w^*=(x^*,y^*,\lambda^*)\in\Omega^*$, we have*

$$\begin{aligned}&2(w^{k+1}-w^*)^\top D_k(w^{k+1}-w^k)\\&\le 2(By^{k+1}-By^k)^\top(\lambda^k-\lambda^{k+1})-2\beta(By^{k+1}-By^*)^\top(By^{k+1}-By^k)\\&\quad-2\|u^{k+1}-u^*\|_\Sigma^2.\end{aligned}\tag{5.2.15}$$

Proof. Noting that $Ax^*+By^*-b=0$, the twice of the right hand of (5.2.8) is written as

$$\begin{aligned}&2\beta(Ax^{k+1}-Ax^*)^\top(By^{k+1}-By^k)\\&=2\beta(Ax^{k+1}+By^*-b+By^{k+1}-By^{k+1})^\top(By^{k+1}-By^k)\end{aligned}$$

$$
\begin{aligned}
&= 2\beta(Ax^{k+1}+By^{k+1}-b)^\top(By^{k+1}-By^k)-2\beta(By^{k+1}-By^*)^\top(By^{k+1}-By^k)\\
&= 2(By^{k+1}-By^k)^\top(\lambda^k-\lambda^{k+1})-2\beta(By^{k+1}-By^*)^\top(By^{k+1}-By^k),
\end{aligned}
$$

where the last equality follows from (5.1.3c). Then the assertion is directly obtained from (5.2.8). ■

Next we give a simple but important lemma.

Lemma 5.2.3. *For vectors $a, b \in \mathbb{R}^n$, and symmetric positive semidefinite matrices $M_1, M_2 \in \mathbb{R}^{n\times n}$, we have that*

$$
a^\top M_1 b - a^\top M_2 b \le \frac{1}{2}a^\top(M_1+M_2)a + \frac{1}{2}b^\top(M_1+M_2)b. \tag{5.2.16}
$$

Proof. For a positive semidefinite matrix M_1, we have

$$
0 \le \frac{1}{2}\|a-b\|_{M_1}^2 = \frac{1}{2}a^\top M_1 a + \frac{1}{2}b^\top M_1 b - a^\top M_1 b,
$$

which implies

$$
a^\top M_1 b \le \frac{1}{2}a^\top M_1 a + \frac{1}{2}b^\top M_1 b. \tag{5.2.17}
$$

In a similar way for M_2, we have

$$
-a^\top M_2 b \le \frac{1}{2}a^\top M_2 a + \frac{1}{2}b^\top M_2 b. \tag{5.2.18}
$$

The assertion immediately follows by adding (5.2.17) and (5.2.18). ■

In order to bound $(w^{k+1}-w^*)^\top D_k(w^{k+1}-w^k)$ further, we now give two technical lemmas to estimate upper-bounds for the crossing term $(By^{k+1}-By^k)^\top(\lambda^k-\lambda^{k+1})$ in (5.2.15).

Lemma 5.2.4. *Let $\{w^k\}$ be generated by the scheme* (5.1.3). *Suppose that the proximal sequence $\{T_k\}$ satisfies Condition 5.2.1. Then it holds that*

$$
\begin{aligned}
&(By^{k+1}-By^k)^\top(\lambda^k-\lambda^{k+1})\\
&\le \frac{1}{2}\|y^{k-1}-y^k\|_{\Gamma_{k-1}}^2 - \frac{1}{2}\|y^{k+1}-y^k\|_{\Gamma_k}^2 - \|y^{k+1}-y^k\|_{\Lambda_k}^2,
\end{aligned} \tag{5.2.19}
$$

where Γ_k and Λ_k are defined in (5.2.7).

Proof. From the optimality condition (5.2.3) for y^{k+1}, we can easily derive the optimality condition for y^k as

$$
(y-y^k)^\top(\xi_y^k - B^\top\lambda^k + T_{k-1}(y^k-y^{k-1})) \ge 0, \quad \forall y \in \mathbb{R}^n. \tag{5.2.20}
$$

Choosing $y = y^k$ in (5.2.3), we have

$$
0 \le (y^k-y^{k+1})^\top(\xi_y^{k+1} - B^\top\lambda^{k+1} + T_k(y^{k+1}-y^k))
$$

$$= (y^{k+1} - y^k)^\top(-\xi_y^{k+1} + B^\top\lambda^{k+1} - T_k(y^{k+1} - y^k)). \tag{5.2.21}$$

Moreover, letting $y = y^{k+1}$ in (5.2.20), we have

$$0 \leq (y^{k+1} - y^k)^\top(\xi_y^k - B^\top\lambda^k + T_{k-1}(y^k - y^{k-1})). \tag{5.2.22}$$

Summing inequalities (5.2.21) and (5.2.22), we obtain that

$$\begin{aligned}&(By^{k+1} - By^k)^\top(\lambda^k - \lambda^{k+1})\\ &\leq -\|y^{k+1} - y^k\|_{T_k}^2 + (y^{k+1} - y^k)^\top T_{k-1}(y^k - y^{k-1}) - \|y^{k+1} - y^k\|_{\Sigma_g}^2.\end{aligned} \tag{5.2.23}$$

Recall that $T_{k-1} = T_+^{k-1} - T_-$ from (c) in Condition 5.2.1 and $T_+^{k-1}, T_- \succeq 0$. Then we have

$$\begin{aligned}&(y^{k+1} - y^k)^\top T_{k-1}(y^k - y^{k-1})\\ &= (y^{k+1} - y^k)^\top T_+^{k-1}(y^k - y^{k-1}) - (y^{k+1} - y^k)^\top T_-(y^k - y^{k-1})\\ &\leq \frac{1}{2}\|y^{k+1} - y^k\|_{T_+^{k-1}+T_-}^2 + \frac{1}{2}\|y^{k-1} - y^k\|_{T_+^{k-1}+T_-}^2,\end{aligned} \tag{5.2.24}$$

where the inequality follows from (5.2.16) with $a = (y^{k+1} - y^k)$, $b = (y^k - y^{k-1})$, $M_1 = T_+^{k-1}$ and $M_2 = T_-$.

We then have from (5.2.23) that

$$\begin{aligned}&(By^{k+1} - By^k)^\top(\lambda^k - \lambda^{k+1})\\ &\leq -\|y^{k+1} - y^k\|_{T_k}^2 + (y^{k+1} - y^k)^\top T_{k-1}(y^k - y^{k-1}) - \|y^{k+1} - y^k\|_{\Sigma_g}^2\\ &\leq -\|y^{k+1} - y^k\|_{T_+^k - T_-}^2 + \frac{1}{2}\|y^{k+1} - y^k\|_{T_+^{k-1}+T_-}^2 + \frac{1}{2}\|y^{k-1} - y^k\|_{T_+^{k-1}+T_-}^2\\ &\quad - \|y^{k+1} - y^k\|_{\Sigma_g}^2\\ &\leq -\|y^{k+1} - y^k\|_{T_+^k - T_-}^2 + \frac{1}{2}\|y^{k+1} - y^k\|_{(1+\gamma_{k-1})T_+^k+T_-}^2 + \frac{1}{2}\|y^{k-1} - y^k\|_{T_+^{k-1}+T_-}^2\\ &\quad - \|y^{k+1} - y^k\|_{\Sigma_g}^2\\ &= \frac{1}{2}\|y^{k-1} - y^k\|_{T_+^{k-1}+T_-}^2 - \frac{1}{2}\|y^{k+1} - y^k\|_{T_+^k+T_-}^2 - \|y^{k+1} - y^k\|_{-\frac{\gamma_{k-1}}{2}T_+^k - 2T_- + \Sigma_g}^2\\ &= \frac{1}{2}\|y^{k-1} - y^k\|_{\Gamma_{k-1}}^2 - \frac{1}{2}\|y^{k+1} - y^k\|_{\Gamma_k}^2 - \|y^{k+1} - y^k\|_{\Lambda_k}^2,\end{aligned}$$

where the second inequality follows from $T_k = T_+^k - T_-$ and (5.2.24), the third inequality follows from Condition 5.2.1 (d), and the last equality is from the definitions (5.2.7a) and (5.2.7b). Then it shows the assertion (5.2.19). ■

Besides Lemma 5.2.4, we can derive another estimation for the term $(By^{k+1} - By^k)^\top(\lambda^k - \lambda^{k+1})$, which is the result in [63, Lemma 4.4].

Lemma 5.2.5. *Let $\{w^k\}$ be generated by the scheme (5.1.3). Then, for any $c \in (0, 0.5)$, it holds that*

$$\begin{aligned}(By^{k+1} - By^k)^\top(\lambda^k - \lambda^{k+1}) \leq &\left(\frac{1}{4} + \frac{1}{2}c\right)\beta\|By^{k+1} - By^k\|^2\\ &+ (1 - c)\frac{1}{\beta}\|\lambda^{k+1} - \lambda^k\|^2.\end{aligned} \tag{5.2.25}$$

Proof. See [63, Lemma 4.4]. ■

Based on the above two lemmas for $(By^{k+1}-By^k)^\top(\lambda^k-\lambda^{k+1})$, we can further bound $(w^{k+1}-w^*)^\top D_k(w^{k+1}-w^k)$ in (5.2.15) of Lemma 5.2.2.

Lemma 5.2.6. *Let $\{w^k\}$ be generated by* (5.1.3). *Suppose that the proximal sequence $\{T_k\}$ satisfies Condition 5.2.1. Then, for given $w^*=(x^*,y^*,\lambda^*)\in\Omega^*$, we have*

$$\begin{aligned}&2(w^{k+1}-w^*)^\top D_k(w^{k+1}-w^k)\\&\le \frac{1}{2}\|y^{k-1}-y^k\|^2_{\Gamma_{k-1}}-\frac{1}{2}\|y^{k+1}-y^k\|^2_{\Gamma_k}-\|y^{k+1}-y^k\|^2_{\Lambda_k}\\&\quad+\left(\frac{1}{4}+\frac{1}{2}c\right)\beta\|By^{k+1}-By^k\|^2+\frac{(1-c)}{\beta}\|\lambda^{k+1}-\lambda^k\|^2\\&\quad-2\beta(By^{k+1}-By^*)^\top(By^{k+1}-By^k)-2\|u^{k+1}-u^*\|^2_{\Sigma}.\end{aligned}\tag{5.2.26}$$

Proof. The term $2(By^{k+1}-By^k)^\top(\lambda^k-\lambda^{k+1})$ in inequality (5.2.15) can be bounded by the above lemmas (5.2.19) and (5.2.25), and then the assertion is obtained. ■

5.2.4 Global Convergence of the VMIP-ADMM

In this subsection we show the global convergence based on the results in the previous subsection and Condition 5.2.1. Firstly, we obtain the following contractive result, which will play a key role in proving the convergence of (5.1.3).

Lemma 5.2.7. *Let $w^*=(x^*,y^*,\lambda^*)\in\Omega^*$, and let $\{w^k\}$ be generated by the scheme* (5.1.3). *Suppose that the proximal sequence $\{T_k\}$ satisfies Condition 5.2.1. Then we have*

$$\begin{aligned}&\|w^k-w^*\|^2_{G_k}+\frac{1}{2}\|y^{k-1}-y^k\|^2_{\Gamma_{k-1}}-\left(\|w^{k+1}-w^*\|^2_{G_k}+\frac{1}{2}\|y^{k+1}-y^k\|^2_{\Gamma_k}\right)\\&\ge \underbrace{\|x^{k+1}-x^k\|^2_{S+\frac{1}{2}\Sigma_f}+\|y^{k+1}-y^k\|^2_{\Delta_k}+\frac{c}{\beta}\|\lambda^{k+1}-\lambda^k\|^2}_{\Upsilon_k},\end{aligned}\tag{5.2.27}$$

where Γ_k and Δ_k are given in (5.2.7).

Proof. By the identity $\|a+b\|^2=\|a\|^2-\|b\|^2+2(a+b)^\top b$, we get

$$\begin{aligned}&\|w^{k+1}-w^*\|^2_{D_k}+\beta\|By^{k+1}-By^*\|^2\\&=\|w^k-w^*\|^2_{D_k}+\beta\|By^k-By^*\|^2-\left(\|w^{k+1}-w^k\|^2_{D_k}+\beta\|By^{k+1}-By^k\|^2\right)\\&\quad+2(w^{k+1}-w^*)^\top D_k(w^{k+1}-w^k)+2\beta(By^{k+1}-By^*)^\top(By^{k+1}-By^k).\end{aligned}\tag{5.2.28}$$

Since the term $2(w^{k+1}-w^*)^\top D_k(w^{k+1}-w^k)$ in equality (5.2.28) can be bounded by (5.2.26) in Lemma 5.2.6, we can rearrange (5.2.28) as

$$
\begin{aligned}
&\|w^{k+1}-w^*\|_{D_k}^2+\beta\|By^{k+1}-By^*\|^2\\
&\le \|w^k-w^*\|_{D_k}^2+\beta\|By^k-By^*\|^2-\|w^{k+1}-w^k\|_{D_k}^2-\beta\|By^{k+1}-By^k\|^2\\
&\quad+\frac{1}{2}\|y^{k-1}-y^k\|_{\Gamma_{k-1}}^2-\frac{1}{2}\|y^{k+1}-y^k\|_{\Gamma_k}^2-\|y^{k+1}-y^k\|_{\Lambda_k}^2\\
&\quad+\left(\frac{1}{4}+\frac{c}{2}\right)\beta\|By^{k+1}-By^k\|^2+\frac{1-c}{\beta}\|\lambda^{k+1}-\lambda^k\|^2-2\|u^{k+1}-u^*\|_\Sigma^2\\
&=\|w^k-w^*\|_{D_k}^2+\beta\|By^k-By^*\|^2\\
&\quad-\|u^{k+1}-u^k\|_{P_k}^2-\left(\frac{3}{4}-\frac{1}{2}c\right)\beta\|By^{k+1}-By^k\|^2-\frac{c}{\beta}\|\lambda^{k+1}-\lambda^k\|^2\\
&\quad+\frac{1}{2}\|y^{k-1}-y^k\|_{\Gamma_{k-1}}^2-\frac{1}{2}\|y^{k+1}-y^k\|_{\Gamma_k}^2-\|y^{k+1}-y^k\|_{\Lambda_k}^2-2\|u^{k+1}-u^*\|_\Sigma^2,
\end{aligned}\tag{5.2.29}
$$

where the last equality follows from the definitions of P_k and D_k in (5.2.6). Rearranging (5.2.29) further, we have

$$
\begin{aligned}
&\|w^{k+1}-w^*\|_{D_k}^2+\beta\|By^{k+1}-By^*\|^2+\frac{1}{2}\|y^{k+1}-y^k\|_{\Gamma_k}^2\\
&\le \|w^k-w^*\|_{D_k}^2+\beta\|By^k-By^*\|^2+\frac{1}{2}\|y^{k-1}-y^k\|_{\Gamma_{k-1}}^2-2\|u^{k+1}-u^*\|_\Sigma^2\\
&\quad-\left(\|u^{k+1}-u^k\|_{P_k}^2+\frac{c}{\beta}\|\lambda^{k+1}-\lambda^k\|^2+\|y^{k+1}-y^k\|_{\Lambda_k+(\frac{3}{4}-\frac{1}{2}c)\beta B^\top B}^2\right),
\end{aligned}
$$

that is,

$$
\begin{aligned}
&\|w^k-w^*\|_{D_k}^2+\beta\|By^k-By^*\|^2+\frac{1}{2}\|y^{k-1}-y^k\|_{\Gamma_{k-1}}^2+\|u^k-u^*\|_\Sigma^2\\
&\quad-\left(\|w^{k+1}-w^*\|_{D_k}^2+\beta\|By^{k+1}-By^*\|^2+\frac{1}{2}\|y^{k+1}-y^k\|_{\Gamma_k}^2+\|u^{k+1}-u^*\|_\Sigma^2\right)\\
&\ge \|u^{k+1}-u^k\|_{P_k}^2+\frac{c}{\beta}\|\lambda^{k+1}-\lambda^k\|^2+\|y^{k+1}-y^k\|_{\Lambda_k+(\frac{3}{4}-\frac{1}{2}c)\beta B^\top B}^2\\
&\quad+\|u^k-u^*\|_\Sigma^2-\|u^{k+1}-u^*\|_\Sigma^2+2\|u^{k+1}-u^*\|_\Sigma^2.
\end{aligned}\tag{5.2.30}
$$

From the definition of G_k in (5.2.6), inequality (5.2.30) can be written as

$$
\begin{aligned}
&\|w^k-w^*\|_{G_k}^2+\frac{1}{2}\|y^{k-1}-y^k\|_{\Gamma_{k-1}}^2-\left(\|w^{k+1}-w^*\|_{G_k}^2+\frac{1}{2}\|y^{k+1}-y^k\|_{\Gamma_k}^2\right)\\
&\ge \|u^{k+1}-u^k\|_{P_k}^2+\frac{c}{\beta}\|\lambda^{k+1}-\lambda^k\|^2+\|y^{k+1}-y^k\|_{\Lambda_k+(\frac{3}{4}-\frac{1}{2}c)\beta B^\top B}^2\\
&\quad+\|u^k-u^*\|_\Sigma^2+\|u^{k+1}-u^*\|_\Sigma^2\\
&\ge \|u^{k+1}-u^k\|_{P_k}^2+\frac{c}{\beta}\|\lambda^{k+1}-\lambda^k\|^2+\|y^{k+1}-y^k\|_{\Lambda_k+(\frac{3}{4}-\frac{1}{2}c)\beta B^\top B}^2\\
&\quad+\frac{1}{2}\|u^{k+1}-u^k\|_\Sigma^2\\
&=\|x^{k+1}-x^k\|_{S+\frac{\Sigma_f}{2}}^2+\|y^{k+1}-y^k\|_{T_k+\Lambda_k+(\frac{3}{4}-\frac{c}{2})\beta B^\top B+\frac{\Sigma_g}{2}}^2+\frac{c}{\beta}\|\lambda^{k+1}-\lambda^k\|^2,
\end{aligned}
$$

where the second inequality follows from the well-known inequality $\|a\|_M^2+\|b\|_M^2\ge\frac{1}{2}\|a-b\|_M^2$ with $M=\Sigma$, $a=u^k-u^*$ and $b=u^{k+1}-u^*$.

From the definitions (5.2.7b) and (5.2.7c), we have that

$$\Delta_k = T_k + \frac{1}{2}\Sigma_g + \Lambda_k + (\frac{3}{4} - \frac{1}{2}c)\beta B^\top B.$$

Thus the proof is completed. ■

Condition 5.2.1 (a) implies $\|x^{k+1}-x^k\|^2_{S+\frac{1}{2}\Sigma_f} \geq 0$ for all k. Moreover, Condition 5.2.1 (g) implies $\|y^{k+1} - y^k\|^2_{\Delta_k} \geq 0$ for all k. Therefore, Υ_k in (5.2.27) is always nonnegative, which indicates the contraction of the sequence $\{w_k\}$.

It follows from the definition of $\{G_k\}$ and Condition 5.2.1 (a), (c) and (e) that $0 \preceq G_{k+1} \preceq (1+\gamma_k)G_k$ for all k. We define two constants C_s and C_p as follows:

$$C_s\colon = \sum_{k=0}^{\infty} \gamma_k \ \text{ and } \ C_p\colon = \prod_{k=0}^{\infty}(1+\gamma_k).$$

From the assumption $\sum_0^\infty \gamma_k < \infty$ and $\gamma_k \geq 0$, we have $0 \leq C_s < \infty$ and $1 \leq C_p < \infty$. Moreover, we can easily get

$$0 \preceq G_k \preceq C_p G_0, \ \ \forall k \geq 0,$$

which means that the sequences $\{G_k\}$ is bounded.

Now we give the main convergent theorem of this subsection.

Theorem 5.2.1. *Let $w^* = (x^*, y^*, \lambda^*) \in \Omega^*$, and let $\{w^k\}$ be a sequence generated by (5.1.3). Suppose that $\{T_k\}$ is a sequence satisfying Condition 5.2.1. Then the sequence $\{w^k\}$ converges to a point $w^* \in \Omega^*$.*

Proof. First we show that the sequence $\{w^k\}$ is bounded. Since $0 \preceq G_{k+1} \preceq (1+\gamma_k)G_k$, we have

$$\|w^{k+1} - w^*\|^2_{G_{k+1}} \leq (1+\gamma_k)\|w^{k+1} - w^*\|^2_{G_k}. \tag{5.2.31}$$

Combining the inequality (5.2.31) with (5.2.27) in Lemma 5.2.7, we have

$$\begin{aligned}
&\|w^{k+1} - w^*\|^2_{G_{k+1}} + \frac{1}{2}\|y^{k+1} - y^k\|^2_{\Gamma_k} \\
&\overset{(5.2.31)}{\leq} (1+\gamma_k)\left(\|w^{k+1} - w^*\|^2_{G_k} + \frac{1}{2}\|y^{k+1} - y^k\|^2_{\Gamma_k}\right) \\
&\overset{(5.2.27)}{\leq} (1+\gamma_k)\left(\|w^{k} - w^*\|^2_{G_k} + \frac{1}{2}\|y^{k-1} - y^k\|^2_{\Gamma_{k-1}}\right) - (1+\gamma_k)\Upsilon_k \\
&\leq (1+\gamma_k)\left(\|w^{k} - w^*\|^2_{G_k} + \frac{1}{2}\|y^{k-1} - y^k\|^2_{\Gamma_{k-1}}\right) - \Upsilon_k.
\end{aligned} \tag{5.2.32}$$

It then follows that for all k,

$$\|w^{k+1} - w^*\|^2_{G_{k+1}} + \frac{1}{2}\|y^{k+1} - y^k\|^2_{\Gamma_k} \leq \left(\prod_{i=0}^{k}(1+\gamma_i)\right)\left(\|w^0 - w^*\|^2_{G_0} + \frac{1}{2}\|y^0 - y^1\|^2_{\Gamma_0}\right)$$

$$\leq C_p \left(\|w^0 - w^*\|^2_{G_0} + \frac{1}{2}\|y^0 - y^1\|^2_{\Gamma_0} \right). \tag{5.2.33}$$

Note that

$$\|w^{k+1} - w^*\|^2_{G_{k+1}} = \|x^{k+1} - x^*\|^2_{S+\Sigma_f} + \|y^{k+1} - y^*\|^2_{T_k+\Sigma_g+\beta B^\top B} + \frac{1}{\beta}\|\lambda^{k+1} - \lambda^*\|^2, \tag{5.2.34}$$

where $(T_k + \Sigma_g + \beta B^\top B)$ is positive definite from Condition 5.2.1, and $C_p \left(\|w^0 - w^*\|^2_{G_0} + \frac{1}{2}\|y^0 - y^1\|^2_{\Gamma_0}\right)$ is a constant. It then follows from (5.2.33) that $\{y^k\}$ and $\{\lambda^k\}$ are bounded. We now show that $\{x^k\}$ is also bounded.

From (5.2.32) and (5.2.33), we have

$$\begin{aligned}
\Upsilon_k &= \|x^{k+1} - x^k\|^2_{S+\frac{1}{2}\Sigma_f} + \|y^{k+1} - y^k\|^2_{\Delta_k} + \frac{c}{\beta}\|\lambda^{k+1} - \lambda^k\|^2 \\
&\leq \|w^k - w^*\|^2_{G_k} - \|w^{k+1} - w^*\|^2_{G_{k+1}} + \frac{1}{2}\|y^{k-1} - y^k\|^2_{\Gamma_{k-1}} \\
&\quad - \frac{1}{2}\|y^{k+1} - y^k\|^2_{\Gamma_k} + \gamma_k \left(\|w^k - w^*\|^2_{G_k} + \frac{1}{2}\|y^{k-1} - y^k\|^2_{\Gamma_{k-1}} \right) \\
&\leq \|w^k - w^*\|^2_{G_k} - \|w^{k+1} - w^*\|^2_{G_{k+1}} + \frac{1}{2}\|y^{k-1} - y^k\|^2_{\Gamma_{k-1}} \\
&\quad - \frac{1}{2}\|y^{k+1} - y^k\|^2_{\Gamma_k} + C_p \left(\|w^0 - w^*\|^2_{G_0} + \frac{1}{2}\|y^0 - y^1\|^2_{\Gamma_0} \right).
\end{aligned}$$

Summing up the inequalities, we obtain

$$\begin{aligned}
&\sum_{k=1}^{\infty} \left(\|x^{k+1} - x^k\|^2_{S+\frac{1}{2}\Sigma_f} + \|y^{k+1} - y^k\|^2_{\Delta_k} + \frac{c}{\beta}\|\lambda^{k+1} - \lambda^k\|^2 \right) \\
&\leq \|w^0 - w^*\|^2_{G_0} + \frac{1}{2}\|y^0 - y^1\|^2_{\Gamma_0} + \left(\sum_{k=0}^{\infty} \gamma_k \right) C_p \left(\|w^0 - w^*\|^2_{G_0} + \frac{1}{2}\|y^0 - y^1\|^2_{\Gamma_0} \right) \\
&\leq (1 + C_s C_p) \left(\|w^0 - w^*\|^2_{G_0} + \frac{1}{2}\|y^0 - y^1\|^2_{\Gamma_0} \right).
\end{aligned}$$

Since $(1 + C_s C_p)\left(\|w^0 - w^*\|^2_{G_0} + \frac{1}{2}\|y^0 - y^1\|^2_{\Gamma_0}\right)$ is a finite constant, we have

$$\lim_{k\to\infty} \|x^{k+1} - x^k\|^2_{S+\frac{1}{2}\Sigma_f} + \|y^{k+1} - y^k\|^2_{\Delta_k} + \frac{c}{\beta}\|\lambda^{k+1} - \lambda^k\|^2 = 0,$$

which indicates that

$$\lim_{k\to\infty} \|\lambda^{k+1} - \lambda^k\| = \lim_{k\to\infty} \beta\|Ax^{k+1} + By^{k+1} - b\| = 0. \tag{5.2.35}$$

Note that $Ax^* + By^* - b = 0$, and

$$\begin{aligned}
\|Ax^{k+1} - Ax^*\| &= \|Ax^{k+1} + By^{k+1} - b - B(y^{k+1} - y^*)\| \\
&\leq \|Ax^{k+1} + By^{k+1} - b\| + \|B(y^{k+1} - y^*)\|.
\end{aligned}$$

It then follows from (5.2.35) that $\|A(x^{k+1} - x^*)\|$ is bounded. Moreover, inequalities (5.2.33) and (5.2.34) imply $\|x^{k+1} - x^*\|^2_{S+\Sigma_f}$ is bounded. Therefore $\|x^{k+1} - x^*\|^2_{S+\Sigma_f+\beta A^\top A}$ is bounded since

$$\|x^{k+1} - x^*\|^2_{S+\Sigma_f+\beta A^\top A} = \|x^{k+1} - x^*\|^2_{S+\Sigma_f} + \beta\|A(x^{k+1} - x^*)\|^2.$$

From the positive definiteness of $S + \Sigma_f + \beta A^\top A$ in Condition 5.2.1 (b), it shows that $\{x^k\}$ is also bounded. Consequently, the sequence $\{w^k\}$ is bounded.

Next we should show that any cluster point of the sequence $\{w^k\}$ is an optimal solution of (5.1.1) and the sequence $\{w^k\}$ has only one cluster point. This can be done in a way similar to the proof of that in Chapter 3. ■

5.3 VMIP-ADMM with the BFGS update

As shown in Chapters 3 and 4, a special variable metric proximal term via the BFGS update can get a solution faster on the iteration and CPU time than the proximal ADMM [40, 63] with a fixed proximal matrix T. Moreover, in their experiments, a slightly indefinite variable also performs well without the theoretical analysis. Note that this choice should have an assumption that the y-subproblems (5.1.3b) should be unconstrained quadratic programming problems. Based on the analysis above and the previous studies, we propose indefinite proximal terms $\{T_k\}$ updated by the BFGS update, and show that $\{T_k\}$ satisfies Condition 5.2.1.

5.3.1 Construction of the indefinite proximal matrix T_k via the BFGS update

Inspired by the semidefinite proximal ADMM with the BFGS update in Chapters 3 and 4, we construct the indefinite matrix T_k by the BFGS update.

We first explain the pure BFGS update for the following unconstrained quadratic optimization:

$$\min \frac{1}{2} x^\top M x,$$

where $M \in \mathbb{R}^{n\times n}$ is a positive definite matrix. Let $s \in \mathbb{R}^n$ and $l = Ms$. Note that $s^\top l > 0$ when $s \neq 0$. The BFGS update generates a sequence of approximate matrices $\{B_k\}$ of M, and its inverse $H_k = B_k^{-1}$. For a given matrix B_k, the BFGS update generates B_{k+1}^{BFGS} and H_{k+1}^{BFGS} with s and l as follows:

$$B_{k+1}^{\text{BFGS}} = B_k + \frac{ll^\top}{l^\top s} - \frac{B_k ss^\top B_k^\top}{s^\top B_k s}, \tag{5.3.1}$$

$$H_{k+1}^{\text{BFGS}} = \left(I - \frac{sl^\top}{s^\top l}\right) H_k \left(I - \frac{ls^\top}{s^\top l}\right) + \frac{ss^\top}{s^\top l}. \tag{5.3.2}$$

Note that B_{k+1}^{BFGS} and H_{k+1}^{BFGS} are positive definite whenever $B_k, H_k \succ 0$ since $s^\top l > 0$. Note also that $H_{k+1}^{\text{BFGS}} l = s = M^{-1} l$.

We now explain how to construct T_k via the BFGS update. Throughout this section we suppose that g in the objective function (5.1.1) is a convex quadratic function. Then y-subproblems (5.1.3b) are unconstrained quadratic programming problems, and the Hessian matrix of the augmented Lagrangian function (5.1.2) is a constant matrix given as

$$M\colon = \nabla^2_{yy}\mathcal{L}_\beta(x,y,\lambda) = \bar{M} + \beta B^\top B,$$

where $\bar{M}\colon = \nabla^2_{yy} g(y)$. Note that M is always positive semidefinite since $\bar{M} \succeq 0$.

We consider a perturbed matrix $M^\delta\colon = M + \delta I \succ 0$ with a sufficiently small $\delta > 0$, and construct an approximate matrix B_k of M^δ via the BFGS update (5.3.1). Let $s_k = x^{k+1} - x^k$, where $\{x^k\}$ is a sequence generated by (5.1.3). We propose that $\{B_k\}$ is generated as

$$B_{k+1} = B_k + c_k\left(\frac{\tilde{l}_k\tilde{l}_k^\top}{\tilde{l}_k^\top s_k} - \frac{B_k s_k s_k^\top B_k^\top}{s_k^\top B_k s_k}\right), \tag{5.3.3}$$

where $\tilde{l}_k = M s_k + \delta s_k = M^\delta s_k$, and $\{c_k\}$ is a sequence such that $c_k \in [0,1]$, and $\sum\limits_{k=0}^{\infty} c_k < \infty$. We can rewrite the update formula (5.3.3) as

$$B_{k+1} = B_k + c_k(B_{k+1}^{\mathrm{BFGS}} - B_k),$$

where B_{k+1}^{BFGS} is updated by the pure BFGS update (5.3.1) with respect to M^δ at every iteration. Note that $B_{k+1} = B_{k+1}^{\mathrm{BFGS}}$ when $c_k = 1$.

We then propose the following construction of T_k via the BFGS update.

Construction of T_k via the BFGS update

1 Let $\delta \in (0,\infty)$, $\tau \in (\frac{3}{4}, 1)$ and $B_0 \succeq \tau M$;

2 Let c_k be a sequence such that $c_k \in [0,1]$ and $\sum\limits_{k=0}^{\infty} c_k < \infty$;

3 If $s_k \neq 0$, then set $\tilde{l}_k = M^\delta s_k$ and update B_{k+1} via

$$B_{k+1} = B_k + c_k\left(\frac{\tilde{l}_k\tilde{l}_k^\top}{\tilde{l}_k^\top s_k} - \frac{B_k s_k s_k^\top B_k^\top}{s_k^\top B_k s_k}\right);$$

4 Otherwise

$$B_{k+1} = B_k;$$

5 Construct T_{k+1} as

$$T_{k+1} = B_{k+1} - M.$$

As shown in Chapter 3, $\delta = 0$ also works well in the experiment results. However, the positive requirement is necessary for the convergence analysis.

5.3.2 Discussion on the Condition 5.2.1 for the indefinite matrix T_k

We now consider matrices $\{T_+^k\}$ and T_- such that $T_k = T_+^k - T_-$, $T_+^k \succeq 0$, $T_- \succeq 0$ in Condition 5.2.1 (c). Let

$$T_+^k = B_k - \tau M \ \text{ and } \ T_- = (1-\tau)M, \ \text{ with } \ \tau \in [0,1).$$

Note that $T_k = T_+^k - T_- = B_k - M$ and $T_- \succeq 0$. Thus we only show that T_+^k is positive semidefinite.

To this end, we give an extension result related to Theorem 3.2.1 in Chapter 3.

Lemma 5.3.1. *Let $M \in \mathbb{R}^{n\times n}$ be a positive definite matrix. Let $s \in \mathbb{R}^n$ such that $s \neq 0$, and let $l = Ms$. If a given matrix $H_k \in \mathbb{R}^{n\times n}$ satisfies $H_k \preceq \tau_1 M^{-1}$ with $\tau_1 \geq 1$, then H_{k+1}^{BFGS} which is generated by the BFGS update* (5.3.2) *with respect to M also satisfies $H_{k+1}^{\mathrm{BFGS}} \preceq \tau_1 M^{-1}$.*

Proof. Let v be an arbitrary nonzero vector in $\mathbb{R}^n$, and $\Psi = \{z \in \mathbb{R}^n \mid s^\top z = 0\}$. As shown in 3.2.1, there exist $c \in \mathbb{R}$ and $z \in \Psi$ such that $v = cl + z$. Together with $H_{k+1}^{\mathrm{BFGS}} l = s = M^{-1}l$ and $s^\top z = 0$, we can obtain that for any $\tau_1 \geq 1$,

$$\begin{aligned}
v^\top H_{k+1}^{\mathrm{BFGS}} v &= (cl+z)^\top H_{k+1}^{\mathrm{BFGS}}(cl+z) \\
&= c^2 l^\top s + 2cs^\top z + z^\top H_{k+1}^{\mathrm{BFGS}} z \\
&= c^2 l^\top M^{-1} l + z^\top H_{k+1}^{\mathrm{BFGS}} z \\
&= c^2 l^\top M^{-1} l + z^\top H_k z - 2z^\top \left(\frac{sl^\top}{s^\top l} H_k\right) z + z^\top \left(\frac{sl^\top}{s^\top l} H_k \frac{ls^\top}{s^\top l}\right) z + \frac{z^\top s s^\top z}{s^\top l} \\
&= c^2 l^\top M^{-1} l + z^\top H_k z \\
&\leq c^2 l^\top \tau_1 M^{-1} l + z^\top \tau_1 M^{-1} z \\
&= (cl+z)^\top \tau_1 M^{-1}(cl+z) - 2\tau_1 c l^\top M^{-1} z \\
&= v^\top \tau_1 M^{-1} v,
\end{aligned}$$

where the forth equality follows from (5.3.2), and the inequality follows from the positive definiteness of M^{-1} and the assumption that $H_k \preceq \tau_1 M^{-1}$. Since v is arbitrary, we have $H_{k+1}^{\mathrm{BFGS}} \preceq \tau_1 M^{-1}$. ■

Lemma 5.3.1 implies that $B_{k+1}^{\mathrm{BFGS}} \succeq \tau M^\delta$ when $B_k \succeq \tau M^\delta$ with $\tau = \frac{1}{\tau_1} \leq 1$, and hence

$$B_{k+1} = (1-c_k)B_k + c_k B_{k+1}^{\mathrm{BFGS}} \succeq \tau M^\delta.$$

That is, if $B_0 \succeq \tau M^\delta$ and $\tau \leq 1$, we have $B_k \succeq \tau M^\delta$ for all k, and hence $T_+^k \succeq 0$ for all k. When $\tau = 1$, it is reduced to the variable metric semi-proximal ADMM in Chapter 3.

For instance, we can choose the initial matrix B_0 as

$$B_0 = \xi I, \ \text{ with } \xi = \tau \lambda_{\max}(M^\delta), \ \ \tau \in (0,1).$$

It is easy to see that $B_0 \succeq \tau M^\delta$.

Next we show that the T_k, T_+^k and T_- satisfy Condition 5.2.1 (d)-(g). We suppose that $B_0 \succeq \tau M^\delta$ and $\tau \in (\frac{3}{4}, 1)$.

First we show Condition 5.2.1 (e). Note that $s_k^\top B_k s_k \geq \tau s_k^\top M^\delta s_k \geq \tau\delta\|s_k\|^2$, $\widetilde{l}_k^\top s_k = s_k^\top M s_k + \delta\|s_k\|^2 \geq \delta\|s_k\|^2$, and M is a constant matrix. Therefore, we can suppose that $\|B_{k+1}^{\text{BFGS}} - B_k\|$ is bounded above by some constant $Q > 0$, that is, $-QI \preceq B_{k+1}^{\text{BFGS}} - B_k \preceq QI$. Moreover, $T_+^k = B_k - \tau M \succeq \tau M^\delta - \tau M \succeq \tau\delta I$. Then we can obtain that

$$\begin{aligned}
T_+^{k+1} &= B_{k+1} - \tau M \\
&= B_k + c_k(B_{k+1}^{\text{BFGS}} - B_k) - \tau M \\
&= T_+^k + c_k(B_{k+1}^{\text{BFGS}} - B_k) \\
&\preceq T_+^k + \frac{c_k Q}{\tau\delta}\tau\delta I \\
&\preceq T_+^k + \frac{c_k Q}{\tau\delta} T_+^k \\
&= (1 + \frac{c_k Q}{\tau\delta}) T_+^k.
\end{aligned}$$

On the other hand, we have

$$\begin{aligned}
T_+^k &= T_+^{k+1} - c_k(B_{k+1}^{\text{BFGS}} - B_k) \\
&\preceq T_+^{k+1} + \frac{c_k Q}{\tau\delta}\tau\delta I \\
&\preceq T_+^{k+1} + \frac{c_k Q}{\tau\delta} T_+^{k+1} \\
&= (1 + \frac{c_k Q}{\tau\delta}) T_+^{k+1}.
\end{aligned}$$

Let $\gamma_k = \frac{Q}{\tau\delta} c_k$. Then we have

$$\frac{1}{1+\gamma_k} T_+^k \preceq T_+^{k+1} \preceq (1+\gamma_k) T_+^k \ \text{ for all } \ k. \tag{5.3.4}$$

Note that $\bar{M} = \nabla_{yy}^2 g(y) = \Sigma_g$. Then

$$T_k + \Sigma_g + \beta B^\top B = B_k - M + \Sigma_g + \beta B^\top B = B_k \succ 0,$$

which shows that Condition (d) holds.

Next we show Condition (f). Since (5.3.4) implies that

$$B_{k+1} - \tau M = T_+^{k+1} \preceq (1+\gamma_k) T_+^k = (1+\gamma_k)(B_k - \tau M),$$

and M is positive semidefinite, then we have

$$B_{k+1} \preceq (1+\gamma_k) B_k - \gamma_k \tau M \preceq (1+\gamma_k) B_k.$$

Obviously,

$$T_{k+1} + \Sigma_g + \beta B^\top B = B_{k+1} \preceq (1+\gamma_k) B_k = (1+\gamma_k)(T_k + \Sigma_g + \beta B^\top B).$$

Finally, we show Condition (g). From the definition of M, we have

$$\begin{aligned}
&T_k + \frac{3}{2}\Sigma_g - \frac{\gamma_{k-1}}{2}T_+^k - 2T_- + (\frac{3}{4} - \frac{1}{2}c)\beta B^\top B \\
&= B_k - M + \frac{3}{2}\bar{M} - \frac{\gamma_{k-1}}{2}(B_k - \tau M) - 2(1-\tau)M + (\frac{3}{4} - \frac{1}{2}c)\beta B^\top B \\
&= \left(1 - \frac{\gamma_{k-1}}{2}\right) B_k + \frac{3}{2}\bar{M} - M + \frac{\gamma_{k-1}}{2}\tau M - 2(1-\tau)M + (\frac{3}{4} - \frac{1}{2}c)\beta B^\top B \\
&\succeq \left(1 - \frac{\gamma_{k-1}}{2}\right) \tau M + \frac{3}{2}\bar{M} - M + \frac{\gamma_{k-1}}{2}\tau M - 2(1-\tau)M + (\frac{3}{4} - \frac{1}{2}c)\beta B^\top B \\
&= (3\tau - 3)(\bar{M} + \beta B^\top B) + \frac{3}{2}\bar{M} + (\frac{3}{4} - \frac{1}{2}c)\beta B^\top B \\
&= (3\tau - \frac{3}{2})\bar{M} + (3\tau - \frac{9}{4} - \frac{1}{2}c)\beta B^\top B,
\end{aligned}$$

where the matrix inequality follows from $B_k \succeq \tau M^\delta = \tau M + \tau\delta I \succeq \tau M$. Note that there exist $\bar{k}$ such that $\gamma_k \le 1$ for all $k \ge \bar{k}$. Without loss of generality, we assume $\bar{k} = 0$ and thus $\left(1 - \frac{\gamma_{k-1}}{2}\right) \ge 0$ for all k.

Let $c = 2(\tau - \frac{3}{4})$. It is easy to see that $c \in (0, \frac{1}{2})$. Moreover, $3\tau - \frac{3}{2} > 0$ and $3\tau - \frac{9}{4} - \frac{1}{2}c = 2\tau - \frac{3}{2} > 0$.

As a conclusion of the above discussion, the indefinite proximal term T_k generated via the BFGS update can satisfy Condition 5.2.1. Obviously, the VMIP-ADMM can cover the general indefinite proximal ADMM as the following remark.

Remark 5.3.1. *When $\{T_k\}$ be a constant sequence for all k, that is, $T_k = T$, then we can write $T = T_+ - T_-$, where $T_+, T_- \succeq 0$. It is easy to check that the boundness conditions (e) and (f) immediately hold when $\gamma_k \equiv 0$. Let $T_+ = \tau(rI - \beta B^\top B) \succ 0$ and $T_- = (1-\tau)\beta B^\top B \succeq 0$, we choose*

$$T = T_+ - T_- = \tau r I - \beta B^\top B, \quad \text{with} \quad r > \beta\|B^T B\|.$$

Condition (d) holds. For $\tau \in (0.75, 1)$, taking $c = 2(\tau - \frac{3}{4})$, then Condition (g) turns to be

$$\begin{aligned}
&T + \frac{3}{2}\Sigma_g - 2T_- + (\frac{3}{4} - \frac{1}{2}c)\beta B^\top B \\
&\succ \tau\beta B^\top B - \beta B^\top B - 2(1-\tau)\beta B^\top B + (\frac{3}{4} - \frac{1}{2}c)\beta B^\top B \\
&= (3\tau - \frac{9}{4} - \frac{1}{2}c)\beta B^\top B \\
&\succ 0.
\end{aligned}$$

It is reduced to the indefinite proximal ADMM in [63].

5.4 Numerical results

In this section, we evaluate our VMIP-ADMM using several datasets. Effectiveness, efficiency and convergence properties of the proposed algorithm are compared with some existing methods.

All experiments are conducted on 64-bit Windows with Intel(R) Core(TM) i7-8680U CPU. We implement all the algorithms by Matlab R2018b.

5.4.1 Experiment setup

We consider the Lasso problem:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|Ax-b\|_2^2 + \rho\|x\|_1, \tag{5.4.1}$$

where $A \in \mathbb{R}^{m\times n}$ is a given data matrix; $x \in \mathbb{R}^n$ is a vector of feature coefficients to be estimated; $b \in \mathbb{R}^m$ is an observation vector and $\rho \in \mathbb{R}$ is a positive regularization parameter; m is the number of data points, and n is the number of features. The Lasso model provides a sparse estimation of x when there are more features than data points (i.e., $n > m$).

By introducing an auxiliary variable $y \in \mathbb{R}^m$, we reformulate problem (5.4.1) as

$$\min_{x\in\mathbb{R}^n,\ y\in\mathbb{R}^n} \frac{1}{2}\|y\|_2^2 + \rho\|x\|_1 \quad \text{s.t.} \quad Ax-b=y. \tag{5.4.2}$$

Data

In this experiment, two simple real-world datasets are firstly used for performance evaluations: Boston house prices[1] and California housing[2]. Besides, we also test the proposed algorithm with several synthetic data.

1. Boston house prices is a small standard dataset, which is useful to quickly illustrate the behaviors of the various algorithms. It includes 506 instances and 13 attributes. It was firstly created by Harrison, D. and Rubinfeld, D.L. [58].

2. California housing dataset is a larger dataset consisting of 20,640 samples and 8 features. This dataset was derived from the 1990 U.S. census [95].

3. We randomly generate A and b with several larger sizes.

Settings

Now we specify the settings for the synthetic data. Firstly, we random generate the sizes m and n. Let $x_0 \in \mathbb{R}^n$ be a random vector normally distributed with sparsity 0.1; a matrix A is drawn from standard normal $\mathcal{N}(0,1)$ distribution. Then the vector b is generated by $b = Ax_0 + \varrho$, where ϱ is a noise under $\mathcal{N}(0,10^{-3})$ distribution, and the regularization parameter is set to be $\rho = 0.1\|A^\top b\|_\infty$ [73].

We always choose $S = 0$ in (5.1.3b). We set the initial points as $x^0 = y^0 = 0$ and $\lambda^0 = 0$. The maximum iterations are set to be 20000 in all experiments. We adopt the same stopping criterion

[1]UCI ML housing dataset. https://archive.ics.uci.edu/ml/machine-learning-databases/housing/

[2]This dataset was obtained from the StatLib repository. http://lib.stat.cmu.edu/datasets/

as in [13] for all the numerical experiments, that is, the primal and dual residuals r^k and σ^k should satisfy

$$\|r^k\|_2 \leq \epsilon_k^{\text{pri}} \quad \text{and} \quad \|\sigma^k\|_2 \leq \epsilon_k^{\text{dual}}, \tag{5.4.3}$$

and $\epsilon^{\text{pri}} > 0$ and $\epsilon^{\text{dual}} > 0$ are chosen using an absolute tolerance $\epsilon^{\text{abs}} > 0$ and relative tolerance $\epsilon^{\text{rel}} > 0$ from the suggestion in [13].The stopping criterions are shown in experiments tables.

Comparison Methods

As we known that the classical ADMM always can get a solution with tens of iterations by choosing a suitable parameter β, but sometimes it is time-consuming for large scale problems. We chose the classical ADMM as the baseline. Since this chapter focuses on the proximal ADMM, we compare it with several state-of-the-art proximal ADMMs. All the parameters are chosen as above. The comparison methods are described as follows.

1. ADMM: the classical ADMM [46, 49] applied for problem (5.4.2). The computation of the inverse matrix follows the Cholesky factorization shown in Chapter 3.

2. SPADM: the semi-proximal ADMM [40] with a semidefinite proximal matrix T where $\tau = 1.01$. The computation of the maximum eigenvalues follows that in Chapter 3.

3. IPADM: the indefinite proximal ADMM [77, 63] with an indefinite proximal matrix T that $\tau = 0.8$.

4. IADMB: the indefinite proximal ADMM with the BFGS update for problem (5.4.2) with an indefinite proximal matrix sequence $\{T_k\}$. The τ is chosen as $\tau = 0.8$.

5. PADML: the semi-proximal ADMM with Limited memory BFGS (LBFGS) update in Chapter 3 with a semidefinite proximal matrix sequence $\{T_k\}$ where $\tau = 1.01$. The memory can be set to a positive value in range $[3, \infty)$. We chose the memory to be 5 for the smaller real datasets and 10 for the larger synthetic data.

6. IADML: the indefinite proximal ADMM with the LBFGS update with $\tau = 0.8$.

5.4.2 Experimental results

In this subsection, experimental results of the proposed VMIP-ADMM algorithm are presented against comparison methods. All the results are averaged over 10 trials to reduce the computer errors.

Notations in tables: (CPU time is recorded in seconds)

- Iter.: the iteration steps for each algorithm;
- T-LU: the CPU time for the Cholesky factorization and the calculation of $AA^\top$ or $A^\top A$;
- T-ME: the CPU time of computing for the maximum eigenvalue;
- T-A: the CPU time for the algorithm proceed without T-LU or T-ME;

Real Data

Firstly, we show the performances of our proposed algorithm and other existing methods in Table 5.2 of the real datasets. Figure 5.1 shows the curves of the objective function values with respect to iterations. Since the sizes of these two datasets are small, the computations of the inverse matrix and maximum eigenvalues could be ignored. Except for the classical ADMM, the best iteration of all is colored with red, and the shortest time is colored with blue. Overall, the proposed IADMB outperforms most proximal ADMMs except the classical ADMM on the iterations for both Boston and California datasets. The iterations of IADMB and IADML are close to those of classical ADMM. The CPU time of IPADM is shorter than that of IADMB for the Boston dataset, while the California dataset is the opposite.

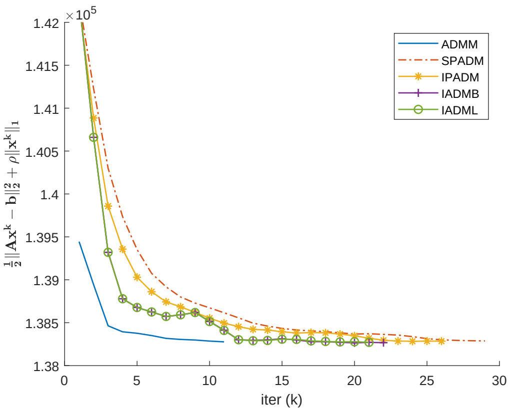
Table 5.2: Results for real datasets

Datasets	Tolerance	β	ADMM		SPADM		IPADM		IADMB		IADML	
			Iter.	T-A(s)	Iter.	T-A(s)	Iter.	T-A(s)	Iter.	T-A(s)	Iter.	T-A(s)
Boston-house	$\epsilon^{\text{rel}} = 10^{-2}$	100	22.0	0.0010	70.0	0.0015	48.0	0.0013	26.0	0.0015	25.0	0.0018
$m = 506$	$\epsilon^{\text{abs}} = 10^{-3}$	300	9.0	0.0010	37.0	0.0013	32.0	0.0012	20.0	0.0015	21.0	0.0016
$n = 13$		500	11.0	0.0010	29.0	0.0012	26.0	0.0011	20.0	0.0012	19.0	0.0014
	$\epsilon^{\text{abs}} = 10^{-3}$	300	16.0	0.0010	69.0	0.0014	60.0	0.0013	29.0	0.0014	34.0	0.0017
	$\epsilon^{\text{abs}} = 10^{-4}$	500	18.0	0.0011	75.0	0.0015	65.0	0.0014	28.0	0.0015	30.0	0.0016
		700	25.0	0.0010	86.0	0.0017	75.0	0.0015	36.0	0.0017	36.0	0.0017
California-housing	$\epsilon^{\text{rel}} = 10^{-2}$	1000	36.0	0.0014	89.0	0.0069	77.0	0.0064	38.0	0.0022	38.0	0.0054
$m = 20640$	$\epsilon^{\text{abs}} = 10^{-3}$	1500	24.0	0.0012	73.0	0.0057	64.0	0.0056	26.0	0.0022	27.0	0.0050
$n = 8$		1700	21.0	0.0013	67.0	0.0056	59.0	0.0049	24.0	0.0018	24.0	0.0041
	$\epsilon^{\text{abs}} = 10^{-3}$	5000	23.0	0.0013	55.0	0.0046	49.0	0.0044	28.0	0.0020	28.0	0.0048
	$\epsilon^{\text{abs}} = 10^{-4}$	5300	22.0	0.0014	53.0	0.0045	47.0	0.0043	27.0	0.0023	27.0	0.0042
		5500	22.0	0.0012	52.0	0.0041	47.0	0.0042	26.0	0.0018	26.0	0.0036

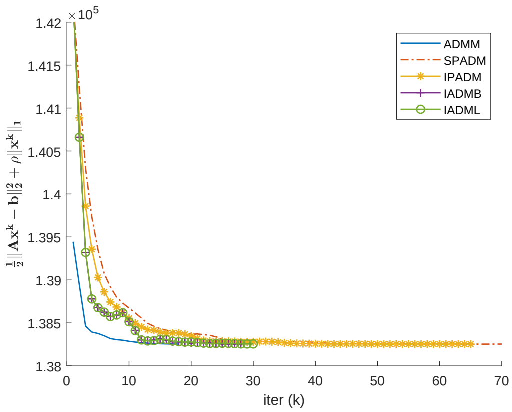
Synthetic Data

In this subsection, the performances of all the methods for synthetic datasets are explored. We randomly generate four different sizes of the data matrix A. For the larger size matrix, the IADMB spends tens or hundreds times longer than the IADM update every step, and thus we replace it by PADML. The computational results for the four datasets are shown in Table 5.3. Also, Figure 5.2 shows the objective function values with respect to iterations with $\beta = 500, 500, 1000, 1000$, respectably.

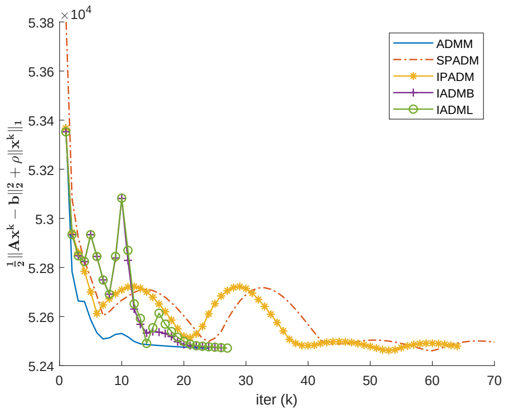
Firstly, focusing on the iteration steps, PADML and IADML use nearly the same iterations as the classical one and almost half of those of general proximal ADMMs for ordinary β. They always outperform the SPADM and IPADM, and the IADML performs better than PADML slightly. The best iteration is colored with red except the classical ADMM.



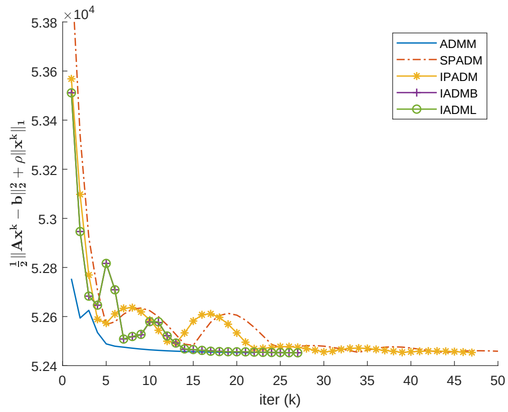
(a) Boston, tol1, $\beta = 500$



(b) Boston, tol2, $\beta = 500$



(c) California, tol1, $\beta = 1500$



(d) California, tol2, $\beta = 5300$

Figure 5.1: Evolution of the objective function values with respect to iterations on real datasets

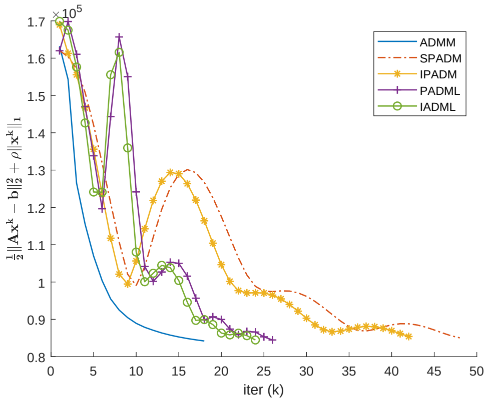
Secondly, paying attention to the CPU time, we see that the averages CPU time of all the algorithms processing are almost the same. The shortest time T-A is colored with blue except the classical ADMM. The CPU time comparison on T-A is given in the left figure of Figure 5.3. The '1-500' in the label means the result of the first randomly generated matrix with $\beta = 500$, and the others mean the same as this. As the matrix size increases, the CPU time T-LU which is for the classical ADMM is quite longer than T-ME that for all the proximal ADMMs. The total time also makes a huge difference, which can be seen from the right figure in Figure 5.3.

Table 5.3: Results for Synthetic datasets ($\epsilon^{\text{rel}} = 10^{-2}$, $\epsilon^{\text{abs}} = 10^{-3}$)

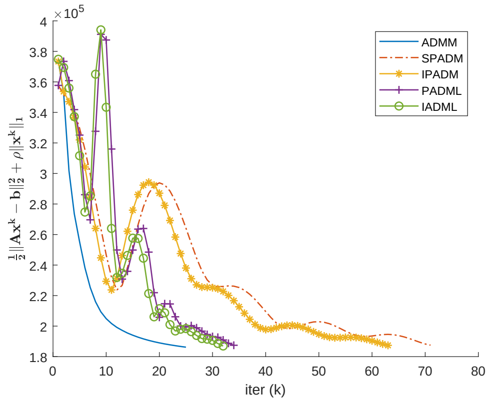
Data	T-LU/ME	β	ADMM		SPADM		IPADM		PADML		IADML	
			Iter.	T-A(s)	Iter.	T-A(s)	Iter.	T-A(s)	Iter.	T-A(s)	Iter.	T-A(s)
$m = 2664$	T-LU = 10.315s	300	30.7	0.5069	75.2	0.5898	66.9	0.5286	36.3	0.2987	34.1	0.2784
$n = 2778$	T-ME = 1.980s	500	17.9	0.3010	49.9	0.4102	42.5	0.3555	26.5	0.2260	24.5	0.2034
		800	11.5	0.1888	36.0	0.2854	31.7	0.2539	20.1	0.1695	18.3	0.1507
		1000	10.3	0.1716	31.6	0.2537	27.3	0.2232	18.4	0.1578	17.1	0.1437
$m = 3580$	T-LU = 28.516s	300	44.9	1.4417	111.5	1.9481	95.9	1.6807	50.2	0.9149	47.8	0.8656
$n = 4620$	T-ME = 5.378s	500	27.9	0.9129	73.1	1.2882	62.1	1.0911	36.0	0.6608	33.7	0.6198
		800	17.9	0.5888	53.2	0.9468	42.5	0.7578	27.9	0.5178	25.4	0.4786
		1000	14.3	0.4787	41.1	0.7307	35.1	0.6266	24.6	0.4576	22.9	0.4221
$m = 7498$	T-LU = 118.826s	500	46.8	1.7680	86.9	4.0866	67.2	3.1514	47.2	2.2569	47.0	2.2453
$n = 5633$	T-ME = 16.141s	1000	24.3	0.9105	51.0	2.3636	45.0	2.0981	27.6	1.3902	26.1	1.2590
		1500	16.1	0.6303	40.9	1.9333	36.0	1.6943	20.7	0.9949	19.5	0.9493
		2000	12.1	0.4829	28.9	1.3641	25.9	1.2279	18.8	0.9107	17.0	0.8338
$m = 4774$	T-LU = 158.214s	500	48.0	4.7976	134.3	8.7370	116.7	7.6785	61.6	4.1815	59.7	3.9039
$n = 10368$	T-ME = 24.249s	1000	23.4	2.3597	80.7	5.3301	70.8	5.7369	41.6	3.6014	38.1	2.6846
		1500	17.4	1.4561	55.7	3.0283	49.0	2.6745	31.4	1.7645	29.1	1.6197
		2000	13.7	1.1631	47.0	2.6077	41.2	2.2898	29.1	1.6362	26.3	1.4693

5.5 Conclusions

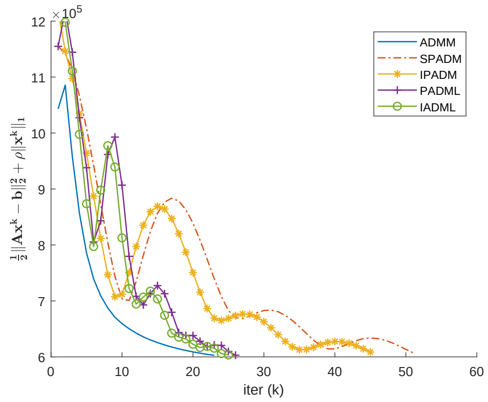
In this chapter, we proposed a variable metric indefinite proximal ADMM whose indefinite proximal term can be chosen differently at every iterative step. We proved the global convergence of the proposed method under some requirements by applying an analysis technique which split the proximal matrix T_k to two parts. Moreover, for a special problem whose y-subproblems are unconstrained quadratic programming problem, we proposed to construct the indefinite term T_k via the BFGS update. We showed that such construction can satisfy the general convergent conditions. We apply our algorithm for a Lasso problem on two real Boston house prices and California housing datasets, along with four synthetic datasets. Experimental results demonstrate the effectiveness of the proposed indefinite proximal ADMM with BFGS or LBFGS update.



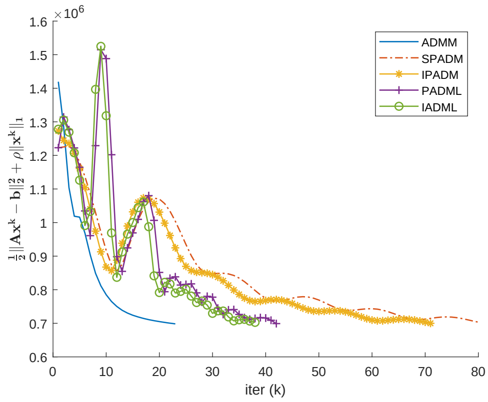
(a) $m = 2664$, $n = 2778$, $\beta = 500$



(b) $m = 3580$, $n = 4620$, $\beta = 500$

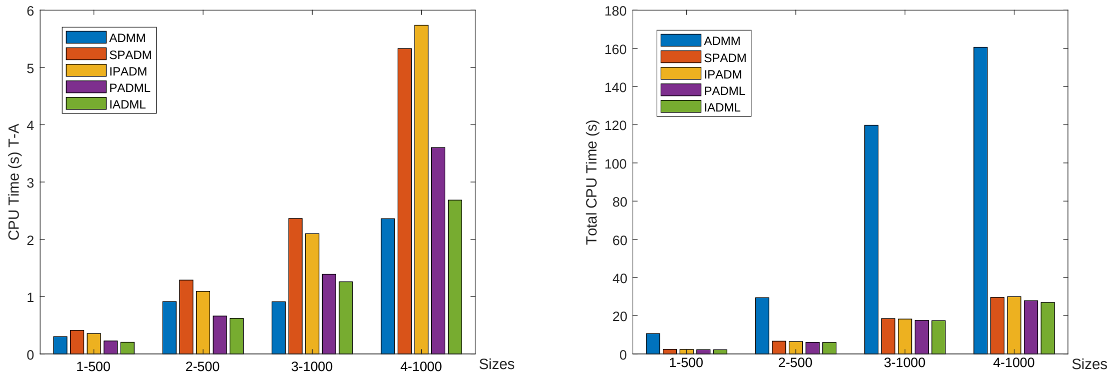


(c) $m = 7498$, $n = 5633$, $\beta = 1000$



(d) $m = 4774$, $n = 10368$, $\beta = 1000$

Figure 5.2: Evolution of the objective function values with respect to iterations on synthetic datasets



(a) CPU time for the algorithm processing

(b) Total CPU time

Figure 5.3: Comparison on the CPU time

How to choose an adjusted proximal term is important to design a more efficient algorithm. The BFGS update provides better performance for some special problems whose y-subproblem is quadratic problem. It is worth developing some efficient proximal term for a general nonlinear subproblem.

Chapter 6

An indefinite-proximal-based strictly contractive Peaceman-Rachford splitting method

6.1 Introduction

In this Chapter, we consider the convex minimization problem with linear constraints and a separable objective function

$$\min \theta_1(x) + \theta_2(y), \quad \text{s.t.} \quad Ax + By = b, \; x \in \mathcal{X}, \; y \in \mathcal{Y}, \tag{6.1.1}$$

where $\theta_1 : \mathbb{R}^{n_1} \to \mathbb{R}$ and $\theta_2 : \mathbb{R}^{n_2} \to \mathbb{R}$ are continuous closed convex (could be nonsmooth) functions; $A \in \mathbb{R}^{m \times n_1}$ and $B \in \mathbb{R}^{m \times n_2}$ are given matrices; $b \in \mathbb{R}^m$ is a given vector; $\mathcal{X}$ and $\mathcal{Y}$ are nonempty closed convex subsets of $\mathbb{R}^{n_1}$ and $\mathbb{R}^{n_2}$, respectively. Throughout, the solution set of (6.1.1) is assumed to be nonempty; and $\mathcal{X}$ and $\mathcal{Y}$ are assumed to be simple in the sense that it is easy to compute the projections under the Euclidean norm onto them (e.g., positive orthant, spheroidal or box areas).

As discussed in Introduction 1.2.4, the Peaceman-Rachford operator splitting method (PRSM) (1.2.12) was proposed, and two different step sizes α and γ adding to (1.2.12b) and (1.2.12d) was also proposed [54, 62]. The convergence results, including global convergence and the worst-case $O(1/t)$ convergence rate in the ergodic and nonergodic sense, have been established in [54].

Considering that in many cases the subproblem in the ADMM and PRSM schemes might be difficult to solve and that in some applications θ_1 or θ_2 is a convex quadratic function, we gave the semi-proximal ADMM scheme (1.2.7) in the previous Chapters. The semidefinite requirements of the proximal terms have also been relaxed to be indefiniteness [63, 77]. The numerical results in [77] showed that the (majorized) ADMM with an indefinite proximal term always performs better than that with semidefinite proximal terms. He et al. [63] obtained a linearized ADMM with an optimal indefinite proximal term. In their method, $S = 0$ and T is chosen by

$$T = \tau r I_{n_2} - \beta B^\top B \;\text{ with }\; r > \beta \|B^\top B\|, \;\; \tau \in (0.75, 1). \tag{6.1.2}$$

Note that they require that the dual stepsize $\gamma = 1$ in (1.2.7). The small value $\tau \in (0.75, 1)$ can ensure that the proximal term has less weight for the y-subproblem (1.2.7b), and thus allows for a larger step.

It is natural to extend the proximal ADMM to the proximal Peaceman-Rachford splitting method. For convenience, we first introduce the whole update scheme of the *indefinite-proximal-based strictly contractive Peaceman-Rachford splitting method (iPSPR):*

$$\begin{cases} x^{k+1} = \arg\min\limits_{x\in\mathcal{X}} \ \mathcal{L}_\beta(x, y^k, \lambda^k) + \dfrac{1}{2}\|x - x^k\|_S^2, & \text{(6.1.3a)} \\ \lambda^{k+\frac{1}{2}} = \lambda^k - \alpha\beta(Ax^{k+1} + By^k - b), & \text{(6.1.3b)} \\ y^{k+1} = \arg\min\limits_{y\in\mathcal{Y}} \ \mathcal{L}_\beta(x^{k+1}, y, \lambda^{k+\frac{1}{2}}) + \dfrac{1}{2}\|y - y^k\|_T^2, & \text{(6.1.3c)} \\ \lambda^{k+1} = \lambda^{k+\frac{1}{2}} - \gamma\beta(Ax^{k+1} + By^{k+1} - b), & \text{(6.1.3d)} \end{cases}$$

where S and T are symmetric and possibly indefinite. Gao et al. [47] considered the generalized ADMM with indefinite proximal term, which corresponds to (6.1.3) with $S = 0$ and $\gamma = 1$. The proximal term T takes a similar formulation as (6.1.2) but with $\tau \in [\frac{\alpha^2-\alpha+4}{\alpha^2-2\alpha+5}, 1)$. Jiang et al. [72] considered the same generalized ADMM as in [47], but they give an optimal bound of τ as $\tau \in (\frac{3+\alpha}{4}, 1)$. For other related works one can refer to [78, 103].

In this chapter, we focus on (6.1.3) with indefinite S and T. Our main contributions are two-fold. Firstly, motivated by the nice analysis techniques in [61] and [114], we prove the global convergence of iPSPR under some assumptions on S and T, see (6.3.32) and (6.3.33), in which the stepsizes α and γ are in the range

$$(\alpha, \gamma) \in \mathbb{D} := \left\{ (\alpha, \gamma) : 0 \le \alpha < 1, 0 \le \gamma < \frac{1 - \alpha + \sqrt{(1+\alpha)^2 + 4(1-\alpha^2)}}{2}, \alpha + \gamma > 0 \right\}. \quad (6.1.4)$$

With some additional mild requirements, see (6.4.6), we prove that the iPSPR is $o(1/t)$ sublinearly convergent in the nonergodic sense. Secondly, our proposed requirements on the proximal T can cover some existing results, such as the special linearized choice (6.1.2) in [63, 72]. More importantly, our proposed requirements on the proximal T employ both the Hessian information of the objective function and the information of $\beta B^\top B$ for the first time. Note that He et al. [63] only uses the information of $\beta B^\top B$, while Li et al. [77] only considers the Hessian information of the objective function.

The rest of this chapter is organized as follows. In section 6.2, we give the optimality condition of (6.1.1) by using the variational inequality and also list some assertions which will be used in later analysis. In section 6.3, we first give the contraction analysis of iPSPR (6.1.3), and then establish the global convergence. We will discuss how to choose T in the end of section 6.3. The detailed formulae will be given for the different ranges of the parameters α and γ. We discuss the nonergodic sublinear convergence rate in section 6.4. In section 6.5, we test the l_1 regularized least square problem to show the efficiency of the proposed iPSPR (6.1.3). Finally, we make some conclusions in section 6.6.

6.2 Preliminaries

In this section, we give the optimality condition of (6.1.1) and some notations or relations which will be frequently used in our analysis. Denote $\Omega = \mathcal{X} \times \mathcal{Y} \times \mathbb{R}^m$. Let $\mathcal{S}$ be the feasible set of (6.1.1), namely, $\mathcal{S} = \{(x, y) : Ax + By = b, x \in \mathcal{X}, y \in \mathcal{Y}\}$ and denote $\mathcal{D} = \mathcal{S} \times \mathbb{R}^m$. Throughout this chapter, we make the following assumption.

Assumption 6.2.1. *Let $\Omega^* \subset \mathcal{D}$ be the set whose elements are the optimal solutions of* (6.1.1) *and the associating dual solutions of* (6.1.1)*. Throughout the chapter, we assume that Ω^* is non-empty.*

6.2.1 Optimality condition of (6.1.1)

Owing to the convexity of $\theta_1(\cdot)$ and $\theta_2(\cdot)$, there exist two positive semidefinite matrices Σ_1 and Σ_2 such that for all $x, x' \in \mathbb{R}^{n_1}$ and $\xi_x \in \partial\theta_1(x)$, $\xi'_x \in \partial\theta_1(x')$,

$$\langle x - x', \xi_x - \xi'_x \rangle \geq \|x - x'\|^2_{\Sigma_1}, \tag{6.2.1}$$

and for all $y, y' \in \mathbb{R}^{n_2}$, $\xi_y \in \partial\theta_2(y)$, $\xi'_y \in \partial\theta_2(y')$,

$$\langle y - y', \xi_y - \xi'_y \rangle \geq \|y - y'\|^2_{\Sigma_2}. \tag{6.2.2}$$

Denote $u = \begin{pmatrix} x \\ y \end{pmatrix}$, $v = \begin{pmatrix} y \\ \lambda \end{pmatrix}$ and $w = \begin{pmatrix} x \\ y \\ \lambda \end{pmatrix}$. For given w, and some specific subgradients $\xi_x \in \partial\theta_1(x)$ and $\xi_y \in \partial\theta_2(x)$, we define $F(w, \xi_x, \xi_y) = \begin{pmatrix} \xi_x - A^\top\lambda \\ \xi_y - B^\top\lambda \\ Ax + By - b \end{pmatrix}$. Due to the convexity of $\theta_1(\cdot)$ and $\theta_2(\cdot)$, it is easy to show that the operator $F(\cdot)$ is monotonic. Specifically, for any $w, w' \in \mathcal{D}$, we have

$$\langle w - w', F(w, \xi_x, \xi_y) - F(w', \xi'_x, \xi'_y) \rangle = \left\langle \begin{pmatrix} x - x' \\ y - y' \end{pmatrix}, \begin{pmatrix} \xi_x - \xi'_x \\ \xi_y - \xi'_y \end{pmatrix} \right\rangle \geq \|u - u'\|^2_{\Sigma}, \tag{6.2.3}$$

where $\Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix}$ and the inequality is due to (6.2.1) and (6.2.2).

Following Theorem 3.1.24 in [90], we say that $w^* \in \Omega^*$ if and only if there exists $\xi^*_x \in \partial\theta_1(x^*)$ and $\xi^*_y \in \partial\theta_2(y^*)$ such that $\langle x - x^*, \xi^*_x \rangle + \langle y - y^*, \xi^*_y \rangle \geq 0$, which is further equivalent to

$$\big\langle w - w^*, F(w^*, \xi^*_x, \xi^*_y) \big\rangle \geq 0, \quad \forall w \in \mathcal{D}, \tag{6.2.4}$$

since $\big\langle w - w^*, F(w^*, \xi^*_x, \xi^*_y) \big\rangle = \langle x - x^*, \xi^*_x \rangle + \langle y - y^*, \xi^*_y \rangle \geq 0$, $\forall w \in \mathcal{D}$.

6.2.2 Some notations

We use the symbol 0 to denote a zero matrix, whose size can be always easily identified form the context. We use $\|\cdot\|$ to denote the 2-norm of a vector. We denote $\|z\|_G^2 = z^\top G z$ for $z \in \mathbb{R}^n$ and $G \in \mathbb{R}^{n\times n}$. For a real symmetric matrix Z, we mark $Z \succeq 0$ (resp. $Z \succ 0$) if Z is positive semidefinite (resp. positive definite). For any given symmetric matrix T, we decompose it as

$$T = T_+ - T_- \quad \text{with} \quad T_+ \succeq 0 \quad \text{and} \quad T_- \succeq 0.$$

To make the analysis more elegantly, we use $r^k = Ax^k + By^k - b$ for short. Similarly, for any $w \in \Omega$, we denote $r(w) = Ax + By - b$. Obviously, there holds that $r(w) = 0$ for any $w \in \mathcal{D}$. For ease of the analysis, we define

$$H = \frac{1}{\alpha+\gamma}\begin{pmatrix} (\alpha+\gamma-\alpha\gamma)\beta B^\top B & -\alpha B^\top \\ -\alpha B & \frac{1}{\beta} I_m \end{pmatrix} \tag{6.2.5}$$

and

$$\hat{H} := \begin{pmatrix} T+\Sigma_2 & 0 \\ 0 & 0 \end{pmatrix} + H = \begin{pmatrix} T+\Sigma_2 + \frac{\alpha+\gamma-\alpha\gamma}{\alpha+\gamma}\beta B^\top B & -\frac{\alpha}{\alpha+\gamma}B^\top \\ -\frac{\alpha}{\alpha+\gamma}B & \frac{1}{(\alpha+\gamma)\beta}I_m \end{pmatrix}. \tag{6.2.6}$$

Denote $P = \begin{pmatrix} S & 0 \\ 0 & T \end{pmatrix}$ and define

$$G := \begin{pmatrix} P & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & H \end{pmatrix} = \begin{pmatrix} S & 0 & 0 \\ 0 & T + \frac{\alpha+\gamma-\alpha\gamma}{\alpha+\gamma}\beta B^\top B & -\frac{\alpha}{\alpha+\gamma}B^\top \\ 0 & -\frac{\alpha}{\alpha+\gamma}B & \frac{1}{(\alpha+\gamma)\beta}I_m \end{pmatrix} \tag{6.2.7}$$

and

$$\widehat{G} := \begin{pmatrix} \Sigma & 0 \\ 0 & 0 \end{pmatrix} + G = \begin{pmatrix} S+\Sigma_1 & 0 & 0 \\ 0 & T+\Sigma_2 + \frac{\alpha+\gamma-\alpha\gamma}{\alpha+\gamma}\beta B^\top B & -\frac{\alpha}{\alpha+\gamma}B^\top \\ 0 & -\frac{\alpha}{\alpha+\gamma}B & \frac{1}{(\alpha+\gamma)\beta}I_m \end{pmatrix} = \begin{pmatrix} S+\Sigma_1 & 0 \\ 0 & \hat{H} \end{pmatrix}. \tag{6.2.8}$$

It follows from (6.2.7) and (6.2.8) that for any $w, w' \in \Omega$

$$\|w - w'\|_G^2 = \|u - u'\|_P^2 + \|v - v'\|_H^2 \tag{6.2.9}$$

and

$$\|w - w'\|_{\widehat{G}}^2 = \|u - u'\|_\Sigma^2 + \|w - w'\|_G^2 = \|x - x'\|_{S+\Sigma_1}^2 + \|v - v'\|_{\hat{H}}^2. \tag{6.2.10}$$

With the update schemes (6.1.3b) and (6.1.3d), it is easy to have

$$\lambda^k = \lambda^{k+1} + (\alpha+\gamma)\beta r^{k+1} + \alpha\beta B(y^k - y^{k+1}). \tag{6.2.11}$$

With (6.2.5) and (6.2.11), we thus have

$$\|v^k - v^{k+1}\|_H^2 = (1-\alpha)\beta\|B(y^k - y^{k+1})\|^2 + (\alpha+\gamma)\beta\|r^{k+1}\|^2. \tag{6.2.12}$$

Finally, it is easy to have the following proposition.

Proposition 6.2.1. *If $0 \le \alpha \le 1$ and $\gamma > 0$, then $H \succeq 0$. If $T + \Sigma_2 + (1-\alpha)\beta B^\top B \succ 0$, then $\hat{H} \succ 0$. If $T + \Sigma_2 + (1-\alpha)\beta B^\top B \succ 0$ and $S + \Sigma_1 \succeq 0$, then $\widehat{G} \succeq 0$.*

6.3 Convergence of iPSPR

In this section, we first show that a sequence related to $\{w_k\}$ generated by iPSPR (6.1.3) is strictly contractive in section 6.3.1 and then establish the global convergence of the method in section 6.3.2, and discuss the choices of the proximal terms in section 6.3.3. Note that the contraction property is also helpful to establish the convergence rate in the nonergodic sense.

6.3.1 Contraction analysis

To establish the strictly contractive property of the sequence $\{\Phi^k_{\alpha,\gamma}(w^*)\}$ (see (6.3.21) for the definition), we first give a rough estimation of $\|w^k - w^*\|^2_{\widehat{G}} - \|w^{k+1} - w^*\|^2_{\widehat{G}}$ based on the optimality conditions of (6.1.3a) and (6.1.3c).

Lemma 6.3.1. *Let the sequence $\{w^k\}$ be generated by iPSPR* (6.1.3)*. If we choose $(\alpha, \gamma) \in \mathbb{D}$, then there holds that*

$$\begin{aligned}\|w^k - w^*\|^2_{\widehat{G}} - \|w^{k+1} - w^*\|^2_{\widehat{G}} \geq \|x^k - x^{k+1}\|^2_{S+\frac{1}{2}\Sigma_1} + \|y^k - y^{k+1}\|^2_{T+\frac{1}{2}\Sigma_2} + (1-\alpha)\beta\|B(y^k - y^{k+1})\|^2 \\ + (2-\alpha-\gamma)\beta\|r^{k+1}\|^2 + 2(1-\alpha)\beta\left\langle r^{k+1}, B(y^k - y^{k+1})\right\rangle\end{aligned} \tag{6.3.1}$$

and

$$\begin{aligned}&\|w^k - w^*\|^2_{\widehat{G}} - \|w^{k+1} - w^*\|^2_{\widehat{G}} \\ &\geq \|x^k - x^{k+1}\|^2_{S+\frac{1}{2}\Sigma_1} + \|y^k - y^{k+1}\|^2_{T+\frac{1}{2}\Sigma_2} + \frac{\alpha^2(1-\gamma)+\gamma^2(1-\alpha)}{(\alpha+\gamma)^2}\beta\|B(y^k - y^{k+1})\|^2 \\ &\quad + \frac{2-\alpha-\gamma}{(\alpha+\gamma)^2\beta}\left\|\lambda^k - \lambda^{k+1}\right\|^2 + \frac{2(\gamma-\alpha)}{(\alpha+\gamma)^2}\left\langle B(y^k - y^{k+1}), \lambda^k - \lambda^{k+1}\right\rangle.\end{aligned} \tag{6.3.2}$$

Proof. The proof of (6.3.1) consists of three steps.

I). We give a rough lower bound estimation of the term $\|w^k - w\|^2_{\widehat{G}} - \|w^{k+1} - w\|^2_{\widehat{G}}$. Following from the first equality of (6.2.10), we have

$$\|w^k - w\|^2_{\widehat{G}} - \|w^{k+1} - w\|^2_{\widehat{G}} = \|w^k - w\|^2_G - \|w^{k+1} - w\|^2_G + \|u^k - u\|^2_\Sigma - \|u^{k+1} - u\|^2_\Sigma. \tag{6.3.3}$$

The Cauchy-Schwartz inequality tells $\|u^k - u\|^2_\Sigma + \|u^{k+1} - u\|^2_\Sigma \geq \frac{1}{2}\|u^k - u^{k+1}\|^2_\Sigma$. Thus, we have

$$\|u^k - u\|^2_\Sigma - \|u^{k+1} - u\|^2_\Sigma \geq \frac{1}{2}\|u^k - u^{k+1}\|^2_\Sigma - 2\|u^{k+1} - u\|^2_\Sigma. \tag{6.3.4}$$

Using the identity $\|a\|^2_G - \|b\|^2_G = \|a-b\|^2_G + 2b^\top G(a-b)$ with $a = w - w^k$ and $b = w - w^{k+1}$, we have

$$\|w^k - w\|^2_G - \|w^{k+1} - w\|^2_G = \|w^k - w^{k+1}\|^2_G + 2(w - w^{k+1})^\top G(w^{k+1} - w^k). \tag{6.3.5}$$

Substituting (6.3.4) and (6.3.5) into (6.3.3), and using (6.2.9) and (6.2.12), we have that for any $w \in \Omega$,

$$\|w^k - w\|^2_{\widehat{G}} - \|w^{k+1} - w\|^2_{\widehat{G}}$$

$$\geq \|x^k - x^{k+1}\|^2_{S+\frac{1}{2}\Sigma_1} + \|y^k - y^{k+1}\|^2_{T+\frac{1}{2}\Sigma_2} + (1-\alpha)\beta\|B(y^k - y^{k+1})\|^2 + (\alpha+\gamma)\beta\|r^{k+1}\|^2$$
$$+2(w - w^{k+1})^\top G(w^{k+1} - w^k) - 2\|u^{k+1} - u\|^2_\Sigma. \tag{6.3.6}$$

II). We focus on the estimation of $(w - w^{k+1})^\top G(w^{k+1} - w^k)$. From the optimality conditions of (6.1.3a) and (6.1.3c), we know that there exist $\xi_x^{k+1} \in \partial\theta_1(x^{k+1})$ and $\xi_y^{k+1} \in \partial\theta_2(y^{k+1})$ such that

$$\left\langle x - x^{k+1}, S(x^{k+1} - x^k) + \xi_x^{k+1} - A^\top\lambda^k + \beta A^\top(Ax^{k+1} + By^k - b)\right\rangle \geq 0, \quad \forall x \in \mathcal{X}$$

and

$$\left\langle y - y^{k+1}, T(y^{k+1} - y^k) + \xi_y^{k+1} - B^\top\lambda^{k+\frac{1}{2}} + \beta B^\top r^{k+1}\right\rangle \geq 0, \quad \forall y \in \mathcal{Y}. \tag{6.3.7}$$

Substituting (6.2.11) into (6.3.7) and noting that $r^{k+1} = Ax^{k+1} + By^{k+1} - b$, we have

$$\left\langle x - x^{k+1}, S(x^{k+1} - x^k) + \xi_x^{k+1} - A^\top\lambda^{k+1} + (1-\alpha-\gamma)\beta A^\top r^{k+1} + (1-\alpha)\beta A^\top B(y^k - y^{k+1})\right\rangle \geq 0,$$
$$\forall x \in \mathcal{X}. \tag{6.3.8}$$

Substituting $\lambda^{k+\frac{1}{2}} = \lambda^{k+1} + \gamma\beta r^{k+1}$ into (6.3.7), we have

$$\left\langle y - y^{k+1}, T(y^{k+1} - y^k) + \xi_y^{k+1} - B^\top\lambda^{k+1} + (1-\gamma)\beta B^\top r^{k+1}\right\rangle \geq 0, \quad \forall y \in \mathcal{Y}. \tag{6.3.9}$$

Rewrite (6.2.11) to be

$$r^{k+1} - \frac{\alpha}{\alpha+\gamma}B(y^{k+1} - y^k) + \frac{1}{(\alpha+\gamma)\beta}(\lambda^{k+1} - \lambda^k) = 0. \tag{6.3.10}$$

Combing (6.3.8), (6.3.9) and (6.3.10) in a suitable way, and recalling the definitions of w and $F(\cdot)$, for any $w \in \Omega$ there holds that

$$\left\langle w - w^{k+1}, \begin{pmatrix} S(x^{k+1} - x^k) \\ T(y^{k+1} - y^k) \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \alpha\beta B^\top r^{k+1} + (1-\alpha)\beta B^\top B(y^{k+1} - y^k) \\ -\frac{\alpha}{\alpha+\gamma}B(y^{k+1} - y^k) + \frac{1}{(\alpha+\gamma)\beta}(\lambda^{k+1} - \lambda^k) \end{pmatrix} \right\rangle$$
$$\geq \left\langle w^{k+1} - w, \begin{pmatrix} A^\top \\ B^\top \\ 0 \end{pmatrix} \left[(1-\alpha-\gamma)\beta r^{k+1} + (1-\alpha)\beta B(y^k - y^{k+1})\right] \right\rangle$$
$$+ \left\langle w^{k+1} - w, F(w^{k+1}, \xi_x^{k+1}, \xi_y^{k+1})\right\rangle. \tag{6.3.11}$$

With the assertion (6.3.10), we have $\alpha\beta B^\top r^{k+1} + (1-\alpha)\beta B^\top B(y^{k+1} - y^k) = \frac{\alpha+\gamma-\alpha\gamma}{\alpha+\gamma}\beta B^\top B(y^{k+1} - y^k) - \frac{\alpha}{\alpha+\gamma}B^\top(\lambda^{k+1} - \lambda^k)$. Using the definition (6.2.5) of H, the definition (6.2.7) of G and the definition of r^{k+1} and $r(w)$, we can rewrite (6.3.11) as

$$(w - w^{k+1})^\top G(w^{k+1} - w^k) \geq \left\langle r^{k+1} - r(w), (1-\alpha-\gamma)\beta r^{k+1} + (1-\alpha)\beta B(y^k - y^{k+1})\right\rangle$$
$$+ \left\langle w^{k+1} - w, F(w^{k+1}, \xi_x^{k+1}, \xi_y^{k+1})\right\rangle. \tag{6.3.12}$$

Noting that $r(w) = 0$ for any $w \in \mathcal{D}$, we have from (6.3.12) that for any $w \in \mathcal{D}$

$$(w - w^{k+1})^\top G(w^{k+1} - w^k) \geq (1 - \alpha - \gamma)\beta\|r^{k+1}\|^2 + (1 - \alpha)\beta \left\langle r^{k+1}, B(y^k - y^{k+1})\right\rangle + \left\langle w^{k+1} - w, F(w^{k+1}, \xi_x^{k+1}, \xi_y^{k+1})\right\rangle. \tag{6.3.13}$$

III). Plugging (6.3.13) into (6.3.6), we have for any $w \in \mathcal{D}$

$$\begin{aligned}&\|w^k - w\|_{\widehat{G}}^2 - \|w^{k+1} - w\|_{\widehat{G}}^2 \\ &\geq \|x^k - x^{k+1}\|_{S+\frac{1}{2}\Sigma_1}^2 + \|y^k - y^{k+1}\|_{T+\frac{1}{2}\Sigma_2}^2 + (1 - \alpha)\beta\|B(y^k - y^{k+1})\|^2 \\ &\quad + (2 - \alpha - \gamma)\beta\|r^{k+1}\|^2 + 2(1 - \alpha)\beta \left\langle r^{k+1}, B(y^k - y^{k+1})\right\rangle + \Delta(w^{k+1}, w),\end{aligned} \tag{6.3.14}$$

where $\Delta(w^{k+1}, w) := 2\left\langle w^{k+1} - w, F(w^{k+1}, \xi_x^{k+1}, \xi_y^{k+1})\right\rangle - 2\|u^{k+1} - u\|_\Sigma^2$. Taking $w = w^{k+1}$ and $w' = w^*$ in (6.2.3), we have from (6.2.3) that

$$\left\langle w^{k+1} - w^*, F(w^{k+1}, \xi_x^{k+1}, \xi_y^{k+1})\right\rangle \geq \left\langle w^{k+1} - w^*, F(w^*, \xi_x^*, \xi_y^*)\right\rangle + \|u^{k+1} - u^*\|_\Sigma^2 \geq \|u^{k+1} - u^*\|_\Sigma^2,$$

where $\xi_x^* \in \partial\theta_1(x^*)$, $\xi_y^* \in \partial\theta_2(y^*)$ and the second inequality is due to the optimality condition (6.2.4) of w^*. This further means that $\Delta(w^{k+1}, w^*) \geq 0$. Setting $w = w^*$ in (6.3.14), we have (6.3.1).

The proof of (6.3.2) follows directly from (6.3.1) and $r^{k+1} = \frac{\alpha}{\alpha+\gamma}B(y^{k+1} - y^k) - \frac{1}{(\alpha+\gamma)\beta}(\lambda^{k+1} - \lambda^k)$ which comes from (6.3.10). The proof is completed.

■

We now need to give a careful estimation of the intersection term $\langle r^{k+1}, B(y^k - y^{k+1})\rangle$, which is useful to establish the strictly contractive property of $\{\Phi_{\alpha,\gamma}^k(w^*)\}$ when $(\alpha, \gamma) \in \mathbb{D}_1 \cup \mathbb{D}_2$ (see (6.3.20) for the definition).

Lemma 6.3.2. *Let the sequence $\{w^k\}$ be generated by iPSPR (6.1.3). If $\alpha \geq 0$ and $\gamma > 0$, then there holds that*

$$\begin{aligned}&\left\langle r^{k+1}, B(y^k - y^{k+1})\right\rangle \\ &\geq \frac{1-\gamma}{1+\alpha}\left\langle r^k, B(y^k - y^{k+1})\right\rangle - \frac{\alpha}{1+\alpha}\|B(y^k - y^{k+1})\|^2 + \frac{1}{1+\alpha}\cdot\frac{1}{\beta}\|y^k - y^{k+1}\|_{-2T_-+\Sigma_2}^2 \\ &\quad + \frac{1}{2(1+\alpha)}\cdot\frac{1}{\beta}\left(\|y^k - y^{k+1}\|_{T_++T_-}^2 - \|y^{k-1} - y^k\|_{T_++T_-}^2\right).\end{aligned} \tag{6.3.15}$$

Proof. From the optimality conditions of (6.1.3c) with $k := k - 1$, we know that there exists $\xi_y^k \in \partial\theta_2(y^k)$ such that

$$\left\langle y - y^k, T(y^k - y^{k-1}) + \xi_y^k - B^\top\lambda^k + (1 - \gamma)\beta B^\top r^k\right\rangle \geq 0, \quad \forall y \in \mathcal{Y}. \tag{6.3.16}$$

Choosing y to be y^k and y^{k+1} in (6.3.9) and (6.3.16) and then rearranging the obtained inequalities suitably, respectively, we have that

$$\left\langle B(y^k - y^{k+1}), -\lambda^{k+1} + (1 - \gamma)\beta r^{k+1}\right\rangle \geq \|y^k - y^{k+1}\|_T^2 - \left\langle y^k - y^{k+1}, \xi_y^{k+1}\right\rangle \tag{6.3.17}$$

and

$$\left\langle B(y^k - y^{k+1}), \lambda^k - (1-\gamma)\beta r^k \right\rangle \geq -\left\langle y^k - y^{k+1}, T(y^{k-1} - y^k) \right\rangle + \left\langle y^k - y^{k+1}, \xi_y^k \right\rangle. \quad (6.3.18)$$

Summing (6.3.17) and (6.3.18) over the both sides yields

$$\begin{aligned} &\left\langle B(y^k - y^{k+1}), \lambda^k - \lambda^{k+1} \right\rangle + (1-\gamma)\beta \left\langle B(y^k - y^{k+1}), r^{k+1} \right\rangle - (1-\gamma)\beta \left\langle B(y^k - y^{k+1}), r^k \right\rangle \\ &\geq \|y^k - y^{k+1}\|_T^2 - \left\langle y^k - y^{k+1}, T(y^{k-1} - y^k) \right\rangle + \left\langle y^k - y^{k+1}, \xi_y^k - \xi_y^{k+1} \right\rangle. \end{aligned} \quad (6.3.19)$$

Recalling that $T = T_+ - T_-$, we know from the Cauchy-Schwarz inequality that

$$\begin{aligned} -\left\langle y^k - y^{k+1}, T(y^{k-1} - y^k) \right\rangle &= -\left\langle y^k - y^{k+1}, T_+(y^{k-1} - y^k) \right\rangle + \left\langle y^k - y^{k+1}, T_-(y^{k-1} - y^k) \right\rangle \\ &\geq -\frac{1}{2}\|y^k - y^{k+1}\|_{T_+ + T_-}^2 - \frac{1}{2}\|y^{k-1} - y^k\|_{T_+ + T_-}^2, \end{aligned}$$

which with (6.2.2) implies that

$$\text{RHS of (6.3.19)} \geq \frac{1}{2}\left(\|y^k - y^{k+1}\|_{T_+ + T_-}^2 - \|y^{k-1} - y^k\|_{T_+ + T_-}^2 \right) + \|y^k - y^{k+1}\|_{-2T_- + \Sigma_2}^2.$$

This with relations (6.2.11) and (6.3.19) implies that (6.3.15). The proof is completed. ■

We now decompose the domain $\mathbb{D}$ (see (6.1.4) for its definition) as $\mathbb{D} = \mathbb{D}_1 \cup \mathbb{D}_2 \cup \mathbb{D}_3 \cup \mathbb{D}_4$ with

$$\mathbb{D}_1 = \left\{ (\alpha, \gamma) : 0 \leq \alpha < 1 < \gamma < \frac{1 - \alpha + \sqrt{(1+\alpha)^2 + 4(1-\alpha^2)}}{2} \right\},$$

$$\mathbb{D}_2 = \{(\alpha, \gamma) : 0 \leq \alpha < 1, \gamma = 1\}, \quad \mathbb{D}_3 = \{(\alpha, \gamma) : 0 \leq \alpha < 1, 0 \leq \gamma < 1, \alpha + \gamma > 0, \alpha \neq \gamma\}, \quad (6.3.20)$$

and

$$\mathbb{D}_4 = \{(\alpha, \gamma) : 0 < \alpha = \gamma < 1\}.$$

For a given $w \in \mathcal{D}$, we define $\Phi_{\alpha,\gamma}^k(w)$ as

$$\Phi_{\alpha,\gamma}^k(w) := \|w^k - w\|_{\widehat{G}}^2 + \rho_1^{\alpha,\gamma}\|y^{k-1} - y^k\|_{T_+ + T_-}^2 + \rho_2^{\alpha,\gamma}\beta\|r^k\|^2, \quad (6.3.21)$$

where the constants

$$\rho_1^{\alpha,\gamma} = \begin{cases} \frac{1-\alpha}{1+\alpha} & (\alpha, \gamma) \in \mathbb{D}_1, \\ \frac{1-\alpha}{2(1+\alpha)} & (\alpha, \gamma) \in \mathbb{D}_2, \\ 0 & (\alpha, \gamma) \in \mathbb{D}_3 \cup \mathbb{D}_4, \end{cases} \qquad \rho_2^{\alpha,\gamma} = \begin{cases} \frac{(\gamma-1)^2}{(1-c^{\alpha,\gamma})(1+\alpha)} & (\alpha, \gamma) \in \mathbb{D}_1, \\ 0 & (\alpha, \gamma) \in \mathbb{D}_2 \cup \mathbb{D}_3 \cup \mathbb{D}_4, \end{cases} \quad (6.3.22)$$

in which the constant $c^{\alpha,\gamma}$ is defined as

$$c^{\alpha,\gamma} \in \begin{cases} \left(0, \frac{1-\alpha^2+\alpha-(\alpha-1)\gamma-\gamma^2}{(2-\alpha-\gamma)(1+\alpha)}\right) & (\alpha, \gamma) \in \mathbb{D}_1, \\ (0, 1) & (\alpha, \gamma) \in \mathbb{D}_2 \cup \mathbb{D}_3. \end{cases}$$

We are now ready to have the following theorem.

Theorem 6.3.1. *Given $w^* \in \Omega^*$, let the sequence $\{w^k\}$ be generated by iPSPR* (6.1.3). *If we choose $(\alpha,\gamma)\in\mathbb{D}$, then there holds that*

$$\Phi^k_{\alpha,\gamma}(w^*) - \Phi^{k+1}_{\alpha,\gamma}(w^*) \geq \|x^k - x^{k+1}\|^2_{S+\frac{1}{2}\Sigma_1} + \|y^k - y^{k+1}\|^2_{T+\frac{1}{2}\Sigma_2+\kappa_1^{\alpha,\gamma}(-2T_-+\Sigma_2)+\kappa_2^{\alpha,\gamma}\beta B^\top B} + \frac{\kappa_3^{\alpha,\gamma}}{\beta}\|\lambda^k - \lambda^{k+1}\|^2 + \kappa_4^{\alpha,\gamma}\|r^{k+1}\|^2, \tag{6.3.23}$$

where the constants

$$\kappa_1^{\alpha,\gamma} = \begin{cases} \frac{2(1-\alpha)}{1+\alpha} & (\alpha,\gamma)\in\mathbb{D}_1, \\ \frac{1-\alpha}{1+\alpha} & (\alpha,\gamma)\in\mathbb{D}_2, \\ 0 & (\alpha,\gamma)\in\mathbb{D}_3\cup\mathbb{D}_4, \end{cases} \qquad \kappa_2^{\alpha,\gamma} = \begin{cases} \frac{c^{\alpha,\gamma}(1-\alpha)^2}{1+\alpha} & (\alpha,\gamma)\in\mathbb{D}_1, \\ \frac{c^{\alpha,\gamma}(1-\alpha)(3-\alpha)}{4(1+\alpha)} & (\alpha,\gamma)\in\mathbb{D}_2, \\ \frac{c^{\alpha,\gamma}(1-\alpha)(1-\gamma)}{(2-\gamma-\alpha)} & (\alpha,\gamma)\in\mathbb{D}_3, \\ \frac{1-\alpha}{2} & (\alpha,\gamma)\in\mathbb{D}_4, \end{cases} \tag{6.3.24}$$

and

$$\kappa_3^{\alpha,\gamma} = \begin{cases} 0 & (\alpha,\gamma)\in\mathbb{D}_1\cup\mathbb{D}_2, \\ \frac{(1-c^{\alpha,\gamma})(1-\alpha)(1-\gamma)(2-\alpha-\gamma)}{(\gamma-\alpha)^2+(1-c^{\alpha,\gamma})(1-\alpha)(1-\gamma)(\alpha+\gamma)^2} & (\alpha,\gamma)\in\mathbb{D}_3, \\ \frac{1-\alpha}{2\alpha^2} & (\alpha,\gamma)\in\mathbb{D}_4, \end{cases} \tag{6.3.25}$$

and

$$\kappa_4^{\alpha,\gamma} = \begin{cases} 2-\alpha-\gamma-\frac{(\gamma-1)^2}{(1-c^{\alpha,\gamma})(1+\alpha)} & (\alpha,\gamma)\in\mathbb{D}_1, \\ \frac{(1-c^{\alpha,\gamma})(1-\alpha)(3-\alpha)}{(1+\alpha)+(1-c^{\alpha,\gamma})(3-\alpha)} & (\alpha,\gamma)\in\mathbb{D}_2, \\ 0 & (\alpha,\gamma)\in\mathbb{D}_3\cup\mathbb{D}_4. \end{cases} \tag{6.3.26}$$

Proof. We consider four cases.

I). $(\alpha,\gamma)\in\mathbb{D}_1$. By combining (6.3.15) and (6.3.1), we derive

$$\begin{aligned}&\left(\|w^k - w^*\|^2_{\widehat{G}} + \frac{1-\alpha}{1+\alpha}\|y^{k-1} - y^k\|^2_{T_++T_-}\right) - \left(\|w^{k+1} - w^*\|^2_{\widehat{G}} + \frac{1-\alpha}{1+\alpha}\|y^k - y^{k+1}\|^2_{T_++T_-}\right) \\ &\geq \|x^k - x^{k+1}\|^2_{S+\frac{1}{2}\Sigma_1} + \|y^k - y^{k+1}\|^2_{T+\frac{1}{2}\Sigma_2} + \frac{2(1-\alpha)}{1+\alpha}\|y^k - y^{k+1}\|^2_{-2T_-+\Sigma_2} \\ &\quad + \frac{(1-\alpha)^2}{1+\alpha}\beta\|B(y^k - y^{k+1})\|^2 + (2-\alpha-\gamma)\beta\|r^{k+1}\|^2 - 2(\gamma-1)\frac{1-\alpha}{1+\alpha}\beta\left\langle r^k, B(y^k - y^{k+1})\right\rangle.\end{aligned} \tag{6.3.27}$$

Note that in this case $0 < c^{\alpha,\gamma} < \frac{1-\alpha^2+\alpha-(\alpha-1)\gamma-\gamma^2}{(2-\alpha-\gamma)(1+\alpha)} < 1$, with the Cauchy-Schwarz inequality, we have

$$-2\left\langle r^k, B(y^k - y^{k+1})\right\rangle \geq -\frac{\gamma-1}{(1-\alpha)(1-c^{\alpha,\gamma})}\cdot\|r^k\|^2 - \frac{(1-\alpha)(1-c^{\alpha,\gamma})}{\gamma-1}\cdot\|B(y^k - y^{k+1})\|^2.$$

Plugging the above inequality into (6.3.27), we obtain (6.3.23) in this case.

II). $(\alpha,\gamma)\in\mathbb{D}_2$. For this case, (6.3.15) reduces to

$$\left\langle r^{k+1}, B(y^k - y^{k+1})\right\rangle \geq -\frac{\alpha}{1+\alpha}\|B(y^k - y^{k+1})\|^2 + \frac{1}{(1+\alpha)\beta}\|y^k - y^{k+1}\|^2_{-2T_-+\Sigma_2}$$

$$+\frac{1}{2(1+\alpha)\beta}\left(\|y^k-y^{k+1}\|^2_{T_++T_-}-\|y^{k-1}-y^k\|^2_{T_++T_-}\right). \tag{6.3.28}$$

On the other hand, by the Cauchy-Schwartz inequality, we have

$$\left\langle r^{k+1}, B(y^k-y^{k+1})\right\rangle \geq -\delta\|B(y^k-y^{k+1})\|^2-\frac{1}{4\delta}\|r^{k+1}\|^2, \tag{6.3.29}$$

where $\delta=\frac{(1+\alpha)+(1-c^{\alpha,\gamma})(3-\alpha)}{4(1+\alpha)}$. Combing (6.3.28) and (6.3.29), and using (6.3.1), we have

$$\begin{aligned}
&\left(\|w^k-w^*\|^2_{\widehat{G}}+\frac{1-\alpha}{2(1+\alpha)}\|y^{k-1}-y^k\|^2_{T_++T_-}\right)-\left(\|w^{k+1}-w^*\|^2_{\widehat{G}}+\frac{1-\alpha}{2(1+\alpha)}\|y^k-y^{k+1}\|^2_{T_++T_-}\right)\\
&\geq \|x^k-x^{k+1}\|^2_{S+\frac{1}{2}\Sigma_1}+\|y^k-y^{k+1}\|^2_{T+\frac{1}{2}\Sigma_2}+\frac{1-\alpha}{1+\alpha}\|y^k-y^{k+1}\|^2_{-2T_-+\Sigma_2}\\
&\quad+\frac{c^{\alpha,\gamma}(1-\alpha)(3-\alpha)}{4(1+\alpha)}\beta\|B(y^k-y^{k+1})\|^2+\frac{(1-c^{\alpha,\gamma})(1-\alpha)(3-\alpha)}{(1+\alpha)+(1-c^{\alpha,\gamma})(3-\alpha)}\|r^{k+1}\|^2.
\end{aligned} \tag{6.3.30}$$

This means (6.3.23) holds in this case.

III). $(\alpha,\gamma)\in\mathbb{D}_3$. Noting that $c^{\alpha,\gamma}\in(0,1)$ and letting $\tilde{c}=\frac{(\gamma-\alpha)^2}{(\gamma-\alpha)^2+(1-c^{\alpha,\gamma})(1-\alpha)(1-\gamma)(\alpha+\gamma)^2}$, we have from the Cauchy-Schwarz inequality that

$$2(\gamma-\alpha)\left\langle B(y^k-y^{k+1}),\lambda^k-\lambda^{k+1}\right\rangle \geq -\frac{(\alpha-\gamma)^2\beta}{\tilde{c}(2-\gamma-\alpha)}\|B(y^k-y^{k+1})\|^2-\frac{\tilde{c}(2-\gamma-\alpha)}{\beta}\|\lambda^k-\lambda^{k+1}\|^2,$$

which with (6.3.2) and the equality $[\alpha^2(1-\gamma)+\gamma^2(1-\alpha)](2-\alpha-\gamma)=(\gamma-\alpha)^2+(1-\alpha)(1-\gamma)(\alpha+\gamma)^2$ implies that (6.3.23) holds in this case.

IV). $(\alpha,\gamma)\in\mathbb{D}_4$. Note that $\alpha=\gamma$ in this case. It is easy to see from (6.3.2) that

$$\begin{aligned}
&\|w^k-w^*\|^2_{\widehat{G}}-\|w^{k+1}-w^*\|^2_{\widehat{G}}\\
&\geq \|x^k-x^{k+1}\|^2_{S+\frac{1}{2}\Sigma_1}+\|y^k-y^{k+1}\|^2_{T+\frac{1}{2}\Sigma_2}+\frac{1-\alpha}{2}\beta\|B(y^k-y^{k+1})\|^2+\frac{1-\alpha}{2\alpha^2\beta}\left\|\lambda^k-\lambda^{k+1}\right\|^2,
\end{aligned} \tag{6.3.31}$$

which means that (6.3.23) holds in this case.

The proof is completed. ■

6.3.2 Global convergence

Theorem 6.3.2. *Let the sequence $\{w^k\}$ be generated by iPSPR (6.1.3). If the stepsizes $(\alpha,\gamma)\in\mathbb{D}$ and the proximal terms S, T are chosen such that*

$$S+\frac{1}{2}\Sigma_1\succeq 0,\quad S+\frac{1}{2}\Sigma_1+\beta A^\top A\succ 0 \tag{6.3.32}$$

and

$$T+\Sigma_2+(1-\alpha)\beta B^\top B\succ 0,\quad T+\frac{1}{2}\Sigma_2+\kappa_1^{\alpha,\gamma}(-2T_-+\Sigma_2)+\kappa_2^{\alpha,\gamma}\beta B^\top B\succ 0, \tag{6.3.33}$$

then $\{w^k\}$ converges to an optimal solution of (6.1.1).

Proof. The first conditions in (6.3.32) and (6.3.33) guarantee $\hat{G} \succeq 0$ and $\hat{H} \succ 0$, see Proposition 6.2.1. We divide the proof into three steps.

I) We show that the sequences $\{w^k\}$ is bounded.

It is straightforward to see from (6.3.23), (6.3.32) and (6.3.33) that $\Phi^k_{\alpha,\gamma}(w^*)$ is monotone decreasing. This with $T_+, T_- \succeq 0$ and the definition (6.3.21) means that $\|w^k - w^*\|^2_{\widehat{G}}$ is bounded. With the second equality of (6.2.10), we have $\|w^k - w^*\|^2_{\widehat{G}} = \|x^k - x^*\|^2_{S+\Sigma_1} + \|v^k - v^*\|^2_{\hat{H}}$, which means that $\|x^k - x^*\|_{S+\Sigma_1}$ and $\|v^k - v^*\|_{\hat{H}}$ are all bounded. Besides, with the positiveness of $\hat{H}$, we know that the sequences $\{\lambda^k\}$ and $\{y^k\}$ are bounded. Following from (6.3.23), (6.3.32) and (6.3.33), we also have

$$\lim_{k\to\infty} \frac{\kappa_3^{\alpha,\gamma}}{\beta}\|\lambda^k - \lambda^{k+1}\|^2 + \kappa_4^{\alpha,\gamma}\|r^{k+1}\|^2 = 0. \tag{6.3.34}$$

Noting that $\kappa_3^{\alpha,\gamma} + \kappa_4^{\alpha,\gamma} > 0$, with (6.2.11) and (6.3.34) and the boundness of y^k, we can see that $\{r^k\}$ is bounded. With the definition of r^k, we know that $\|Ax^k - Ax^*\| = \|r^k - B(y^k - y^*)\| \leq \|r^k\| + \|B(y^k - y^*)\|$, which with the boundness of r^k and y^k implies that $\|x^k - x^*\|_{\beta A^\top A}$ is bounded. Recalling that $S + \frac{1}{2}\Sigma_1 + \beta A^\top A \succ 0$ and $\|x^k - x^*\|_{S+\Sigma_1}$ is bounded, it is safe to say that $\{x^k\}$ is also bounded.

II) We argue that any cluster point of the sequence $\{w^k\}$ is an optimal solution of (6.1.1). Let $\{w^{k_i}\}$ be a subsequence of the sequence $\{w^k\}$ and $\lim\limits_{k_i\to\infty} w^{k_i} = w^\infty$. Following from (6.3.23), (6.3.32) and (6.3.33), we have

$$\lim_{k\to\infty} \|x^k - x^{k+1}\|_{S+\frac{1}{2}\Sigma_1} = \lim_{k\to\infty} \|y^k - y^{k+1}\|_{T+\frac{1}{2}\Sigma_2+\kappa_1^{\alpha,\gamma}(-2T_-+\Sigma_2)+\kappa_2^{\alpha,\gamma}\beta B^\top B} = 0. \tag{6.3.35}$$

With the second condition on T in (6.3.33), we know from the second equality in (6.3.35) that

$$\lim_{k\to\infty} \|y^k - y^{k+1}\| = 0. \tag{6.3.36}$$

Again using $\kappa_3^{\alpha,\gamma} + \kappa_4^{\alpha,\gamma} > 0$, with (6.2.11) and (6.3.34), it is easy to see that

$$\lim_{k\to\infty} \|r^k\| = \lim_{k\to\infty} \|\lambda^k - \lambda^{k+1}\| = 0. \tag{6.3.37}$$

On the other hand, with the definition of r^k, we have $A(x^k - x^{k+1}) = r^k - r^{k+1} - B(y^k - y^{k+1})$. Therefore, we know from (6.3.36) and (6.3.37) that $\lim\limits_{k\to\infty} \|A(x^k - x^{k+1})\| = 0$, which with the first equality in (6.3.35) implies $\lim\limits_{k\to\infty} \|x^k - x^{k+1}\|_{S+\frac{1}{2}\Sigma_1+\beta A^T A} = 0$. This with the second condition on S in (6.3.32) implies

$$\lim_{k\to\infty} \|x^k - x^{k+1}\| = 0. \tag{6.3.38}$$

Since the graphs of $\partial\theta_1(\cdot)$ and $\partial\theta_2(\cdot)$ are both closed, taking the limit with respect $k_i \to \infty$ on both sides of (6.3.11) and by using (6.3.36), (6.3.37) and (6.3.38), we know that there exists ξ_x^∞ and ξ_y^∞ such that

$$(w - w^\infty)^\top F(w^\infty, \xi_x^\infty, \xi_y^\infty) \geq 0, \quad \forall w \in \mathcal{D},$$

which means that w^∞ is an optimal solution of (6.1.1).

III) We finally prove that the sequence $\{w^k\}$ has only one cluster point. We first replace w^* with w^∞ in the analysis of Steps I) and II). It follows from $\lim\limits_{k_i\to\infty} w^{k_i} = w^\infty$ and (6.3.36), (6.3.37) that $\lim\limits_{k_i\to\infty} \Phi^{k_i}_{\alpha,\gamma}(w^\infty) = 0$. Owing to the decreasing monotonicity of the sequence $\Phi^k_{\alpha,\gamma}(w^\infty)$, we can see that

$$\lim_{k\to\infty} \Phi^k_{\alpha,\gamma}(w^\infty) = 0.$$

This together with $T_+, T_- \succeq 0$ and $\|w^k - w^\infty\|^2_{\widehat{G}} = \|x^k - x^\infty\|^2_{S+\Sigma_1} + \|v^k - v^\infty\|^2_{\hat{H}}$ and $\hat{H} \succ 0$ shows that

$$\lim_{k\to\infty} \|x^k - x^\infty\|_{S+\Sigma_1} = \lim_{k\to\infty} \|y^k - y^\infty\| = \lim_{k\to\infty} \|\lambda^k - \lambda^\infty\|. \tag{6.3.39}$$

With (6.2.12), we further have $\lim_{k\to\infty} \|r^k\| = 0$. Using again the inequality $\|Ax^k - Ax^\infty\| = \|r^k - B(y^k - y^\infty)\| \le \|r^k\| + \|B(y^k - y^\infty)\|$, which with (6.3.39) and (6.3.37) implies

$$\lim_{k\to\infty} \|A(x^k - x^\infty)\| = 0. \tag{6.3.40}$$

Combing (6.3.39) and (6.3.40), and using that $S + \frac{1}{2}\Sigma_1 + \beta A^\top A \succ 0$, we immediately have

$$\lim_{k\to\infty} w^k = w^\infty.$$

The proof is completed. ∎

Remark 6.3.1. *If the condition* (6.3.32) *is replaced by* $S \succeq 0$, *we can have from* $\lim\limits_{k\to\infty} \|x^k - x^{k+1}\|_{S+\frac{1}{2}\Sigma_1} = 0$ *that* $\lim\limits_{k\to\infty} S(x^k - x^{k+1}) = 0$. *Using the similar analysis to the above proof, we can show that* $\{v^k\}$ *converges to some* $v^* = \begin{pmatrix} y^* \\ \lambda^* \end{pmatrix}$, *where* $w^* = \begin{pmatrix} x^* \\ v^* \end{pmatrix}$ *is an optimal solution of problem* (6.1.1).

6.3.3 Choices of proximal terms

When the proximal terms S and T satisfy conditions (6.3.32) and (6.3.33), it is easy to see that the objective functions of subproblems (6.1.3a) and (6.1.3c) are strongly convex, which make the corresponding problems much easier to solve. Note that by allowing S or T indefinite, we can always take a larger step on updating the variable x or y. Besides, we next show that for some special cases, with particularly chosen proximal term T, the subproblem (6.1.3c) is easy to solve or even takes a closed form solution. Note that the discussion for the proximal term S is omitted since it is similar.

We consider to choose T as

$$T = rI - \left(\Sigma_2 + \beta B^\top B\right) \quad \text{with} \quad r = \lambda_{\max}\left(\frac{1}{2}\Sigma_2 + \tau\beta B^\top B\right), \tag{6.3.41}$$

where $\tau \in (0, 1]$. We decompose $T = T_+ - T_-$ with $T_+ = rI - (\frac{1}{2}\Sigma_2 + \tau\beta B^\top B)$ and $T_- = \frac{1}{2}\Sigma_2 + (1-\tau)\beta B^\top B$. Note that $T_+, T_- \succeq 0$. By some direct calculations, we have

$$T + \Sigma_2 + (1-\alpha)\beta B^\top B = rI - \alpha\beta B^\top B \tag{6.3.42}$$

and

$$T+\frac{1}{2}\Sigma_2+\kappa_1^{\alpha,\gamma}(-2T_-+\Sigma_2)+\kappa_2^{\alpha,\gamma}\beta B^\top B = rI-\left(\frac{1}{2}\Sigma_2+(1+2\kappa_1^{\alpha,\gamma}(1-\tau)-\kappa_2^{\alpha,\gamma})\beta B^\top B\right). \quad (6.3.43)$$

For given $(\alpha,\gamma)\in\mathbb{D}$ and a fixed $c^{\alpha,\gamma}$, by (6.3.42) and (6.3.43), we know that if we choose $\tau>\alpha$ and $\tau>1-\frac{\kappa_2^{\alpha,\gamma}}{1+2\kappa_1^{\alpha,\gamma}}$, then (6.3.33) must hold. Note that the number $1-\frac{\kappa_2^{\alpha,\gamma}}{1+2\kappa_1^{\alpha,\gamma}}$ is decreasing with respect to $c^{\alpha,\gamma}$ which is defined over an open interval. Hence, we can argue that if

$$1\ge\tau>\max\left\{\alpha,\inf_{c^{\alpha,\gamma}}\left\{1-\frac{\kappa_2^{\alpha,\gamma}}{1+2\kappa_1^{\alpha,\gamma}}\right\}\right\},$$

namely,

$$1\ge\tau>\underline{\tau}^{\alpha,\gamma}:=\begin{cases}1-(1-\alpha)^2\frac{1-\alpha^2-(\gamma-1)(\alpha+\gamma)}{(2-\alpha-\gamma)(1+\alpha)(5-3\alpha)} & (\alpha,\gamma)\in\mathbb{D}_1,\\ \frac{3+\alpha}{4} & (\alpha,\gamma)\in\mathbb{D}_2,\\ \frac{1-\alpha\gamma}{2-\alpha-\gamma} & (\alpha,\gamma)\in\mathbb{D}_3,\\ \frac{1+\alpha}{2} & (\alpha,\gamma)\in\mathbb{D}_4,\end{cases} \quad (6.3.44)$$

then (6.3.33) must hold.

Consider the case when $\theta_2(y)=\frac{1}{2}y^\top My+h(y)$, where M is symmetric positive semidefinite and the nonsmooth convex function $h(y)$ is simple in the sense that $\min_{y\in\mathcal{Y}}h(y)+\frac{1}{2}\|y-d\|^2$ is easy to compute. Here $d\in\mathbb{R}^{n_2}$ is a given vector. In this case, we have $\Sigma_2=M$ and the subproblem (6.1.3b) with T chosen according to (6.3.41) and (6.3.44) takes the following form

$$y^{k+1}=\arg\min_{y\in\mathcal{Y}}\ h(y)+\frac{1}{2}\|y-d^k\|^2$$

with $d^k=Ty^k+B^\top\left(\lambda^{k+\frac{1}{2}}-\beta(Ax^{k+1}-b)\right)$.

To end this subsection, some comments are listed in order. Firstly, if $\alpha=0,\gamma=1$ and $\Sigma_2=0$, (6.3.44) becomes $0.75<\tau\le1$, which recovers the optimal bound of τ for the linearized ADMM in [63]; if $\alpha\in(0,1),\gamma=1$ and $\Sigma_2=0$, (6.3.44) becomes $(3+\alpha)/4<\tau\le1$, which partially recovers the optimal bound of τ for the linearized version of the generalized ADMM in [72]. Note that in [72], they allowed $\alpha\in(-1,1)$. Secondly, if $(\alpha,\gamma)\in\mathbb{D}_2\cup\mathbb{D}_3$, it is easy to see that $\frac{1-\alpha\gamma}{2-\alpha-\gamma}\ge\frac{2+\alpha+\gamma}{4}$ and the equality holds if and only if $\alpha=\gamma$, namely, $(\alpha,\gamma)\in\mathbb{D}_4$. Thirdly, when the subproblem (6.1.3c) does not take a closed form solution, as done in [40, 77, 118], we can consider the majorized version of iPSPR. The techniques for constructing the indefinite proximal T in [22, 77] can be explored to construct T. We leave this for future investigation.

6.4 Sublinear convergence of iPSPR

The rate of convergence of an algorithm can help us have a deeper understanding of the algorithm. In this section, motivated by [29, 77], we establish the $o(1/t)$ sublinear rate of convergence of iPSPR in the nonergodic sense.

We first give a new optimality condition of (6.1.1) as follows.

Lemma 6.4.1. *Let the sequence $\{w^k\}$ be generated by iPSPR* (6.1.3)*. We choose $(\alpha,\gamma)\in\mathbb{D}$ and the proximal terms S, T are chosen such that* (6.3.32) *and* (6.3.33) *hold. Then $w^{k+1}\in\Omega^*$, namely, w^{k+1} is one optimal solution of* (6.1.1)*, if*

$$\|w^k - w^{k+1}\|_{\widehat{G}} = 0.$$

Proof. The proof is similar to the second part of the proof of Theorem 6.3.2, we omit the details here. ∎

Following from (6.2.6), (6.2.10) and (6.2.12), we have

$$\|w^k - w^{k+1}\|_{\widehat{G}}^2 = \|x^k - x^{k+1}\|_{S+\Sigma_1}^2 + \|y^k - y^{k+1}\|_{T+\Sigma_2+(1-\alpha)\beta B^T B}^2 + (\alpha+\gamma)\beta\|r^{k+1}\|^2.$$

Hence, Lemma 6.4.1 provides a practical stopping condition for iPSPR (6.1.3), which is shown as

$$\max\{\|x^k - x^{k+1}\|_{S+\Sigma_1}^2, \|y^k - y^{k+1}\|_{T+\Sigma_2+(1-\alpha)\beta B^T B}^2, \beta\|r^{k+1}\|^2\} \le \mathrm{tol}, \tag{6.4.1}$$

where tol is some tolerance.

Theorem 6.4.1. *Let the sequence $\{w^k\}$ be generated by iPSPR* (6.1.3) *with $(\alpha,\gamma)\in\mathbb{D}$. Suppose that the proximal terms S, T are chosen such that* (6.3.32), (6.3.33) *and*

$$S + \frac{1}{2}\Sigma_1 \succeq \frac{1}{2}c\Sigma_1 \tag{6.4.2}$$

hold, where c is a positive constant. We have that

$$\min_{1\le i\le t} \|w^i - w^{i+1}\|_{\widehat{G}}^2 = o(1/t). \tag{6.4.3}$$

Proof. With conditions (6.4.2) on S, we know that $S+\Sigma_1 \preceq (1+c^{-1})(S+\frac{1}{2}\Sigma_1)$. With the condition (6.3.33) on T, we know that there exists a positive constant c_1 such that

$$\hat{H} \preceq c_1 \begin{pmatrix} T + \frac{1}{2}\Sigma_2 + \kappa_1^{\alpha,\gamma}(-2T_- + \Sigma_2) + \kappa_2^{\alpha,\gamma}\beta B^\top B & 0 \\ 0 & \frac{\bar{\kappa}_3^{\alpha,\gamma}}{\beta} I \end{pmatrix},$$

which with (6.2.8) implies that

$$\widehat{G} \le \max\{1+c^{-1}, c_1\} \begin{pmatrix} S + \frac{1}{2}\Sigma_1 & 0 & 0 \\ 0 & T + \frac{1}{2}\Sigma_2 + \kappa_1^{\alpha,\gamma}(-2T_- + \Sigma_2) + \kappa_2^{\alpha,\gamma}\beta B^\top B & 0 \\ 0 & 0 & \frac{\bar{\kappa}_3^{\alpha,\gamma}}{\beta} I \end{pmatrix}.$$

This with (6.3.23) implies that

$$\Phi_{\alpha,\gamma}^k(w^*) - \Phi_{\alpha,\gamma}^{k+1}(w^*) \ge \frac{1}{\max\{c, c_1\}} \|w^k - w^{k+1}\|_{\widehat{G}}^2. \tag{6.4.4}$$

Summing (6.4.4) over $k = 1, \dots, +\infty$ leads to

$$\frac{1}{c} \cdot \sum_{k=1}^{+\infty} \|w^{k+1} - w^k\|_{\widehat{G}}^2 \le \Phi_{\alpha,\gamma}^1(w^*). \tag{6.4.5}$$

Using Lemma 3 in [77], we have (6.4.3). ■

Now we show that if $(\alpha,\gamma)\in\mathbb{D}_2\cup\mathbb{D}_3\cup\mathbb{D}_4$ and some additional requirement is made on T, we can have a stronger result than (6.4.3). We first show that the sequence $\{\|w^k-w^{k+1}\|^2_{\widehat{G}}\}$ is non-increasing.

Lemma 6.4.2. *Let the sequence $\{w^k\}$ be generated by iPSPR* (6.1.3)*. If $(\alpha,\gamma)\in\mathbb{D}_2\cup\mathbb{D}_3\cup\mathbb{D}_4$ and the proximal terms S, T are chosen such that* (6.3.32), (6.3.33) *and*

$$T+\frac{1}{2}\Sigma_2+\frac{(1-\alpha)(1-\gamma)}{2-\alpha-\gamma}\beta B^\top B\succeq 0 \tag{6.4.6}$$

hold then there holds that

$$\|w^k-w^{k+1}\|^2_{\widehat{G}}\geq\|w^{k+1}-w^{k+2}\|^2_{\widehat{G}}. \tag{6.4.7}$$

Proof. Note that (6.3.12) also holds with $k:=k+1$, then we have

$$\begin{aligned}(w-w^{k+2})^\top G(w^{k+2}-w^{k+1})\geq{}&\left\langle r^{k+2}-r(w),(1-\alpha-\gamma)\beta r^{k+2}+(1-\alpha)\beta B(y^{k+1}-y^{k+2})\right\rangle\\&+\left\langle w^{k+2}-w,F(w^{k+2},\xi_x^{k+2},\xi_y^{k+2})\right\rangle,\end{aligned}\tag{6.4.8}$$

where $\xi_x^{k+2}\in\partial\theta_1(x^{k+2})$ and $\xi_y^{k+2}\in\partial\theta_2(y^{k+2})$.

By choosing w to be w^{k+2} and w^{k+1}, respectively, in (6.3.12) and (6.4.8), we have

$$\begin{aligned}(w^{k+2}-w^{k+1})^\top G(w^{k+1}-w^k)\geq{}&\left\langle r^{k+1}-r^{k+2},(1-\alpha-\gamma)\beta r^{k+1}+(1-\alpha)\beta B(y^k-y^{k+1})\right\rangle\\&+\left\langle w^{k+1}-w^{k+2},F(w^{k+1},\xi_x^{k+1},\xi_y^{k+1})\right\rangle.\end{aligned}\tag{6.4.9}$$

and

$$\begin{aligned}(w^{k+1}-w^{k+2})^\top G(w^{k+2}-w^{k+1})\geq{}&\left\langle r^{k+2}-r^{k+1},(1-\alpha-\gamma)\beta r^{k+2}+(1-\alpha)\beta B(y^{k+1}-y^{k+2})\right\rangle\\&+\left\langle w^{k+2}-w^{k+1},F(w^{k+2},\xi_x^{k+2},\xi_y^{k+2})\right\rangle.\end{aligned}\tag{6.4.10}$$

Adding (6.4.9) and (6.4.10) and noting from (6.2.3) that $\left\langle w^{k+2}-w^{k+1},F(w^{k+2},\xi_x^{k+2},\xi_y^{k+2})-F(w^{k+1},\xi_x^{k+1},\xi_y^{k+1})\right\rangle\geq\|u^{k+2}-u^{k+1}\|^2_\Sigma$, we obtain that

$$\begin{aligned}&(w^{k+2}-w^{k+1})^\top G\left[(w^{k+1}-w^k)-(w^{k+2}-w^{k+1})\right]\\&\geq(1-\alpha-\gamma)\beta\|r^{k+1}-r^{k+2}\|^2+(1-\alpha)\beta\left\langle B\left[(y^k-y^{k+1})-(y^{k+1}-y^{k+2})\right],r^{k+1}-r^{k+2}\right\rangle\\&\quad+\|u^{k+2}-u^{k+1}\|^2_\Sigma.\end{aligned}\tag{6.4.11}$$

Following the deriving process of (6.2.9) and (6.2.12), we have that

$$\begin{aligned}&\|(w^k-w^{k+1})-(w^{k+1}-w^{k+2})\|^2_G\\&=\|(u^k-u^{k+1})-(u^{k+1}-u^{k+2})\|^2_P+(1-\alpha)\beta\left\|B[(y^k-y^{k+1})-(y^{k+1}-y^{k+2})]\right\|^2\\&\quad+(\alpha+\gamma)\beta\|r^{k+1}-r^{k+2}\|^2.\end{aligned}\tag{6.4.12}$$

Thus we conclude that

$$\begin{aligned}
&\|w^k - w^{k+1}\|^2_{\widehat{G}} - \|w^{k+1} - w^{k+2}\|^2_{\widehat{G}} \\
&= (\|w^k - w^{k+1}\|^2_G + \|u^k - u^{k+1}\|^2_\Sigma) - (\|w^{k+1} - w^{k+2}\|^2_G + \|u^{k+1} - u^{k+2}\|^2_\Sigma) \\
&= 2(w^{k+2} - w^{k+1})^\top G\left[(w^{k+1} - w^k) - (w^{k+2} - w^{k+1})\right] + \|(w^{k+1} - w^k) - (w^{k+2} - w^{k+1})\|^2_G \\
&\quad + \|u^k - u^{k+1}\|^2_\Sigma - \|u^{k+1} - u^{k+2}\|^2_\Sigma \\
&\geq (2-\alpha-\gamma)\beta\|r^{k+1} - r^{k+2}\|^2 + 2(1-\alpha)\beta\left\langle B\left[(y^k - y^{k+1}) - (y^{k+1} - y^{k+2})\right], r^{k+1} - r^{k+2}\right\rangle \\
&\quad + (1-\alpha)\beta\left\|B[(y^k - y^{k+1}) - (y^{k+1} - y^{k+2})]\right\|^2 + \|(u^k - u^{k+1}) - (u^{k+1} - u^{k+2})\|^2_P \\
&\quad + \|u^k - u^{k+1}\|^2_\Sigma + \|u^{k+1} - u^{k+2}\|^2_\Sigma \\
&\geq \frac{(1-\alpha)(1-\gamma)}{2-\alpha-\gamma}\beta\left\|B[(y^k - y^{k+1}) - (y^{k+1} - y^{k+2})]\right\|^2 + \|(u^k - u^{k+1}) - (u^{k+1} - u^{k+2})\|^2_P \\
&\quad + \frac{1}{2}\|(u^k - u^{k+1}) - (u^{k+1} - u^{k+2})\|^2_\Sigma \\
&= \|(x^k - x^{k+1}) - (x^{k+1} - x^{k+2})\|^2_{S+\frac{1}{2}\Sigma_1} + \|(y^k - y^{k+1}) - (y^{k+1} - y^{k+2})\|^2_{T+\frac{1}{2}\Sigma_2+\frac{(1-\alpha)(1-\gamma)}{2-\alpha-\gamma}\beta B^\top B} \\
&\geq 0.
\end{aligned} \tag{6.4.13}$$

where the first inequality is due to (6.4.11) and (6.4.12), the second inequality follows from the Cauchy-Schwarz inequality and the last inequality is due to $P = \begin{pmatrix} S & 0 \\ 0 & T \end{pmatrix}$, $S + \frac{1}{2}\Sigma_1 \succeq 0$ and (6.4.6). The proof is completed. ■

Theorem 6.4.2. *Let the sequence* $\{w^k\}$ *be generated by iPSPR* (6.1.3) *with* $(\alpha, \gamma) \in \mathbb{D}_2 \cup \mathbb{D}_3 \cup \mathbb{D}_4$. *Suppose that the proximal term* S *is chosen according to* (6.3.32) *and* (6.4.2) *and the proximal term* T *is chosen according to* (6.3.33) *and* (6.4.6). *We have*

$$\|w^t - w^{t+1}\|^2_{\widehat{G}} = o(1/t). \tag{6.4.14}$$

Proof. If follows from Theorem 6.4.1 that $\min_{1\leq i\leq t}\|w^i - w^{i+1}\|^2_{\widehat{G}} = o(1/t)$, which with (6.4.7) implies (6.4.14). The proof is completed. ■

6.5 Numerical results

In this section, we demonstrate the potential efficiency of our method iPSPR (6.1.3) by solving the following l_1 regularized least square problem

$$\min\ \frac{1}{2}\|Qy - c\|^2 + \rho\|y\|_1, \quad \text{s.t.} \quad By \leq b, \tag{6.5.1}$$

where $y \in \mathbb{R}^n, c \in \mathbb{R}^p$, $Q \in \mathbb{R}^{p\times n}$ and $B \in \mathbb{R}^{m\times n}$. Problem (6.5.1) is an constrained extension of the ordinary unconstrained l_1 regularized least square problem and it was considered in [77]. By introducing an auxiliary variable $x \in \mathbb{R}^m$, we rewrite (6.5.1) as

$$\min \ \frac{1}{2}\|Qy - c\|^2 + \rho\|y\|_1, \quad \text{s.t.} \quad x + By = b, \ x \geq 0, \tag{6.5.2}$$

which is a special instance of (6.1.1).

For our method iPSPR (6.1.3), we set $S = 0$ and choose T according to (6.3.41), namely,

$$T = rI - (Q^\top Q + \beta B^\top B) \quad \text{with} \quad r = \lambda_{\max}\left(\frac{1}{2}Q^\top Q + \tau\beta B^\top B\right) \tag{6.5.3}$$

with $\tau = 1.001\underline{\tau}^{\alpha,\gamma}$, where $\underline{\tau}^{\alpha,\gamma}$ is defined in (6.3.44). Our method iPSPR (6.1.3) for solving (6.5.2) is then given as

$$\begin{cases} x^{k+1} = \mathcal{P}_+\left[b - By^k + \lambda^k/\beta\right], \\ \lambda^{k+\frac{1}{2}} = \lambda^k - \alpha\beta(x^{k+1} + By^k - b), \\ y^{k+1} = \mathcal{S}_{\rho/r}\left[y^k + \frac{1}{r}\left(B^\top\left(\lambda^{k+\frac{1}{2}} - \beta(x^{k+1} + By^k - b)\right) + Q^\top(c - Qy^k)\right)\right], \\ \lambda^{k+1} = \lambda^{k+\frac{1}{2}} - \gamma\beta(x^{k+1} + By^{k+1} - b), \end{cases} \tag{6.5.4}$$

where the projection operator $\mathcal{P}_+(z) = \max(z, 0)$ and the shrinkage operator $\mathcal{S}_\nu(z) := \text{sgn}(z) \odot \max\{|z| - \nu, 0\}$. Note that for problem (6.5.2), the majorized indefinite proximal ADMM in [77] coincides with our iPSPR (6.5.4) with $\alpha = 0$ and $(\alpha, \gamma) \in \mathbb{D}_1$ since the smooth part of the objective function is quadratic with respect to y. If the proximal parameter $r = 1.001\lambda_{\max}(Q^\top Q + \beta B^\top B)$, iPSPR becomes the semidefinite proximal-based strictly contractive Peaceman-Rachford splitting method (sPSPR). To make a fair comparison, as done in (85) of [77], we stop iPSPR or sPSPR when the KKT residual is less than 10^{-6}.

All the experiments are preformed in Ubuntu 16.04 LTS a Dell workstation with a 3.5 GHz Intel Xeon E3-1240 v5 processor with access to 32 GB of RAM. All the methods are implemented in MATLAB (R2016b). Given m and n, as done in [77], we set $p = 0.1n$, $\rho = 5\sqrt{n}$ and generate the data as

```
B = sprandn(m, n, 0.2);  yy = randn(n, 1);  b = B * yy + max(randn(m,1), 0);
Q = sprandn(p, n, 0.1);  c = Q * yy.
```

In our tests, we set $m = 2000$ and $n = 1000, 2000, 4000, 8000$. For each m and n, we use the above scheme to generate 50 groups of instances and will always report the average performance for methods iPSPR and sPSPR. For each instance, we fix the sum $\alpha + \gamma$ to be $\{1.9, 1.8, 1.618, 1\}$ and always choose the special cases with $\alpha = \gamma$, $\alpha = 0$ or $\gamma = 1$. In total, we have nine groups of choices of α and γ. In our tests, the penalty parameter β is fixed during the iterations. Generally, choosing the best penalty parameter β is not easy and it might be problem dependent [61]. We spent some efforts to choose the penalty parameter β from a large number of candidates. For each given m, n and α, γ, we report the performance of iPSPR or sPSPR with four choices of β. Note that in our tests, the second choice is the best in the candidates for $\alpha + \gamma = 1.9$ and almost the best

Table 6.1: The results for $m = 2000, n = 1000$ over 50 runs. The CPU time is in seconds.

		$\beta = 0.50$			$\beta = 1.50$			$\beta = 3.00$			$\beta = 5.00$		
(α, γ)	method	iter	r	t(s)	iter	r	t(s)	iter	r	t(s)	iter	r	t(s)
(0.950, 0.950)	iPSPR	8769.0	5.23e2	4.9	**4750.1**	1.56e3	2.7	5876.0	3.11e3	3.3	8576.2	5.18e3	4.8
(0.950, 0.950)	sPSPR	8877.3	5.47e2	5.0	**4815.4**	1.60e3	2.7	6000.3	3.19e3	3.4	8810.7	5.31e3	5.0
(0.900, 1.000)	iPSPR	8769.8	5.23e2	5.0	**4752.9**	1.56e3	2.7	5878.7	3.11e3	3.4	8579.8	5.18e3	4.9
(0.900, 1.000)	sPSPR	8878.4	5.47e2	5.0	**4817.9**	1.60e3	2.8	6001.5	3.19e3	3.4	8811.9	5.31e3	5.0
(0.900, 0.900)	iPSPR	9090.5	5.10e2	5.1	**4884.2**	1.52e3	2.8	5815.1	3.03e3	3.3	8375.5	5.04e3	4.7
(0.900, 0.900)	sPSPR	9252.9	5.47e2	5.2	**4982.8**	1.60e3	2.8	6075.2	3.19e3	3.4	8880.5	5.31e3	5.0
(0.800, 1.000)	iPSPR	9094.7	5.10e2	5.2	**4887.6**	1.52e3	2.8	5823.3	3.03e3	3.3	8405.3	5.04e3	4.8
(0.800, 1.000)	sPSPR	9256.5	5.47e2	5.2	**4985.6**	1.60e3	2.8	6079.5	3.19e3	3.5	8894.2	5.31e3	5.0
(0.809, 0.809)	iPSPR	9706.1	4.86e2	5.5	**5250.0**	1.44e3	3.0	5667.2	2.88e3	3.2	8094.9	4.80e3	4.6
(0.809, 0.809)	sPSPR	10058.7	5.47e2	5.7	**5388.7**	1.60e3	3.1	6240.7	3.19e3	3.6	8960.9	5.31e3	5.1
(0.618, 1.000)	iPSPR	9722.5	4.86e2	5.5	**5245.5**	1.44e3	3.0	5694.4	2.88e3	3.2	8124.6	4.80e3	4.6
(0.618, 1.000)	sPSPR	10074.1	5.47e2	5.7	**5392.1**	1.60e3	3.1	6261.2	3.19e3	3.6	9007.0	5.31e3	5.1
(0.000, 1.618)	iPSPR	10216.2	5.37e2	5.8	**5511.2**	1.60e3	3.1	6602.8	3.19e3	3.8	9583.0	5.31e3	5.4
(0.000, 1.618)	sPSPR	10264.2	5.47e2	5.8	**5517.4**	1.60e3	3.1	6605.5	3.19e3	3.8	9583.1	5.31e3	5.4
(0.000, 1.000)	iPSPR	13521.7	4.05e2	7.6	7761.5	1.20e3	4.4	**5967.7**	2.39e3	3.4	7667.3	3.98e3	4.3
(0.000, 1.000)	sPSPR	14513.1	5.47e2	8.2	8041.4	1.60e3	4.6	**7311.9**	3.19e3	4.2	9930.7	5.31e3	5.6
(0.500, 0.500)	iPSPR	13308.9	4.05e2	7.5	7724.5	1.20e3	4.4	**5728.7**	2.39e3	3.3	7265.2	3.98e3	4.1
(0.500, 0.500)	sPSPR	14336.6	5.47e2	8.1	7964.6	1.60e3	4.5	**7097.4**	3.19e3	4.0	9540.8	5.31e3	5.4

Table 6.2: The results for $m = 2000, n = 2000$ over 50 runs. The CPU time is in seconds.

		$\beta = 0.10$			$\beta = 0.30$			$\beta = 0.50$			$\beta = 1.00$		
(α, γ)	method	iter	r	t(s)	iter	r	t(s)	iter	r	t(s)	iter	r	t(s)
(0.950, 0.950)	iPSPR	2192.4	2.18e2	3.2	**1012.0**	4.43e2	1.5	1240.6	7.22e2	1.8	2328.8	1.43e3	3.4
(0.950, 0.950)	sPSPR	2357.0	3.83e2	3.4	**1121.2**	5.22e2	1.7	1309.0	7.66e2	1.9	2417.4	1.48e3	3.5
(0.900, 1.000)	iPSPR	2192.8	2.18e2	3.2	**1012.2**	4.43e2	1.5	1240.9	7.22e2	1.9	2329.3	1.43e3	3.4
(0.900, 1.000)	sPSPR	2357.3	3.83e2	3.4	**1121.4**	5.22e2	1.7	1309.2	7.66e2	2.0	2417.9	1.48e3	3.5
(0.900, 0.900)	iPSPR	2294.6	2.17e2	3.3	**1028.0**	4.32e2	1.5	1226.5	7.04e2	1.8	2274.1	1.39e3	3.3
(0.900, 0.900)	sPSPR	2474.9	3.83e2	3.6	**1146.1**	5.22e2	1.7	1324.5	7.66e2	2.0	2423.7	1.48e3	3.5
(0.800, 1.000)	iPSPR	2295.3	2.17e2	3.4	**1028.4**	4.32e2	1.6	1227.1	7.04e2	1.8	2276.1	1.39e3	3.3
(0.800, 1.000)	sPSPR	2475.5	3.83e2	3.6	**1146.8**	5.22e2	1.7	1325.2	7.66e2	2.0	2425.9	1.48e3	3.5
(0.809, 0.809)	iPSPR	2509.4	2.14e2	3.6	**1074.5**	4.13e2	1.6	1201.1	6.71e2	1.8	2170.0	1.33e3	3.2
(0.809, 0.809)	sPSPR	2722.4	3.83e2	3.9	**1201.1**	5.22e2	1.8	1351.8	7.66e2	2.0	2437.7	1.48e3	3.5
(0.618, 1.000)	iPSPR	2511.7	2.14e2	3.6	**1075.4**	4.13e2	1.6	1204.1	6.71e2	1.8	2177.7	1.33e3	3.2
(0.618, 1.000)	sPSPR	2724.1	3.83e2	3.9	**1202.5**	5.22e2	1.8	1354.7	7.66e2	2.0	2445.8	1.48e3	3.6
(0.000, 1.618)	iPSPR	2541.2	2.20e2	3.7	**1138.7**	4.53e2	1.7	1366.0	7.40e2	2.0	2547.4	1.47e3	3.7
(0.000, 1.618)	sPSPR	2733.0	3.83e2	3.9	**1228.7**	5.22e2	1.8	1406.8	7.66e2	2.1	2571.5	1.48e3	3.7
(0.000, 1.000)	iPSPR	3737.8	2.05e2	5.4	1551.7	3.50e2	2.3	**1242.2**	5.60e2	1.8	1910.1	1.10e3	2.8
(0.000, 1.000)	sPSPR	4152.4	3.83e2	5.9	1649.0	5.22e2	2.4	**1542.5**	7.66e2	2.3	2559.0	1.48e3	3.7
(0.500, 0.500)	iPSPR	3716.2	2.05e2	5.3	1546.0	3.50e2	2.3	**1213.4**	5.60e2	1.8	1831.6	1.10e3	2.7
(0.500, 0.500)	sPSPR	4137.3	3.83e2	5.9	1637.5	5.22e2	2.4	**1509.1**	7.66e2	2.2	2481.4	1.48e3	3.6

Table 6.3: The results for $m = 2000, n = 4000$ over 50 runs. The CPU time is in seconds.

		$\beta = 0.08$			$\beta = 0.15$			$\beta = 0.25$			$\beta = 0.50$		
(α, γ)	method	iter	r	t(s)	iter	r	t(s)	iter	r	t(s)	iter	r	t(s)
(0.950, 0.950)	iPSPR	905.6	3.71e2	3.0	**672.0**	4.24e2	2.3	831.3	5.65e2	2.8	1657.8	1.06e3	5.4
(0.950, 0.950)	sPSPR	1207.2	7.03e2	4.0	**1103.8**	7.39e2	3.7	1233.3	8.09e2	4.1	1813.6	1.15e3	5.9
(0.900, 1.000)	iPSPR	905.7	3.71e2	3.0	**672.0**	4.24e2	2.3	831.4	5.65e2	2.8	1657.9	1.06e3	5.4
(0.900, 1.000)	sPSPR	1207.2	7.03e2	4.0	**1103.9**	7.39e2	3.7	1233.3	8.09e2	4.1	1813.7	1.15e3	5.9
(0.900, 0.900)	iPSPR	943.5	3.70e2	3.2	**681.6**	4.21e2	2.3	812.2	5.54e2	2.8	1615.6	1.03e3	5.3
(0.900, 0.900)	sPSPR	1234.3	7.03e2	4.1	**1103.9**	7.39e2	3.7	1231.0	8.09e2	4.1	1815.3	1.15e3	6.0
(0.800, 1.000)	iPSPR	943.7	3.70e2	3.2	**681.7**	4.21e2	2.3	812.3	5.54e2	2.8	1616.2	1.03e3	5.3
(0.800, 1.000)	sPSPR	1234.5	7.03e2	4.1	**1104.4**	7.39e2	3.7	1231.3	8.09e2	4.1	1816.0	1.15e3	6.0
(0.809, 0.809)	iPSPR	1029.0	3.68e2	3.4	**706.1**	4.14e2	2.4	779.1	5.34e2	2.7	1537.7	9.86e2	5.1
(0.809, 0.809)	sPSPR	1294.0	7.03e2	4.3	**1108.9**	7.39e2	3.7	1226.0	8.09e2	4.1	1817.5	1.15e3	6.0
(0.618, 1.000)	iPSPR	1029.6	3.68e2	3.4	**706.8**	4.14e2	2.4	780.9	5.34e2	2.7	1540.7	9.86e2	5.1
(0.618, 1.000)	sPSPR	1294.5	7.03e2	4.3	**1109.9**	7.39e2	3.7	1227.7	8.09e2	4.1	1820.4	1.15e3	6.0
(0.000, 1.618)	iPSPR	1035.6	3.73e2	3.5	**731.1**	4.28e2	2.5	876.7	5.77e2	3.0	1764.3	1.09e3	5.8
(0.000, 1.618)	sPSPR	1300.3	7.03e2	4.3	**1125.7**	7.39e2	3.7	1256.0	8.09e2	4.2	1876.5	1.15e3	6.2
(0.000, 1.000)	iPSPR	1594.1	3.61e2	5.2	927.6	3.95e2	3.1	**759.7**	4.74e2	2.6	1300.8	8.25e2	4.3
(0.000, 1.000)	sPSPR	1727.8	7.03e2	5.7	1232.1	7.39e2	4.1	**1223.3**	8.09e2	4.1	1847.1	1.15e3	6.1
(0.500, 0.500)	iPSPR	1589.2	3.61e2	5.2	922.7	3.95e2	3.1	**747.5**	4.74e2	2.6	1264.4	8.25e2	4.2
(0.500, 0.500)	sPSPR	1724.1	7.03e2	5.7	1224.8	7.39e2	4.1	**1205.5**	8.09e2	4.0	1810.6	1.15e3	6.0

Table 6.4: The results for $m = 2000, n = 8000$ over 50 runs. The CPU time is in seconds.

		$\beta = 0.04$			$\beta = 0.07$			$\beta = 0.15$			$\beta = 0.30$		
(α, γ)	method	iter	r	t(s)	iter	r	t(s)	iter	r	t(s)	iter	r	t(s)
(0.950, 0.950)	iPSPR	889.6	6.80e2	6.6	**759.9**	6.95e2	5.7	861.4	7.55e2	6.4	1236.9	1.05e3	9.1
(0.950, 0.950)	sPSPR	**1487.6**	1.34e3	10.7	1556.1	1.36e3	11.2	1658.1	1.40e3	11.9	1819.2	1.52e3	13.1
(0.900, 1.000)	iPSPR	889.7	6.80e2	6.6	**760.0**	6.95e2	5.7	861.5	7.55e2	6.4	1236.9	1.05e3	9.1
(0.900, 1.000)	sPSPR	**1487.6**	1.34e3	10.7	1556.2	1.36e3	11.2	1658.1	1.40e3	11.9	1819.2	1.52e3	13.1
(0.900, 0.900)	iPSPR	913.9	6.80e2	6.9	**761.9**	6.94e2	5.7	854.8	7.51e2	6.4	1210.7	1.03e3	8.9
(0.900, 0.900)	sPSPR	1488.1	1.34e3	10.9	**1550.8**	1.36e3	11.2	1655.9	1.40e3	11.9	1819.2	1.52e3	13.1
(0.800, 1.000)	iPSPR	913.9	6.80e2	6.9	**762.0**	6.94e2	5.8	854.9	7.51e2	6.4	1211.0	1.03e3	9.0
(0.800, 1.000)	sPSPR	1488.3	1.34e3	10.9	**1550.9**	1.36e3	11.3	1656.0	1.40e3	12.0	1819.6	1.52e3	13.2
(0.809, 0.809)	iPSPR	968.8	6.79e2	7.2	**770.7**	6.93e2	5.8	842.1	7.44e2	6.3	1164.0	9.89e2	8.7
(0.809, 0.809)	sPSPR	1497.5	1.34e3	10.8	**1539.4**	1.36e3	11.2	1650.8	1.40e3	12.0	1819.0	1.52e3	13.2
(0.618, 1.000)	iPSPR	969.2	6.79e2	7.4	**771.2**	6.93e2	5.8	842.9	7.44e2	6.3	1165.4	9.89e2	8.6
(0.618, 1.000)	sPSPR	1497.7	1.34e3	10.2	**1539.8**	1.36e3	11.2	1651.7	1.40e3	12.0	1820.6	1.52e3	13.2
(0.000, 1.618)	iPSPR	972.6	6.81e2	7.3	**779.9**	6.96e2	5.9	873.9	7.59e2	6.6	1289.3	1.07e3	9.5
(0.000, 1.618)	sPSPR	1501.2	1.34e3	11.0	**1546.2**	1.36e3	11.2	1664.1	1.40e3	12.0	1845.1	1.52e3	13.4
(0.000, 1.000)	iPSPR	1406.3	6.76e2	10.9	922.8	6.87e2	6.9	**799.3**	7.25e2	6.1	1030.0	8.76e2	7.7
(0.000, 1.000)	sPSPR	1732.8	1.34e3	13.2	**1500.3**	1.36e3	11.0	1625.9	1.40e3	11.8	1825.7	1.52e3	13.3
(0.500, 0.500)	iPSPR	1403.0	6.76e2	11.0	919.9	6.87e2	6.9	**790.6**	7.25e2	6.0	1014.0	8.76e2	7.6
(0.500, 0.500)	sPSPR	1730.7	1.34e3	13.3	**1496.1**	1.36e3	10.9	1617.5	1.40e3	11.8	1809.7	1.52e3	13.1

choice in the candidates for $\alpha+\gamma \in \{1.618, 1.8\}$; the third choice of β is the best in the candidates for $\alpha+\gamma=1$.

The numerical results are presented in Tables 6.1 - 6.4. In the tables, "iter" means the averaged iteration numbers, "r" denotes the proximal parameter in (6.5.4), and "t" means the CPU time in seconds. From the tables, we can make the following observations. Firstly, iPSPR always performs better than sPSPR. In particular, for $n=4000, \beta=0.15$ and $n=8000, \beta=0.07$, iPSPR can bring about a 40% to 50% reduction in the number of iterations and the CPU time over the sPSPR. For $n=1000$ and 2000, iPSPR with large sum $\alpha+\gamma$ performs only slightly better than sPSPR with the same α and γ. This might be because $\beta B^\top B$ takes a major part in computing r and the parameter τ of iPSPR is near to 1 in this case. Secondly, iPSPR (resp. sPSPR) with $\alpha=\gamma$ performs slightly better among the choices of α and γ with a fixed sum. Thirdly, a large $\alpha+\gamma$ sum (near to 2) always performs better than a small sum for a relatively small β, while a small sum works better than a large sum for a relatively large β. However, if we choose the best β (the corresponding results are marked in bold in each table) for each α and γ, we can see that iPSPR (resp. sPSPR) a large $\alpha+\gamma$ sum always performs better than iPSPR (resp. sPSPR) with a small sum.

6.6 Conclusions

In this chapter, we proposed a modification of the Peaceman-Rachford splitting method by introducing two different parameters α and γ in updating the dual variable, and by introducing indefinite proximal terms to the subproblems in updating the primal variables. We established the relationship between the two parameters α and γ and proved the global convergence of the algorithm under some requirements on the proximal matrices S and T. Moreover, we provided a specific construction of the proximal matrix T and discussed the detailed performance for the variant parameters α and γ, which can unify several existing results. We also analyzed the $o(1/t)$ sublinear rate convergence in the nonergodic sense. Finally, we reported some preliminary numerical results, indicating the efficiency of the proposed algorithm.

Chapter 7

Conclusion

In this thesis, we have proposed two proximal type splitting methods for the structured convex optimization problems with linear constraints. One is the alternating direction methods of multipliers (ADMM), and the other is the Peaceman-Rachford splitting methods (PRSM).

The results obtained in this thesis are summarized as follows.

(a) In Chapter 3, we have proposed a proximal ADMM whose regularized matrix in the proximal term is generated by the BFGS update (or limited memory BFGS) at every iteration. This method turned out to be the variable metric semi-proximal ADMM (VMSP-ADMM). We have shown that the update formula for the positive semidefinite proximal matrices follows $T_k = B_k - M$, where $M = \nabla^2_{xx}\mathcal{L}_\beta(x, y, \lambda)$ and B_k is generated via the BFGS update. We have proved that $B_k \succeq M$ for global convergence, which also means T_k to be positive semidefinite. These types of matrices use second-order information of the objective function, which can speed up the convergence compared with the classical proximal ADMM on the numerical view .

(b) In Chapter 4, for solving more general optimization problems, we have extended the proposed proximal ADMM in Chapter 3 for two general convex problems and further extended the proximal terms by the Broyden family update. For the generic variable metric semi-proximal ADMM, we have also shown the global convergence under the standard assumptions on the proximal matrices. In the numerical experiment, we have implemented the standard ADMM and proximal ADMM and then observed good performance of the proposed method.

(c) In Chapter 5, we have considered a variable metric indefinite proximal ADMM (VMIP-ADMM) which can allow a larger stepsize and can unify several existing ADMMs. We have presented sufficient conditions on the indefinite proximal matrices for the global convergence. Moreover, motivated by the previous two chapters, we have provided a construction of the indefinite term via the BFGS update as $T_k = B_k - M$, where B_k is generated by the BFGS update with respect to τM, $\tau < 1$. We have also shown that this construction of the proximal term satisfies the above conditions for the global convergence property when $\tau \in (0.75, 1)$. Finally, we have examined the behavior of the proposed method by the numerical experiments on real-world datasets and synthetic datasets. The results all demonstrated that our proposed variable metric indefinite proximal ADMM outperforms most of the comparison proximal ADMMs.

(d) In Chapter 6, we have focused on another splitting method for minimizing a convex optimization problem with a separable objective function and linear constraints, which is called the Peaceman-Rachford splitting method (PRSM). We have extended the strictly contractive Peaceman-Rachford splitting method by using two different relaxation factors. Besides, motivated by the recent advances on the ADMM type method with indefinite proximal terms, we have employed the indefinite proximal term in the strictly contractive Peaceman-Rachford splitting method. We have shown that the proposed indefinite-proximal strictly contractive Peaceman-Rachford splitting method is convergent and also proved the $o(1/t)$ convergence rate in the nonergodic sense. The numerical tests on the l_1 regularized least square problem have demonstrated the efficiency of the proposed method.

As we summarized above, we have made several contributions to the splitting methods for the separable optimization problems. However, there are many problems that remain unknown. Finally, we discuss some future research topics based on our current achievements.

Regarding the separability of the objective functions which ADMM can be used, the advantages of ADMM are numerous: 1. it allows a linear scaling as the data is processed in parallel progress; 2. it does not need gradients steps; 3. it is resistant to poor conditioning. However, there are still several challenges that should be overcome to spread the uses of ADMM.

(a) Considering that the main task in the splitting methods is to solve the x- or y-optimization problem, solving them in an inexact manner may improve the efficiency of the algorithm. The approximate version with practical accuracy criteria is one of our future research topics. On the other hand, the parameters of the primal and dual problems are essential to the efficiency of the algorithm, which should be variable along with the iteration. Allowing the parameters varying with the process of the iterate may give us the freedom to choose them in a self-adaptive manner. It is interesting to study such suitable updating rules.

(b) In this thesis, some variants of ADMM for the convex optimization problems have been studied to improve the implantations. Considering the good performance of the ADMM in many machine learning applications, efficient ADMM-based methods are desired. A typical machine learning problem consists of a combination of linear and nonlinear constraints, which make the optimization problem be nonconvex. The general analysis of convex functions is not enough. For a nonconvex problem, classical ADMM could be used. Some additional assumptions should be included to guarantee its global convergence. Developing an improved and efficient ADMM with better global convergent conditions is a challenging topic.

Bibliography

[1] H. Attouch, L. M. Briceno-Arias, and P. L. Combettes, *A parallel splitting method for coupled monotone inclusions*, SIAM Journal on Control and Optimization, 48 (2010), pp. 3246–3270.

[2] J. Bai, J. Li, F. Xu, and H. Zhang, *Generalized symmetric ADMM for separable convex optimization*, Computational Optimization and Applications, 70 (2018), pp. 129–170.

[3] S. Banert, R. I. Boţ, and E. R. Csetnek, *Fixing and extending some recent results on the ADMM algorithm*, Numerical Algorithms, (2016), pp. 1–23.

[4] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, *Convexity, classification, and risk bounds*, Journal of the American Statistical Association, 101 (2006), pp. 138–156.

[5] H. H. Bauschke, P. L. Combettes, et al., *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, vol. 408, Springer, 2011.

[6] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM journal on imaging sciences, 2 (2009), pp. 183–202.

[7] ——, *Gradient-based algorithms with applications to signal recovery*, Convex optimization in signal processing and communications, (2009), pp. 42–88.

[8] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Academic press, 1982.

[9] ——, *Nonlinear Programming*, Athena scientific Belmont, 1999.

[10] D. P. Bertsekas, A. Nedi, A. E. Ozdaglar, et al., *Convex Analysis and Optimization*, Athena Scientific, 2003.

[11] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, vol. 23, Prentice hall Englewood Cliffs, NJ, 1989.

[12] J. Borwein and A. S. Lewis, *Convex Analysis and Nonlinear Optimization: Theory and Examples*, Springer Science & Business Media, 2010.

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Foundations and Trends® in Machine Learning, 3 (2011), pp. 1–122.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge university press, 2004.

[15] A. M. Bruckstein, D. L. Donoho, and M. Elad, *From sparse solutions of systems of equations to sparse modeling of signals and images*, SIAM review, 51 (2009), pp. 34–81.

[16] J.-F. Cai, S. Osher, and Z. Shen, *Split bregman methods and frame based image restoration*, Multiscale modeling & simulation, 8 (2009), pp. 337–369.

[17] X. Cai, D. Han, and X. Yuan, *On the convergence of the direct extension of ADMM for three-block separable convex minimization models with one strongly convex function*, Computational Optimization and Applications, 66 (2017), pp. 39–73.

[18] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, Journal of mathematical imaging and vision, 40 (2011), pp. 120–145.

[19] C. Chen, B. He, Y. Ye, and X. Yuan, *The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent*, Mathematical Programming, 155 (2016), pp. 57–79.

[20] C. Chen, Y. Shen, and Y. You, *On the convergence analysis of the alternating direction method of multipliers with three blocks*, in Abstract and Applied Analysis, vol. 2013, Hindawi, 2013.

[21] G. Chen and M. Teboulle, *A proximal-based decomposition method for convex minimization problems*, Mathematical Programming, 64 (1994), pp. 81–101.

[22] L. Chen, X. Li, D. Sun, and K.-C. Toh, *On the equivalence of inexact proximal ALM and ADMM for a class of convex composite programming*, Mathematical Programming, (2019).

[23] P. Chen, J. Huang, and X. Zhang, *A primal–dual fixed point algorithm for convex separable minimization with applications to image restoration*, Inverse Problems, 29 (2013), p. 025011.

[24] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, SIAM review, 43 (2001), pp. 129–159.

[25] Y. Chen, Y.-H. Dai, and D. Han, *Fiber orientation distribution estimation using a Peaceman-Rachford splitting method*, SIAM Journal on Imaging Sciences, 9 (2016), pp. 573–604.

[26] P. L. Combettes, *Iterative construction of the resolvent of a sum of maximal monotone operators*, Journal of Convex Analysis, 16 (2009), pp. 727–748.

[27] P. L. Combettes and J.-C. Pesquet, *Proximal splitting methods in signal processing*, in Fixed-point algorithms for inverse problems in science and engineering, Springer, 2011, pp. 185–212.

[28] E. Corman and X. Yuan, *A generalized proximal point algorithm and its convergence rate*, SIAM Journal on Optimization, 24 (2014), pp. 1614–1638.

[29] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, *Parallel multi-block admm with $o(1/k)$ convergence*, Journal of Scientific Computing, 71 (2017), pp. 712–736.

[30] W. Deng and W. Yin, *On the global and linear convergence of the generalized alternating direction method of multipliers*, Journal of Scientific Computing, 66 (2016), pp. 889–916.

[31] D. L. Donoho, *Compressed sensing*, IEEE Transactions on information theory, 52 (2006), pp. 1289–1306.

[32] J. Douglas and H. Rachford, *On the numerical solution of heat conduction problems in two and three space variables*, Transactions of the American mathematical Society, (1956), pp. 421–439.

[33] D. Dua and C. Graff, *UCI machine learning repository*, 2017.

[34] J. Eckstein, *Some saddle-function splitting methods for convex programming*, Optimization Methods and Software, 4 (1994), pp. 75–83.

[35] J. Eckstein and D. P. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Mathematical Programming, 55 (1992), pp. 293–318.

[36] J. Eckstein and M. Fukushima, *Some reformulations and applications of the alternating direction method of multipliers*, in Large Scale Optimization, Springer, 1994, pp. 115–134.

[37] J. Eckstein and W. Yao, *Approximate ADMM algorithms derived from Lagrangian splitting*, Computational Optimization and Applications, 68 (2017), pp. 363–405.

[38] ——, *Relative-error approximate versions of Douglas–Rachford splitting and special cases of the ADMM*, Mathematical Programming, 170 (2018), pp. 417–444.

[39] E. Esser, X. Zhang, and T. Chan, *A general framework for a class of first order primal-dual algorithms for tv minimization*, Ucla Cam Report, (2009), pp. 09–67.

[40] M. Fazel, T. K. Pong, D. Sun, and P. Tseng, *Hankel matrix rank minimization with applications to system identification and realization*, SIAM Journal on Matrix Analysis and Applications, 34 (2013), pp. 946–977.

[41] M. C. Ferris and J.-S. Pang, *Engineering and economic applications of complementarity problems*, Siam Review, 39 (1997), pp. 669–713.

[42] R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, 2013.

[43] C. A. Floudas and P. M. Pardalos, *Encyclopedia of Optimization*, vol. 1, Springer Science & Business Media, 2001.

[44] M. Fortin and R. Glowinski, *Chapter iii on decomposition-coordination methods using an augmented lagrangian*, in Studies in Mathematics and Its Applications, vol. 15, Elsevier, 1983, pp. 97–146.

[45] D. Gabay, *Chapter ix applications of the method of multipliers to variational inequalities*, in Studies in Mathematics and Its Applications, vol. 15, Elsevier, 1983, pp. 299–331.

[46] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite element approximation*, Computers & Mathematics with Applications, 2 (1976), pp. 17–40.

[47] B. Gao and F. Ma, *Symmetric alternating direction method with indefinite proximal regularization for linearly constrained convex optimization*, Journal of Optimization Theory and Applications, 176 (2018), pp. 178–204.

[48] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, vol. 9, SIAM, 1989.

[49] R. Glowinski and A. Marroco, *Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires*, ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique, 9 (1975), pp. 41–76.

[50] T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk, *Fast alternating direction optimization methods*, SIAM Journal on Imaging Sciences, 7 (2014), pp. 1588–1623.

[51] T. Goldstein and S. Osher, *The split bregman method for l1-regularized problems*, SIAM journal on imaging sciences, 2 (2009), pp. 323–343.

[52] G. H. Golub and C. F. Van Loan, *Matrix Computations*, vol. 3, JHU Press, 2012.

[53] M. L. N. Gonçalves, M. M. Alves, and J. G. Melo, *Pointwise and ergodic convergence rates of a variable metric proximal alternating direction method of multipliers*, Journal of Optimization Theory and Applications, 177 (2018), pp. 448–478.

[54] Y. Gu and D. Han, *Modified strictly contractive Peaceman–Rachford splitting method (in Chinese)*, 2015.

[55] J. K. Hale and S. M. V. Lunel, *Introduction to Functional Differential Equations*, vol. 99, Springer Science & Business Media, 2013.

[56] D. Han and X. Yuan, *A note on the alternating direction method of multipliers*, Journal of Optimization Theory and Applications, 155 (2012), pp. 227–238.

[57] D. Han, X. Yuan, and W. Zhang, *An augmented lagrangian based parallel splitting method for separable convex minimization with applications to image processing*, Mathematics of Computation, 83 (2014), pp. 2263–2291.

[58] D. Harrison Jr and D. L. Rubinfeld, *Hedonic housing prices and the demand for clean air*, Journal of environmental economics and management, 5 (1978), pp. 81–102.

[59] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Science & Business Media, 2009.

[60] B. He, L.-Z. Liao, D. Han, and H. Yang, *A new inexact alternating directions method for monotone variational inequalities*, Mathematical Programming, 92 (2002), pp. 103–118.

[61] B. He, H. Liu, Z. Wang, and X. Yuan, *A strictly contractive Peaceman–Rachford splitting method for convex programming*, SIAM Journal on Optimization, 24 (2014), pp. 1011–1040.

[62] B. He, F. Ma, and X. Yuan, *Convergence study on the symmetric version of ADMM with larger step sizes*, SIAM Journal on Imaging Sciences, 9 (2016), pp. 1467–1501.

[63] ——, *Optimally linearizing the alternating direction method of multipliers for convex programming*, Computational Optimization and Applications, (2019), pp. 1–28.

[64] B. He, M. Tao, and X. Yuan, *Alternating direction method with gaussian back substitution for separable convex programming*, SIAM Journal on Optimization, 22 (2012), pp. 313–340.

[65] B. He and X. Yuan, *On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method*, SIAM Journal on Numerical Analysis, 50 (2012), pp. 700–709.

[66] ——, *Block-wise alternating direction method of multipliers for multiple-block convex programming and beyond*, SMAI Journal of Computational Mathematics, 1 (2015), pp. 145–174.

[67] ——, *On non-ergodic convergence rate of Douglas–Rachford alternating direction method of multipliers*, Numerische Mathematik, 130 (2015), pp. 567–577.

[68] Y. He, H. Li, and X. Liu, *Relaxed inertial proximal Peaceman–Rachford splitting method for separable convex programming*, Frontiers of Mathematics in China, 13 (2018), pp. 555–578.

[69] M. R. Hestenes, *Multiplier and gradient methods*, Journal of optimization theory and applications, 4 (1969), pp. 303–320.

[70] M. Hong and Z.-Q. Luo, *On the linear convergence of the alternating direction method of multipliers*, Mathematical Programming, 162 (2017), pp. 165–199.

[71] J. Huang, T. Zhang, and D. Metaxas, *Learning with structured sparsity*, Journal of Machine Learning Research, 12 (2011), pp. 3371–3412.

[72] F. Jiang, Z. Wu, and X. Cai, *Generalized ADMM with optimal indefinite proximal term for linearly constrained convex optimization*, Journal of Industrial & Management Optimization, (2018), pp. 183–202.

[73] K. Koh, S.-J. Kim, and S. Boyd, *An interior-point method for large-scale l1-regularized logistic regression*, Journal of Machine learning research, 8 (2007), pp. 1519–1555.

[74] M. Kowalski and B. Torrésani, *Structured sparsity: from mixed norms to structured shrinkage*, in SPARS'09-Signal Processing with Adaptive Sparse Structured Representations, 2009.

[75] G. Li, T. Liu, and T. K. Pong, *Peaceman–Rachford splitting for a class of nonconvex optimization problems*, Computational Optimization and Applications, 68 (2017), pp. 407–436.

[76] M. Li, D. Sun, and K.-C. Toh, *A convergent 3-block semi-proximal ADMM for convex minimization problems with one strongly convex block*, Asia-Pacific Journal of Operational Research, 32 (2015), p. 1550024.

[77] ——, *A majorized admm with indefinite proximal terms for linearly constrained convex composite optimization*, SIAM Journal on Optimization, 26 (2016), pp. 922–950.

[78] X. Li and X. Yuan, *A proximal strictly contractive Peaceman–Rachford splitting method for convex programming with applications to imaging*, SIAM Journal on Imaging Sciences, 8 (2015), pp. 1332–1365.

[79] T. Lin, S. Ma, and S. Zhang, *On the global linear convergence of the ADMM with multiblock variables*, SIAM Journal on Optimization, 25 (2015), pp. 1478–1497.

[80] P.-L. Lions and B. Mercier, *Splitting algorithms for the sum of two nonlinear operators*, SIAM Journal on Numerical Analysis, 16 (1979), pp. 964–979.

[81] R. Lopes, S. A. Santos, and P. J. Silva, *Accelerating block coordinate descent methods with identification strategies*, Computational Optimization and Applications, 72 (2019), pp. 609–640.

[82] P. A. Lotito, L. A. Parente, and M. Solodov, *A class of variable metric decomposition methods for monotone variational inclusions*, J. Convex Anal, 16 (2009), pp. 857–880.

[83] D. G. Luenberger, Y. Ye, et al., *Linear and Nonlinear Programming*, vol. 2, Springer, 1984.

[84] L. Meier, S. Van De Geer, and P. Bühlmann, *The group lasso for logistic regression*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 70 (2008), pp. 53–71.

[85] R. D. Monteiro and B. F. Svaiter, *Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers*, SIAM Journal on Optimization, 23 (2013), pp. 475–507.

[86] A. Nagurney, *Network Economics: A Variational Inequality Approach*, vol. 10, Springer Science & Business Media, 2013.

[87] Y. Nesterov, *A method of solving a convex programming problem with convergence rate* $O(1/k^2)$, Dokl. Akad. Nauk SSSR, 27 (1983), pp. 372–376.

[88] ——, *Gradient methods for minimizing composite functions*, Mathematical Programming, 140 (2013), pp. 125–161.

[89] ——, *Introductory Lectures on Convex Optimization: A Basic Course*, vol. 87, Springer Science & Business Media, 2013.

[90] ——, *Lectures on Convex Optimization*, vol. 137, Springer, 2018.

[91] A. Y. Ng, *Feature selection,* l_1 *vs.* l_2 *regularization, and rotational invariance*, in Proceedings of the twenty-first international conference on Machine learning, 2004, p. 78.

[92] M. K. Ng, F. Wang, and X. Yuan, *Inexact alternating direction methods for image recovery*, SIAM Journal on Scientific Computing, 33 (2011), pp. 1643–1668.

[93] J. Nocedal, *Updating quasi-newton matrices with limited storage*, Mathematics of computation, 35 (1980), pp. 773–782.

[94] J. Nocedal and S. Wright, *Numerical Optimization*, Springer Science & Business Media, 2006.

[95] R. K. Pace and R. Barry, *Sparse spatial autoregressions*, Statistics & Probability Letters, 33 (1997), pp. 291–297.

[96] J.-S. Pang and F. Facchinei, *Finite-dimensional variational inequalities and complementarity problems*, 2003.

[97] N. Parikh and S. Boyd, *Proximal algorithms*, Foundations and Trends in optimization, 1 (2014), pp. 127–239.

[98] D. W. Peaceman and H. H. Rachford, Jr, *The numerical solution of parabolic and elliptic differential equations*, Journal of the Society for Industrial & Applied Mathematics, 3 (1955), pp. 28–41.

[99] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization, Fletcher, R. (ed.), Academic Press, 1969, pp. 283–298.

[100] R. Rockafellar, *Convex Analysis*, Princeton Univ. Press, 1970.

[101] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, vol. 317, Springer Science & Business Media, 2009.

[102] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D: nonlinear phenomena, 60 (1992), pp. 259–268.

[103] H. Sun, M. Tian, and M. Sun, *The symmetric ADMM with indefinite proximal regularization and its application*, Journal of Inequalities and Applications, 172 (2017).

[104] M. Tao and X. Yuan, *Recovering low-rank and sparse components of matrices from incomplete and noisy observations*, SIAM Journal on Optimization, 21 (2011), pp. 57–81.

[105] H. L. Taylor, S. C. Banks, and J. F. McCoy, *Deconvolution with the l_1 norm*, Geophysics, 44 (1979), pp. 39–52.

[106] R. Tibshirani, *Regression shrinkage and selection via the lasso*, Journal of the Royal Statistical Society. Series B (Methodological), (1996), pp. 267–288.

[107] A. N. Tikhonov, A. Goncharsky, V. Stepanov, and A. G. Yagola, *Numerical methods for the solution of ill-posed problems*, vol. 328, Springer Science & Business Media, 2013.

[108] P. Tseng, *Approximation accuracy, gradient methods, and error bound for structured convex optimization*, Mathematical Programming, 125 (2010), pp. 263–295.

[109] P. Tseng and S. Yun, *A coordinate gradient descent method for nonsmooth separable minimization*, Mathematical Programming, 117 (2009), pp. 387–423.

[110] C. workbench team, *A phamacology dataset*, 2008.

[111] S. J. Wright, *Coordinate descent algorithms*, Mathematical Programming, 151 (2015), pp. 3–34.

[112] T. T. Wu, K. Lange, et al., *Coordinate descent algorithms for lasso penalized regression*, The Annals of Applied Statistics, 2 (2008), pp. 224–244.

[113] Z. Wu, M. Li, D. Z. Wang, and D. Han, *A symmetric alternating direction method of multipliers for separable nonconvex minimization problems*, Asia-Pacific Journal of Operational Research, 34 (2017). 1750030.

[114] M. Xu and T. Wu, *A class of linearized proximal alternating direction methods*, Journal of Optimization Theory and Applications, 151 (2011), pp. 321–337.

[115] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, *Bregman iterative algorithms for l_1-minimization with applications to compressed sensing*, SIAM Journal on Imaging sciences, 1 (2008), pp. 143–168.

[116] M. Yuan and Y. Lin, *Model selection and estimation in regression with grouped variables*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 68 (2006), pp. 49–67.

[117] X. Yuan, *The improvement with relative errors of he et al.'s inexact alternating direction method for monotone variational inequalities*, Mathematical and computer modelling, 42 (2005), pp. 1225–1236.

[118] N. Zhang, J. Wu, and L. Zhang, *A linearly convergent majorized ADMM with indefinite proximal terms for convex composite programming and its applications*, Mathematics of Computation, 89 (2020), pp. 1867–1894.