# A TWO-PHASE AUGMENTED LAGRANGIAN METHOD FOR CONVEX COMPOSITE QUADRATIC PROGRAMMING


**LI XUDONG**

*(B.Sc., USTC)*


**A THESIS SUBMITTED**

**FOR THE DEGREE OF DOCTOR OF PHILOSOPHY**


**DEPARTMENT OF MATHEMATICS**

**NATIONAL UNIVERSITY OF SINGAPORE**



**2015**

To my parents

# DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

---

Li, Xudong

21 January, 2015

# Acknowledgements

I would like to express my sincerest thanks to my supervisor Professor Sun Defeng. Without his amazing depth of mathematical knowledge and professional guidance, this work would not have been possible. It is his instruction on mathematical programming, which is the first optimization module I took during my first year in NUS, introduce me into the field of convex optimization, and thus, led me to where I am now. His integrity and enthusiasm for research has a huge impact on me. I owe him a great debt of gratitude.

My deepest gratitude also goes to Professor Toh Kim Chuan, my co-supervisor and my guide to numerical optimization and software. I have benefited a lot from many discussions we had during past three years. It is my great honor to have an opportunity of doing research with him.

My thanks also go to the previous and present members in the optimization group, in particular, Ding Chao, Miao Weimin, Jiang Kaifeng, Gong Zheng, Shi Dongjian, Wu Bin, Chen Caihua, Du Mengyu, Cui Ying, Yang Liuqing and Chen Liang. In particular, I would like to give my special thanks to Wu Bin, Du Mengyu, Cui Ying, Yang Liuqing, and Chen Liang for their enlightening suggestions and helpful discussions in many interesting optimization topics related to my research.

I would like to thank all my friends in Singapore at NUS, in particular, Cai

Ruilun, Gao Rui, Gao Bing, Wang Kang, Jiang Kaifeng, Gong Zheng, Du Mengyu, Ma Jiajun, Sun Xiang, Hou Likun, Li Shangru, for their friendship, the gatherings and chit-chats. I will cherish the memories of my time with them.

I am also grateful to the university and the department for providing me the four-year research scholarship to complete the degree, the financial support for conference trips, and the excellent research conditions.

Although they do not read English, I would like to dedicate this thesis to my parents for their unconditionally love and support. Last but not least, I am also greatly indebted to my fiancée, Chen Xi, for her understanding, encouragement and love.

# Contents

**Acknowledgements** **vii**

**Summary** **xi**

**1 Introduction** **1**

1.1 Motivations and related methods . . . . . . . . . . . . . . . . . . . . . 2

1.1.1 Convex quadratic semidefinite programming . . . . . . . . . . 2

1.1.2 Convex quadratic programming . . . . . . . . . . . . . . . . . . 8

1.2 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

1.3 Thesis organization . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

**2 Preliminaries** **15**

2.1 Notations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

2.2 The Moreau-Yosida regularization . . . . . . . . . . . . . . . . . . . . 17

2.3 Proximal ADMM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

2.3.1 Semi-proximal ADMM . . . . . . . . . . . . . . . . . . . . . . . 22

2.3.2 A majorized ADMM with indefinite proximal terms . . . . . . 27

**3 Phase I: A symmetric Gauss-Seidel based proximal ADMM for convex composite quadratic programming** **33**

3.1 One cycle symmetric block Gauss-Seidel technique . . . . . . . . . . . 34

3.1.1 The two block case . . . . . . . . . . . . . . . . . . . . . . . . . 35

3.1.2 The multi-block case . . . . . . . . . . . . . . . . . . . . . . . . 37

3.2 A symmetric Gauss-Seidel based semi-proximal ALM . . . . . . . . . 44

3.3 A symmetric Gauss-Seidel based proximal ADMM . . . . . . . . . . . 50

3.4 Numerical results and examples . . . . . . . . . . . . . . . . . . . . . 60

3.4.1 Convex quadratic semidefinite programming (QSDP) . . . . . 61

3.4.2 Nearest correlation matrix (NCM) approximations . . . . . . 75

3.4.3 Convex quadratic programming (QP) . . . . . . . . . . . . . . 79

**4 Phase II: An inexact proximal augmented Lagrangian method for convex composite quadratic programming** **89**

4.1 A proximal augmented Lagrangian method of multipliers . . . . . . . 90

4.1.1 An inexact alternating minimization method for inner subproblems . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96

4.2 The second stage of solving convex QSDP . . . . . . . . . . . . . . . . 100

4.2.1 The second stage of solving convex QP . . . . . . . . . . . . . 107

4.3 Numerical results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 111

**5 Conclusions** **121**

**Bibliography** **123**

# Summary

This thesis is concerned with an important class of high dimensional convex composite quadratic optimization problems with large numbers of linear equality and inequality constraints. The motivation for this work comes from recent interests in important convex quadratic conic programming problems, as well as from convex quadratic programming problems with dual block angular structures arising from network flows problems, two stage stochastic programming problems, etc. In order to solve the targeted problems to desired accuracy efficiently, we introduce a two phase augmented Lagrangian method, with Phase I to generate a reasonably good initial point and Phase II to obtain accurate solutions fast.

In Phase I, we carefully examine a class of convex composite quadratic programming problems and introduce a one cycle symmetric block Gauss-Seidel technique. This technique allows us to design a novel symmetric Gauss-Seidel based proximal ADMM (sGS-PADMM) for solving convex composite quadratic programming problems. The ability of dealing with coupling quadratic term in the objective function makes the proposed algorithm very flexible in solving various multi-block convex optimization problems. The high efficiency of our proposed algorithm for achieving low to medium accuracy solutions is demonstrated by numerical experiments on various large scale examples including convex quadratic semidefinite programming

(QSDP) problems, convex quadratic programming (QP) problems and some other extensions.

In Phase II, in order to obtain more accurate solutions for convex composite quadratic programming problems, we propose an inexact proximal augmented Lagrangian method (pALM). We study the global and local convergence of our proposed algorithm based on the classic results of proximal point algorithms. We propose to solve the inner subproblems by inexact alternating minimization method. Then, we specialize the proposed pALM algorithm to convex QSDP problems and convex QP problems. We discuss the implementation of a semismooth Newton-CG method and an inexact accelerated proximal gradient (APG) method for solving the resulted inner subproblems. We also show that how the aforementioned symmetric Gauss-Seidel technique can be intelligently incorporated in the implementation of our Phase II algorithm. Numerical experiments on a variety of high dimensional convex QSDP problems and convex QP problems show that our proposed two phase framework is very efficient and robust.

Chapter 1

# Introduction

In this thesis, we focus on designing algorithms for solving large scale convex composite quadratic programming problems. In particular, we are interested in convex quadratic semidefinite programming (QSDP) problems and convex quadratic programming (QP) problems with large numbers of linear equality and inequality constraints. The general convex composite quadratic optimization model we considered in this thesis is given as follows:

$$\begin{array}{ll} \min & \theta(y_1) + f(y_1, y_2, \ldots, y_p) + \varphi(z_1) + g(z_1, z_2, \ldots, z_q) \\ \text{s.t.} & \mathcal{A}_1^* y_1 + \mathcal{A}_2^* y_2 + \cdots + \mathcal{A}_p^* y_p + \mathcal{B}_1^* z_1 + \mathcal{B}_2^* z_2 + \cdots + \mathcal{B}_q^* z_q = c, \end{array} \tag{1.1}$$

where $p$ and $q$ are given nonnegative integers, $\theta : \mathcal{Y}_1 \to (-\infty, +\infty]$ and $\varphi : \mathcal{Z}_1 \to (-\infty, +\infty]$ are simple closed proper convex function in the sense that their proximal mappings are relatively easy to compute, $f : \mathcal{Y}_1 \times \mathcal{Y}_2 \times \ldots \times \mathcal{Y}_p \to \Re$ and $g : \mathcal{Z}_1 \times \mathcal{Z}_2 \times \ldots \times \mathcal{Z}_q \to \Re$ are convex quadratic, possibly nonseparable, functions, $\mathcal{A}_i : \mathcal{X} \to \mathcal{Y}_i$, $i = 1, \ldots, p$, and $\mathcal{B}_j : \mathcal{X} \to \mathcal{Z}_j$, $j = 1, \ldots, q$, are linear maps, $c \in \mathcal{X}$ is given data, $\mathcal{Y}_1, \ldots, \mathcal{Y}_p, \mathcal{Z}_1, \ldots, \mathcal{Z}_q$ and $\mathcal{X}$ are real finite dimensional Euclidean spaces each equipped with an inner product $\langle \cdot, \cdot \rangle$ and its induced norm $\| \cdot \|$. In this thesis, we aim to design efficient algorithms for finding a solution of medium to high accuracy to convex composite quadratic programming problems.

## 1.1 Motivations and related methods

The motivation for studying general convex composite quadratic programming model (1.1) comes from recent interests in the following convex composite quadratic conic programming problem:

$$\begin{array}{ll} \min & \theta(y_1) + \dfrac{1}{2}\langle y_1,\, \mathcal{Q}y_1\rangle + \langle c,\, y_1\rangle \\ \text{s.t.} & y_1 \in \mathcal{K}_1, \quad \mathcal{A}_1^* y_1 - b \in \mathcal{K}_2, \end{array} \tag{1.2}$$

where $\mathcal{Q} : \mathcal{Y}_1 \to \mathcal{Y}_1$ is a self-adjoint positive semidefinite linear operator, $c \in \mathcal{Y}_1$ and $b \in \mathcal{X}$ are given data, $\mathcal{K}_1 \subseteq \mathcal{Y}_1$ and $\mathcal{K}_2 \subseteq \mathcal{X}$ are closed convex cones. The Lagrangian dual of problem (1.2) is given by

$$\begin{array}{ll} \max & -\theta^*(-s) - \dfrac{1}{2}\langle w,\, \mathcal{Q}w\rangle + \langle b,\, x\rangle \\ \text{s.t.} & s + z - \mathcal{Q}w + \mathcal{A}_1 x = c, \\ & z \in \mathcal{K}_1^*, \quad w \in \mathcal{W}, \quad x \in \mathcal{K}_2^*, \end{array}$$

where $\mathcal{W} \subseteq \mathcal{Y}_1$ is any subspace such that $\text{Range}(\mathcal{Q}) \subseteq \mathcal{W}$, $\mathcal{K}_1^*$ and $\mathcal{K}_2^*$ are the dual cones of $\mathcal{K}_1$ and $\mathcal{K}_2$, respectively, i.e., $\mathcal{K}_1^* := \{d \in \mathcal{Y}_1 \mid \langle d,\, y_1\rangle \geq 0\ \forall y_1 \in \mathcal{K}_1\}$, $\theta^*(\cdot)$ is the Fenchel conjugate function [53] of $\theta(\cdot)$ defined by $\theta^*(s) = \sup_{y_1 \in \mathcal{Y}_1}\{\langle s,\, y_1\rangle - \theta(y_1)\}$.

Below we introduce several prominent special cases of the model (1.2) including convex quadratic semidefinite programming problems and convex quadratic programming problems.

### 1.1.1 Convex quadratic semidefinite programming

An important special case of convex composite quadratic conic programming is the following convex quadratic semidefinite programming (QSDP)

$$\begin{array}{ll} \min & \frac{1}{2}\langle X,\, \mathcal{Q}X\rangle + \langle C,\, X\rangle \\ \text{s.t.} & \mathcal{A}_E X = b_E, \quad \mathcal{A}_I X \geq b_I, \quad X \in \mathcal{S}_+^n \cap \mathcal{K}\,, \end{array} \tag{1.3}$$

where $\mathcal{S}^n_+$ is the cone of $n \times n$ symmetric and positive semidefinite matrices in the space of $n\times n$ symmetric matrices $\mathcal{S}^n$ endowed with the standard trace inner product $\langle \cdot, \cdot \rangle$ and the Frobenius norm $\| \cdot \|$, $\mathcal{Q}$ is a self-adjoint positive semidefinite linear operator from $\mathcal{S}^n$ to $\mathcal{S}^n$, $\mathcal{A}_E : \mathcal{S}^n \to \Re^{m_E}$ and $\mathcal{A}_I : \mathcal{S}^n \to \Re^{m_I}$ are two linear maps, $C \in \mathcal{S}^n$, $b_E \in \Re^{m_E}$ and $b_I \in \Re^{m_I}$ are given data, $\mathcal{K}$ is a nonempty simple closed convex set, e.g., $\mathcal{K} = \{W \in \mathcal{S}^n : \ L \leq W \leq U\}$ with $L, U \in \mathcal{S}^n$ being given matrices. The dual of problem (1.3) is given by

$$\begin{array}{rl} \max & -\delta^*_{\mathcal{K}}(-Z) - \frac{1}{2}\langle X', \, \mathcal{Q}X' \rangle + \langle b_E, \, y_E \rangle + \langle b_I, \, y_I \rangle \\ \text{s.t.} & Z - \mathcal{Q}X' + S + \mathcal{A}^*_E y_E + \mathcal{A}^*_I y_I = C, \\ & X' \in \mathcal{S}^n, \quad y_I \geq 0, \quad S \in \mathcal{S}^n_+ \, , \end{array} \tag{1.4}$$

where for any $Z \in \mathcal{S}^n$, $\delta^*_{\mathcal{K}}(-Z)$ is given by

$$\delta^*_{\mathcal{K}}(-Z) = - \inf_{W \in \mathcal{K}} \langle Z, \, W \rangle = \sup_{W \in \mathcal{K}} \langle -Z, \, W \rangle. \tag{1.5}$$

Note that, in general, problem (1.4) does not fit our general convex composite quadratic programming model (1.1) unless $y_I$ is vacuous from the model or $\mathcal{K} \equiv \mathcal{S}^n$. However, one can always reformulate problem (1.4) equivalently as

$$\begin{array}{rl} \min & (\delta^*_{\mathcal{K}}(-Z) + \delta_{\Re^{m_I}_+}(u)) + \frac{1}{2}\langle X', \, \mathcal{Q}X' \rangle + \delta_{\mathcal{S}^n_+}(S) - \langle b_E, \, y_E \rangle - \langle b_I, \, y_I \rangle \\ \text{s.t.} & Z - \mathcal{Q}X' + S + \mathcal{A}^*_E y_E + \mathcal{A}^*_I y_I = C, \\ & u - y_I = 0, \quad X' \in \mathcal{S}^n, \end{array} \tag{1.6}$$

where $\delta_{\Re^{m_I}_+}(\cdot)$ is the indicator function over $\Re^{m_I}_+$, i.e., $\delta_{\Re^{m_I}_+}(u) = 0$ if $u \in \Re^{m_I}_+$ and $\delta_{\Re^{m_I}_+}(u) = \infty$ if $u \notin \Re^{m_I}_+$. Now, one can see that problem (1.6) satisfies our general optimization model (1.1). Actually, the introduction of the variable $u$ in (1.6) not only fits our model but also makes the computations more efficient. Specifically, in applications, the largest eigenvalue of $\mathcal{A}_I \mathcal{A}^*_I$ is normally very large. Thus, to make the variable $y_I$ in (1.6) to be of free sign is critical for efficient numerical computations.

Due to its wide applications and mathematical elegance [1, 26, 31, 50], QSDP has been extensively studied both theoretically and numerically in the literature. For the recent theoretical developments, one may refer to [49, 61, 2] and references therein. From the numerical aspect, below we briefly review some of the methods available for solving QSDP problems. In (1.6), if there are no inequality constraints (i.e., $\mathcal{A}_I$ and $b_I$ are vacuous and $\mathcal{K} = \mathcal{S}^n$), Toh et al [63] and Toh [65] proposed inexact primal-dual path-following methods, which belong to the category of interior point methods, to solve this special class of convex QSDP problems. In theory, these methods can be used to solve QSDP with any numbers of inequality constraints. However, in practice, as far as we know, the interior point based methods can only solve moderate scale QSDP problems. In her PhD thesis, Zhao [72] designed a semismooth Newton-CG augmented Lagrangian (NAL) method and analyzed its convergence for solving the primal formulation of QSDP problems (1.3). However, NAL algorithm may encounter numerical difficulty when the nonnegative constraints are present. Later, Jiang et al [29] proposed an inexact accelerated proximal gradient method mainly for least squares semidefinite programming without inequality constraints. Note that it is also designed to solve the primal formulation of QSDP. To the best of our knowledge, there are no existing methods which can efficiently solve the general QSDP model (1.3).

There are many convex optimization problems related to convex quadratic conic programming which fall within our general convex composite quadratic programming model. One example comes from the matrix completion with fixed basis coefficients [42, 41, 68]. Indeed the nuclear semi-norm penalized least squares model in [41] can be written as

$$\begin{array}{ll} \min\limits_{X \in \Re^{m\times n}} & \frac{1}{2}\|\mathcal{A}_F X - d\|^2 + \rho(\|X\|_* - \langle C, X\rangle) \\ \text{s.t.} & \mathcal{A}_E X = b_E, \quad X \in \mathcal{K} := \{X \mid \|\mathcal{R}_\Omega X\|_\infty \le \alpha\}, \end{array} \tag{1.7}$$

where $\|X\|_*$ is the nuclear norm of $X$ defined as the sum of all its singular values, $\|\cdot\|_\infty$ is the element-wise $l_\infty$ norm defined by $\|X\|_\infty := \max\limits_{i=1,\ldots,m} \max\limits_{j=1,\ldots,n} |X_{ij}|$, $\mathcal{A}_F : \Re^{m\times n} \to \Re^{n_F}$ and $\mathcal{A}_E : \Re^{m\times n} \to \Re^{n_E}$ are two linear maps, $\rho$ and $\alpha$ are two given positive parameters, $d \in \Re^{n_F}$, $C \in \Re^{m\times n}$ and $b_E \in \Re^{n_E}$ are given data, $\Omega \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$ is the set of the indices relative to which the basis coefficients are not fixed, $\mathcal{R}_\Omega : \Re^{m\times n} \to \Re^{|\Omega|}$ is the linear map such that $\mathcal{R}_\Omega X := (X_{ij})_{ij\in\Omega}$. Note that when there are no fixed basis coefficients (i.e., $\Omega = \{1,\ldots,m\}\times\{1,\ldots,n\}$ and $\mathcal{A}_E$ are vacuous), the above problem reduces to the model considered by Negahban and Wainwright in [45] and Klopp in [30]. By introducing slack variables $\eta$, $R$ and $W$, we can reformulate problem (1.7) as

$$\begin{aligned} \min \quad & \tfrac{1}{2}\|\eta\|^2 + \rho\big(\|R\|_* - \langle C,\, X\rangle\big) + \delta_{\mathcal{K}}(W) \\ \text{s.t.} \quad & \mathcal{A}_F X - d = \eta, \quad \mathcal{A}_E X = b_E, \quad X = R, \quad X = W. \end{aligned} \tag{1.8}$$

The dual of problem (1.8) takes the form of

$$\begin{aligned} \max \quad & -\delta^*_{\mathcal{K}}(-Z) - \tfrac{1}{2}\|\xi\|^2 + \langle d,\, \xi\rangle + \langle b_E,\, y_E\rangle \\ \text{s.t.} \quad & Z + \mathcal{A}_F^*\xi + S + \mathcal{A}_E^* y_E = -\rho C, \quad \|S\|_2 \le \rho, \end{aligned} \tag{1.9}$$

where $\|S\|_2$ is the operator norm of $S$, which is defined to be its largest singular value.

Another compelling example is the so called robust PCA (principle component analysis) considered in [66]:

$$\begin{aligned} \min \quad & \|A\|_* + \lambda_1\|E\|_1 + \frac{\lambda_2}{2}\|Z\|_F^2 \\ \text{s.t.} \quad & A + E + Z = W, \quad A, E, Z \in \Re^{m\times n}, \end{aligned} \tag{1.10}$$

where $W \in \Re^{m\times n}$ is the observed data matrix, $\|\cdot\|_1$ is the elementwise $l_1$ norm given by $\|E\|_1 := \sum_{i=1}^m \sum_{j=1}^n |E_{ij}|$, $\|\cdot\|_F$ is the Frobenius norm, $\lambda_1$ and $\lambda_2$ are two positive parameters. There are many different variants to the robust PCA model. For example, one may consider the following model where the observed data matrix $W$ is incomplete:

$$\begin{aligned} \min \quad & \|A\|_* + \lambda_1\|E\|_1 + \frac{\lambda_2}{2}\|\mathcal{P}_\Omega(Z)\|_F^2 \\ \text{s.t.} \quad & \mathcal{P}_\Omega(A + E + Z) = \mathcal{P}_\Omega(W), \quad A, E, Z \in \Re^{m\times n}, \end{aligned} \tag{1.11}$$

i.e. one assumes that only a subset $\Omega \subseteq \{1,\ldots,m\}\times\{1,\ldots,n\}$ of the entries of $W$ can be observed. Here $\mathcal{P}_\Omega : \Re^{m\times n} \to \Re^{m\times n}$ is the orthogonal projection operator defined by

$$\mathcal{P}_\Omega(X) = \begin{cases} X_{ij} & \text{if } (i,j) \in \Omega, \\ 0 & \text{otherwise.} \end{cases} \tag{1.12}$$

In [62], Tao and Yuan tested one of the equivalent forms of problem (1.11). In the numerical section, we will see other interesting examples.

Due to the fact that the objective functions in all above examples are separable, these examples can also be viewed as special cases of the following block-separable convex optimization problem:

$$\min\left\{\sum_{i=1}^{n}\phi_i(w_i) \mid \sum_{i=1}^{n}\mathcal{H}_i^* w_i = c\right\}, \tag{1.13}$$

where for each $i \in \{1,\ldots,n\}$, $\mathcal{W}_i$ is a finite dimensional real Euclidean space equipped with an inner product $\langle\cdot,\,\cdot\rangle$ and its induced norm $\|\cdot\|$, $\phi_i : \mathcal{W}_i \to (-\infty,+\infty]$ is a closed proper convex function, $\mathcal{H}_i : \mathcal{X} \to \mathcal{W}_i$ is a linear map and $c \in \mathcal{X}$ is given. Note that the quadratic structure in all the mentioned examples is hidden in the sense that each $\phi_i$ will be treated equally. However, this special quadratic structure will be thoroughly exploited in our search for an efficient yet simple algorithm with guaranteed convergence.

Let $\sigma > 0$ be a given parameter. The augmented Lagrangian function for (1.13) is defined by

$$\mathcal{L}_\sigma(w_1,\ldots,w_n;x) := \textstyle\sum_{i=1}^{n}\phi_i(w_i) + \langle x,\, \sum_{i=1}^{n}\mathcal{H}_i^* w_i - c\rangle + \frac{\sigma}{2}\|\sum_{i=1}^{n}\mathcal{H}_i^* w_i - c\|^2$$

for $w_i \in \mathcal{W}_i$, $i = 1,\ldots,n$ and $x \in \mathcal{X}$. Choose any initial points $w_i^0 \in \mathrm{dom}(\phi_i)$, $i = 1,\ldots,q$ and $x^0 \in \mathcal{X}$. The classical augmented Lagrangian method consists of the following iterations:

$$(w_1^{k+1},\ldots,w_n^{k+1}) = \operatorname{argmin} \mathcal{L}_\sigma(w_1,\ldots,w_n;x^k), \tag{1.14}$$

$$x^{k+1} = x^k + \tau\sigma\left(\sum_{i=1}^{n}\mathcal{H}_i^* w_i^{k+1} - c\right), \tag{1.15}$$

where $\tau \in (0,2)$ guarantees the convergence. Due to the non-separability of the quadratic penalty term in $\mathcal{L}_\sigma$, it is generally a challenging task to solve the joint minimization problem (1.14) exactly or approximately with high accuracy. To overcome this difficulty, one may consider the following $n$-block alternating direction methods of multipliers (ADMM):

$$
\begin{aligned}
w_1^{k+1} &= \operatorname{argmin} \mathcal{L}_\sigma(w_1, w_2^k \ldots, w_n^k; x^k), \\
&\ \vdots \\
w_i^{k+1} &= \operatorname{argmin} \mathcal{L}_\sigma(w_1^{k+1}, \ldots, w_{i-1}^{k+1}, w_i, w_{i+1}^k, \ldots, w_n^k; x^k), \\
&\ \vdots \\
w_n^{k+1} &= \operatorname{argmin} \mathcal{L}_\sigma(w_1^{k+1}, \ldots, w_{n-1}^{k+1}, w_n; x^k), \\
x^{k+1} &= x^k + \tau\sigma \left(\sum\nolimits_{i=1}^n \mathcal{H}_i^* w_i^{k+1} - c\right).
\end{aligned}
\tag{1.16}
$$

Note that although the above $n$-block ADMM can not be directly applied to solve general convex composite quadratic programming problem (1.1) due to the nonseparable structure of the objective functions, we still briefly discuss recent developments of this algorithm here as it is close related to our proposed new algorithm. In fact, the above $n$-block ADMM is an direct extension of the ADMM for solving the following 2-block convex optimization problem

$$\min \left\{\phi_1(w_1) + \phi_2(w_2) \mid \mathcal{H}_1^* w_1 + \mathcal{H}_2^* w_2 = c\right\}. \tag{1.17}$$

The convergence of 2-block ADMM has already been extensively studied in [18, 16, 17, 14, 15, 11] and references therein. However, the convergence of the $n$-block ADMM has been ambiguous for a long time. Fortunately this ambiguity has been addressed very recently in [4] where Chen, He, Ye, and Yuan showed that the direct extension of the ADMM to the case of a 3-block convex optimization problem is not necessarily convergent. This seems to suggest that one has to give up the direct extension of $m$-block ($m \geq 3$) ADMM unless if one is willing to take a sufficiently small step-length $\tau$ as was shown by Hong and Luo in [28] or to take a small penalty parameter $\sigma$ if at least $m-2$ blocks in the objective are strongly convex [23, 5, 36, 37, 34]. On the other hand, the $n$-block ADMM with $\tau \geq 1$ often works very well in practice and this fact poses a big challenge if one attempts to develop new ADMM-type algorithms which have convergence guarantee but with competitive numerical efficiency and iteration simplicity as the $n$-block ADMM.

Recently, there is exciting progress in this active research area. Sun, Toh and Yang [59] proposed a convergent semi-proximal ADMM (ADMM+) for convex programming problems of three separable blocks in the objective function with the third part being linear. The convergence proof of ADMM+ presented in [59] is via establishing its equivalence to a particular case of the general 2-block semi-proximal ADMM considered in [13]. Later, Li, Sun and Toh [35] extended the 2-block semi-proximal ADMM in [13] to a majorized ADMM with indefinite proximal terms. In this thesis, inspired by the aforementioned work, we aim to extend the idea in ADMM+ to solve convex composite quadratic programming problems based on the convergence results provided in [35].

### 1.1.2 Convex quadratic programming

As a special class of convex composite quadratic conic programming, the following high dimensional convex quadratic programming (QP) problem is also a strong motivation for us to study the general convex composite quadratic programming problem. The large scale convex quadratic programming with many equality and inequality constraints is given as follows:

$$\min\left\{\frac{1}{2}\langle x,\, Qx\rangle + \langle c,\, x\rangle \mid Ax = b,\ \bar{b} - Bx \in \mathcal{C},\ x \in \mathcal{K}\right\}, \tag{1.18}$$

where vector $c \in \Re^n$ and positive semidefinite matrix $Q \in \mathcal{S}^n_+$ define the linear and quadratic costs for decision variable $x \in \Re^n$, matrices $A \in \Re^{m_E \times n}$ and $B \in \Re^{m_I \times n}$ respectively define the equality and inequality constraints, $\mathcal{C} \subseteq \Re^{m_I}$ is a closed convex cone, e.g., the nonnegative orthant $\mathcal{C} = \{\bar{x} \in \Re^{m_I} \mid \bar{x} \geq 0\}$, $\mathcal{K} \subseteq \Re^n$ is a nonempty simple closed convex set, e.g., $\mathcal{K} = \{x \in \Re^n \mid l \leq x \leq u\}$ with $l, u \in \Re^n$

being given vectors. The dual of (1.18) takes the following form

$$
\begin{array}{ll}
\max & -\delta^*_{\mathcal{K}}(-z) - \frac{1}{2}\langle x',\, Qx'\rangle + \langle b,\, y\rangle + \langle \bar{b},\, \bar{y}\rangle \\
\text{s.t.} & z - Qx' + A^*y + B^*\bar{y} = c, \quad x' \in \Re^n, \quad \bar{y} \in \mathcal{C}^\circ,
\end{array} \tag{1.19}
$$

where $\mathcal{C}^\circ$ is the polar cone [53, Section 14] of $\mathcal{C}$. We are more interested in the case when the dimensions $n$ and/or $m_E + m_I$ are extremely large. Convex QP has been extensively studied for over the last fifty years, see, for examples [60, 19, 20, 21, 8, 7, 9, 10, 70, 67] and references therein. Nowadays, main solvers for convex QP are based on active set methods or interior point methods. One may also refer to `http://www.numerical.rl.ac.uk/people/nimg/qp/qp.html` for more information. Currently, one popular state-of-the-art solver for large scale convex QP problems is the interior point methods based solver Gurobi[22]*. However, for high dimensional convex QP problems with a large number of constraints, the interior point methods based solvers, such as Gurobi, will encounter inherent numerical difficulties as the lack of sparsity of the linear systems to be solved often makes the critical sparse Cholesky factorization fail. This fact indicates that an algorithm which can handle high dimensional convex QP problems with many dense linear constraints is needed.

In order to handle the equality and inequality constraints simultaneously, we propose to add a slack variable $\bar{x}$ to get the following problem:

$$
\begin{array}{ll}
\min & \frac{1}{2}\langle x,\, Qx\rangle + \langle c,\, x\rangle \\
\text{s.t.} & \begin{bmatrix} A & \\ B & I \end{bmatrix} \begin{bmatrix} x \\ \bar{x} \end{bmatrix} = \begin{bmatrix} b \\ \bar{b} \end{bmatrix}, \quad x \in \mathcal{K}, \quad \bar{x} \in \mathcal{C}.
\end{array} \tag{1.20}
$$

The dual of problem (1.20) is given by

$$
\begin{array}{ll}
\max & (-\delta^*_{\mathcal{K}}(-z) - \delta^*_{\mathcal{C}}(-\bar{z})) - \frac{1}{2}\langle x',\, Qx'\rangle + \langle b,\, y\rangle + \langle \bar{b},\, \bar{y}\rangle \\
\text{s.t.} & \begin{bmatrix} z \\ \bar{z} \end{bmatrix} - \begin{bmatrix} Qx' \\ 0 \end{bmatrix} + \begin{bmatrix} A^* & B^* \\ & I \end{bmatrix} \begin{bmatrix} y \\ \bar{y} \end{bmatrix} = \begin{bmatrix} c \\ 0 \end{bmatrix}.
\end{array} \tag{1.21}
$$

*Base on the results presented in `http://plato.asu.edu/ftp/barrier.html`

Thus, problem (1.21) belongs to our general optimization model (1.1). Note that, due to the extremely large problem size, ideally, one should decompose $x'$ into smaller pieces but then the quadratic term about $x'$ in the objective function becomes nonseparable. Thus, one will encounter difficulties while using classic ADMM to solve (1.21) since classic ADMM can not handle nonseparable structures in the objective function. This again calls for new developments of efficient and convergent ADMM type methods.

A prominent example of convex QP comes from the two-stage stochastic optimization problem. Consider the following stochastic optimization problem:

$$\min_x \Big\{\frac{1}{2}\langle x,\, Qx\rangle + \langle c,\, x\rangle + E_\xi\, P(x;\xi) \mid Ax = b,\ x \in \mathcal{K}\Big\}, \tag{1.22}$$

where $\xi$ is a random vector and

$$P(x;\xi) \;=\; \min\left\{\frac{1}{2}\langle \bar{x},\, Q_\xi \bar{x}\rangle + \langle q_\xi,\, \bar{x}\rangle \mid \overline{B}_\xi \bar{x} = \bar{b}_\xi - B_\xi x,\ \bar{x} \in \overline{\mathcal{K}}_\xi\right\},$$

where $\overline{\mathcal{K}}_\xi \in \mathcal{X}$ is a simple closed convex set depending on the random vector $\xi$. By sampling $N$ scenarios for $\xi$, one may approximately solve (1.22) via the following deterministic optimization problem:

$$\begin{array}{ll} \min & \frac{1}{2}\langle x,\, Qx\rangle + \langle c,\, x\rangle + \sum_{i=1}^N \big(\frac{1}{2}\langle \bar{x}_i,\, \overline{Q}_i \bar{x}_i\rangle + \langle \bar{c}_i,\, \bar{x}_i\rangle\big) \\ \text{s.t.} & \begin{bmatrix} A & & & \\ B_1 & \overline{B}_1 & & \\ \vdots & & \ddots & \\ B_N & & & \overline{B}_N \end{bmatrix} \begin{bmatrix} x \\ \bar{x}_1 \\ \vdots \\ \bar{x}_N \end{bmatrix} = \begin{bmatrix} b \\ \bar{b}_1 \\ \vdots \\ \bar{b}_N \end{bmatrix}, \\ & x \in \mathcal{K}, \quad \bar{x} = [\bar{x}_1; \dots; \bar{x}_N] \in \overline{\mathcal{K}} = \overline{\mathcal{K}}_1 \times \cdots \times \overline{\mathcal{K}}_N, \end{array} \tag{1.23}$$

where $\overline{Q}_i = p_i Q_i$ and $\bar{c}_i = p_i q_i$ with $p_i$ being the probability of occurrence of the $i$th scenario, $B_i, \overline{B}_i, \bar{b}_i$ are the data and $\bar{x}_i$ is the second stage decision variable associated with the $i$th scenario. The dual problem of (1.23) is given by

$$\begin{array}{ll}\min & (\sum_{j=1}^{N}\delta^*_{\overline{\mathcal{K}}_j}(-\bar{z}_j)+\delta^*_{\mathcal{K}}(-z))+\frac{1}{2}\langle x',\,Qx'\rangle+\sum_{i=1}^{N}\frac{1}{2}\langle\bar{x}'_i,\,\overline{Q}_i\bar{x}'_i\rangle-\langle b,\,y\rangle-\sum_{j=1}^{N}\langle\bar{b}_j,\,\bar{y}_j\rangle\\ \text{s.t.} & \begin{bmatrix} z\\ \bar{z}_1\\ \vdots\\ \bar{z}_N\end{bmatrix}-\begin{bmatrix} Q & & & \\ & \overline{Q}_1 & & \\ & & \ddots & \\ & & & \overline{Q}_N\end{bmatrix}\begin{bmatrix} x'\\ \bar{x}'_1\\ \vdots\\ \bar{x}'_N\end{bmatrix}+\begin{bmatrix} A^* & B_1^* & \cdots & B_N^*\\ & \overline{B}_1^* & & \\ & & \ddots & \\ & & & \overline{B}_N^*\end{bmatrix}\begin{bmatrix} y\\ \bar{y}_1\\ \vdots\\ \bar{y}_N\end{bmatrix}=\begin{bmatrix} c\\ \bar{c}_1\\ \vdots\\ \bar{c}_N\end{bmatrix}.\end{array}\tag{1.24}$$

Clearly, (1.24) is another perfect example of our general convex composite quadratic programming problems.

## 1.2 Contributions

In order to solve the convex composite quadratic programming problems (1.1) to high accuracy efficiently, we introduce a two-phase augmented Lagrangian method, with Phase I to generate a reasonably good initial point and Phase II to obtain accurate solutions fast. In fact, this two stage framework has been successfully applied to solve semidefinite programming (SDP) problems with partial or full nonnegative constraints where ADMM+ [59] and SDPNAL+ [69] are regraded as Phase I algorithm and Phase II algorithm, respectively. Inspired by the aforementioned work, we propose to extend their ideas to solve large scale convex composite quadratic programming problems including convex QSDP and convex QP.

In Phase I, to solve convex quadratic conic programming, the first question we need to ask is that shall we work on the primal formulation (1.2) or the dual formulation (1.3)? Note that since the objective function in the dual problem contains quadratic functions as the primal problem does and has more blocks, it is natural for people to focus more on primal formulation. Actually, the primal approach has been used to solve special class of QSDP as in [29, 72]. However, as demonstrated in [59, 69], it is usually better to work on the dual formulation than the primal formulation for linear SDP problems with nonegative constraints (SDP+). [59, 69] pose the following question: for general convex quadratic conic programming (1.2), can we work on the dual formulation instead of primal formulation, as for the linear SDP+ problems? So that when the quadratic term in the objective function of QSDP reduced to a linear term, our algorithm is at least comparable with the algorithms proposed [59, 69]. In this thesis, we will resolve this issue in a unified way elegantly. Observe that ADMM+ can only deal with convex programming problems of three separable blocks in the objective function with the third part being linear. Thus, we need to invent new techniques to handle the quadratic terms and the multi-block structure in (1.4). Fortunately, by carefully examining a class of convex composite quadratic programming problems, we are able to design a novel one cycle symmetric block Gauss-Seidel technique to deal with the nonseparable structure in the objective function. Based on this technique, we then propose a symmetric Gauss-Seidel based proximal ADMM (sGS-PADMM) for solving not only the dual formulation of convex quadratic conic programming, which includes the dual formulation of QSDP as a special case, but also the general convex composite quadratic optimization model (1.1). Specifically, when sGS-PADMM is applied to solve high dimensional convex QP problems, the obstacles brought about by the large scale quadratic term, linear equality and inequality constraints can thus be overcome via using sGS-PADMM to decompose these terms into smaller pieces. Extensive numerical experiments on high dimensional QSDP problems, convex QP problems and some extensions demonstrate the efficiency of sGS-PADMM for finding a solution of low to medium accuracy.

In Phase I, the success of sGS-PADMM of being able to decompose the nonseparable structure in the dual formulation of convex quadratic conic programming (1.3) depends on the assumptions that the subspace $\mathcal{W}$ in (1.3) is chosen to be the whole space. This in fact can introduce unfavorable property of the unboundedness of the dual solution $w$ to problem (1.3). Fortunately, it causes no problem in Phase I. However, this unboundedness becomes critical in designing our second phase algorithm. Therefore, in Phase II, we will take $\mathcal{W} = \text{Range}(\mathcal{Q})$ to eliminate the unboundedness of the dual optimal solution $w$. This of course will introduce numerical difficulties as we need to maintain $w \in \text{Range}(\mathcal{Q})$, which, in general, is a difficult task. However, by fully exploring the structure of problem (1.3), we are able to resolve this issue. In this way, we can design an inexact proximal augmented Lagrangian (pALM) method for solving convex composite quadratic programming. The global convergence is analyzed based on the classic results of proximal point algorithms. Under the error bound assumption, we are also able to establish the local linear convergence of our proposed algorithm pALM. Then, we specialize the proposed pALM algorithm to convex QSDP problems and convex QP problems. We discuss in detail the implementation of a semismooth Newton-CG method and an inexact accelerated proximal gradient (APG) method for solving the resulted inner subproblems. We also show that how the aforementioned symmetric Gauss-Seidel technique can be intelligently incorporated in the implementation of our Phase II algorithm. The efficiency and robustness of our proposed two phase framework is then demonstrated by numerical experiments on a variety of high dimensional convex QSDP and convex QP problems.

## 1.3 Thesis organization

The rest of the thesis is organized as follows. In Chapter 2, we present some preliminaries that are relate to the subsequent discussions. We analyze the property of the Moreau-Yosida regularization and review the recent developments of proximal ADMM. In Chapter 3, we introduce the one cycle symmetric block Gauss-Seidel technique. Based on this technique, we are able to present our first phase algorithm, i.e., a symmetric Gauss-Seidel based proximal ADMM (sGS-PADMM), for solving convex composite quadratic programming problems. The efficiency of our proposed algorithm for finding a solution of low to medium accuracy to the tested problems is demonstrated by numerical experiments on various examples including convex QSDP and convex QP. In Chapter 4, for Phase II, we propose an inexact proximal augmented Lagrangian method for solving our convex composite quadratic optimization model and analyze its global and local convergence. The inner subproblems are solved by an inexact alternating minimization method. We also discuss in detail the implementations of our proposed algorithm for convex QSDP and convex QP problems. We also show that how the aforementioned symmetric Gauss-Seidel technique can be wisely incorporated in the proposed algorithms for solving the resulted inner subproblems. Numerical experiments conducted on a variety of large scale convex QSDP and convex QP problems show that our two phase framework is very efficient and robust for finding high accuracy solutions for convex composite quadratic programming problems. We give the final conclusions of the thesis and discuss a few future research directions in Chapter 5.

Chapter 2

# Preliminaries

## 2.1 Notations

Let $\mathcal{X}$ and $\mathcal{Y}$ be finite dimensional real Euclidian spaces each equipped with an inner product $\langle \cdot, \cdot \rangle$ and its induced norm $\| \cdot \|$. Let $\mathcal{M} : \mathcal{X} \to \mathcal{X}$ be a self-adjoint positive semidefinite linear operator. Then, there exists a unique positive semidefinite linear operator $\mathcal{N}$ with $\mathcal{N}^2 = \mathcal{M}$. Thus, we define $\mathcal{M}^{\frac{1}{2}} = \sqrt{\mathcal{M}} = \mathcal{N}$. Define $\langle \cdot, \cdot \rangle_{\mathcal{M}} : \mathcal{X} \times \mathcal{X} \to \Re$ by $\langle x, y \rangle_{\mathcal{M}} = \langle x, \mathcal{M}y \rangle$ for all $x, y \in \mathcal{X}$. Let $\| \cdot \|_{\mathcal{M}} : \mathcal{X} \to \Re$ be defined as $\|x\|_{\mathcal{M}} = \sqrt{\langle x, x \rangle_{\mathcal{M}}}$ for all $x \in \mathcal{X}$. If, $\mathcal{M}$ is further assumed to be positive definite, $\langle \cdot, \cdot \rangle_{\mathcal{M}}$ will be an inner product and $\| \cdot \|_{\mathcal{M}}$ will be its induced norm. Let $\mathcal{S}^n_+$ be the cone of $n \times n$ symmetric and positive semidefinite matrices in the space of $n \times n$ symmetric matrices $\mathcal{S}^n$ endowed with the standard trace inner product $\langle \cdot, \cdot \rangle$ and the Frobenius norm $\| \cdot \|$. Let $\mathbf{svec} : \mathcal{S}^n \to \Re^{n(n+1)/2}$ be the vectorization operator on symmetric matrices defined by $\mathbf{svec}(X) := [X_{11}, \sqrt{2}X_{12}, X_{22}, \ldots, \sqrt{2}X_{1n}, \ldots, \sqrt{2}X_{n-1,n}, X_{nn}]^T$.

**Definition 2.1.** A function $F : \mathcal{X} \to \mathcal{Y}$ is said to be directionally differentiable at $x \in \mathcal{X}$ if

$$F'(x; h) := \lim_{t \to 0^+} \frac{F(x + th) - F(x)}{t} \quad \text{exists}$$

for all $h \in \mathcal{X}$ and $F$ is directionally differentiable if $F$ is directionally differentiable at every $x \in \mathcal{X}$.

Let $F : \mathcal{X} \to \mathcal{Y}$ be a Lipschitz continuous function. By Rademacher's theorem [56, Section 9.J], $F$ is Fréchet differentiable almost everywhere. Let $D_F$ be the set of points in $\mathcal{X}$ where $F$ is differentiable. The Bouligand subdifferential of $F$ at $x \in \mathcal{X}$ is defined by

$$\partial_B F(x) = \left\{ \lim_{x^k \to x} F'(x^k), x^k \in D_F \right\},$$

where $F'(x^k)$ denotes the Jacobian of $F$ at $x^k \in D_F$ and the Clarke's [6] generalized Jacobian of $F$ at $x \in \mathcal{X}$ is defined as the convex hull of $\partial_B F(x)$ as follows

$$\partial F(x) = \mathrm{conv}\{\partial_B F(x)\}.$$

First introduced by Miffin [43] for functionals, the following concept of semismoothness was then extended by Qi and Sun [51] to cases when a vector-valued function is not differentiable, but locally Lipschitz continuous. See also [12, 40]

**Definition 2.2.** Let $F : \mathcal{O} \subseteq \mathcal{X} \to \mathcal{Y}$ be a locally Lipschitz continuous function on the open set $\mathcal{O}$. $F$ is said to be semismooth at a point $x \in \mathcal{O}$ if

1. $F$ is directionally differentiable at $x$; and

2. for any $\Delta x \in \mathcal{X}$ and $V \in \partial F(x + \Delta x)$ with $\Delta x \to 0$,

$$F(x + \Delta x) - F(x) - V\Delta x = o(\|\Delta x\|).$$

Furthermore, $F$ is said to be strongly semismooth at $x \in \mathcal{X}$ if $F$ is semismooth at $x$ and for any $\Delta x \in \mathcal{X}$ and $V \in \partial F(x + \Delta x)$ with $\Delta x \to 0$,

$$F(x + \Delta x) - F(x) - V\Delta x = O(\|\Delta x\|^2).$$

In fact, many functions such as convex functions and smooth functions are semismooth everywhere. Moreover, piecewise linear functions and twice continuously differentiable functions are strongly semismooth functions.

## 2.2 The Moreau-Yosida regularization

In this section, we discuss the Moreau-Yosida regularization which is a useful tool in our subsequent analysis.

**Definition 2.3.** Let $f : \mathcal{X} \to (-\infty, \infty]$ be a closed proper convex function. Let $\mathcal{M} : \mathcal{X} \to \mathcal{X}$ be a self-adjoint positive definite linear operator. The Moreau-Yosida regularization $\varphi^f_{\mathcal{M}} : \mathcal{X} \to \Re$ of $f$ with respect to $\mathcal{M}$ is defined as

$$\varphi^f_{\mathcal{M}}(x) = \min_{z \in \mathcal{X}} \left\{ f(z) + \frac{1}{2}\|z - x\|^2_{\mathcal{M}} \right\}, \quad x \in \mathcal{X}. \tag{2.1}$$

From [44, 71], we have the following proposition.

**Proposition 2.1.** *For any given $x \in \mathcal{X}$, the problem* (2.1) *has a unique optimal solution.*

**Definition 2.4.** The unique optimal solution of problem (2.1), denoted by $\text{prox}^f_{\mathcal{M}}(x)$, is called the proximal point of $x$ associated with $f$. When $\mathcal{M} = \mathcal{I}$, for simplicity, we write $\text{prox}_f(x) \equiv \text{prox}^f_{\mathcal{I}}(x)$ for all $x \in \mathcal{X}$, where $\mathcal{I} : \mathcal{X} \to \mathcal{X}$ is the identity operator.

Below, we list some important properties of the Moreau-Yosida regularization.

**Proposition 2.2.** *Let $g : \mathcal{X} \to (-\infty, +\infty]$ be defined as $g(x) \equiv f(\mathcal{M}^{-\frac{1}{2}}x)$ $\forall x \in \mathcal{X}$. Then,*

$$\text{prox}^f_{\mathcal{M}}(x) = \mathcal{M}^{-\frac{1}{2}}\text{prox}_g(\mathcal{M}^{\frac{1}{2}}x) \quad \forall x \in \mathcal{X}.$$

**Proof.** Note that, for any given $x \in \mathcal{X}$,

$$\begin{aligned} \text{prox}^f_{\mathcal{M}}(x) &= \text{argmin}\{f(z) + \frac{1}{2}\|z - x\|^2_{\mathcal{M}}\} \\ &= \text{argmin}\{f(z) + \frac{1}{2}\|\mathcal{M}^{\frac{1}{2}}z - \mathcal{M}^{\frac{1}{2}}x\|^2\}. \end{aligned}$$

By change of variables, we have $\text{prox}^f_{\mathcal{M}}(x) = \mathcal{M}^{-\frac{1}{2}}y$, where

$$\begin{aligned} y &= \text{argmin}\{f(\mathcal{M}^{-\frac{1}{2}}y) + \frac{1}{2}\|y - \mathcal{M}^{\frac{1}{2}}x\|^2\} = \text{argmin}\{g(y) + \frac{1}{2}\|y - \mathcal{M}^{\frac{1}{2}}x\|^2\} \\ &= \text{prox}_g(\mathcal{M}^{\frac{1}{2}}x). \end{aligned}$$

That is $\text{prox}^f_{\mathcal{M}}(x) = \mathcal{M}^{-\frac{1}{2}}\text{prox}^g_{\mathcal{I}}(\mathcal{M}^{\frac{1}{2}}x)$ for all $x \in \mathcal{X}$. □

**Proposition 2.3.** *[32, Theorem XV.4.1.4 and Theorem XV.4.1.7] Let* $f : \mathcal{X} \to (-\infty, +\infty]$ *be a closed proper convex function. Let* $\mathcal{M} : \mathcal{X} \to \mathcal{X}$ *be a given self-adjoint positive definite linear operator,* $\varphi^f_{\mathcal{M}}(x)$ *be the Moreau-Yosida regularization of* $f$*, and* $\mathrm{prox}^f_{\mathcal{M}} : \mathcal{X} \to \mathcal{X}$ *be the associated proximal mapping. Then the following properties hold.*

(i) $\mathrm{argmin}_{x\in\mathcal{X}} f(x) = \mathrm{argmin}_{x\in\mathcal{X}} \varphi^f_{\mathcal{M}}(x)$.

(ii) *Both* $\mathrm{prox}^f_{\mathcal{M}}$ *and* $Q^f_{\mathcal{M}} := I - \mathrm{prox}^f_{\mathcal{M}}$ *(*$I : \mathcal{X} \to \mathcal{X}$ *is the identity map) are firmly non-expensive, i.e., for any* $x, y \in \mathcal{X}$*,*

$$\|\mathrm{prox}^f_{\mathcal{M}}(x) - \mathrm{prox}^f_{\mathcal{M}}(y)\|^2_{\mathcal{M}} \leq \langle \mathrm{prox}^f_{\mathcal{M}}(x) - \mathrm{prox}^f_{\mathcal{M}}(y),\, x - y\rangle_{\mathcal{M}}\,, \quad (2.2)$$

$$\|Q^f_{\mathcal{M}}(x) - Q^f_{\mathcal{M}}(y)\|^2_{\mathcal{M}} \leq \langle Q^f_{\mathcal{M}}(x) - Q^f_{\mathcal{M}}(y),\, x - y\rangle_{\mathcal{M}}\,. \quad (2.3)$$

(iii) $\varphi^f_{\mathcal{M}}$ *is continuous differentiable, and further more, it holds that*

$$\nabla\varphi^f_{\mathcal{M}}(x) = \mathcal{M}(x - \mathrm{prox}^f_{\mathcal{M}}(x)) \in \partial f(\mathrm{prox}^f_{\mathcal{M}}(x)).$$

*Hence,*

$$f(v) \geq f(\mathrm{prox}^f_{\mathcal{M}}(x)) + \langle x - \mathrm{prox}^f_{\mathcal{M}}(x),\, v - \mathrm{prox}^f_{\mathcal{M}}(x)\rangle_{\mathcal{M}} \quad \forall v \in \mathcal{X}.$$

**Proposition 2.4 (Moreau Decomposition).** *Let* $f : \mathcal{X} \to (-\infty, +\infty]$ *be a closed proper convex function and* $f^*$ *be its conjugate. Then any* $z \in \mathcal{X}$ *has the decomposition*

$$z = \mathrm{prox}^f_{\mathcal{M}}(z) + \mathcal{M}^{-1}\mathrm{prox}^{f^*}_{\mathcal{M}^{-1}}(\mathcal{M}z).$$

**Proof**: For any given $z \in \mathcal{X}$, by definition of $\mathrm{prox}^f_{\mathcal{M}}(z)$, we have

$$0 \in \partial f(\mathrm{prox}^f_{\mathcal{M}}(z)) + \mathcal{M}(\mathrm{prox}^f_{\mathcal{M}}(z) - z),$$

i.e., $z - \mathrm{prox}^f_{\mathcal{M}}(z) \in \mathcal{M}^{-1}\partial f(\mathrm{prox}^f_{\mathcal{M}}(z))$. Define function $g : \mathcal{X} \to (-\infty, +\infty]$ as $g(x) \equiv f(\mathcal{M}^{-1}x)$. By [53, Theorem 9.5], $g$ is also a closed proper convex function. By [53, Theorem 12.3 and Theorem 23.9], we have

$$g^*(y) = f^*(\mathcal{M}y) \quad \text{and} \quad \partial g(x) = \mathcal{M}^{-1}\partial f(\mathcal{M}^{-1}x),$$

respectively. Thus, we obtain

$$z - \mathrm{prox}_{\mathcal{M}}^{f}(z) \in \partial g(\mathcal{M}\mathrm{prox}_{\mathcal{M}}^{f}(z)).$$

Then, by [53, Theorem 23.5 and Theorem 23.9], it is easy to have that

$$\mathcal{M}\mathrm{prox}_{\mathcal{M}}^{f}(z) \in \partial g^*(z - \mathrm{prox}_{\mathcal{M}}^{f}(z)) = \mathcal{M}\partial f^*\big(\mathcal{M}(z - \mathrm{prox}_{\mathcal{M}}^{f}(z))\big).$$

Therefore,

$$\begin{aligned} \mathcal{M}(z - \mathrm{prox}_{\mathcal{M}}^{f}(z)) &= \mathrm{argmin}_{y\in\mathcal{X}} \left\{ f^*(y) + \frac{1}{2}\|y - \mathcal{M}z\|_{\mathcal{M}^{-1}}^2 \right\} \\ &= \mathrm{prox}_{\mathcal{M}^{-1}}^{f^*}(\mathcal{M}z). \end{aligned}$$

Thus, we complete the proof. □

Now let us consider a special application of the aforementioned Moreau-Yosida regularization.

We first focus on the case where the function $f$ is assumed to be the indicator function of a given closed convex set $\mathcal{K}$, i.e., $f(x) = \delta_{\mathcal{K}}(x)$ where $\delta_{\mathcal{K}}(x) = 0$ if $x \in \mathcal{K}$ and $\delta_{\mathcal{K}}(x) = \infty$ if $x \notin \mathcal{K}$. For simplicity, we also let the self-adjoint positive definite linear operator $\mathcal{M}$ to be the identity operator $\mathcal{I}$. Then, the proximal point of $x$ associated with indicator function $f(\cdot) = \delta_{\mathcal{K}}(\cdot)$ with $\mathcal{M} = \mathcal{I}$ is the unique optimal solution, denoted by $\Pi_{\mathcal{K}}(x)$, of the following convex optimization problem:

$$\begin{aligned} \min \quad & \frac{1}{2}\|z - x\|^2 \\ \text{s.t.} \quad & z \in \mathcal{K}. \end{aligned} \tag{2.4}$$

In fact, $\Pi_{\mathcal{K}} : \mathcal{X} \to \mathcal{X}$ is the metric projector over $\mathcal{K}$. Thus, the distance function is defined by $\mathrm{dist}(x, \mathcal{K}) = \|x - \Pi_{\mathcal{K}}(x)\|$. By Proposition 2.3, we know that $\Pi_{\mathcal{K}}(x)$ is Lipschitz continuous with modulus 1. Hence, $\Pi_{\mathcal{K}}(\cdot)$ is almost everywhere Fréchet differentiable in $\mathcal{X}$ and for every $x \in \mathcal{X}$, $\partial\Pi_{\mathcal{K}}(x)$ is well defined. Below, we list the following lemma [40], which provides some important properties of $\partial\Pi_{\mathcal{K}}(\cdot)$.

**Lemma 2.5.** *Let $\mathcal{K} \subseteq \mathcal{X}$ be a closed convex set. Then, for any $x \in \mathcal{X}$ and $\mathcal{V} \in \partial\Pi_{\mathcal{K}}(x)$, it holds that*

1. *$\mathcal{V}$ is self-adjoint.*

2. $\langle h, \mathcal{V}h\rangle \geq 0 \quad \forall h \in \mathcal{X}$.

3. $\langle h, \mathcal{V}h\rangle \geq \|\mathcal{V}h\|^2 \quad \forall h \in \mathcal{X}$.

Let $\mathcal{K} = \{W \in \mathcal{S}^n \mid L \leq W \leq U\}$ with $L, U \in \mathcal{S}^n$ being given matrices. For $X \in \mathcal{S}^n$, let $Y = \Pi_{\mathcal{K}}(X)$ be the metric projection of $X$ onto the subset $\mathcal{K}$ of $\mathcal{S}^n$ under the Frobenius norm. Then, $Y = \min(\max(X, L), U)$. Define linear operator $\mathcal{W}^0 : \mathcal{S}^n \to \mathcal{S}^n$ by

$$\mathcal{W}^0(M) = \Omega \circ M, \quad M \in \mathcal{S}^n,$$

where

$$\Omega_{ij} = \begin{cases} 0 & \text{if} \quad X_{ij} < L_{ij}, \\ 1 & \text{if} \quad L_{ij} \leq X_{ij} \leq U_{ij}, \\ 0 & \text{if} \quad X_{ij} > U_{ij}. \end{cases} \tag{2.5}$$

Observing that $\Pi_{\mathcal{K}}(X)$ now is in fact a piecewise linear function, we have $\mathcal{W}^0$ is an element of the set $\partial\Pi_{\mathcal{K}}(X)$.

Let $\mathcal{K} = \mathcal{S}^n_+$, i.e., the cone of $n \times n$ symmetric and positive semidefinite matrices. Given $X \in \mathcal{S}^n$, let $X_+ = \Pi_{\mathcal{S}^n_+}(X)$ be the projection of $X$ onto $\mathcal{S}^n_+$ under the Frobenius norm. Assume that $X$ has the following spectral decomposition

$$X = P\Lambda P^T,$$

where $\Lambda$ is the diagonal matrix with diagonal entries consisting of the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k > 0 \geq \lambda_{k+1} \geq \ldots \geq \lambda_n$ of $X$ and $P$ is a corresponding orthogonal matrix of eigenvectors. Then

$$X_+ = P\Lambda_+ P^T,$$

where $\Lambda_+ = \max\{\Lambda, 0\}$. Sun and Sun, in their paper [58], show that $\Pi_{\mathcal{S}^n_+}(\cdot)$ is strongly semismooth everywhere in $\mathcal{S}^n$. Define the operator $\mathcal{W}^0 : \mathcal{S}^n \to \mathcal{S}^n$ by

$$\mathcal{W}^0(M) = Q(\Omega \circ (Q^T M Q))Q^T, \quad M \in \mathcal{S}^n, \tag{2.6}$$

where

$$\Omega = \begin{pmatrix} E_k & \overline{\Omega} \\ \overline{\Omega}^T & 0 \end{pmatrix}, \quad \overline{\Omega}_{ij} = \frac{\lambda_i}{\lambda_i - \lambda_j}, i \in \{1, \ldots, k\}, j \in \{k+1, \ldots, n\},$$

where $E_k$ is the square matrix of ones with dimension $k$ (the number of positive eigenvalues), and the matrix $\overline{\Omega}$ has all its entries lying in the interval $[0, 1]$. In their paper [47], Pang, Sun and Sun proved that $\mathcal{W}^0$ is an element of the set $\partial \Pi_{\mathcal{S}^n_+}(X)$.

Next we examine the case when the function $f$ is chosen as follows:

$$f(x) = \delta^*_{\mathcal{K}}(-x) = -\inf_{z \in \mathcal{K}} \langle z,\, x\rangle = \sup_{z \in \mathcal{K}} \langle -z,\, x\rangle, \tag{2.7}$$

where $\mathcal{K}$ is a given closed convex set. Then, by Proportion 2.3 and Proposition 2.4, we have the following useful results.

**Proposition 2.6.** *Let* $\varphi(\bar{x}) := \min \delta^*_{\mathcal{K}}(-x) + \frac{\lambda}{2}\|x - \bar{x}\|^2$, *the following results hold:*

(i) $x^+ = \operatorname{argmin} \delta^*_{\mathcal{K}}(-x) + \frac{\lambda}{2}\|x - \bar{x}\|^2 = \bar{x} + \frac{1}{\lambda}\Pi_{\mathcal{K}}(-\lambda\bar{x})$.

(ii) $\nabla\varphi(\bar{x}) = \lambda(\bar{x} - x^+) = -\Pi_{\mathcal{K}}(-\lambda\bar{x})$.

(iii) $\varphi(\bar{x}) = \langle -x^+,\, \Pi_{\mathcal{K}}(-\lambda\bar{x})\rangle + \frac{1}{2\lambda}\|\Pi_{\mathcal{K}}(-\lambda\bar{x})\|^2 = -\langle \bar{x},\, \Pi_{\mathcal{K}}(-\lambda\bar{x})\rangle - \frac{1}{2\lambda}\|\Pi_{\mathcal{K}}(-\lambda\bar{x})\|^2$.

## 2.3 Proximal ADMM

In this section, we review the convergence results for the proximal alternating direction method of multipliers (ADMM) which will be used in our subsequent analysis.

Let $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ be finite dimensional real Euclidian spaces. Let $F : \mathcal{Y} \to (-\infty, +\infty]$ and $G : \mathcal{Z} \to (-\infty, +\infty]$ be closed proper convex functions, $\mathcal{F} : \mathcal{X} \to \mathcal{Y}$ and $\mathcal{G} : \mathcal{X} \to \mathcal{Z}$ be linear maps. Let $\partial F$ and $\partial G$ be the subdifferential mappings of $F$ and $G$, respectively. Since both $\partial F$ and $\partial G$ are maximally monotone [56, Theorem 12.17], there exist two self-adjoint and positive semidefinite operators $\Sigma_F$ and $\Sigma_G$ [13] such that for all $y, \tilde{y} \in \operatorname{dom}(F)$, $\xi \in \partial F(y)$, and $\tilde{\xi} \in \partial F(\tilde{y})$,

$$\langle \xi - \tilde{\xi}, y - \tilde{y}\rangle \geq \|y - \tilde{y}\|^2_{\Sigma_F} \tag{2.8}$$

and for all $z, \tilde{z} \in \mathrm{dom}(G)$, $\zeta \in \partial G(z)$, and $\tilde{\zeta} \in \partial G(\tilde{z})$,

$$\langle \zeta - \tilde{\zeta}, z - \tilde{z} \rangle \geq \|z - \tilde{z}\|_{\Sigma_G}^2. \tag{2.9}$$

### 2.3.1 Semi-proximal ADMM

Firstly, we discuss the semi-proximal ADMM proposed in [13]. Consider the convex optimization problem with the following 2-block separable structure

$$\begin{array}{ll} \min & F(y) + G(z) \\ \text{s.t.} & \mathcal{F}^* y + \mathcal{G}^* z = c. \end{array} \tag{2.10}$$

The dual of problem (2.10) is given by

$$\min \left\{ \langle c,\, x \rangle + F^*(s) + G^*(t) \mid \mathcal{F}x + s = 0,\ \mathcal{G}x + t = 0 \right\}. \tag{2.11}$$

Let $\sigma > 0$ be given. The augmented Lagrangian function associated with (2.10) is given as follows:

$$\mathcal{L}_\sigma(y, z; x) \;=\; F(y) + G(z) + \langle x,\, \mathcal{F}^* y + \mathcal{G}^* z - c \rangle + \frac{\sigma}{2}\|\mathcal{F}^* y + \mathcal{G}^* z - c\|^2.$$

The semi-proximal ADMM proposed in [13], when applied to (2.10), has the following template. Since the proximal terms added here are allowed to be positive semidefinite, the corresponding method is referred to as semi-proximal ADMM instead of proximal ADMM as in [13].

**Algorithm sPADMM: A generic 2-block semi-proximal ADMM for solving (2.10).**

Let $\sigma > 0$ and $\tau \in (0, \infty)$ be given parameters. Let $\mathcal{T}_F$ and $\mathcal{T}_G$ be given self-adjoint positive semidefinite, not necessarily positive definite, linear operators defined on $\mathcal{Y}$ and $\mathcal{Z}$, respectively. Choose $(y^0, z^0, x^0) \in \mathrm{dom}(F) \times \mathrm{dom}(G) \times \mathcal{X}$. For $k = 0, 1, 2, ...,$ perform the $k$th iteration as follows:

**Step 1.** Compute

$$y^{k+1} = \mathrm{argmin}_y \ \mathcal{L}_\sigma(y, z^k; x^k) + \frac{\sigma}{2}\|y - y^k\|^2_{\mathcal{T}_F}. \tag{2.12}$$

**Step 2.** Compute

$$z^{k+1} = \mathrm{argmin}_z \ \mathcal{L}_\sigma(y^{k+1}, z; x^k) + \frac{\sigma}{2}\|z - z^k\|^2_{\mathcal{T}_G}. \tag{2.13}$$

**Step 3.** Compute

$$x^{k+1} = x^k + \tau\sigma(\mathcal{F}^* y^{k+1} + \mathcal{G}^* z^{k+1} - c). \tag{2.14}$$

In the above 2-block semi-proximal ADMM for solving (2.10), the presence of $\mathcal{T}_F$ and $\mathcal{T}_G$ can help to guarantee the existence of solutions for the subproblems (2.12) and (2.13). In addition, they play important roles in ensuring the boundedness of the two generated sequences $\{y^{k+1}\}$ and $\{z^{k+1}\}$. Hence, these two proximal terms are preferred. The choices of $\mathcal{T}_F$ and $\mathcal{T}_G$ are very much problem dependent. The general principle is that both $\mathcal{T}_F$ and $\mathcal{T}_G$ should be as small as possible while $y^{k+1}$ and $z^{k+1}$ are still relatively easy to compute.

For the convergence of the 2-block semi-proximal ADMM, we need the following assumption.

**Assumption 1.** *There exists* $(\hat{y}, \hat{z}) \in \mathrm{ri}(\mathrm{dom}\, F \times \mathrm{dom}\, G)$ *such that* $\mathcal{F}^*\hat{y} + \mathcal{G}^*\hat{z} = c$.

**Theorem 2.7.** *Let* $\Sigma_F$ *and* $\Sigma_G$ *be the self-adjoint and positive semidefinite operators defined by* (2.8) *and* (2.9), *respectively. Suppose that the solution set of problem*

(2.10) *is nonempty and that Assumption 1 holds. Assume that $\mathcal{T}_F$ and $\mathcal{T}_G$ are chosen such that the sequence $\{(y^k, z^k, x^k)\}$ generated by Algorithm sPADMM is well defined. Then, under the condition either (a) $\tau \in (0, (1+\sqrt{5}\,)/2)$ or (b) $\tau \geq (1+\sqrt{5}\,)/2$ but $\sum_{k=0}^{\infty}(\|\mathcal{G}^*(z^{k+1}-z^k)\|^2 + \tau^{-1}\|\mathcal{F}^*y^{k+1} + \mathcal{G}^*z^{k+1} - c\|^2) < \infty$, the following results hold:*

(i) *If $(y^\infty, z^\infty, x^\infty)$ is an accumulation point of $\{(y^k, z^k, x^k)\}$, then $(y^\infty, z^\infty)$ solves problem* (2.10) *and $x^\infty$ solves* (2.11)*, respectively.*

(ii) *If both $\sigma^{-1}\Sigma_F + \mathcal{T}_F + \mathcal{F}\mathcal{F}^*$ and $\sigma^{-1}\Sigma_G + \mathcal{T}_G + \mathcal{G}\mathcal{G}^*$ are positive definite, then the sequence $\{(y^k, z^k, x^k)\}$, which is automatically well defined, converges to a unique limit, say, $(y^\infty, z^\infty, x^\infty)$ with $(y^\infty, z^\infty)$ solving problem* (2.10) *and $x^\infty$ solving* (2.11)*, respectively.*

(iii) *When the y-part disappears, the corresponding results in parts (i)–(ii) hold under the condition either $\tau \in (0, 2)$ or $\tau \geq 2$ but $\sum_{k=0}^{\infty}\|\mathcal{G}^*z^{k+1} - c\|^2 < \infty$.*

**Remark 2.8.** The conclusions of Theorem 2.7 follow essentially from the results given in [13, Theorem B.1]. See [59] for more detailed discussions.

As a simple application of the aforementioned semi-proximal ADMM algorithm, we present a special semi-proximal augmented Lagrangian method for solving the following block-separable convex optimization problem

$$\begin{array}{ll} \min & \sum_{i=1}^{N} \theta_i(v_i) \\ \text{s.t.} & \sum_{i=1}^{N} \mathcal{A}_i^* v_i = c, \end{array} \tag{2.15}$$

where $N$ is a given positive integer, $\theta_i : \mathcal{V}_i \to (-\infty, +\infty]$, $i = 1, \ldots, N$ are closed proper convex functions, $\mathcal{A}_i : \mathcal{X} \to \mathcal{V}_i$, $i = 1, \ldots, N$ are linear operators, $\mathcal{V}_1, \ldots, \mathcal{V}_N$ are all real finite dimensional Euclidean spaces each equipped with an inner product $\langle \cdot, \cdot \rangle$ and its induced norm $\|\cdot\|$. For notational convenience, let

$\mathcal{V} := \mathcal{V}_1 \times \mathcal{V}_2 \times, \ldots, \mathcal{V}_N$. For any $v \in \mathcal{V}$, we write $v \equiv (v_1, v_2, \ldots, v_N) \in \mathcal{V}$. Define the linear map $\mathcal{A} : \mathcal{X} \to \mathcal{V}$ such that its adjoint is given by

$$\mathcal{A}^* v = \sum_{i=1}^{N} \mathcal{A}_i^* v_i \quad \forall v \in \mathcal{V}.$$

Additionally, let

$$\theta(v) = \sum_{i=1}^{N} \theta_i(v_i) \quad \forall v \in \mathcal{V}.$$

Given $\sigma > 0$, the augmented Lagrange function associated with (2.15) is given as follows:

$$\mathcal{L}_\sigma^\theta(v; x) = \theta(v) + \langle x, \mathcal{A}^* v - c \rangle + \frac{\sigma}{2} \|\mathcal{A}^* v - c\|^2. \tag{2.16}$$

In order to handle the non-separability of the quadratic penalty term in $\mathcal{L}_\sigma^\theta$, as well as to design efficient parallel algorithm for solving problem (2.15), we propose the following novel majorization step

$$\begin{aligned} \mathcal{A}\mathcal{A}^* = & \begin{pmatrix} \mathcal{A}_1\mathcal{A}_1^* & \cdots & \mathcal{A}_1\mathcal{A}_N^* \\ \vdots & \ddots & \vdots \\ \mathcal{A}_N\mathcal{A}_1^* & \ldots & \mathcal{A}_N\mathcal{A}_N^* \end{pmatrix} \\ \preceq & \ \ \mathcal{M} := \operatorname{Diag}(\mathcal{M}_1, \ldots, \mathcal{M}_N), \end{aligned} \tag{2.17}$$

with $\mathcal{M}_i \succeq \mathcal{A}_i\mathcal{A}_i^* + \sum_{j \neq i} (\mathcal{A}_i\mathcal{A}_j^*\mathcal{A}_j\mathcal{A}_i^*)^{\frac{1}{2}}$. Let $\mathcal{S} : \mathcal{Y} \to \mathcal{Y}$ be a self-adjoint linear operator given by

$$\mathcal{S} := \mathcal{M} - \mathcal{A}\mathcal{A}^*. \tag{2.18}$$

Here, we state a useful proposition to show that $\mathcal{S}$ is indeed a self-adjoint positive semidefinite linear operator.

**Proposition 2.9.** *It holds that* $\mathcal{S} = \mathcal{M} - \mathcal{A}\mathcal{A}^* \succeq 0$.

**Proof.** The proposition can be proved by observing that for any given matrix $X \in \Re^{m\times n}$, it holds that

$$\begin{pmatrix} & X \\ X^* & \end{pmatrix} \preceq \begin{pmatrix} (XX^*)^{\frac{1}{2}} & \\ & (X^*X)^{\frac{1}{2}} \end{pmatrix}. \quad \square$$

Define $\mathcal{T}_\theta : \mathcal{V} \to \mathcal{V}$ to be a self-adjoint positive semidefinite, not necessarily positive definite, linear operator given by

$$\mathcal{T}_\theta := \mathrm{Diag}(\mathcal{T}_{\theta_1}, \ldots, \mathcal{T}_{\theta_N}), \tag{2.19}$$

where for $i = 1, \ldots, N$, each $\mathcal{T}_{\theta_i}$ is a self-adjoint positive semidefinite linear operator defined on $\mathcal{V}_i$ and is chosen such that the subproblem (2.20) is relatively easy to solve. Now, we are ready to propose a semi-proximal augmented Lagrangian method with a Jacobi type decomposition for solving (2.15).

**Algorithm sPALMJ: A semi-proximal augmented Lagrangian method with a Jacobi type decomposition for solving (2.15).**

Let $\sigma > 0$ and $\tau \in (0, \infty)$ be given initial parameters. Choose $(v^0, x^0) \in \mathrm{dom}(\theta) \times \mathcal{X}$. For $k = 0, 1, 2, ...$, generate $v^{k+1}$ according to the following iteration:

**Step 1.** For $i = 1, \ldots, N$, compute

$$v_i^{k+1} = \mathrm{argmin}_{v_i} \left\{ \begin{array}{l} \mathcal{L}_\sigma^\theta((v_1^k, \ldots, v_{i-1}^k, v_i, v_{i+1}^k, \ldots, v_N^k); x^k) \\ +\frac{\sigma}{2}\|v_i - v_i^k\|^2_{\mathcal{M}_i - \mathcal{A}_{ii}\mathcal{A}_{ii}^*} + \frac{\sigma}{2}\|v_i - v_i^k\|^2_{\mathcal{T}_{\theta_i}} \end{array} \right\}. \tag{2.20}$$

**Step 2.** Compute

$$x^{k+1} = x^k + \tau\sigma(\mathcal{A}^* v^{k+1} - c). \tag{2.21}$$

The relationship between Algorithm sPALMJ and Algorithm sPADMM for solving (2.15) will be revealed in the next proposition. Hence, the convergence of Algorithm sPALMJ can be easily obtained under certain conditions.

**Proposition 2.10.** *For any $k \geq 0$, the point $(v^{k+1}, x^{k+1})$ obtained by Algorithm sPALMJ for solving problem (2.15) can be generated exactly according to the following iteration:*

$$\begin{aligned} v^{k+1} &= \mathrm{argmin}_v \ \mathcal{L}_\sigma^\theta(v; x^k) + \frac{\sigma}{2}\|v - v^k\|^2_{\mathcal{S}} + \frac{\sigma}{2}\|v - v^k\|^2_{\mathcal{T}_\theta}. \\ x^{k+1} &= x^k + \tau\sigma(\mathcal{A}^* v^{k+1} - c). \end{aligned}$$

**Proof.** The equivalence can be obtained by carefully examining the optimality conditions for subproblems (2.20) in Algorithm sPALMJ. □

### 2.3.2 A majorized ADMM with indefinite proximal terms

Secondly, we discuss the majorized ADMM with indefinite proximal terms proposed in [35]. Here, we assume that the convex functions $F(\cdot)$ and $G(\cdot)$ take the following composite form:

$$F(y) = p(y) + f(y) \quad \text{and} \quad G(z) = q(z) + g(z),$$

where $p : \mathcal{Y} \to (-\infty, +\infty]$ and $q : \mathcal{Z} \to (-\infty, +\infty]$ are closed proper convex (not necessarily smooth) functions; $f : \mathcal{Y} \to (-\infty, +\infty]$ and $g : \mathcal{Z} \to (-\infty, +\infty]$ are closed proper convex functions with Lipschitz continuous gradients on some open neighborhoods of $\text{dom}(p)$ and $\text{dom}(q)$, respectively. Problem (2.10) now takes the form of

$$\begin{array}{ll} \min & p(y) + f(y) + q(z) + g(z) \\ \text{s.t.} & \mathcal{F}^* y + \mathcal{G}^* z = c. \end{array} \tag{2.22}$$

Since both $f(\cdot)$ and $g(\cdot)$ are assumed to be smooth convex functions with Lipschitz continuous gradients, we know that there exist two self-adjoint and positive semidefinite linear operators $\Sigma_f$ and $\Sigma_g$ such that for any $y, y' \in \mathcal{Y}$ and any $z, z' \in \mathcal{Z}$,

$$f(y) \;\geq\; f(y') + \langle y - y',\ \nabla f(y') \rangle + \frac{1}{2}\|y - y'\|^2_{\Sigma_f}, \tag{2.23}$$

$$g(z) \;\geq\; g(z') + \langle z - z',\ \nabla g(z') \rangle + \frac{1}{2}\|z - z'\|^2_{\Sigma_g}; \tag{2.24}$$

moreover, there exist self-adjoint and positive semidefinite linear operators $\widehat{\Sigma}_f \succeq \Sigma_f$ and $\widehat{\Sigma}_g \succeq \Sigma_g$ such that for any $y, y' \in \mathcal{Y}$ and any $z, z' \in \mathcal{Z}$,

$$f(y) \;\leq\; \hat{f}(y; y') := f(y') + \langle y - y',\ \nabla f(y') \rangle + \frac{1}{2}\|y - y'\|^2_{\widehat{\Sigma}_f}, \tag{2.25}$$

$$g(z) \;\leq\; \hat{g}(z; z') := g(z') + \langle z - z',\ \nabla g(z') \rangle + \frac{1}{2}\|z - z'\|^2_{\widehat{\Sigma}_g}. \tag{2.26}$$

The two functions $\hat{f}$ and $\hat{g}$ are called the majorized convex functions of $f$ and $g$, respectively. Given $\sigma > 0$, the augmented Lagrangian function is given by

$$\mathcal{L}_\sigma(y,z;x) := p(y) + f(y) + q(z) + g(z) + \langle x,\, \mathcal{F}^*y + \mathcal{G}^*z - c\rangle + \frac{\sigma}{2}\|\mathcal{F}^*y + \mathcal{G}^*z - c\|^2.$$

Similarly, for given $(y',z') \in \mathcal{Y} \times \mathcal{Z}$, $\sigma \in (0,+\infty)$ and any $(x,y,z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, define the majorized augmented Lagrangian function as follows:

$$\widehat{\mathcal{L}}_\sigma(y,z;(x,y',z')) := \left\{ \begin{array}{l} p(y) + \hat{f}(y;y') + q(z) + \hat{g}(z;z') \\ +\langle x, \mathcal{F}^*y + \mathcal{G}^*z - c\rangle + \frac{\sigma}{2}\|\mathcal{F}^*y + \mathcal{G}^*z - c\|^2 \end{array} \right\}, \qquad (2.27)$$

where the two majorized convex functions $\hat{f}$ and $\hat{g}$ are defined by (2.25) and (2.26), respectively. The majorized ADMM with indefinite proximal terms proposed in [35], when applied to (2.22), has the following template.

**Algorithm Majorized iPADMM: A majorized ADMM with indefinite proximal terms for solving (2.22).**

Let $\sigma > 0$ and $\tau \in (0,\infty)$ be given parameters. Let $\mathcal{S}$ and $\mathcal{T}$ be given self-adjoint, possibly indefinite, linear operators defined on $\mathcal{Y}$ and $\mathcal{Z}$, respectively such that

$$\mathcal{M} := \widehat{\Sigma}_f + \mathcal{S} + \sigma\mathcal{F}\mathcal{F}^* \succeq 0 \quad \text{and} \quad \mathcal{N} := \widehat{\Sigma}_g + \mathcal{T} + \sigma\mathcal{G}\mathcal{G}^* \succeq 0.$$

Choose $(y^0, z^0, x^0) \in \text{dom}(p) \times \text{dom}(q) \times \mathcal{X}$. For $k = 0,1,2,...$, perform the $k$th iteration as follows:

**Step 1.** Compute

$$y^{k+1} = \text{argmin}_y\, \widehat{\mathcal{L}}_\sigma(y, z^k; (x^k, y^k, z^k)) + \frac{1}{2}\|y - y^k\|^2_{\mathcal{S}}. \qquad (2.28)$$

**Step 2.** Compute

$$z^{k+1} = \text{argmin}_z\, \widehat{\mathcal{L}}_\sigma(y^{k+1}, z; (x^k, y^k, z^k)) + \frac{1}{2}\|z - z^k\|^2_{\mathcal{T}}. \qquad (2.29)$$

**Step 3.** Compute

$$x^{k+1} = x^k + \tau\sigma(\mathcal{F}^*y^{k+1} + \mathcal{G}^*z^{k+1} - c). \qquad (2.30)$$

There are two important differences between the Majorized iPADMM and the semi-proximal ADMM. Firstly, the majorization technique is imposed in the Majorized iPADMM to make the correspond subproblems in the semi-proximal ADMM more amenable to efficient computations, especially when the functions $f$ and $g$ are not quadratic or linear functions. Secondly, the Majorized iPADMM allows the added proximal terms to be indefinite.

Note that in the context of the 2-block convex composite optimization problem (2.22), Assumption 1 takes the following form:

**Assumption 2.** *There exists* $(\hat{y}, \hat{z}) \in \mathrm{ri}(\mathrm{dom}\, p \times \mathrm{dom}\, q)$ *such that* $\mathcal{F}^*\hat{y} + \mathcal{G}^*\hat{z} = c$.

**Theorem 2.11.** *[35, Theorem 4.1, Remark 4.4] Suppose that the solution set of problem* (2.22) *is nonempty and that Assumption 2 holds. Assume that* $\mathcal{S}$ *and* $\mathcal{T}$ *are chosen such that the sequence* $\{(y^k, z^k, x^k)\}$ *generated by Algorithm sPADMM is well defined. Then, the following results hold:*

(i) *Assume that* $\tau \in (0, (1+\sqrt{5})/2)$ *and for some* $\alpha \in (\tau/\min(1+\tau, 1+\tau^{-1}),\ 1]$,

$$\widehat{\Sigma}_f + \mathcal{S} \succeq 0, \quad \frac{1}{2}\Sigma_f + \mathcal{S} + \frac{(1-\alpha)\sigma}{2}\mathcal{F}\mathcal{F}^* \succeq 0, \quad \frac{1}{2}\Sigma_f + \mathcal{S} + \sigma\mathcal{F}\mathcal{F}^* \succ 0$$

*and*

$$\frac{1}{2}\widehat{\Sigma}_g + \mathcal{T} \succeq 0, \quad \frac{1}{2}\Sigma_g + \mathcal{T} + \min(\tau,\ 1+\tau-\tau^2)\sigma\alpha\mathcal{G}\mathcal{G}^* \succ 0.$$

*Then, the sequence* $\{(y^k, z^k)\}$ *converges to an optimal solution of problem* (2.22) *and* $\{x^k\}$ *converges to an optimal solution of the dual of problem* (2.22).

(ii) *Suppose that* $\mathcal{G}$ *is vacuous,* $q \equiv 0$ *and* $g \equiv 0$. *Then, the corresponding results in part (i) hold under the condition that* $\tau \in (0, 2)$ *and for some* $\alpha \in (\tau/2, 1]$,

$$\widehat{\Sigma}_f + \mathcal{S} \succeq 0, \quad \frac{1}{2}\Sigma_f + \mathcal{S} + \frac{(1-\alpha)\sigma}{2}\mathcal{F}\mathcal{F}^* \succeq 0, \quad \frac{1}{2}\Sigma_f + \mathcal{S} + \sigma\mathcal{F}\mathcal{F}^* \succ 0.$$

In order to discuss the worst-case iteration complexity of the Majorized iPADMM, we need to rewrite the optimization problem (2.22) as the following variational inequality problem: find a vector find a vector $\bar{w} := (\bar{y}, \bar{z}, \bar{x}) \in \mathcal{W} := \mathcal{Y} \times \mathcal{Z} \times \mathcal{X}$ such that

$$\theta(u) - \theta(\bar{u}) + \langle w - \bar{w}, H(\bar{w}) \rangle \geq 0 \qquad \forall\, w \in \mathcal{W} \tag{2.31}$$

with

$$u := \begin{pmatrix} y \\ z \end{pmatrix}, \quad \theta(u) := p(y) + q(z), \quad w := \begin{pmatrix} y \\ z \\ x \end{pmatrix} \text{ and } H(w) := \begin{pmatrix} \nabla f(y) + \mathcal{F}x \\ \nabla g(z) + \mathcal{G}x \\ -(\mathcal{F}^* y + \mathcal{G}^* z - c) \end{pmatrix}. \tag{2.32}$$

Denote by VI($\mathcal{W}, H, \theta$) the variational inequality problem (2.31)-(2.32); and by $\mathcal{W}^*$ the solution set of VI($\mathcal{W}, H, \theta$), which is nonempty under Assumption 2 and the fact that the solution set of problem (2.22) is assumed to be nonempty. Since the mapping $H(\cdot)$ in (2.32) is monotone with respect to $\mathcal{W}$, we have, by [12, Theorem 2.3.5], the solution set $\mathcal{W}^*$ of VI($\mathcal{W}, H, \theta$) is closed and convex and can be characterized as follows:

$$\mathcal{W}^* := \bigcap_{w \in \mathcal{W}} \{\tilde{w} \in \mathcal{W} \mid \theta(u) - \theta(\tilde{u}) + \langle w - \tilde{w}, H(w) \rangle \geq 0\}.$$

Similarly as [46, Definition 1], the definition for an $\varepsilon$-approximation solution of the variational inequality problem is given as following.

**Definition 2.5.** $\tilde{w} \in \mathcal{W}$ is an $\varepsilon$-approximation solution of VI($\mathcal{W}, H, \theta$) if it satisfies

$$\sup_{w \in \mathcal{B}(\tilde{w})} \big\{\theta(\tilde{u}) - \theta(u) + \langle \tilde{w} - w, H(w) \rangle\big\} \leq \varepsilon, \quad \text{where } \mathcal{B}(\tilde{w}) := \big\{w \in \mathcal{W} \mid \|w - \tilde{w}\| \leq 1\big\}.$$

By this definition, the worst-case $O(1/k)$ ergodic iteration-complexity of the Algorithm Majorized iPADMM will be presented in the sense that one can find a $\tilde{w} \in \mathcal{W}$ such that

$$\theta(\tilde{u}) - \theta(u) + \langle \tilde{w} - w, F(w) \rangle \leq \varepsilon \qquad \forall\, w \in \mathcal{B}(\tilde{w})$$

with $\varepsilon = O(1/k)$, after $k$ iterations. Denote

$$\begin{aligned} &\tilde{x}^{k+1} := x^k + \sigma(\mathcal{F}^* y^{k+1} + \mathcal{G}^* z^{k+1} - c), \quad \hat{x}^k = \frac{1}{k}\sum_{i=1}^{k} \tilde{x}^{i+1}, \\ &\hat{y}^k = \frac{1}{k}\sum_{i=1}^{k} y^{i+1}, \quad \hat{z}^k = \frac{1}{k}\sum_{i=1}^{k} z^{i+1}. \end{aligned} \tag{2.33}$$

**Theorem 2.12.** *[35, Theorem 4.3] Suppose that Assumption 2 holds. For* $\tau \in (0, \frac{1+\sqrt{5}}{2})$*, under the same conditions in Theorem 2.11, we have that for any iteration point* $\{(y^k, z^k, x^k)\}$ *generated by Majorized iPADMM,* $(\hat{y}^k, \hat{z}^k, \hat{x}^k)$ *is an* $O(1/k)$*-approximate solution of the first order optimality condition in variational inequality form.*

Chapter 3

# Phase I: A symmetric Gauss-Seidel based proximal ADMM for convex composite quadratic programming

In this chapter, we focus on designing the Phase I algorithm, i.e., a simple yet efficient algorithm to generate a good initial point for our general convex composite quadratic optimization model. Recall the general convex composite quadratic optimization model given in the Chapter 1:

$$\begin{array}{ll} \min & \theta(y_1) + f(y_1, y_2, \ldots, y_p) + \varphi(z_1) + g(z_1, z_2, \ldots, z_q) \\ \text{s.t.} & \mathcal{A}_1^* y_1 + \mathcal{A}_2^* y_2 + \cdots + \mathcal{A}_p^* y_p + \mathcal{B}_1^* z_1 + \mathcal{B}_2^* z_2 + \cdots + \mathcal{B}_q^* z_q = c, \end{array} \tag{3.1}$$

where $p$ and $q$ are given nonnegative integers, $\theta : \mathcal{Y}_1 \to (-\infty, +\infty]$ and $\varphi : \mathcal{Z}_1 \to (-\infty, +\infty]$ are simple closed proper convex function in the sense that their proximal mappings can be relatively easy to compute, $f : \mathcal{Y}_1 \times \mathcal{Y}_2 \times \ldots \times \mathcal{Y}_p \to \Re$ and $g : \mathcal{Z}_1 \times \mathcal{Z}_2 \times \ldots \times \mathcal{Z}_q \to \Re$ are convex quadratic, possibly nonseparable, functions, $\mathcal{A}_i : \mathcal{X} \to \mathcal{Y}_i$, $i = 1, \ldots, p$ and $\mathcal{B}_j : \mathcal{X} \to \mathcal{Z}_j$, $j = 1, \ldots, q$ are linear maps, $\mathcal{Y}_1, \ldots, \mathcal{Y}_p, \mathcal{Z}_1, \ldots, \mathcal{Z}_q$ and $\mathcal{X}$ are all real finite dimensional Euclidean spaces each equipped with an inner product $\langle \cdot, \cdot \rangle$ and its induced norm $\| \cdot \|$. Note that, the functions $f$ and $g$ are also coupled with non-smooth functions $\theta$ and $\varphi$ through the variables $y_1$ and $z_1$, respectively.

For notational convenience, we let $\mathcal{Y} := \mathcal{Y}_1 \times \mathcal{Y}_2 \times, \ldots, \mathcal{Y}_p$, $\mathcal{Z} := \mathcal{Z}_1 \times \mathcal{Z}_2 \times, \ldots, \mathcal{Z}_q$. We write $y \equiv (y_1, y_2, \ldots, y_p) \in \mathcal{Y}$ and $z \equiv (z_1, z_2, \ldots, z_q) \in \mathcal{Z}$. Define the linear maps $\mathcal{A} : \mathcal{X} \to \mathcal{Y}$ and $\mathcal{B} : \mathcal{X} \to \mathcal{Z}$ such that the adjoint maps are given by

$$\mathcal{A}^* y = \sum_{i=1}^{p} \mathcal{A}_i^* y_i \quad \forall y \in \mathcal{Y}, \qquad \mathcal{B}^* z = \sum_{j=1}^{q} \mathcal{B}_j^* z_j \quad \forall z \in \mathcal{Z}.$$

## 3.1 One cycle symmetric block Gauss-Seidel technique

Let $s \geq 2$ be a given integer and $\mathcal{D} := \mathcal{D}_1 \times \mathcal{D}_2 \times \ldots \times \mathcal{D}_s$ with all $\mathcal{D}_i$ being assumed to be real finite dimensional Euclidean spaces. For any $d \in \mathcal{D}$, we write $d \equiv (d_1, d_2, \ldots, d_s) \in \mathcal{D}$. Let $\mathcal{H} : \mathcal{D} \to \mathcal{D}$ be a given self-adjoint positive semidefinite linear operator. Consider the following block decomposition

$$\mathcal{H}d \equiv \begin{pmatrix} \mathcal{H}_{11} & \mathcal{H}_{12} & \cdots & \mathcal{H}_{1s} \\ \mathcal{H}_{12}^* & \mathcal{H}_{22} & \cdots & \mathcal{H}_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{H}_{1s}^* & \mathcal{H}_{2s}^* & \cdots & \mathcal{H}_{ss} \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_s \end{pmatrix},$$

where $\mathcal{H}_{ii} : \mathcal{D}_i \to \mathcal{D}_i,\ i = 1, \ldots, s$ are self-adjoint positive semidefinite linear operators, $\mathcal{H}_{ij} : \mathcal{D}_j \to \mathcal{D}_i,\ i = 1, \ldots, s-1,\ j > i$ are linear maps. Let $r \equiv (r_1, r_2, \ldots, r_s) \in \mathcal{D}$ be given. Define the convex quadratic function $h : \mathcal{D} \to \Re$ by

$$h(d) := \frac{1}{2}\langle d, \mathcal{H}d\rangle - \langle r, d\rangle, \quad d \in \mathcal{D}.$$

Let $\phi : \mathcal{D}_1 \to (-\infty, +\infty]$ be a given closed proper convex function.

### 3.1.1 The two block case

In this subsection, we consider the case for $s = 2$. Assume that $\mathcal{H}_{22} \succ 0$. Define the self-adjoint positive semidefinite linear operator $\widehat{\mathcal{O}} : \mathcal{D}_1 \to \mathcal{D}_1$ by

$$\widehat{\mathcal{O}} = \mathcal{H}_{12}\mathcal{H}_{22}^{-1}\mathcal{H}_{12}^{*}.$$

Let $r_1 \in \mathcal{D}_1$ and $r_2 \in \mathcal{D}_2$ be given. Let $\delta_1^+ \in \mathcal{D}_1$ be an error tolerance vector in $\mathcal{D}_1$, $\delta_2'$ and $\delta_2^+$ be two error tolerance vectors in $\mathcal{D}_2$, which all can be zero vectors. Define

$$\eta(\delta_2', \delta_2^+) = \begin{pmatrix} \mathcal{H}_{12}\mathcal{H}_{22}^{-1}(\delta_2' - \delta_2^+) \\ -\delta_2^+ \end{pmatrix}.$$

Let $(\bar{d}_1, \bar{d}_2) \in \mathcal{D}_1 \times \mathcal{D}_2$ be given two vectors. Define $(d_1^+, d_2^+) \in \mathcal{D}_1 \times \mathcal{D}_2$ by

$$(d_1^+, d_2^+) = \operatorname{argmin}_{d_1,d_2}\ \phi(d_1) + h(d_1, d_2) + \frac{1}{2}\|d_1 - \bar{d}_1\|^2_{\widehat{\mathcal{O}}} - \langle \delta_1^+,\, d_1\rangle + \langle \eta(\delta_2', \delta_2^+),\, d\rangle. \tag{3.2}$$

**Proposition 3.1.** *Suppose that $\mathcal{H}_{22}$ is a self-adjoint positive definite linear operator defined on $\mathcal{D}_2$. Define $d_2' \in \mathcal{D}_2$ by*

$$d_2' = \operatorname{argmin}_{d_2}\ \phi(\bar{d}_1) + h(\bar{d}_1, d_2) - \langle \delta_2',\, d_2\rangle = \mathcal{H}_{22}^{-1}(r_2 + \delta_2' - \mathcal{H}_{12}^{*}\bar{d}_1). \tag{3.3}$$

*Then the optimal solution $(d_1^+, d_2^+)$ to problem (3.2) is generated exactly by the following procedure*

$$\begin{cases} d_1^+ &=& \operatorname{argmin}_{d_1}\ \phi(d_1) + h(d_1, d_2') - \langle \delta_1^+,\, d_1\rangle, \\ d_2^+ &=& \operatorname{argmin}_{d_2}\ \phi(d_1^+) + h(d_1^+, d_2) - \langle \delta_2^+,\, d_2\rangle \ = \ \mathcal{H}_{22}^{-1}(r_2 + \delta_2^+ - \mathcal{H}_{12}^{*}d_1^+). \end{cases} \tag{3.4}$$

*Furthermore, let $\bar{\delta} := \mathcal{H}_{12}\mathcal{H}_{22}^{-1}(r_2 + \delta_2' - \mathcal{H}_{12}^{*}\bar{d}_1 - \mathcal{H}_{22}\bar{d}_2)$, then $(d_1^+, d_2^+)$ can also be obtained by the following equivalent procedure*

$$\begin{cases} d_1^+ &=& \operatorname{argmin}_{d_1}\ \phi(d_1) + h(d_1, \bar{d}_2) + \langle \bar{\delta},\, d_1\rangle - \langle \delta_1^+,\, d_1\rangle, \\ d_2^+ &=& \operatorname{argmin}_{d_2}\ \phi(d_1^+) + h(d_1^+, d_2) - \langle \delta_2^+,\, d_2\rangle \ = \ \mathcal{H}_{22}^{-1}(r_2 + \delta_2^+ - \mathcal{H}_{12}^{*}d_1^+). \end{cases} \tag{3.5}$$

**Proof.** First we show the equivalence between (3.2) and (3.4). Note that (3.4) can be equivalently rewritten as

$$0 \in \partial\phi(d_1^+) + \mathcal{H}_{11}d_1^+ + \mathcal{H}_{12}d_2' - r_1 - \delta_1^+, \tag{3.6}$$

$$d_2^+ = \mathcal{H}_{22}^{-1}(r_2 + \delta_2^+ - \mathcal{H}_{12}^* d_1^+). \tag{3.7}$$

By using the definition of $d_2' = \mathcal{H}_{22}^{-1}(r_2+\delta_2'-\mathcal{H}_{12}^*\bar{d}_1)$, we know that (3.6) is equivalent to

$$0 \in \partial\phi(d_1^+) + \mathcal{H}_{11}d_1^+ + \mathcal{H}_{12}\mathcal{H}_{22}^{-1}(r_2 + \delta_2' - \mathcal{H}_{12}^*\bar{d}_1) - r_1 - \delta_1^+, \tag{3.8}$$

which, in view of (3.7), can be equivalently recast as follows

$$0 \in \partial\phi(d_1^+) + \mathcal{H}_{11}d_1^+ + \mathcal{H}_{12}d_2^+ + \mathcal{H}_{12}\mathcal{H}_{22}^{-1}\mathcal{H}_{12}^*(d_1^+ - \bar{d}_1) + \mathcal{H}_{12}\mathcal{H}_{22}^{-1}(\delta_2' - \delta_2^+) - r_1 - \delta_1^+.$$

Thus, we have

$$\begin{cases} 0 \in \partial\phi(d_1^+) + \mathcal{H}_{11}d_1^+ + \mathcal{H}_{12}d_2^+ + \mathcal{H}_{12}\mathcal{H}_{22}^{-1}(\delta_2' - \delta_2^+) - r_1 - \delta_1^+ + \widehat{\mathcal{O}}(d_1^+ - \bar{d}_1), \\ d_2^+ = \mathcal{H}_{22}^{-1}(r_2 + \delta_2^+ - \mathcal{H}_{12}^* d_1^+), \end{cases}$$

which are equivalently to

$$(d_1^+, d_2^+) = \operatorname{argmin}_{d_1,d_2} \left\{ \begin{array}{l} \phi(d_1) + h(d_1, d_2) - \langle \delta_1^+, d_1\rangle + \frac{1}{2}\|d_1 - \bar{d}_1\|_{\widehat{\mathcal{O}}}^2 \\ +\langle \mathcal{H}_{12}\mathcal{H}_{22}^{-1}(\delta_2' - \delta_2^+), d_1\rangle - \langle \delta_2^+, d_2\rangle \end{array} \right\}.$$

Next, we prove the equivalence between (3.4) and (3.5). By using the definition of $\bar{\delta} := \mathcal{H}_{12}\mathcal{H}_{22}^{-1}(r_2 + \delta_2' - \mathcal{H}_{12}^*\bar{d}_1 - \mathcal{H}_{22}\bar{d}_2)$, we have that (3.8) is equivalent to

$$0 \in \partial\phi(d_1^+) + \mathcal{H}_{11}d_1^+ + \mathcal{H}_{12}\bar{d}_2 - r_1 - \delta_1^+ + \bar{\delta},$$

i.e.,

$$d_1^+ = \operatorname{argmin}_{d_1} \phi(d_1) + h(d_1, \bar{d}_2) + \langle \bar{\delta}, d_1\rangle - \langle \delta_1^+, d_1\rangle.$$

Thus, we obtain the equivalence between (3.4) and (3.5). □

**Remark 3.2.** Under the setting of Proposition 3.1, if $\phi(d_1) \equiv 0$, $\delta_1^+ = 0$, $\delta_2' = \delta_2^+ = 0$ and $\mathcal{H}_{11} \succ 0$, then, by Proposition 3.1, we have $(d_1^+, d_2^+) = \mathrm{argmin}_{d_1,d_2} h(d_1, d_2) + \frac{1}{2}\|d_1 - \bar{d}_1\|^2_{\widehat{\mathcal{O}}}$ and

$$\left\{\begin{array}{lcl} d_2' & = & \mathcal{H}_{22}^{-1}(r_2 - \mathcal{H}_{12}^*\bar{d}_1), \\ d_1^+ & = & \mathcal{H}_{11}^{-1}(r_1 - \mathcal{H}_{12}d_2'), \\ d_2^+ & = & \mathcal{H}_{22}^{-1}(r_2 - \mathcal{H}_{12}^*d_1^+). \end{array}\right. \tag{3.9}$$

Note that, procedure (3.9) is exactly one cycle symmetric block Gauss-Seidel iteration for the following linear system

$$\mathcal{H}d \equiv \begin{pmatrix} \mathcal{H}_{11} & \mathcal{H}_{12} \\ \mathcal{H}_{12}^* & \mathcal{H}_{22} \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} \tag{3.10}$$

with the starting point chosen as $(\bar{d}_1, \bar{d}_2)$.

### 3.1.2 The multi-block case

Now we consider the multi-block case for $s \geq 2$. Here, we further assume that $\mathcal{H}_{ii}$, $i = 2, \ldots, s$ are positive definite. Define

$$d_{\leq i} := (d_1, d_2, \ldots, d_i), \quad d_{\geq i} := (d_i, d_{i+1}, \ldots, d_s),\ i = 0, \ldots, s+1$$

with the convention that $d_0 = d_{s+1} = d_{\leq 0} = d_{\geq s+1} = \emptyset$. Let

$$\mathcal{O}_i := \begin{pmatrix} \mathcal{H}_{1i} \\ \vdots \\ \mathcal{H}_{(i-1)i} \end{pmatrix} \mathcal{H}_{ii}^{-1} \begin{pmatrix} \mathcal{H}_{1i}^* & \cdots & \mathcal{H}_{(i-1)i}^* \end{pmatrix}, \qquad i = 2, \ldots, s.$$

Define the following self-adjoint linear operators: $\widehat{\mathcal{O}}_2 := \mathcal{O}_2$.

$$\widehat{\mathcal{O}}_i := \mathrm{diag}(\widehat{\mathcal{O}}_{i-1}, 0) + \mathcal{O}_i, \qquad i = 3, \ldots, s. \tag{3.11}$$

Let $\delta_1^+ \in \mathcal{D}_1$ and $\delta_i', \delta_i^+ \in \mathcal{D}_i$, $i = 2, \ldots, s$ be given error tolerance vectors. Let

$$\eta_i(\delta_i', \delta_i^+) := \begin{pmatrix} \mathcal{H}_{1i}\mathcal{H}_{ii}^{-1}(\delta_i' - \delta_i^+) \\ \vdots \\ \mathcal{H}_{(i-1)i}\mathcal{H}_{ii}^{-1}(\delta_i' - \delta_i^+) \\ -\delta_i^+ \end{pmatrix}, \quad i = 2, \ldots, s.$$

Define the following linear functions:

$$\Delta_2(d_1, d_2) := -\langle \delta_1^+, d_1 \rangle + \langle \eta_2(\delta_2', \delta_2^+),\, d_{\leq 2} \rangle$$

and for $i = 3, \ldots, s$,

$$\Delta_i(d_{\leq i}) := \Delta_{i-1}(d_{\leq i-1}) + \langle \eta_i(\delta_i', \delta_i^+),\, d_{\leq i} \rangle \tag{3.12}$$

for any $d \in \mathcal{D}$. Write $\delta_{\geq 2}' \equiv (\delta_2', \ldots, \delta_s')$, $\delta_{\geq 2}^+ \equiv (\delta_2^+, \ldots, \delta_s^+)$ and $\delta^+ \equiv (\delta_1^+, \ldots, \delta_s^+)$. By simple calculations, we have that

$$\Delta_s(d) = -\langle \delta^+,\, d \rangle + \Big\langle \mathcal{M}_s(\delta_{\geq 2}' - \delta_{\geq 2}^+),\, d_{\leq s-1} \Big\rangle$$

with

$$\mathcal{M}_s = \begin{pmatrix} \mathcal{H}_{12} & \cdots & \mathcal{H}_{1s} \\ & \ddots & \vdots \\ & & \mathcal{H}_{(s-1)s} \end{pmatrix} \begin{pmatrix} \mathcal{H}_{22}^{-1} & & \\ & \ddots & \\ & & \mathcal{H}_{ss}^{-1} \end{pmatrix}.$$

Let $\bar{d} \in \mathcal{D}$ be given. Define

$$d^+ := \operatorname{argmin}_d \Big\{ \phi(d_1) + h(d) + \frac{1}{2}\| d_{\leq s-1} - \bar{d}_{\leq s-1} \|_{\widehat{\mathcal{O}}_s}^2 + \Delta_s(d) \Big\}. \tag{3.13}$$

The following theorem describing an equivalent procedure for computing $d^+$ is the key ingredient for our subsequent algorithmic developments. The idea of proving this proposition is quite simple: use Proposition 3.1 repeatedly though the proof itself is rather lengthy due to the multi-layered nature of the problems involved. For (3.13), we first express $d_s$ as a function of $d_{\leq s-1}$ to obtain a problem involving only $d_{\leq s-1}$, and from the resulting problem, express $d_{s-1}$ as a function of $d_{\leq s-2}$ to get another problem involving only $d_{\leq s-2}$. We continue this way until we get a problem involving only $(d_1, d_2)$.

**Theorem 3.3.** *Assume that the self-adjoint linear operators $\mathcal{H}_{ii}$, $i = 2, \ldots, s$ are positive definite. For $i = s, \ldots, 2$, define $d'_i \in \mathcal{D}_i$ by*

$$\begin{aligned} d'_i &:= \operatorname{argmin}_{d_i} \phi(\bar{d}_1) + h(\bar{d}_{\le i-1}, d_i, d'_{\ge i+1}) - \langle \delta'_i,\, d_i \rangle \\ &= \mathcal{H}_{ii}^{-1}\big(r_i + \delta'_i - \sum_{j=1}^{i-1} \mathcal{H}_{ji}^* \bar{d}_j - \sum_{j=i+1}^{s} \mathcal{H}_{ij} d'_j\big). \end{aligned} \tag{3.14}$$

(i) *Then the optimal solution $d^+$ defined by* (3.13) *can be obtained exactly via*

$$\begin{cases} d_1^+ &= \operatorname{argmin}_{d_1} \phi(d_1) + h(d_1, d'_{\ge 2}) - \langle \delta_1^+,\, d_1 \rangle, \\ d_i^+ &= \operatorname{argmin}_{d_i} \phi(d_1^+) + h(d_{\le i-1}^+, d_i, d'_{\ge i+1}) - \langle \delta_i^+,\, d_i \rangle \\ &= \mathcal{H}_{ii}^{-1}(r_i + \delta_i^+ - \sum_{j=1}^{i-1} \mathcal{H}_{ji}^* d_j^+ - \sum_{j=i+1}^{s} \mathcal{H}_{ij} d'_j), \quad i = 2, \ldots, s. \end{cases} \tag{3.15}$$

(ii) *It holds that*

$$\mathcal{H} + \operatorname{diag}(\widehat{\mathcal{O}}_s, 0) \succ 0 \Leftrightarrow \mathcal{H}_{11} \succ 0. \tag{3.16}$$

**Proof.** We will separate our proof into two parts.

**Part (i).** We prove our conclusions by induction. Firstly, the case for $s = 2$ has been proven in Proposition 3.1.

Assume now that the equivalence between (3.13) and (3.15) holds for all $s \le l$. We need to show that for $s = l + 1$, this equivalence also holds. For this purpose, we define the following quadratic function with respect to $d_{\le l}$ and $d_{l+1}$

$$h_{l+1}(d_{\le l}, d_{l+1}) := h(d_{\le l}, d_{l+1}) + \frac{1}{2}\|d_{\le l-1} - \bar{d}_{\le l-1}\|^2_{\widehat{\mathcal{O}}_l} + \Delta_l(d_{\le l}). \tag{3.17}$$

By using the definitions (3.11) and (3.12) and noting that

$$\frac{1}{2}\|d_{\le l} - \bar{d}_{\le l}\|^2_{\widehat{\mathcal{O}}_{l+1}} = \frac{1}{2}\|d_{\le l-1} - \bar{d}_{\le l-1}\|^2_{\widehat{\mathcal{O}}_l} + \frac{1}{2}\|d_{\le l} - \bar{d}_{\le l}\|^2_{\mathcal{O}_{l+1}}$$

and

$$\Delta_{l+1}(d_{\le l+1}) = \Delta_l(d_{\le l}) + \langle \eta_{l+1}(\delta'_{l+1}, \delta^+_{l+1}),\, d_{\le l+1} \rangle,$$

we can rewrite the optimization problem (3.13) for $s = l + 1$ equivalently as

$$(d^+_{\le l}, d^+_{l+1}) = \operatorname{argmin}_{(d_{\le l}, d_{l+1})} \left\{ \begin{array}{l} \phi(d_1) + h_{l+1}(d_{\le l}, d_{l+1}) + \frac{1}{2}\|d_{\le l} - \bar{d}_{\le l}\|^2_{\mathcal{O}_{l+1}} \\ + \langle \eta_{l+1}(\delta'_{l+1}, \delta^+_{l+1}),\, d_{\le l+1} \rangle \end{array} \right\}. \tag{3.18}$$

Now, from Proposition 3.1, we know that the optimal solution $(d^+_{\leq l}, d^+_{l+1})$ to problem (3.18) is generated exactly by the following procedure

$$\begin{aligned} d'_{l+1} &= \operatorname{argmin}_{d_{l+1}} \phi(\bar{d}_1) + h_{l+1}(\bar{d}_{\leq l}, d_{l+1}) - \langle \delta'_{l+1}, d_{l+1} \rangle \\ &= \operatorname{argmin}_{d_{l+1}} \phi(\bar{d}_1) + h(\bar{d}_{\leq l}, d_{l+1}) - \langle \delta'_{l+1}, d_{l+1} \rangle, \qquad (3.19) \\ d^+_{\leq l} &= \operatorname{argmin}_{d_{\leq l}} \phi(d_1) + h_{l+1}(d_{\leq l}, d'_{l+1}), \qquad (3.20) \\ d^+_{l+1} &= \operatorname{argmin}_{d_{l+1}} \phi(d^+_1) + h_{l+1}(d^+_{\leq l}, d_{l+1}) - \langle \delta^+_{l+1}, d_{l+1} \rangle \\ &= \operatorname{argmin}_{d_{l+1}} \phi(d^+_1) + h(d^+_{\leq l}, d_{l+1}) - \langle \delta^+_{l+1}, d_{l+1} \rangle. \qquad (3.21) \end{aligned}$$

In order to apply our induction hypothesis to problem (3.20), we need to construct a corresponding quadratic function. For this purpose, let the self-dual positive semidefinite linear operator $\widetilde{\mathcal{H}}$ be defined by

$$\widetilde{\mathcal{H}} \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_l \end{pmatrix} := \begin{pmatrix} \mathcal{H}_{11} & \mathcal{H}_{12} & \cdots & \mathcal{H}_{1l} \\ \mathcal{H}^*_{12} & \mathcal{H}_{22} & \cdots & \mathcal{H}_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{H}^*_{1l} & \mathcal{H}^*_{2l} & \cdots & \mathcal{H}_{ll} \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_l \end{pmatrix}.$$

Consider the following quadratic function with respect to $d_{\leq l}$, which is obtained from $h(d_{\leq l}, d'_{l+1})$,

$$\widetilde{h}(d_{\leq l}; d'_{l+1}) := \frac{1}{2}\langle d_{\leq l}, \widetilde{\mathcal{H}} d_{\leq l} \rangle - \langle r_{\leq l} - (\mathcal{H}^*_{1,l+1}, \ldots, \mathcal{H}^*_{l,l+1})^* d'_{l+1}, d_{\leq l} \rangle. \qquad (3.22)$$

Note that

$$h_{l+1}(d_{\leq l}, d'_{l+1}) = \left\{ \begin{array}{l} \widetilde{h}(d_{\leq l}; d'_{l+1}) + \frac{1}{2}\|d_{\leq l-1} - \bar{d}_{\leq l-1}\|^2_{\widehat{\mathcal{O}}_l} + \Delta_l(d_{\leq l}) \\ +\frac{1}{2}\langle d'_{l+1}, \mathcal{H}_{l+1,l+1} d'_{l+1} \rangle - \langle r_{l+1}, d'_{l+1} \rangle \end{array} \right\}.$$

Therefore, problem (3.20) can be equivalently recast as

$$d^+_{\leq l} = \operatorname{argmin}_{d_{\leq l}} \phi(d_1) + \widetilde{h}(d_{\leq l}; d'_{l+1}) + \frac{1}{2}\|d_{\leq l-1} - \bar{d}_{\leq l-1}\|^2_{\widehat{\mathcal{O}}_l} + \Delta_l(d_{\leq l}). \qquad (3.23)$$

By applying our induction hypothesis on (3.23), we obtain equivalently that

$$\widetilde{d}'_i = \mathrm{argmin}_{d_i} \left\{ \begin{array}{l} \phi(\bar{d}_1) + \widetilde{h}(\bar{d}_{\le i-1}, d_i, (\widetilde{d}'_{i+1}, \ldots, \widetilde{d}'_l); d'_{l+1}) \\ -\langle \delta'_i,\, d_i\rangle \end{array} \right\}, i = l, \ldots, 2, \tag{3.24}$$

$$d_1^+ = \mathrm{argmin}_{d_1}\ \phi(d_1) + \widetilde{h}(d_1, (\widetilde{d}'_2, \ldots, \widetilde{d}'_l); d'_{l+1}) - \langle \delta_1^+,\, d_1\rangle, \tag{3.25}$$

$$d_i^+ = \mathrm{argmin}_{d_i} \left\{ \begin{array}{l} \phi(d_1^+) + \widetilde{h}(d^+_{\le i-1}, d_i, (\widetilde{d}'_{i+1}, \ldots, \widetilde{d}'_l); d'_{l+1}) \\ -\langle \delta_i^+,\, d_i\rangle \end{array} \right\}, i = 2, \ldots, l. \tag{3.26}$$

Next we need to prove that

$$\widetilde{d}'_i = d'_i \quad \forall i = l, \ldots, 2. \tag{3.27}$$

By using the definition of the quadratic function $\widetilde{h}$ in (3.22) and the definition of $d'$ in (3.14), we have that

$$\widetilde{d}'_l = \mathcal{H}_{ll}^{-1}\big(r_l + \delta'_l - \mathcal{H}_{l,l+1}d'_{l+1} - \sum_{j=1}^{l-1} \mathcal{H}_{jl}^* \bar{d}_j\big) = d'_l.$$

That is, (3.27) holds for $i = l$. Now assume that we have proven $\widetilde{d}'_i = d'_i$ for all $i \ge k+1$ with $k+1 \le l$. We shall next prove that (3.27) holds for $i = k$. Again, by using the definition of $\widetilde{h}$ and $d'$, we obtain that

$$\begin{aligned} \widetilde{d}'_k &= \mathcal{H}_{kk}^{-1}\big(r_k + \delta'_k - \mathcal{H}_{k,l+1}d'_{l+1} - \sum_{j=1}^{k-1} \mathcal{H}_{jk}^* \bar{d}_j - \sum_{j=k+1}^{l} \mathcal{H}_{kj}\widetilde{d}'_j\big) \\ &= \mathcal{H}_{kk}^{-1}\big(r_k + \delta'_k - \mathcal{H}_{k,l+1}d'_{l+1} - \sum_{j=1}^{k-1} \mathcal{H}_{jk}^* \bar{d}_j - \sum_{j=k+1}^{l} \mathcal{H}_{kj} d'_j\big) \\ &= d'_k, \end{aligned}$$

which shows that (3.27) holds for $i = k$. Thus, (3.27) holds. Note that by the definition of $\widetilde{h}$ and direct calculations, we have that

$$h(d_{\le l}, d'_{l+1}) = \widetilde{h}(d_{\le l}; d'_{l+1}) + \frac{1}{2}\langle d'_{l+1},\, \mathcal{H}_{l+1,l+1}d'_{l+1}\rangle - \langle r_{l+1},\, d'_{l+1}\rangle. \tag{3.28}$$

Thus, by using (3.27) and (3.28), we know that (3.25) and (3.26) can be rewritten as

$$\left\{\begin{array}{lcl} d_i' & = & \mathrm{argmin}_{d_i}\ \phi(\bar{d}_1) + h(\bar{d}_{\le i-1}, d_i, d'_{\ge i+1}) - \langle \delta_i',\, d_i\rangle, \quad i = l,\ldots,2, \\ d_1^+ & = & \mathrm{argmin}_{d_1} \phi(d_1) + h(d_1, d'_{\ge 2}) - \langle \delta_1^+,\, d_1\rangle, \\ d_i^+ & = & \mathrm{argmin}_{d_i}\ \phi(d_1^+) + h(d^+_{\le i-1}, d_i, d'_{\ge i+1}) - \langle \delta_i^+,\, d_i\rangle, \quad i = 2,\ldots,l, \end{array}\right.$$

which together with (3.19) and (3.21) shows that the equivalence between (3.13) and (3.15) holds for $s = l+1$. Thus, the proof of the first part is completed.

**Part (ii).** Now we prove the second part. If $s = 2$, we have

$$\mathcal{H} + \mathrm{diag}(\widehat{\mathcal{O}}_2, 0) = \begin{pmatrix} \mathcal{H}_{11} + \widehat{\mathcal{O}}_2 & \mathcal{H}_{12} \\ \mathcal{H}_{12}^* & \mathcal{H}_{22} \end{pmatrix}.$$

Since $\mathcal{H}_{22} \succ 0$, by the Schur complement condition for ensuring the positive definiteness of linear operators, we get

$$\begin{pmatrix} \mathcal{H}_{11} + \widehat{\mathcal{O}}_2 & \mathcal{H}_{12} \\ \mathcal{H}_{12}^* & \mathcal{H}_{22} \end{pmatrix} \succ 0 \ \Leftrightarrow\ \mathcal{H}_{11} + \widehat{\mathcal{O}}_2 - \mathcal{H}_{12}\mathcal{H}_{22}^{-1}\mathcal{H}_{12}^* = \mathcal{H}_{11} \succ 0. \tag{3.29}$$

Thus, we complete the proof the case of $s = 2$.

For the case $s \ge 3$, let $\widehat{\mathcal{H}}_1 = \mathcal{H}_{11}$. For $i = 1,\ldots,s-1$, define

$$\mathcal{H}_{\le i,i+1} := \begin{pmatrix} \mathcal{H}_{1(i+1)} \\ \vdots \\ \mathcal{H}_{i(i+1)} \end{pmatrix} \quad \text{and} \quad \widehat{\mathcal{H}}_{i+1} := \begin{pmatrix} \widehat{\mathcal{H}}_i & \mathcal{H}_{\le i,i+1} \\ \mathcal{H}^*_{\le i,i+1} & \mathcal{H}_{(i+1)(i+1)} \end{pmatrix}.$$

Since $\mathcal{H}_{ii} \succ 0$ for all $i \ge 2$, by the Schur complement condition for ensuring the positive definiteness of linear operators, we obtain, for $i = 2,\ldots,s-1$,

$$\widehat{\mathcal{H}}_{i+1} + \mathrm{diag}(\widehat{\mathcal{O}}_{i+1}, 0) = \begin{pmatrix} \widehat{\mathcal{H}}_i + \widehat{\mathcal{O}}_{i+1} & \mathcal{H}_{\le i,i+1} \\ \mathcal{H}^*_{\le i,i+1} & \mathcal{H}_{(i+1),(i+1)} \end{pmatrix} \succ 0$$
$$\Updownarrow$$
$$\widehat{\mathcal{H}}_i + \widehat{\mathcal{O}}_{i+1} - \mathcal{H}_{\le i,i+1}\mathcal{H}^{-1}_{(i+1),(i+1)}\mathcal{H}^*_{\le i,i+1} = \widehat{\mathcal{H}}_i + \mathrm{diag}(\widehat{\mathcal{O}}_i, 0) \succ 0.$$

Therefore, by taking $i = 2$, we obtain that

$$\mathcal{H} + \mathrm{diag}(\widehat{\mathcal{O}}_s, 0) \succ 0 \ \Leftrightarrow\ \begin{pmatrix} \widehat{\mathcal{H}}_{11} + \widehat{\mathcal{O}}_2 & \mathcal{H}_{\le 1,2} \\ \mathcal{H}^*_{\le 1,2} & \mathcal{H}_{22} \end{pmatrix} = \begin{pmatrix} \mathcal{H}_{11} + \widehat{\mathcal{O}}_2 & \mathcal{H}_{12} \\ \mathcal{H}_{12}^* & \mathcal{H}_{22} \end{pmatrix} \succ 0,$$

i.e.,

$$\mathcal{H} + \operatorname{diag}(\widehat{\mathcal{O}}_s, 0) \succ 0 \;\Leftrightarrow\; \mathcal{H}_{11} \succ 0.$$

This completes the proof to the second part of this theorem. □

**Remark 3.4.** Under the setting of Theorem 3.3, if $\phi(d_1) \equiv 0$, $\delta_1^+ = 0$, $\delta_i' = \delta_i^+ = 0, i = 2, \ldots, s$ and $\mathcal{H}_{11} \succ 0$, then we know from Proposition 3.3 that

$$\left\{ \begin{array}{lcl} d_i' & = & \mathcal{H}_{ii}^{-1}\big(r_i - \sum_{j=1}^{i-1} \mathcal{H}_{ji}^* \bar{d}_j - \sum_{j=i+1}^{s} \mathcal{H}_{ij} d_j'\big), \quad i = s, \ldots, 2, \\ d_1^+ & = & \mathcal{H}_{11}^{-1}\big(r_1 - \sum_{j=2}^{s} \mathcal{H}_{1j} d_j'\big), \\ d_i^+ & = & \mathcal{H}_{ii}^{-1}(r_i - \sum_{j=1}^{i-1} \mathcal{H}_{ji}^* d_j^+ - \sum_{j=i+1}^{s} \mathcal{H}_{ij} d_j'), \quad i = 2, \ldots, s. \end{array} \right. \tag{3.30}$$

The procedure (3.30) is exactly one cycle symmetric block Gauss-Seidel iteration for the following linear system

$$\mathcal{H}d \equiv \begin{pmatrix} \mathcal{H}_{11} & \mathcal{H}_{12} & \cdots & \mathcal{H}_{1s} \\ \mathcal{H}_{12}^* & \mathcal{H}_{22} & \cdots & \mathcal{H}_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{H}_{1s}^* & \mathcal{H}_{2s}^* & \cdots & \mathcal{H}_{ss} \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_s \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_s \end{pmatrix} \tag{3.31}$$

with the initial point chosen as $\bar{d}$. Therefore, one can see that using the symmetric Gauss-Seidel method for solving the linear system (3.31) can equivalently be regarded as solving exactly a sequence of quadratic programming problems of the form (3.13). Specifically, given $d^0 \in \mathcal{D}$, for $k = 0, 1, \ldots$, compute

$$d^{k+1} = \operatorname{argmin}_d \Big\{ h(d) + \frac{1}{2}\|d_{\le s-1} - d_{\le s-1}^k\|_{\widehat{\mathcal{O}}_s}^2 \Big\}.$$

As far as we are aware of, this is the first time that the symmetric block Gauss-Seidel algorithm is interpreted, from the optimization perspective, as a sequential quadratic programming procedure.

## 3.2 A symmetric Gauss-Seidel based semi-proximal ALM

Before we introduce our approach for the general multi-block case, we shall first pay particular attention to a special case of the general convex composite quadratic optimization model (3.1). More specifically, we consider a simple yet important convex composite quadratic optimization problem with the following 2-block separable structure

$$\begin{array}{ll} \min & \theta(y_1) + \rho(y_2) \\ \text{s.t.} & \mathcal{A}_1^* y_1 + \mathcal{A}_2^* y_2 = c, \end{array} \tag{3.32}$$

i.e., in (3.1), $p = 2$, $\mathcal{B}$ is vacuous, $\varphi \equiv 0$, $g \equiv 0$ and $\rho(y_2) \equiv f(y_1, y_2)$ $\forall (y_1, y_2) \in \mathcal{Y}_1 \times \mathcal{Y}_2$ is a convex quadratic function depending only on $y_2$:

$$\rho(y_2) = \frac{1}{2}\langle y_2,\, \Sigma_2 y_2\rangle - \langle b,\, y_2\rangle, \quad y_2 \in \mathcal{Y}_2,$$

where $\Sigma_2$ is a self-adjoint positive semidefinite linear operator defined on $\mathcal{Y}_2$ and $b \in \mathcal{Y}_2$ is a given vector. Let $\partial\theta$ be the subdifferential mapping of $\theta$. Since $\partial\theta$ is maximally monotone [53, Corollary 31.5.2], there exists a self-adjoint and positive semidefinite operator $\Sigma_1$ such that for all $y_1, \tilde{y}_1 \in \mathrm{dom}(\theta)$, $\xi \in \partial\theta(y_1)$, and $\tilde{\xi} \in \partial\theta(\tilde{y}_1)$,

$$\langle \xi - \tilde{\xi}, y_1 - \tilde{y}_1\rangle \geq \|y_1 - \tilde{y}_1\|_{\Sigma_1}^2.$$

Given $\sigma > 0$, the augmented Lagrangian function associated with (3.32) is given as follows:

$$\mathcal{L}_\sigma(y_1, y_2; x) \;=\; \theta(y_1) + \rho(y_2) + \langle x,\, \mathcal{A}_1^* y_1 + \mathcal{A}_2^* y_2 - c\rangle + \frac{\sigma}{2}\|\mathcal{A}_1^* y_1 + \mathcal{A}_2^* y_2 - c\|^2.$$

Here, we consider using Algorithm sPADMM, proposed in [13] and reviewed in Chapter 2, to solve problem (3.32). In order to solve the subproblem associated with $y_2$ in Algorithm sPADMM, we need to solve a linear system with the linear operator given by $\sigma^{-1}\Sigma_2 + \mathcal{A}_2\mathcal{A}_2^*$. Hence, an appropriate proximal term should be chosen such that the corresponding subproblem can be solved efficiently. Here, we choose $\mathcal{T}_2$ as follows. Let $\mathcal{E}_2 : \mathcal{Y}_2 \to \mathcal{Y}_2$ be a self-adjoint positive definite linear operator such that it is a majorization of $\sigma^{-1}\Sigma_2 + \mathcal{A}_2\mathcal{A}_2^*$, i.e.,

$$\mathcal{E}_2 \succeq \sigma^{-1}\Sigma_2 + \mathcal{A}_2\mathcal{A}_2^*.$$

We choose $\mathcal{E}_2$ such that its inverse can be computed at a moderate cost. Define

$$\mathcal{T}_2 := \mathcal{E}_2 - \sigma^{-1}\Sigma_2 - \mathcal{A}_2\mathcal{A}_2^* \succeq 0. \tag{3.33}$$

Note that for numerical efficiency, we need the self-adjoint positive semidefinite linear operator $\mathcal{T}_2$ to be as small as possible. In order to fully exploit the structure of the quadratic function $\rho(\cdot)$, we add, instead of a naive proximal term, a proximal term based on the symmetric Gauss-Seidel technique as follows. For a given $\mathcal{T}_1 \succeq 0$, we define the self-adjoint positive semidefinite linear operator

$$\widehat{\mathcal{T}}_1 := \mathcal{T}_1 + \mathcal{A}_1\mathcal{A}_2^*\mathcal{E}_2^{-1}\mathcal{A}_2\mathcal{A}_1^*. \tag{3.34}$$

Now, we can propose our symmetric Gauss-Seidel based semi-proximal augmented Lagrangian method (sGS-sPALM) to solve (3.32) with a specially chosen proximal term involving $\widehat{\mathcal{T}}_1$ and $\mathcal{T}_2$.

**Algorithm sGS-sPALM: A symmetric Gauss-Seidel based semi-proximal augmented Lagrangian method for solving (3.32).**

Let $\sigma > 0$ and $\tau \in (0, \infty)$ be given parameters. Choose $(y_1^0, y_2^0, x^0) \in \mathrm{dom}(\theta) \times \mathcal{Y}_2 \times \mathcal{X}$. For $k = 0, 1, 2, ...$, perform the $k$th iteration as follows:

**Step 1.** Compute

$$(y_1^{k+1}, y_2^{k+1}) = \mathrm{argmin}_{y_1,y_2} \left\{ \begin{array}{l} \mathcal{L}_\sigma(y_1, y_2; x^k) + \frac{\sigma}{2}\|y_1 - y_1^k\|^2_{\widehat{\mathcal{T}}_1} \\ +\frac{\sigma}{2}\|y_2 - y_2^k\|^2_{\mathcal{T}_2} \end{array} \right\}. \tag{3.35}$$

**Step 2.** Compute

$$x^{k+1} = x^k + \tau\sigma(\mathcal{F}^* u^{k+1} + \mathcal{G}^* v^{k+1} - c). \tag{3.36}$$

Note that problem (3.35) in Step 1 is well defined if $\sigma^{-1}\Sigma_1 + \mathcal{T}_1 + \mathcal{A}_1\mathcal{A}_1^* \succ 0$.

For the convergence of the sGS-sPALM, we need the following assumption.

**Assumption 3.** *There exists* $(\hat{y}_1, \hat{y}_2) \in \mathrm{ri}(\mathrm{dom}\,\theta) \times \mathcal{Y}_2$ *such that* $\mathcal{A}_1^*\hat{y}_1 + \mathcal{A}_2^*\hat{y}_2 = c$.

Now, we are ready to establish our convergence results for Algorithm sGS-sPALM for solving (3.32).

**Theorem 3.5.** *Suppose that the solution set of problem* (3.32) *is nonempty and that Assumption 3 holds. Assume that* $\mathcal{T}_1$ *is chosen such that the sequence* $\{(y_1^k, y_2^k, x^k)\}$ *generated by Algorithm sGS-sPALM is well defined. Then, under the condition either (a)* $\tau \in (0, 2)$ *or (b)* $\tau \geq 2$ *but* $\sum_{k=0}^{\infty} \|\mathcal{A}_1^* y_1^{k+1} + \mathcal{A}_2^* y_2^{k+1} - c\|^2 < \infty$, *the following results hold:*

(i) *If* $(y_1^\infty, y_2^\infty, x^\infty)$ *is an accumulation point of* $\{(y_1^k, y_2^k, x^k)\}$, *then* $(y_1^\infty, y_2^\infty)$ *solves problem* (3.32) *and* $x^\infty$ *solves its dual problem, respectively.*

(ii) *If* $\sigma^{-1}\Sigma_1 + \mathcal{T}_1 + \mathcal{A}_1\mathcal{A}_1^*$ *is positive definite, then the sequence* $\{(y_1^k, y_2^k, x^k)\}$ *is well defined and it converges to a unique limit, say,* $(y_1^\infty, y_2^\infty, x^\infty)$ *with* $(y_1^\infty, y_2^\infty)$ *solving problem* (3.32) *and* $x^\infty$ *solving the corresponding dual problem, respectively.*

**Proof.** By combining Theorem 2.7 and the fact that

$$\begin{aligned} & \begin{pmatrix} \mathcal{A}_1 \\ \mathcal{A}_2 \end{pmatrix} \begin{pmatrix} \mathcal{A}_1 \\ \mathcal{A}_2 \end{pmatrix}^* + \sigma^{-1} \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix} + \begin{pmatrix} \widehat{\mathcal{T}}_1 & \\ & \mathcal{T}_2 \end{pmatrix} \succ 0 \\ \Longleftrightarrow \quad & \mathcal{A}_1\mathcal{A}_1^* + \sigma^{-1}\Sigma_1 + \mathcal{T}_1 \succ 0, \end{aligned}$$

one can prove the results of this theorem directly. □

Now we are able to apply our one cycle symmetric Gauss-Seidel technique on the subproblem (3.35). Let $\delta_\rho : \mathcal{Y}_1 \times \mathcal{Y}_2 \times \mathcal{X} \to \mathcal{Y}_1$ be an auxiliary linear function associated with (3.35) defined by

$$\delta_\rho(y_1, y_2, x) \ := \ \mathcal{A}_1\mathcal{A}_2^*\mathcal{E}_2^{-1}(b - \mathcal{A}_2 x - \Sigma_2 y_2 + \sigma\mathcal{A}_2(c - \mathcal{A}_1^* y_1 - \mathcal{A}_2^* y_2)). \tag{3.37}$$

**Proposition 3.6.** *Let $\delta_\rho^k := \delta_\rho(y_1^k, y_2^k, x^k)$ for $k = 0, 1, 2, \ldots$. We have that $y_1^{k+1}$ and $y_2^{k+1}$ obtained by Algorithm sGS-sPALM for solving* (3.32) *can be generated exactly according to the following procedure:*

$$\begin{cases} \bar{y}_2^k &= \operatorname{argmin}_{y_2} \mathcal{L}_\sigma(y_1^k, y_2; x^k) + \frac{\sigma}{2}\|y_2 - y_2^k\|_{\mathcal{T}_2}^2, \\ y_1^{k+1} &= \operatorname{argmin}_{y_1} \mathcal{L}_\sigma(y_1, \bar{y}_2^k; x^k) + \frac{\sigma}{2}\|y_1 - y_1^k\|_{\mathcal{T}_1}^2, \\ y_2^{k+1} &= \operatorname{argmin}_{y_2} \mathcal{L}_\sigma(y_1^{k+1}, y_2; x^k) + \frac{\sigma}{2}\|y_2 - y_2^k\|_{\mathcal{T}_2}^2, \\ x^{k+1} &= x^k + \tau\sigma(\mathcal{A}_1^* y_1^{k+1} + \mathcal{A}_2^* y_2^{k+1} - c). \end{cases} \tag{3.38}$$

*Equivalently, $(y_1^{k+1}, y_2^{k+1})$ can also be obtained exactly via:*

$$\begin{cases} y_1^{k+1} &= \operatorname{argmin}_{y_1} \mathcal{L}_\sigma(y_1, y_2^k; x^k) + \langle \delta_\rho^k,\, y_1\rangle + \frac{\sigma}{2}\|y_1 - y_1^k\|_{\mathcal{T}_1}^2, \\ y_2^{k+1} &= \operatorname{argmin}_{y_2} \mathcal{L}_\sigma(y_1^{k+1}, y_2; x^k) + \frac{\sigma}{2}\|y_2 - y_2^k\|_{\mathcal{T}_2}^2, \\ x^{k+1} &= x^k + \tau\sigma(\mathcal{A}_1^* y_1^{k+1} + \mathcal{A}_2^* y_2^{k+1} - c). \end{cases} \tag{3.39}$$

**Proof.** The results follow directly from (3.4) and (3.5) in Proposition 3.1 with all the error tolerance vectors $(\delta_1^+, \delta_2', \delta_2^+)$ chosen to be zero vectors. □

**Remark 3.7.** (i) Note that comparing to the Algorithm sPADMM, the first subproblem of (3.39) has an extra linear term $\langle \delta_\rho^k, \cdot\rangle$. This linear term will vanish if $\Sigma_2 = 0$, $\mathcal{E}_2 = \mathcal{A}_2\mathcal{A}_2^* \succ 0$ and a proper starting point $(y_1^0, y_2^0, x^0)$ is chosen. Specifically, if we choose $x^0 \in \mathcal{X}$ such that $\mathcal{A}_2 x^0 = b$ and $(y_1^0, y_2^0) \in \operatorname{dom}(\theta) \times \mathcal{Y}_2$ such that $y_2^0 = \mathcal{E}_2^{-1}\mathcal{A}_2(c - \mathcal{A}_1^* y_1^0)$, then it holds that $\mathcal{A}_2 x^k = b$ and $y_2^k = \mathcal{E}_2^{-1}\mathcal{A}_2(c - \mathcal{A}_1^* y_1^k)$, which imply that $\delta_\rho^k = 0$.

(ii) Observe that when $\mathcal{T}_1$ and $\mathcal{T}_2$ are chosen to be 0 in (3.39), apart from the range of $\tau$, our Algorithm sGS-sPALM differs from the classical 2-block ADMM for solving problem (3.32) only in the linear term $\langle \delta_\rho^k, \cdot\rangle$. This shows that the classical 2-block ADMM for solving problem (3.32) has an unremovable deviation from the augmented Lagrangian method. This may explain why even when ADMM type methods suffer from slow local convergence, the latter can still enjoy fast local convergence.

In the following, we compare our symmetric Gauss-Seidel based proximal term $\frac{\sigma}{2}\|y_1 - y_1^k\|^2_{\mathcal{T}_1} + \frac{\sigma}{2}\|y_2 - y_2^k\|^2_{\mathcal{T}_2}$ used to derive the scheme (3.39) for solving (3.32) with the following proximal term which allows one to update $y_1$ and $y_2$ simultaneously:

$$\frac{\sigma}{2}(\|(y_1, y_2) - (y_1^k, y_2^k)\|^2_{\mathcal{M}} + \|y_1 - y_1^k\|^2_{\mathcal{T}_1} + \|y_2 - y_2^k\|^2_{\mathcal{T}_2}) \quad \text{with} \tag{3.40}$$
$$\mathcal{M} = \begin{pmatrix} \mathcal{D}_1 & -\mathcal{A}_1\mathcal{A}_2^* \\ -\mathcal{A}_2\mathcal{A}_1^* & \mathcal{D}_2 \end{pmatrix} \succeq 0,$$

where $\mathcal{D}_1 : \mathcal{Y}_1 \to \mathcal{Y}_1$ and $\mathcal{D}_2 : \mathcal{Y}_2 \to \mathcal{Y}_2$ are two self-adjoint positive semidefinite linear operators satisfying

$$\mathcal{D}_1 \succeq \sqrt{(\mathcal{A}_1\mathcal{A}_2^*)(\mathcal{A}_1\mathcal{A}_2^*)^*} \quad \text{and} \quad \mathcal{D}_2 \succeq \sqrt{(\mathcal{A}_2\mathcal{A}_1^*)(\mathcal{A}_2\mathcal{A}_1^*)^*}\,.$$

A common and naive choice will be $\mathcal{D}_1 = \lambda_{\max}\mathcal{I}_1$ and $\mathcal{D}_2 = \lambda_{\max}\mathcal{I}_2$ where $\lambda_{\max} = \|\mathcal{A}_1\mathcal{A}_2^*\|_2$, $\mathcal{I}_1 : \mathcal{Y}_1 \to \mathcal{Y}_1$ and $\mathcal{I}_2 : \mathcal{Y}_2 \to \mathcal{Y}_2$ are identity maps. By Proposition 2.10, we have that the resulting semi-proximal augmented Lagrangian method generates $(y_1^{k+1}, y_2^{k+1}, x^{k+1})$ as follows:

$$\left\{\begin{array}{lcl} y_1^{k+1} & = & \mathrm{argmin}_{y_1}\ \mathcal{L}_\sigma(y_1, y_2^k; x^k) + \frac{\sigma}{2}\|y_1 - y_1^k\|^2_{\mathcal{D}_1+\mathcal{T}_1}, \\ y_2^{k+1} & = & \mathrm{argmin}_{y_2}\ \mathcal{L}_\sigma(y_1^k, y_2; x^k) + \frac{\sigma}{2}\|y_2 - y_2^k\|^2_{\mathcal{D}_2+\mathcal{T}_2}, \\ x^{k+1} & = & x^k + \tau\sigma(\mathcal{A}_1^* y_1^{k+1} + \mathcal{A}_2^* y_2^{k+1} - c). \end{array}\right. \tag{3.41}$$

To ensure that the subproblems in (3.41) are well defined, we may require the following sufficient conditions to hold:

$$\sigma^{-1}\Sigma_1 + \mathcal{T}_1 + \mathcal{A}_1\mathcal{A}_1^* + \mathcal{D}_1 \succ 0 \quad \text{and} \quad \sigma^{-1}\Sigma_2 + \mathcal{T}_2 + \mathcal{A}_2\mathcal{A}_2^* + \mathcal{D}_2 \succ 0.$$

Comparing the proximal terms used in (3.35) and (3.40), we can easily see that the difference is:

$$\|y_1 - y_1^k\|^2_{\mathcal{A}_1\mathcal{A}_2^*\mathcal{E}_2^{-1}\mathcal{A}_2\mathcal{A}_1^*} \quad \text{vs.} \quad \|(y_1, y_2) - (y_1^k, y_2^k)\|^2_{\mathcal{M}}.$$

To simplify the comparison, we assume that

$$\mathcal{D}_1 = \sqrt{(\mathcal{A}_1\mathcal{A}_2^*)(\mathcal{A}_1\mathcal{A}_2^*)^*} \quad \text{and} \quad \mathcal{D}_2 = \sqrt{(\mathcal{A}_2\mathcal{A}_1^*)(\mathcal{A}_2\mathcal{A}_1^*)^*}\,.$$

By rescaling the equality constraint in (3.32) if necessary, we may also assume that $\|\mathcal{A}_1\| = 1$. Now, we have that

$$\mathcal{A}_1\mathcal{A}_2^*\mathcal{E}_2^{-1}\mathcal{A}_2\mathcal{A}_1^* \preceq \mathcal{A}_1\mathcal{A}_1^*$$

and

$$\|y_1 - y_1^k\|^2_{\mathcal{A}_1\mathcal{A}_2^*\mathcal{E}_2^{-1}\mathcal{A}_2\mathcal{A}_1^*} \;\leq\; \|y_1 - y_1^k\|^2_{\mathcal{A}_1\mathcal{A}_1^*} \leq \|y_1 - y_1^k\|^2.$$

In contrast, we have

$$\|(y_1, y_2) - (y_1^k, y_2^k)\|^2_{\mathcal{M}} \leq 2\left(\|y_1 - y_1^k\|^2_{\mathcal{D}_1} + \|y_2 - y_2^k\|^2_{\mathcal{D}_2}\right)$$

$$\leq \;\; 2\|\mathcal{A}_1\mathcal{A}_2^*\|\left(\|y_1 - y_1^k\|^2 + \|y_2 - y_2^k\|^2\right) \;\leq\; 2\|\mathcal{A}_2\|\left(\|y_1 - y_1^k\|^2 + \|y_2 - y_2^k\|^2\right),$$

which is larger than the former upper bound $\|y_1 - y_1^k\|^2$ if $\|\mathcal{A}_2\| \geq 1/2$. Thus we can conclude safely that the proximal term $\|y_1 - y_2^k\|^2_{\mathcal{A}_1\mathcal{A}_2^*\mathcal{E}_2^{-1}\mathcal{A}_2\mathcal{A}_1^*}$ can be potentially much smaller than $\|(y_1, y_2) - (y_1^k, y_2^k)\|^2_{\mathcal{M}}$ unless $\|\mathcal{A}_2\|$ is very small. In fact, as is already presented in (2.17), for the general multi-block case, one can always design a proximal term $\mathcal{M}$ to obtain an algorithm with a Jacobian type decomposition.

The above mentioned upper bounds difference is of course due to the fact that the sGS semi-proximal augmented Lagrangian method takes advantage of the fact that $\rho$ is assumed to be a convex quadratic function. However, the key difference lies in the fact that (3.41) is a splitting version of the semi-proximal augmented Lagrangian method with a Jacobi type decomposition, whereas Algorithm sGS-sPALM is a splitting version of semi-proximal augmented Lagrangian method with a Gauss-Seidel type decomposition. It is this fact that provides us with the key idea to design symmetric Gauss-Seidel based proximal terms for multi-block composite convex quadratic optimization problems in the next section.

## 3.3 A symmetric Gauss-Seidel based proximal ADMM

Here, we rewrite the general convex composite quadratic optimization model (3.1) in a more compact form:

$$\begin{aligned} \min \quad & \theta(y_1) + f(y) + \varphi(z_1) + g(z) \\ \text{s.t.} \quad & \mathcal{A}^* y + \mathcal{B}^* z = c, \end{aligned} \tag{3.42}$$

where the convex quadratic functions $f : \mathcal{Y} \to \Re$ and $g : \mathcal{Z} \to \Re$ are given by

$$f(y) = \frac{1}{2}\langle y,\, \mathcal{P}y\rangle - \langle b_y,\, y\rangle \quad \text{and} \quad g(z) = \frac{1}{2}\langle z,\, \mathcal{Q}z\rangle - \langle b_z,\, z\rangle$$

with $b_y \in \mathcal{Y}$ and $b_z \in \mathcal{Z}$ as given data. Here, $\mathcal{P}$ and $\mathcal{Q}$ are two self-adjoint positive semidefinite linear operators. For later discussions, we write $\mathcal{P}$ and $\mathcal{Q}$ as follows:

$$\mathcal{P} := \begin{pmatrix} \mathcal{P}_{11} & \mathcal{P}_{12} & \cdots & \mathcal{P}_{1p} \\ \mathcal{P}_{12}^* & \mathcal{P}_{22} & \cdots & \mathcal{P}_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{P}_{1p}^* & \mathcal{P}_{2p}^* & \cdots & \mathcal{P}_{pp} \end{pmatrix} \quad \text{and} \quad \mathcal{Q} := \begin{pmatrix} \mathcal{Q}_{11} & \mathcal{Q}_{12} & \cdots & \mathcal{Q}_{1q} \\ \mathcal{Q}_{12}^* & \mathcal{Q}_{22} & \cdots & \mathcal{Q}_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{Q}_{1q}^* & \mathcal{Q}_{2q}^* & \cdots & \mathcal{Q}_{qq} \end{pmatrix},$$

where $\mathcal{H}_{ij} : \mathcal{Y}_j \to \mathcal{Y}_i$ for $i = 1, \ldots, p,\ j \le i$ and $\mathcal{Q}_{mn} : \mathcal{Z}_n \to \mathcal{Z}_m$ for $m = 1, \ldots, q,\ n \le m$ are linear operators. For notational convenience, we further write

$$\theta_f(y) := \theta(y_1) + f(y) \quad \forall y \in \mathcal{Y} \quad \text{and} \quad \varphi_g(z) := \varphi(z_1) + g(z) \quad \forall z \in \mathcal{Z}. \tag{3.43}$$

Let $\sigma > 0$ be given. The augmented Lagrangian function associated with (3.42) is given as follows:

$$\mathcal{L}_\sigma(y, z; x) = \theta_f(y) + \varphi_g(z) + \langle x,\, \mathcal{A}^* y + \mathcal{B}^* z - c\rangle + \frac{\sigma}{2}\|\mathcal{A}^* y + \mathcal{B}^* z - c\|^2.$$

Recall the majorized ADMM with indefinite proximal terms proposed in [35], when applied to (3.42), has the following template. Note that now since $f$ and $g$ are convex quadratic functions, the majorization step is omitted.

**iPADMM: An ADMM with indefinite proximal terms for solving problem (3.42).**

Let $\sigma > 0$ and $\tau \in (0, \infty)$ be given parameters. Let $\mathcal{M}$ and $\mathcal{N}$ be given self-adjoint, possibly indefinite, linear operators defined on $\mathcal{Y}$ and $\mathcal{Z}$, respectively such that

$$\sigma^{-1}\mathcal{P} + \mathcal{M} + \mathcal{A}\mathcal{A}^* \succeq 0 \quad \text{and} \quad \sigma^{-1}\mathcal{Q} + \mathcal{N} + \mathcal{B}\mathcal{B}^* \succeq 0.$$

Choose $(y^0, z^0, x^0) \in \text{dom}(\theta_f) \times \text{dom}(\varphi_g) \times \mathcal{X}$. For $k = 0, 1, 2, ...$, generate $(y^{k+1}, z^{k+1})$ and $x^{k+1}$ according to the following iteration.

**Step 1.** Compute

$$y^{k+1} = \text{argmin}_y \, \mathcal{L}_\sigma(y, z^k; x^k) + \frac{\sigma}{2}\|y - y^k\|^2_{\mathcal{M}}.$$

**Step 2.** Compute

$$z^{k+1} = \text{argmin}_z \, \mathcal{L}_\sigma(y^{k+1}, z; x^k) + \frac{\sigma}{2}\|z - z^k\|^2_{\mathcal{N}}.$$

**Step 3.** Compute

$$x^{k+1} = x^k + \tau\sigma(\mathcal{A}^* y^{k+1} + \mathcal{B}^* z^{k+1} - c).$$

**Remark 3.8.** In the above iPADMM for solving problem (3.42), the presence of two self-adjoint linear operator $\mathcal{M}$ and $\mathcal{N}$ not only helps to ensure the well-definedness and convergence of the algorithm but also, as will be demonstrated later, is the key for us to use the symmetric Gauss-Seidel idea from the previous section. The general principle is that both $\mathcal{M}$ and $\mathcal{N}$ should be chosen such that $y^{k+1}$ and $z^{k+1}$ take larger step-lengths while they are still relatively easy to compute. From the numerical point of view, it is therefore advantageous to pick indefinite $\mathcal{M}$ and $\mathcal{N}$ whenever possible.

For the convergence and the iteration complexity of the iPADMM, we need the following assumption.

**Assumption 4.** *There exists* $(\hat{y}, \hat{z}) \in \mathrm{ri}(\mathrm{dom}\,\theta_f) \times \mathrm{ri}(\mathrm{dom}\,\varphi_g)$ *such that* $\mathcal{A}^*\hat{y} + \mathcal{B}^*\hat{z} = c$.

We also denote

$$\begin{aligned} &\tilde{x}^{k+1} := x^k + \sigma(\mathcal{A}^* y^{k+1} + \mathcal{B}^* z^{k+1} - c), \quad \hat{x}^k = \frac{1}{k}\sum_{i=1}^{k} \tilde{x}^{i+1}, \\ &\hat{y}^k = \frac{1}{k}\sum_{i=1}^{k} y^{i+1}, \quad \hat{z}^k = \frac{1}{k}\sum_{i=1}^{k} z^{i+1}. \end{aligned} \tag{3.44}$$

Now we are ready to show the global convergence property and the $O(1/k)$ iteration complexity of the iPADMM.

**Theorem 3.9.** *Suppose that the solution set of problem* (3.42) *is nonempty and that Assumption 4 holds. Assume that* $\mathcal{M}$ *and* $\mathcal{N}$ *are chosen such that the sequence* $\{(y^k, z^k, x^k)\}$ *generated by Algorithm iPADMM is well defined. Let* $\tau \in (0, (1+\sqrt{5})/2)$, *if*

$$\frac{1}{2}\sigma^{-1}\mathcal{P} + \mathcal{M} \succeq 0, \quad \frac{1}{2}\sigma^{-1}\mathcal{P} + \mathcal{M} + \mathcal{A}\mathcal{A}^* \succ 0 \tag{3.45}$$

*and*

$$\frac{1}{2}\sigma^{-1}\mathcal{Q} + \mathcal{N} \succeq 0, \quad \frac{1}{2}\sigma^{-1}\mathcal{Q} + \mathcal{N} + \mathcal{B}\mathcal{B}^* \succ 0, \tag{3.46}$$

*we have:*

(a) *The sequence* $\{(y^k, z^k, x^k)\}$ *converges to a unique limit, say,* $(y^\infty, z^\infty, x^\infty)$ *with* $(y^\infty, z^\infty)$ *solving problem* (3.42) *and* $x^\infty$ *solving its dual problem, respectively.*

(b) *For any iteration point* $\{(y^k, z^k, x^k)\}$ *generated by iPADMM,* $(\hat{y}^k, \hat{z}^k, \hat{x}^k)$ *is an approximate solution of the first order optimality condition in variational inequality form with* $O(1/k)$ *iteration complexity.*

**Remark 3.10.** The conclusion of Theorem 3.9 follows essentially from Theorem 2.11 and Theorem 2.12. See [35] for more detailed discussions.

From Remark 3.8, here, we propose to split $\mathcal{M}$ into the sum of two self-adjoint linear operators. In order to take the larger step-length, the first linear operator, denoted by $\mathcal{S}$, is chosen to be indefinite. Meanwhile, the second linear operator is chosen to be positive semidefinite and is specially designed such that the joint minimization subproblem corresponding to $y$ can be decoupled by our symmetric Gauss-Seidel based decomposition technique. Using the similar idea, $\mathcal{N}$ can again be decomposed as the sum of a self-adjoint indefinite linear operator $\mathcal{T}$ and a specially designed self-adjoint positive semidefinite linear operator. In this thesis, to simplify the analysis, we made the following assumption.

**Assumption 5.** *For any given* $\alpha \in [0, \frac{1}{2}]$*, assume*

$$\mathcal{S} = -\sigma^{-1}\alpha\mathcal{P} \quad \text{and} \quad \mathcal{T} = -\sigma^{-1}\alpha\mathcal{Q}.$$

Note that, in this way, the conditions $\frac{1}{2}\sigma^{-1}\mathcal{P} + \mathcal{M} \succeq 0$ and $\frac{1}{2}\sigma^{-1}\mathcal{Q} + \mathcal{N} \succeq 0$ are always guaranteed. Below, we focus on the design of the rest parts of $\mathcal{M}$ and $\mathcal{N}$.

Given $\alpha \in [0, \frac{1}{2}]$, we first define two self-adjoint semidefinite linear operators $\mathcal{S}_1$ and $\mathcal{T}_1$ to handle the convex, possibly nonsmooth, functions $\theta(y_1)$ and $\varphi(z_1)$. Let $\mathcal{E}_{y_1}, \mathcal{S}_1$ be self-adjoint semidefinite linear operators defined on $\mathcal{Y}_1$ such that

$$\mathcal{E}_{y_1} := \mathcal{S}_1 + \sigma^{-1}(1-\alpha)\mathcal{P}_{11} + \mathcal{A}_1\mathcal{A}_1^* \succeq 0, \tag{3.47}$$

and the following well-defined optimization problem can easily be solved

$$\min_{y_1} \; \theta(y_1) + \frac{\sigma}{2}\|y_1 - \bar{y}_1\|^2_{\mathcal{E}_{y_1}}.$$

Similarly, define self-adjoint semidefinite linear operators $\mathcal{E}_{z_1}, \mathcal{T}_1$ on $\mathcal{Z}_1$ such that

$$\mathcal{E}_{z_1} := \mathcal{T}_1 + \sigma^{-1}(1-\alpha)\mathcal{Q}_{11} + \mathcal{B}_1\mathcal{B}_1^* \succeq 0, \tag{3.48}$$

and the optimal solution to the following problem can be easily obtained

$$\min_{z_1} \; \varphi(z_1) + \frac{\sigma}{2}\|z_1 - \bar{z}_1\|^2_{\mathcal{E}_{z_1}}.$$

Then, for $i = 2, \ldots, p$, let $\mathcal{E}_{y_i}$ be a self-adjoint positive definite linear operator on $\mathcal{Y}_i$ such that it is a majorization of $\sigma^{-1}(1-\alpha)\mathcal{P}_{ii} + \mathcal{A}_i\mathcal{A}_i^*$, i.e.,

$$\mathcal{E}_{y_i} \succeq \sigma^{-1}(1-\alpha)\mathcal{P}_{ii} + \mathcal{A}_i\mathcal{A}_i^*.$$

In practice, we would choose $\mathcal{E}_{y_i}$ in such a way that its inverse can be computed at a moderate cost. Define

$$\mathcal{S}_i := \mathcal{E}_{y_i} - \sigma^{-1}(1-\alpha)\mathcal{P}_{ii} - \mathcal{A}_i\mathcal{A}_i^* \succeq 0, \quad i = 1, \ldots, p. \tag{3.49}$$

Note that for numerical efficiency, we need the self-adjoint positive semidefinite linear operator $\mathcal{S}_i$ to be as small as possible for each $i = 1, \ldots, p$. Similarly, for $j = 2, \ldots, q$, let $\mathcal{E}_{z_j}$ be a self-adjoint positive definite linear operator on $\mathcal{Z}_j$ that majorizes $\sigma^{-1}(1-\alpha)\mathcal{Q}_{jj} + \mathcal{B}_j\mathcal{B}_j^*$ in such a way that $\mathcal{E}_{z_j}^{-1}$ can be computed relatively easily. Define

$$\mathcal{T}_j := \mathcal{E}_{z_j} - \sigma^{-1}(1-\alpha)\mathcal{Q}_{jj} - \mathcal{B}_j\mathcal{B}_j^* \succeq 0, \quad j = 1, \ldots, q. \tag{3.50}$$

Again, we need the self-adjoint positive semidefinite linear operator $\mathcal{T}_j$ to be as small as possible for each $j = 1, \ldots, q$.

Now we are ready to present our sGS-PADMM (symmetric Gauss-Seidel based proximal alternating direction method of multipliers) algorithm for solving (3.42).

**Algorithm sGS-PADMM: A symmetric Gauss-Seidel based proximal ADMM for solving** (3.42)**.** Let $\sigma > 0$ and $\tau \in (0, \infty)$ be given parameters. Choose $(y^0, z^0, x^0) \in \mathrm{dom}(\theta_f) \times \mathrm{dom}(\varphi_g) \times \mathcal{X}$. For $k = 0, 1, 2, ...$, generate $(y^{k+1}, z^{k+1})$ and $x^{k+1}$ according to the following iteration.

**Step 1.** (Backward GS sweep) Compute for $i = p, \dots, 2$,

$$\overline{y}_i^k = \mathrm{argmin}_{y_i} \left\{ \begin{array}{l} \mathcal{L}_\sigma((y^k_{\le i-1}, y_i, \overline{y}^k_{\ge i+1}), z^k; x^k) \\ +\frac{\sigma}{2}\|(y^k_{\le i-1}, y_i, \overline{y}^k_{\ge i+1}) - y^k\|^2_{\mathcal{S}} + \frac{\sigma}{2}\|y_i - y_i^k\|^2_{\mathcal{S}_i} \end{array} \right\}.$$

Then compute

$$y_1^{k+1} = \mathrm{argmin}_{y_1} \left\{ \begin{array}{l} \mathcal{L}_\sigma((y_1, \overline{y}^k_{\ge 2}), z^k; x^k) + \frac{\sigma}{2}\|(y_1, \overline{y}^k_{\ge 2}) - y^k\|^2_{\mathcal{S}} \\ +\frac{\sigma}{2}\|y_1 - y_1^k\|^2_{\mathcal{S}_1} \end{array} \right\}.$$

**Step 2.** (Forward GS sweep) Compute for $i = 2, \dots, p$,

$$y_i^{k+1} = \mathrm{argmin}_{y_i} \left\{ \begin{array}{l} \mathcal{L}_\sigma((y^{k+1}_{\le i-1}, y_i, \overline{y}^k_{\ge i+1}), z^k; x^k) \\ +\frac{\sigma}{2}\|(y^{k+1}_{\le i-1}, y_i, \overline{y}^k_{\ge i+1}) - y^k\|^2_{\mathcal{S}} + \frac{\sigma}{2}\|y_i - y_i^k\|^2_{\mathcal{S}_i} \end{array} \right\}.$$

**Step 3.** (Backward GS sweep) Compute for $j = q, \dots, 2$,

$$\overline{z}_j^k = \mathrm{argmin}_{z_j} \left\{ \begin{array}{l} \mathcal{L}_\sigma(y^{k+1}, (z^k_{\le j-1}, z_j, \overline{z}^k_{\ge j+1}); x^k) \\ +\frac{\sigma}{2}\|(z^k_{\le j-1}, z_j, \overline{z}^k_{\ge j+1}) - z^k\|^2_{\mathcal{T}} + \frac{\sigma}{2}\|z_j - z_j^k\|^2_{\mathcal{T}_j} \end{array} \right\}.$$

Then compute

$$z_1^{k+1} = \mathrm{argmin}_{z_1} \left\{ \begin{array}{l} \mathcal{L}_\sigma(y^{k+1}, (z_1, \overline{z}^k_{\ge 2}); x^k) \\ +\frac{\sigma}{2}\|(z_1, \overline{z}^k_{\ge 2}) - z^k\|^2_{\mathcal{T}} + \frac{\sigma}{2}\|z_1 - z_1^k\|^2_{\mathcal{T}_1} \end{array} \right\}.$$

**Step 4.** (Forward GS sweep) Compute for $j = 2, \dots, q$,

$$z_j^{k+1} = \mathrm{argmin}_{z_j} \left\{ \begin{array}{l} \mathcal{L}_\sigma(y^{k+1}, (z^{k+1}_{\le j-1}, z_j, \overline{z}^k_{\ge j+1}); x^k) \\ +\frac{\sigma}{2}\|(z^{k+1}_{\le j-1}, z_j, \overline{z}^k_{\ge j+1})) - z^k\|^2_{\mathcal{T}} + \frac{\sigma}{2}\|z_j - z_j^k\|^2_{\mathcal{T}_j} \end{array} \right\}.$$

**Step 5.** Compute

$$x^{k+1} = x^k + \tau\sigma(\mathcal{A}^* y^{k+1} + \mathcal{B}^* z^{k+1} - c).$$

In order to prove the convergence of Algorithm sGS-PADMM for solving (3.42), we need first to study the relationship between sGS-PADMM and the generic 2-block iPADMM for solving a two-block convex optimization problem.

For given $\alpha \in [0, \frac{1}{2}]$, define the following linear operators:

$$\mathcal{M}_i := \sigma^{-1}(1-\alpha) \begin{pmatrix} \mathcal{P}_{1i} \\ \vdots \\ \mathcal{P}_{(i-1)i} \end{pmatrix} + \begin{pmatrix} \mathcal{A}_1 \\ \vdots \\ \mathcal{A}_{i-1} \end{pmatrix} \mathcal{A}_i^*, \quad i = 2, \ldots, p.$$

Similarly, let

$$\mathcal{N}_j := \sigma^{-1}(1-\alpha) \begin{pmatrix} \mathcal{Q}_{1j} \\ \vdots \\ \mathcal{Q}_{(j-1)j} \end{pmatrix} + \begin{pmatrix} \mathcal{B}_1 \\ \vdots \\ \mathcal{B}_{i-1} \end{pmatrix} \mathcal{B}_j^*, \quad j = 2, \ldots, q.$$

For the given self-adjoint semidefinite linear operators $\mathcal{S}_1$ and $\mathcal{T}_1$, define $\widehat{\mathcal{S}}_2 := \mathcal{S}_1 + \mathcal{M}_2 \mathcal{E}_2^{-1} \mathcal{M}_2^*$,

$$\widehat{\mathcal{S}}_i := \operatorname{diag}(\widehat{\mathcal{S}}_{i-1}, \mathcal{S}_{i-1}) + \mathcal{M}_i \mathcal{E}_{y_i}^{-1} \mathcal{M}_i^*, \qquad i = 3, \ldots, p$$

and $\widehat{\mathcal{T}}_2 := \mathcal{T}_1 + \mathcal{N}_2 \mathcal{E}_{z_2}^{-1} \mathcal{N}_2^*$,

$$\widehat{\mathcal{T}}_j := \operatorname{diag}(\widehat{\mathcal{T}}_{j-1}, \mathcal{T}_{j-1}) + \mathcal{N}_j \mathcal{E}_{z_j}^{-1} \mathcal{N}_j^*, \qquad j = 3, \ldots, q.$$

**Proposition 3.11.** *For any $k \geq 0$, the point $(x^{k+1}, y^{k+1}, z^{k+1})$ obtained by Algorithm sGS-PADMM for solving problem* (3.42) *can be generated exactly according to the following iteration:*

$$\begin{aligned} y^{k+1} &= \operatorname{argmin}_y \left\{ \begin{array}{l} \mathcal{L}_\sigma(y, z^k; x^k) + \frac{\sigma}{2}\|y - y^k\|_{\mathcal{S}}^2 \\ + \frac{\sigma}{2}\|y_{\leq p-1} - y_{\leq p-1}^k\|_{\widehat{\mathcal{S}}_p}^2 + \frac{\sigma}{2}\|y_p - y_p^k\|_{\mathcal{S}_p}^2 \end{array} \right\}, \qquad (3.51) \\ z^{k+1} &= \operatorname{argmin}_z \left\{ \begin{array}{l} \mathcal{L}_\sigma(y^{k+1}, z; x^k) + \frac{\sigma}{2}\|z - z^k\|_{\mathcal{T}}^2 \\ + \frac{\sigma}{2}\|z_{\leq q-1} - z_{\leq q-1}^k\|_{\widehat{\mathcal{T}}_q}^2 + \frac{\sigma}{2}\|z_q - z_q^k\|_{\mathcal{T}_q}^2 \end{array} \right\}, \qquad (3.52) \\ x^{k+1} &= x^k + \tau\sigma(\mathcal{A}^* y^{k+1} + \mathcal{B}^* z^{k+1} - c). \end{aligned}$$

**Proof.** We only need to prove the $y^{k+1}$ part as the $z^{k+1}$ part can be obtained in the similar manner. Let

$$\Delta\mathcal{S}_p := \widehat{\mathcal{S}}_p - \mathrm{diag}(\mathcal{S}_1, \ldots, \mathcal{S}_{p-1}).$$

Note that problem (3.51) can equivalently be rewritten as

$$y^{k+1} = \mathrm{argmin}_y \left\{ \begin{array}{l} \mathcal{L}_\sigma(y, z^k; x^k) + \frac{\sigma}{2}\|y_1 - y_1^k\|^2_{\mathcal{S}_1} + \frac{\sigma}{2}\sum_{i=2}^p \|y_i - y_i^k\|^2_{\mathcal{S}_i} \\ +\frac{\sigma}{2}\|y - y^k\|^2_{\mathcal{S}_0} + \frac{\sigma}{2}\|y_{\leq p-1} - y^k_{\leq p-1}\|^2_{\Delta\mathcal{S}_p} \end{array} \right\}. \quad (3.53)$$

The equivalence then follows directly by applying Theorem 3.3 with all the error tolerance vectors $(\delta^+, \delta'_{\geq 2})$ chosen to be zero for problem (3.53). The proof of this proposition is completed. □

**Remark 3.12.** Note that in the proof for Proposition 3.11, all the error tolerance vectors $(\delta^+, \delta'_{\geq 2})$ are set to zero. Naturally, one may ask the following question: Why these error tolerance vectors are included in Theorem 3.3? As can be seen later, these error terms play important roles in the designing of a special inexact accelerated proximal gradient (APG) algorithm in Phase II. In fact, these error tolerance vectors also open up many possibilities of designing inexact ADMM type methods which will allow the inexact solution for each subproblem and have attainable stopping conditions.

In fact, we have finished the design of $\mathcal{M}$ and $\mathcal{N}$. From Proposition 3.11, we have

$$\mathcal{M} = -\sigma^{-1}\alpha\mathcal{P} + \mathrm{diag}(\widehat{\mathcal{S}}_p, \mathcal{S}_p) \quad (3.54)$$

and

$$\mathcal{N} = -\sigma^{-1}\alpha\mathcal{Q} + \mathrm{diag}(\widehat{\mathcal{T}}_p, \mathcal{T}_p). \quad (3.55)$$

Next, we study the conditions which will guarantee the convergence of our proposed Algorithm sGS-PADMM.

In order to prove the convergence of Algorithm sGS-PADMM for solving problem (3.42), the following proposition is needed.

**Proposition 3.13.** *For any given* $\alpha \in [0, \frac{1}{2})$*, it holds that*

$$\begin{aligned} & \mathcal{A}\mathcal{A}^* + \sigma^{-1}(\tfrac{1}{2} - \alpha)\mathcal{P} + \mathrm{diag}(\widehat{\mathcal{S}}_p, \mathcal{S}_p) \succ 0 \\ \Leftrightarrow \quad & \mathcal{A}_1\mathcal{A}_1^* + \sigma^{-1}(1-\alpha)\mathcal{P}_{11} + \mathcal{S}_1 \succ 0, \end{aligned} \tag{3.56}$$

$$\begin{aligned} & \mathcal{B}\mathcal{B}^* + \sigma^{-1}(\tfrac{1}{2} - \alpha)\mathcal{Q} + \mathrm{diag}(\widehat{\mathcal{T}}_q, \mathcal{T}_q) \succ 0 \\ \Leftrightarrow \quad & \mathcal{B}_1\mathcal{B}_1^* + \sigma^{-1}(1-\alpha)\mathcal{Q}_{11} + \mathcal{T}_1 \succ 0. \end{aligned} \tag{3.57}$$

**Proof.** Note the fact that if $\mathcal{A}$ and $\mathcal{B}$ are two positive semidefinite linear operators, then

$$\begin{aligned} & (\forall \alpha_1 > 0, \alpha_2 > 0) \quad \alpha_1\mathcal{A} + \alpha_2\mathcal{B} \succ 0 \\ \Leftrightarrow \quad & (\exists \alpha_1 > 0, \alpha_2 > 0) \quad \alpha_1\mathcal{A} + \alpha_2\mathcal{B} \succ 0 \\ \Leftrightarrow \quad & \mathcal{A} + \mathcal{B} \succ 0. \end{aligned}$$

Hence, to prove (3.56) and (3.57), we only need to prove

$$\left\{ \begin{array}{l} \mathcal{A}\mathcal{A}^* + \sigma^{-1}(1-\alpha)\mathcal{P} + \mathrm{diag}(\widehat{\mathcal{S}}_p, \mathcal{S}_p) \succ 0 \Leftrightarrow \mathcal{A}_1\mathcal{A}_1^* + \sigma^{-1}(1-\alpha)\mathcal{P}_{11} + \mathcal{S}_1 \succ 0, \\ \mathcal{B}\mathcal{B}^* + \sigma^{-1}(1-\alpha)\mathcal{Q} + \mathrm{diag}(\widehat{\mathcal{T}}_q, \mathcal{T}_q) \succ 0 \Leftrightarrow \mathcal{B}_1\mathcal{B}_1^* + \sigma^{-1}(1-\alpha)\mathcal{Q}_{11} + \mathcal{T}_1 \succ 0. \end{array} \right. \tag{3.58}$$

Note that (3.58) can be readily obtained by using part (ii) of Theorem 3.3. Thus, we prove the proposition. □

After all these preparations, we can finally state our main convergence theorem.

**Theorem 3.14.** *Suppose that the solution set of problem* (3.42) *is nonempty and that Assumption 4 and 5 hold. Assume that the sequence* $\{(y^k, z^k, x^k)\}$ *generated by Algorithm sGS-PADMM is well defined. Let* $\tau \in (0, (1+\sqrt{5})/2)$*. Then, the following conclusion holds:*

(a) *For* $\alpha \in [0, 1/2)$*, under the condition that*

$$\mathcal{A}_1\mathcal{A}_1^* + \sigma^{-1}(1-\alpha)\mathcal{P}_{11} + \mathcal{S}_1 \succ 0 \quad \text{and} \quad \mathcal{B}_1\mathcal{B}_1^* + \sigma^{-1}(1-\alpha)\mathcal{Q}_{11} + \mathcal{T}_1 \succ 0,$$

*the sequence* $\{(y^k, z^k)\}$*, which is automatically well defined, converges to an optimal solution of problem* (3.42) *and* $\{x^k\}$ *converges to an optimal solution of the corresponding dual problem, respectively.*

(b) *For $\alpha = \frac{1}{2}$, under the condition that*

$$\mathcal{A}\mathcal{A}^* + \mathrm{diag}(\widehat{\mathcal{S}}_p, \mathcal{S}_p) \succ 0 \quad \text{and} \quad \mathcal{B}\mathcal{B}^* + \mathrm{diag}(\widehat{\mathcal{T}}_q, \mathcal{T}_q) \succ 0,$$

*the sequence $\{(y^k, z^k)\}$, which is automatically well defined, converges to an optimal solution of problem (3.42) and $\{x^k\}$ converges to an optimal solution of the corresponding dual problem, respectively.*

**Proof.** Note that, conditions (3.45) and (3.46) now become

$$\begin{cases} \mathcal{A}\mathcal{A}^* + \sigma^{-1}(\frac{1}{2} - \alpha)\mathcal{P} + \mathrm{diag}(\widehat{\mathcal{S}}_p, \mathcal{S}_p) \succ 0, \\ \mathcal{B}\mathcal{B}^* + \sigma^{-1}(\frac{1}{2} - \alpha)\mathcal{Q} + \mathrm{diag}(\widehat{\mathcal{T}}_q, \mathcal{T}_q) \succ 0. \end{cases} \tag{3.59}$$

When $\alpha \in [0, \frac{1}{2})$, by Proposition 3.13, conditions (3.59) are equivalent to

$$\mathcal{A}_1\mathcal{A}_1^* + \sigma^{-1}(1 - \alpha)\mathcal{P}_{11} + \mathcal{S}_1 \succ 0 \quad \text{and} \quad \mathcal{B}_1\mathcal{B}_1^* + \sigma^{-1}(1 - \alpha)\mathcal{Q}_{11} + \mathcal{T}_1 \succ 0.$$

On the other hand, if $\alpha = \frac{1}{2}$, conditions (3.59) reduce to

$$\mathcal{A}\mathcal{A}^* + \mathrm{diag}(\widehat{\mathcal{S}}_p, \mathcal{S}_p) \succ 0 \quad \text{and} \quad \mathcal{B}\mathcal{B}^* + \mathrm{diag}(\widehat{\mathcal{T}}_q, \mathcal{T}_q) \succ 0.$$

Then by combing part (a) of Theorem 3.9 with Proposition 3.11, we can readily obtain the conclusions of this theorem. □

In the next theorem, we shall show that the sGS-PADMM for solving problem (3.42) has $O(1/k)$ ergodic iteration complexity.

**Theorem 3.15.** *Suppose that Assumption 4 holds. For $\tau \in (0, \frac{1+\sqrt{5}}{2})$, under the same conditions in Theorem 3.14, we have that for any iteration point $\{(y^k, z^k, x^k)\}$ generated by sGS-PADMM, $(\hat{y}^k, \hat{z}^k, \hat{x}^k)$ is an approximate solution of the first order optimality condition in variational inequality form with $O(1/k)$ iteration complexity.*

**Proof.** By by combing part (b) of Theorem 3.9 with Proposition 3.11, we know that the conclusion of this theorem holds. □

## 3.4 Numerical results and examples

Recall the definitions of $\theta_f(\cdot)$ and $\varphi_g(\cdot)$ in (3.43), our general convex quadratic composite optimization model can be recast as

$$\begin{aligned} \min \quad & \theta_f(y) + \varphi_g(z) \\ \text{s.t.} \quad & \mathcal{A}^* y + \mathcal{B}^* z = c \end{aligned} \tag{3.60}$$

and its dual is given by

$$\max \big\{ -\langle c,\, x\rangle - \theta_f^*(-\mathcal{A}x) - \varphi_g^*(-\mathcal{B}x) \big\}. \tag{3.61}$$

We first examine the optimality condition for the general problem (3.60) and its dual (3.61). Suppose that the solution set of problem (3.60) is nonempty and that Assumption 4 holds. Then in order that $(y^*, z^*)$ be an optimal solution for (3.60) and $x^*$ be an optimal solution for (3.60), it is necessary and sufficient that $(y^*, z^*)$ and $x^*$ satisfy

$$\begin{cases} \mathcal{A}^* y + \mathcal{B}^* z = c, \\ \theta_f(y) + \theta_f^*(-\mathcal{A}x) = \langle y,\, -\mathcal{A}x\rangle, \\ \varphi_g(z) + \varphi_g^*(-\mathcal{B}x) = \langle z,\, -\mathcal{B}x\rangle. \end{cases} \tag{3.62}$$

We will measure the accuracy of an approximate solution based on the above optimality condition. If the given problem is properly scaled, the following relative residual is a natural choice to be used in our stopping criterion:

$$\eta = \max\{\eta_P, \eta_{\theta_f}, \eta_{\varphi_g}\}, \tag{3.63}$$

where

$$\begin{aligned} \eta_P &= \frac{\|\mathcal{A}^* y + \mathcal{B}^* z - c\|}{1 + \|c\|}, \\ \eta_{\theta_f} &= \frac{\|y - \mathrm{prox}_{\theta_f}(y - \mathcal{A}x)\|}{1 + \|y\| + \|\mathcal{A}x\|}, \\ \eta_{\varphi_g} &= \frac{\|z - \mathrm{prox}_{\varphi_g}(z - \mathcal{B}x)\|}{1 + \|z\| + \|\mathcal{B}x\|}. \end{aligned}$$

Additionally, we compute the relative gap by

$$\eta_{\text{gap}} = \frac{\text{obj}_P - \text{obj}_D}{1 + |\text{obj}_P| + |\text{obj}_D|},$$

where $\text{obj}_P := \theta(y_1) + f(y) + \varphi(z_1) + g(z)$ and $\text{obj}_D := -\langle c,\, x\rangle - \theta_f^*(-\mathcal{A}x) - \varphi_g^*(-\mathcal{B}x)$. In order to demonstrate the efficiency of our proposed algorithms in Phase I, we test the following problem sets. Note that, for simplicity, we set $\alpha = 0$ in our Algorithm sGS-padmm, i.e., we add only semidefinite proximal terms.

### 3.4.1 Convex quadratic semidefinite programming (QSDP)

As a very important example of the convex composite quadratic optimization problems, in this subsection, we consider the following convex quadratic semidefinite programming problem:

$$\begin{array}{ll} \min & \frac{1}{2}\langle X,\, \mathcal{Q}X\rangle + \langle C,\, X\rangle \\ \text{s.t.} & \mathcal{A}_E X = b_E, \quad \mathcal{A}_I X \geq b_I, \quad X \in \mathcal{S}^n_+ \cap \mathcal{K}, \end{array} \tag{3.64}$$

where $\mathcal{Q}$ is a self-adjoint positive semidefinite linear operator from $\mathcal{S}^n$ to $\mathcal{S}^n$, $\mathcal{A}_E : \mathcal{S}^n \to \Re^{m_E}$ and $\mathcal{A}_I : \mathcal{S}^n \to \Re^{m_I}$ are two linear maps, $C \in \mathcal{S}^n$, $b_E \in \Re^{m_E}$ and $b_I \in \Re^{m_I}$ are given data, $\mathcal{K}$ is a nonempty simple closed convex set, e.g., $\mathcal{K} = \{X \in \mathcal{S}^n \mid L \leq X \leq U\}$ with $L, U \in \mathcal{S}^n$ being given matrices. The dual problem associated with (3.64) is given by

$$\begin{array}{ll} \max & -\delta^*_{\mathcal{K}}(-Z) - \frac{1}{2}\langle X',\, \mathcal{Q}X'\rangle + \langle b_E,\, y_E\rangle + \langle b_I,\, y_I\rangle \\ \text{s.t.} & Z - \mathcal{Q}X' + S + \mathcal{A}_E^* y_E + \mathcal{A}_I^* y_I = C, \\ & X' \in \mathcal{S}^n, \quad y_I \geq 0, \quad S \in \mathcal{S}^n_+\,. \end{array} \tag{3.65}$$

We use $X'$ here to indicate the fact that $X'$ can be different from the primal variable $X$. Despite this fact, we have that at the optimal point, $\mathcal{Q}X = \mathcal{Q}X'$. Since $\mathcal{Q}$ is only assumed to be a self-adjoint positive semidefinite linear operator, the augmented Lagrangian function associated with (3.65) may not be strongly convex with respect to $X'$. Without further adding a proximal term, we propose the following strategy to rectify this difficulty. Since $\mathcal{Q}$ is positive semidefinite, $\mathcal{Q}$ can be decomposed as $\mathcal{Q} = \mathcal{B}^*\mathcal{B}$ for some linear map $\mathcal{B}$. By introducing a new variable $\Xi = -\mathcal{B}X'$, the problem (3.65) can be rewritten as follows:

$$\begin{array}{ll} \max & -\delta_{\mathcal{K}}^*(-Z) - \frac{1}{2}\|\Xi\|_F^2 + \langle b_E,\, y_E\rangle + \langle b_I,\, y_I\rangle \\ \text{s.t.} & Z + \mathcal{B}^*\Xi + S + \mathcal{A}_E^* y_E + \mathcal{A}_I^* y_I = C, \quad y_I \geq 0, \quad S \in \mathcal{S}_+^n\,. \end{array} \tag{3.66}$$

Note that now the augmented Lagrangian function associated with (3.66) is strongly convex with respect to $\Xi$. Surprisingly, much to our delight, we can update the iterations in our sGS-PADMM without explicitly computing $\mathcal{B}$ or $\mathcal{B}^*$. Given $\overline{Z}, \bar{y}_I, \overline{S}, \bar{y}_E$ and $\overline{X}$, denote

$$\begin{array}{rcl} \Xi^+ & := & \text{argmin}_{\Xi}\ \dfrac{1}{2}\|\Xi\|^2 + \dfrac{\sigma}{2}\|\overline{Z} + \mathcal{A}_I^*\bar{y}_I + \mathcal{B}^*\Xi + \overline{S} + \mathcal{A}_E^*\bar{y}_E - C + \sigma^{-1}\overline{X}\|^2 \\ & = & -(\mathcal{I} + \sigma\mathcal{B}\mathcal{B}^*)^{-1}\mathcal{B}\overline{R}, \end{array}$$

where $\overline{R} = \overline{X} + \sigma(\overline{Z} + \mathcal{A}_I^*\bar{y}_I + \overline{S} + \mathcal{A}_E^*\bar{y}_E - C)$. In updating the sGS-PADMM iterations, we actually do not need $\Xi^+$ explicitly, but only need $\Upsilon^+ := -\mathcal{B}^*\Xi^+$. From the condition that $(\mathcal{I} + \sigma\mathcal{B}\mathcal{B}^*)(-\Xi^+) = \mathcal{B}\overline{R}$, we get $(\mathcal{I} + \sigma\mathcal{B}^*\mathcal{B})(-\mathcal{B}^*\Xi^+) = \mathcal{B}^*\mathcal{B}\overline{R}$. Hence we can compute $\Upsilon^+$ via $\mathcal{Q}$:

$$\Upsilon^+ = (\mathcal{I} + \sigma\mathcal{Q})^{-1}(\mathcal{Q}\overline{R}).$$

In fact, $\Upsilon := -\mathcal{B}^*\Xi$ can be viewed as the shadow of $\mathcal{Q}X'$. Meanwhile, for the function $\delta_{\mathcal{K}}^*(-Z)$, we have the following useful observation that for any $\lambda > 0$,

$$Z^+ = \text{argmin}\ \delta_{\mathcal{K}}^*(-Z) + \frac{\lambda}{2}\|Z - \overline{Z}\|^2 = \overline{Z} + \frac{1}{\lambda}\Pi_{\mathcal{K}}(-\lambda\overline{Z}), \tag{3.67}$$

where (3.67) follows from Proposition 2.6.

Here, in our numerical experiments, we test QSDP problems without inequality constraints (i.e., $\mathcal{A}_I$ and $b_I$ are vacuous). We consider first the linear operator $\mathcal{Q}$ given by

$$\mathcal{Q}(X) = \frac{1}{2}(BX + XB) \tag{3.68}$$

for a given matrix $B \in \mathcal{S}^n_+$. Suppose that we have the eigenvalue decomposition $B = P\Lambda P^T$, where $\Lambda = \mathrm{diag}(\lambda)$ and $\lambda = (\lambda_1, \ldots, \lambda_n)^T$ is the vector of eigenvalues of $B$. Then

$$\begin{aligned}\langle X,\, \mathcal{Q}X\rangle &= \frac{1}{2}\langle \widehat{X},\, \Lambda\widehat{X} + \widehat{X}\Lambda\rangle = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\widehat{X}_{ij}^2(\lambda_i + \lambda_j)\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\widehat{X}_{ij}^2 H_{ij}^2 = \langle X,\, \mathcal{B}^*\mathcal{B}X\rangle,\end{aligned}$$

where $\widehat{X} = P^T X P$, $H_{ij} = \sqrt{\frac{\lambda_i+\lambda_j}{2}}$, $\mathcal{B}X = H \circ (P^T X P)$ and $\mathcal{B}^*\Xi = P(H \circ \Xi)P^T$. In our numerical experiments, the matrix $B$ is a low rank random symmetric positive semidefinite matrix. Note that when $\mathrm{rank}(B) = 0$ and $\mathcal{K}$ is a polyhedral cone, problem (3.64) reduces to the SDP problem considered in [59]. In our experiments, we test both of the cases where $\mathrm{rank}(B) = 5$ and $\mathrm{rank}(B) = 10$. All the linear constraints are extracted from the numerical test examples in [59] (Section 4.1). More specifically, we construct the following problem sets:

(i) The QSDP-BIQ problem is given by:

$$\begin{aligned} \min\quad & \tfrac{1}{2}\langle X,\, \mathcal{Q}X\rangle + \tfrac{1}{2}\langle Q,\, X_0\rangle + \langle c,\, x\rangle \\ \text{s.t.}\quad & \mathrm{diag}(X_0) - x = 0, \quad \alpha = 1, \\ & X = \begin{pmatrix} X_0 & x \\ x^T & \alpha \end{pmatrix} \in \mathcal{S}^n_+, \quad X \in \mathcal{K} := \{X \in \mathcal{S}^n \mid X \geq 0\}. \end{aligned} \tag{3.69}$$

In our numerical experiments, the test data for $Q$ and $c$ are taken from Biq Mac Library maintained by Wiegele, which is available at `http://biqmac.uni-klu.ac.at/biqmaclib.html`.

(ii) Given a graph $G$ with edge set $\mathcal{E}$, the QSDP-$\theta_+$ problem is constructed by:

$$\begin{aligned} \min\quad & \tfrac{1}{2}\langle X,\, \mathcal{Q}X\rangle - \langle ee^T,\, X\rangle \\ \text{s.t.}\quad & \langle E_{ij},\, X\rangle = 0,\ (i,j) \in \mathcal{E}, \quad \langle I,\, X\rangle = 1, \\ & X \in \mathcal{S}^n_+, \quad X \in \mathcal{K} := \{X \in \mathcal{S}^n \mid X \geq 0\}, \end{aligned} \tag{3.70}$$

where $E_{ij} = e_i e_j^T + e_j e_i^T$ and $e_i$ denotes the $i$th column of the identity matrix. In our numerical experiments, we test the graph instances $G$ considered in [57, 64, 39].

(iii) The QSDP-RCP problem is constructed based on the formula presented in [48, eq. (13)] as following:

$$\begin{aligned} \min \quad & \tfrac{1}{2}\langle X,\, \mathcal{Q}X\rangle - \langle W,\, X\rangle \\ \text{s.t.} \quad & Xe = e,\ \langle I,\, X\rangle = K, \\ & X \in \mathcal{S}^n_+, \quad X \in \mathcal{K} := \{X \in \mathcal{S}^n \mid X \geq 0\}, \end{aligned} \tag{3.71}$$

where $W$ is the so-called affinity matrix whose entries represent the similarities of the objects in the dataset, $e$ is the vector of ones, and $K$ is the number of clusters. All the data sets we tested are from the UCI Machine Learning Repository (available at `http://archive.ics.uci.edu/ml/datasets.html`). For some large data instances, we only select the first $n$ rows. For example, the original data instance "`spambase`" has 4601 rows, we select the first 1500 rows to obtain the test problem "`spambase-large.2`" for which the number "2" means that there are $K = 2$ clusters.

Here we compare our algorithm sGS-PADMM with the directly extended ADMM (with step length $\tau = 1$) and the convergent alternating direction method with a Gaussian back substitution proposed in [24] (we call the method ADMMGB here and use the parameter $\alpha = 0.99$ in the Gaussian back substitution step). We have implemented all the algorithms sGS-PADMM, ADMM and ADMMGB in MATLAB version 7.13. The numerical results reported later are obtained from a PC with 24 GB memory and 2.80GHz dual-core CPU running on 64-bit Windows Operating System.

We measure the accuracy of an approximate optimal solution $(X, Z, \Xi, S, y_E)$ for QSDP (3.64) and its dual (3.66) by using the following relative residual obtained from the general optimality condition (3.62):

$$\eta_{\text{qsdp}} = \max\{\eta_P, \eta_D, \eta_Z, \eta_{S_1}, \eta_{S_2}\}, \tag{3.72}$$

where

$$\eta_P = \frac{\|\mathcal{A}_E X - b_E\|}{1 + \|b_E\|}, \quad \eta_D = \frac{\|Z + \mathcal{B}^*\Xi + S + \mathcal{A}_E^* y_E - C\|}{1 + \|C\|}, \quad \eta_Z = \frac{\|X - \Pi_{\mathcal{K}}(X - Z)\|}{1 + \|X\| + \|Z\|},$$

$$\eta_{S_1} = \frac{|\langle S,\, X\rangle|}{1 + \|S\| + \|X\|}, \quad \eta_{S_2} = \frac{\|X - \Pi_{\mathcal{S}_+^n}(X)\|}{1 + \|X\|}.$$

We terminate the solvers sGS-PADMM, ADMM and ADMMGB when $\eta_{\rm qsdp} < 10^{-6}$ with the maximum number of iterations set at 25000.

Table 3.1 reports detailed numerical results for sGS-PADMM, ADMM and ADMMGB in solving some large scale QSDP problems. Here, we only list the results for the case of $\text{rank}(B) = 10$, since we obtain similar results for the case of $\text{rank}(B) = 5$. Our numerical experience also indicates that the order of solving the subproblems has generally no influence on the performance of sGS-PADMM . From the numerical results, one can observe that sGS-PADMM is generally the fastest in terms of the computing time, especially when the problem size is large. In addition, we can see that sGS-PADMM and ADMM solved all instances to the required accuracy, while ADMMGB failed in certain cases.

Figure 3.1 shows the performance profiles in terms of the number of iterations and computing time for sGS-PADMM, ADMM and ADMMGB, for all the tested large scale QSDP problems. We recall that a point $(x, y)$ is in the performance profiles curve of a method if and only if it can solve $(100y)\%$ of all the tested problems no slower than $x$ times of any other methods. We may observe that for the majority of the tested problems, sGS-PADMM takes the least number of iterations. Besides, in terms of computing time, it can be seen that both sGS-PADMM and ADMM outperform ADMMGB by a significant margin, even though ADMM has no convergence guarantee.

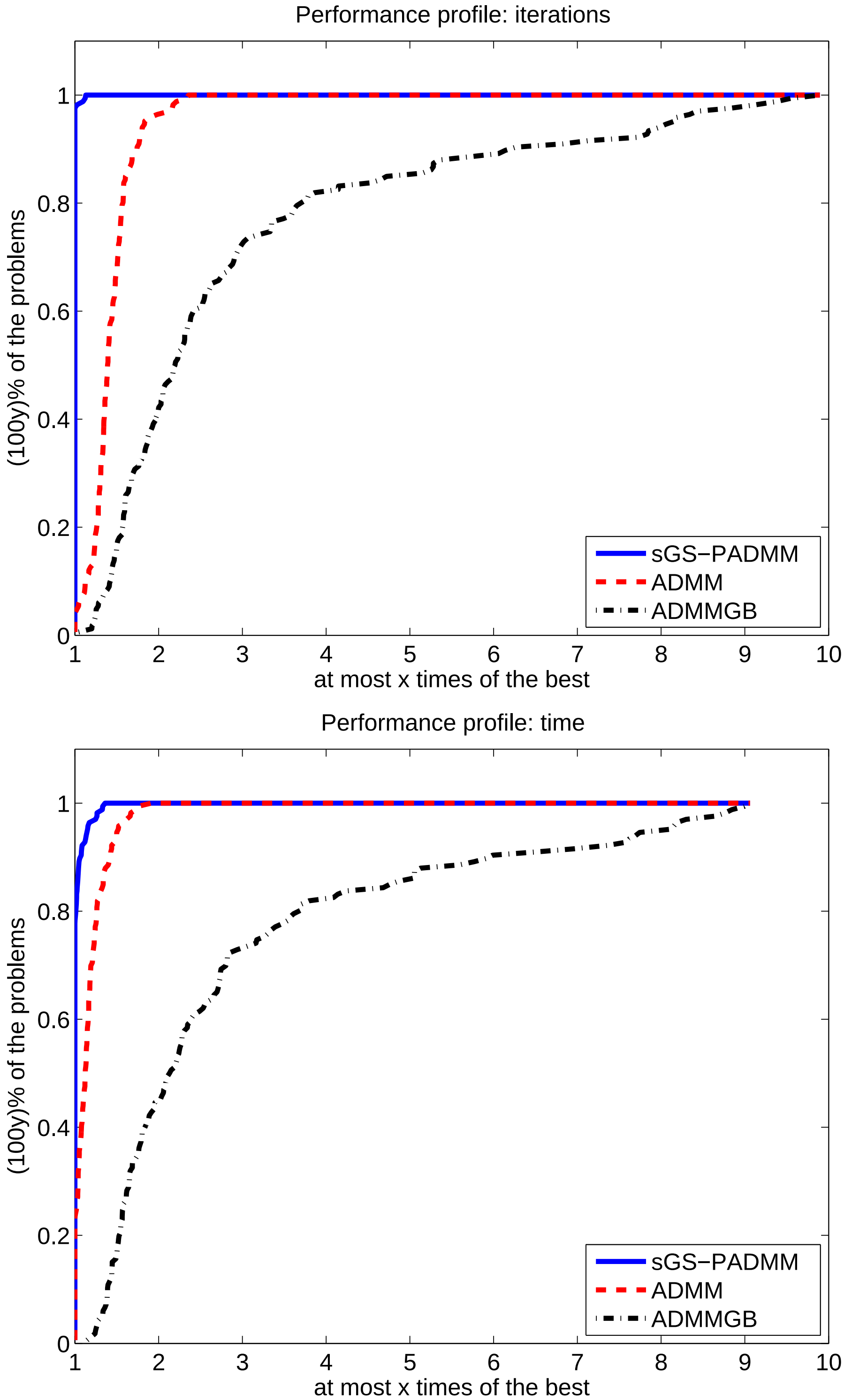


Figure 3.1: Performance profiles of sGS-PADMM, ADMM and ADMMGB for the tested large scale QSDP.

Table 3.1: The performance of sGS-PADMM, ADMM, ADMMGB on QSDP-$\theta_+$, QSDP-BIQ and QSDP-RCP problems (accuracy = $10^{-6}$). In the table, "sgs" stands for sGS-PADMM and "gb" stands for ADMMGB, respectively. The computation time is in the format of "hours:minutes:seconds".

| | | | iteration | $\eta_{\text{qsdp}}$ | $\eta_{\text{gap}}$ | time |
|---|---|---|---|---|---|---|
| problem | $m_E; n_s$ | rank(B) | sgs\|admm\|gb | sgs\|admm\|gb | sgs\|admm\|gb | sgs\|admm\|gb |
| theta6 | 4375 ; 300 | 10 | 311 \| 407 \| 549 | 7.9-7 \| 9.7-7 \| 9.9-7 | 2.1-6 \| -1.6-6 \| -6.2-7 | 08 \| 09 \| 14 |
| theta62 | 13390 ; 300 | 10 | 153 \| 196 \| 229 | 9.6-7 \| 9.9-7 \| 9.6-7 | -1.1-7 \| 9.6-8 \| -4.5-7 | 04 \| 05 \| 06 |
| theta8 | 7905 ; 400 | 10 | 314 \| 384 \| 616 | 9.5-7 \| 9.6-7 \| 9.5-7 | 2.7-6 \| -1.3-6 \| -5.4-7 | 17 \| 18 \| 33 |
| theta82 | 23872 ; 400 | 10 | 158 \| 179 \| 234 | 9.5-7 \| 9.7-7 \| 9.9-7 | -3.7-8 \| -2.8-7 \| -8.2-7 | 10 \| 09 \| 13 |
| theta83 | 39862 ; 400 | 10 | 200 \| 177 \| 219 | 9.3-7 \| 9.6-7 \| 9.4-7 | 6.2-9 \| 1.4-7 \| -1.2-7 | 11 \| 09 \| 14 |
| theta10 | 12470 ; 500 | 10 | 329 \| 439 \| 614 | 9.0-7 \| 8.5-7 \| 9.7-7 | -2.5-6 \| 1.5-6 \| 5.8-7 | 27 \| 33 \| 50 |
| theta102 | 37467 ; 500 | 10 | 150 \| 187 \| 235 | 8.7-7 \| 9.4-7 \| 9.9-7 | 6.4-7 \| 2.9-7 \| -9.3-7 | 15 \| 15 \| 21 |
| theta103 | 62516 ; 500 | 10 | 202 \| 184 \| 222 | 9.8-7 \| 9.5-7 \| 9.9-7 | -4.2-8 \| 6.9-8 \| -1.6-7 | 20 \| 15 \| 21 |
| theta104 | 87245 ; 500 | 10 | 181 \| 181 \| 242 | 9.4-7 \| 9.5-7 \| 9.9-7 | 6.9-8 \| 2.0-7 \| -2.8-7 | 20 \| 15 \| 23 |
| theta12 | 17979 ; 600 | 10 | 343 \| 441 \| 703 | 9.9-7 \| 8.3-7 \| 9.9-7 | 3.0-6 \| 1.4-6 \| -8.8-7 | 42 \| 48 \| 1:27 |
| theta123 | 90020 ; 600 | 10 | 204 \| 205 \| 213 | 9.7-7 \| 9.8-7 \| 9.9-7 | -9.1-8 \| 6.6-8 \| -1.9-7 | 29 \| 25 \| 31 |
| san200-0.7-1 | 5971 ; 200 | 10 | 2150 \| 4758 \| 5172 | 9.8-7 \| 9.9-7 \| 9.9-7 | 5.1-6 \| 2.0-6 \| -3.5-6 | 15 \| 26 \| 36 |
| sanr200-0.7 | 6033 ; 200 | 10 | 177 \| 223 \| 280 | 9.6-7 \| 9.7-7 \| 9.7-7 | 1.9-7 \| -6.0-8 \| 1.7-8 | 02 \| 02 \| 03 |
| c-fat200-1 | 18367 ; 200 | 10 | 2257 \| 3027 \| 3268 | 9.9-7 \| 9.7-7 \| 9.9-7 | -2.6-6 \| -2.0-6 \| -2.2-6 | 24 \| 26 \| 35 |
| hamming-8-4 | 11777 ; 256 | 10 | 2820 \| 2945 \| 3517 | 9.9-7 \| 9.9-7 \| 9.9-7 | -6.0-7 \| -6.4-7 \| -1.1-6 | 53 \| 49 \| 1:09 |
| hamming-9-8 | 2305 ; 512 | 10 | 3891 \| 4980 \| 5577 | 9.9-7 \| 9.9-7 \| 9.9-7 | -3.4-6 \| -5.8-7 \| 9.9-7 | 3:54 \| 4:12 \| 5:50 |
| hamming-8-3-4 | 16129 ; 256 | 10 | 202 \| 220 \| 294 | 4.8-7 \| 8.9-7 \| 9.8-7 | 4.5-6 \| 5.9-7 \| 2.2-7 | 04 \| 04 \| 06 |
| hamming-9-5-6 | 53761 ; 512 | 10 | 436 \| 535 \| 684 | 8.5-7 \| 8.7-7 \| 9.6-7 | 1.1-5 \| -1.7-6 \| -1.6-7 | 36 \| 37 \| 57 |
| brock200-1 | 5067 ; 200 | 10 | 198 \| 210 \| 291 | 9.7-7 \| 9.4-7 \| 9.8-7 | 9.9-8 \| -2.9-7 \| -6.9-10 | 02 \| 02 \| 03 |
| brock200-4 | 6812 ; 200 | 10 | 209 \| 186 \| 263 | 9.8-7 \| 9.9-7 \| 9.8-7 | 1.2-7 \| -2.6-9 \| -1.1-7 | 03 \| 02 \| 03 |

Table 3.1: The performance of sGS-PADMM, ADMM, ADMMGB on QSDP-$\theta_+$, QSDP-BIQ and QSDP-RCP problems (accuracy $= 10^{-6}$). In the table, "sgs" stands for sGS-PADMM and "gb" stands for ADMMGB, respectively. The computation time is in the format of "hours:minutes:seconds".

| | | | iteration | | | $\eta_{\text{qsdp}}$ | | | $\eta_{\text{gap}}$ | | | time | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| problem | $m_E; n_s$ | rank(B) | sgs | admm | gb | sgs | admm | gb | sgs | admm | gb | sgs | admm | gb |
| brock400-1 | 20078 ; 400 | 10 | 168 | 217 | 275 | 9.0-7 | 9.6-7 | 9.7-7 | 8.6-7 | -4.9-8 | 6.2-9 | 11 | 10 | 15 |
| keller4 | 5101 ; 171 | 10 | 669 | 909 | 963 | 9.9-7 | 9.9-7 | 9.9-7 | -1.3-8 | 4.6-9 | -8.4-8 | 06 | 07 | 09 |
| p-hat300-1 | 33918 ; 300 | 10 | 468 | 829 | 2501 | 9.9-7 | 9.9-7 | 8.3-7 | -8.7-7 | 2.1-7 | -1.0-6 | 14 | 20 | 1:09 |
| be250.1 | 251 ; 251 | 10 | 4126 | 7439 | 25000 | 9.6-7 | 9.9-7 | 1.3-6 | -5.8-7 | -8.6-7 | -1.3-8 | 59 | 1:27 | 5:41 |
| be250.2 | 251 ; 251 | 10 | 3604 | 6504 | 16322 | 9.8-7 | 9.9-7 | 9.9-7 | -4.9-7 | -6.8-7 | -7.4-9 | 52 | 1:18 | 3:40 |
| be250.3 | 251 ; 251 | 10 | 3562 | 5712 | 8501 | 9.9-7 | 9.9-7 | 9.7-7 | -9.2-7 | -9.4-7 | 9.3-7 | 52 | 1:08 | 1:57 |
| be250.4 | 251 ; 251 | 10 | 4072 | 7668 | 25000 | 9.9-7 | 9.9-7 | 1.4-6 | -2.1-6 | 2.8-6 | -9.4-9 | 57 | 1:32 | 5:41 |
| be250.5 | 251 ; 251 | 10 | 3210 | 4635 | 7406 | 9.9-7 | 9.9-7 | 9.9-7 | -8.6-7 | -8.8-7 | 1.4-6 | 46 | 55 | 1:41 |
| be250.6 | 251 ; 251 | 10 | 3250 | 5580 | 9812 | 9.9-7 | 9.9-7 | 9.9-7 | -2.8-7 | -3.1-7 | -3.6-7 | 46 | 1:05 | 2:10 |
| be250.7 | 251 ; 251 | 10 | 3699 | 6562 | 13501 | 9.9-7 | 9.9-7 | 9.9-7 | -6.5-7 | -3.8-7 | 5.4-9 | 52 | 1:17 | 3:03 |
| be250.8 | 251 ; 251 | 10 | 3507 | 4712 | 7701 | 9.9-7 | 9.9-7 | 9.6-7 | -9.7-7 | -1.0-6 | 5.1-7 | 50 | 56 | 1:43 |
| be250.9 | 251 ; 251 | 10 | 3678 | 7292 | 21001 | 9.9-7 | 9.9-7 | 9.9-7 | -4.1-7 | -7.2-7 | -1.2-8 | 53 | 1:28 | 4:57 |
| be250.10 | 251 ; 251 | 10 | 3305 | 5752 | 10500 | 9.9-7 | 9.9-7 | 9.9-7 | -1.1-6 | -8.2-7 | -3.7-8 | 49 | 1:06 | 2:19 |
| bqp100-1 | 101 ; 101 | 10 | 1376 | 2134 | 3067 | 9.9-7 | 9.9-7 | 9.9-7 | 2.6-7 | -1.9-7 | -5.1-7 | 05 | 06 | 10 |
| bqp100-2 | 101 ; 101 | 10 | 3109 | 4319 | 7107 | 9.9-7 | 9.9-7 | 9.9-7 | -1.8-7 | -7.2-7 | -5.3-7 | 10 | 13 | 22 |
| bqp100-3 | 101 ; 101 | 10 | 1751 | 2371 | 6276 | 9.9-7 | 9.9-7 | 9.9-7 | -2.7-6 | -3.1-6 | 4.7-7 | 06 | 06 | 20 |
| bqp100-4 | 101 ; 101 | 10 | 2646 | 3986 | 13901 | 9.9-7 | 9.9-7 | 9.1-7 | -4.0-7 | -6.6-7 | -3.3-8 | 09 | 11 | 45 |
| bqp100-5 | 101 ; 101 | 10 | 1979 | 3001 | 6901 | 9.9-7 | 9.9-7 | 9.7-7 | -3.7-7 | -1.5-7 | 1.7-8 | 07 | 08 | 22 |
| bqp100-6 | 101 ; 101 | 10 | 1316 | 2083 | 2937 | 9.4-7 | 9.9-7 | 9.9-7 | 1.1-7 | 3.3-7 | -9.5-7 | 05 | 06 | 11 |
| bqp100-7 | 101 ; 101 | 10 | 1787 | 2341 | 3664 | 9.9-7 | 9.9-7 | 9.9-7 | -5.5-7 | -5.1-7 | -1.3-6 | 06 | 06 | 12 |

Table 3.1: The performance of sGS-PADMM, ADMM, ADMMGB on QSDP-$\theta_+$, QSDP-BIQ and QSDP-RCP problems (accuracy = $10^{-6}$). In the table, "sgs" stands for sGS-PADMM and "gb" stands for ADMMGB, respectively. The computation time is in the format of "hours:minutes:seconds".

<table>
<tr><th></th><th></th><th></th><th>iteration</th><th>$\eta_{\text{qsdp}}$</th><th>$\eta_{\text{gap}}$</th><th>time</th></tr>
<tr><th>problem</th><th>$m_E; n_s$</th><th>rank(B)</th><th>sgs|admm|gb</th><th>sgs|admm|gb</th><th>sgs|admm|gb</th><th>sgs|admm|gb</th></tr>
<tr><td>bqp100-8</td><td>101 ; 101</td><td>10</td><td>1820 | 3337 | 9612</td><td>9.9-7 | 9.9-7 | 9.9-7</td><td>7.3-7 | 8.9-8 | 1.1-8</td><td>06 | 09 | 32</td></tr>
<tr><td>bqp100-9</td><td>101 ; 101</td><td>10</td><td>1948 | 4146 | 15901</td><td>9.9-7 | 9.9-7 | 9.9-7</td><td>-2.2-6 | -6.7-7 | 2.6-9</td><td>07 | 11 | 52</td></tr>
<tr><td>bqp100-10</td><td>101 ; 101</td><td>10</td><td>3207 | 5077 | 12101</td><td>9.9-7 | 9.9-7 | 9.9-7</td><td>8.0-8 | 4.3-7 | 2.7-8</td><td>10 | 15 | 38</td></tr>
<tr><td>bqp250-1</td><td>251 ; 251</td><td>10</td><td>3931 | 5941 | 11758</td><td>9.6-7 | 9.9-7 | 9.9-7</td><td>-1.2-6 | -1.5-6 | 1.2-7</td><td>57 | 1:10 | 2:39</td></tr>
<tr><td>bqp250-2</td><td>251 ; 251</td><td>10</td><td>4007 | 5774 | 9704</td><td>9.5-7 | 9.9-7 | 9.9-7</td><td>-6.6-7 | -2.3-7 | -1.2-6</td><td>57 | 1:07 | 2:11</td></tr>
<tr><td>bqp250-3</td><td>251 ; 251</td><td>10</td><td>4112 | 5708 | 12202</td><td>9.9-7 | 9.9-7 | 9.9-7</td><td>-3.9-6 | 3.8-8 | 3.0-6</td><td>57 | 1:05 | 2:40</td></tr>
<tr><td>bqp250-4</td><td>251 ; 251</td><td>10</td><td>3158 | 4290 | 9671</td><td>9.9-7 | 9.9-7 | 9.9-7</td><td>-5.5-7 | -2.4-6 | 4.5-6</td><td>45 | 52 | 2:13</td></tr>
<tr><td>bqp250-5</td><td>251 ; 251</td><td>10</td><td>4430 | 7349 | 22802</td><td>9.9-7 | 9.9-7 | 9.9-7</td><td>-2.0-6 | 3.6-6 | -1.3-8</td><td>1:02 | 1:29 | 5:13</td></tr>
<tr><td>bqp250-6</td><td>251 ; 251</td><td>10</td><td>2871 | 5122 | 7801</td><td>9.9-7 | 9.9-7 | 9.9-7</td><td>-1.2-6 | -1.3-6 | -2.5-7</td><td>42 | 1:01 | 1:47</td></tr>
<tr><td>bqp250-7</td><td>251 ; 251</td><td>10</td><td>3991 | 5570 | 11508</td><td>9.9-7 | 9.9-7 | 9.9-7</td><td>-2.2-6 | -2.0-6 | -2.7-6</td><td>57 | 1:04 | 2:31</td></tr>
<tr><td>bqp250-8</td><td>251 ; 251</td><td>10</td><td>2882 | 4008 | 5501</td><td>9.9-7 | 9.8-7 | 9.8-7</td><td>-2.0-7 | -7.1-7 | -1.0-6</td><td>40 | 45 | 1:14</td></tr>
<tr><td>bqp250-9</td><td>251 ; 251</td><td>10</td><td>4127 | 6279 | 11998</td><td>9.7-7 | 9.9-7 | 9.9-7</td><td>-5.1-7 | -3.9-7 | 3.8-6</td><td>58 | 1:11 | 2:38</td></tr>
<tr><td>bqp250-10</td><td>251 ; 251</td><td>10</td><td>3044 | 4185 | 7986</td><td>9.9-7 | 9.9-7 | 9.9-7</td><td>-9.3-7 | -7.5-7 | -2.5-6</td><td>43 | 48 | 1:43</td></tr>
<tr><td>bqp500-1</td><td>501 ; 501</td><td>10</td><td>6003 | 8391 | 13416</td><td>9.9-7 | 9.9-7 | 9.9-7</td><td>-3.9-7 | -7.3-7 | -5.4-7</td><td>6:01 | 7:05 | 13:34</td></tr>
<tr><td>bqp500-2</td><td>501 ; 501</td><td>10</td><td>6601 | 10203 | 25000</td><td>9.7-7 | 9.9-7 | 3.4-6</td><td>-4.2-7 | -1.2-7 | 1.8-5</td><td>6:52 | 8:43 | 25:23</td></tr>
<tr><td>bqp500-3</td><td>501 ; 501</td><td>10</td><td>7450 | 10517 | 21140</td><td>9.9-7 | 9.9-7 | 9.9-7</td><td>7.6-7 | -4.3-6 | 1.1-6</td><td>7:31 | 8:46 | 21:10</td></tr>
<tr><td>bqp500-4</td><td>501 ; 501</td><td>10</td><td>7035 | 9903 | 23551</td><td>9.6-7 | 9.9-7 | 9.9-7</td><td>-3.3-7 | -1.3-6 | 2.6-6</td><td>7:08 | 8:12 | 23:36</td></tr>
<tr><td>bqp500-5</td><td>501 ; 501</td><td>10</td><td>6164 | 8406 | 20533</td><td>9.9-7 | 9.9-7 | 9.9-7</td><td>-8.8-7 | -4.8-7 | 2.8-6</td><td>6:30 | 7:04 | 20:37</td></tr>
<tr><td>bqp500-6</td><td>501 ; 501</td><td>10</td><td>6905 | 8659 | 25000</td><td>9.8-7 | 9.9-7 | 1.4-4</td><td>-3.8-7 | -1.5-6 | -1.8-4</td><td>7:13 | 7:30 | 25:44</td></tr>
<tr><td>bqp500-7</td><td>501 ; 501</td><td>10</td><td>6587 | 9038 | 18072</td><td>9.9-7 | 9.9-7 | 9.9-7</td><td>-6.8-7 | 2.5-7 | 2.8-6</td><td>6:41 | 7:39 | 18:13</td></tr>
</table>

Table 3.1: The performance of sGS-PADMM, ADMM, ADMMGB on QSDP-$\theta_+$, QSDP-BIQ and QSDP-RCP problems (accuracy $= 10^{-6}$). In the table, "sgs" stands for sGS-PADMM and "gb" stands for ADMMGB, respectively. The computation time is in the format of "hours:minutes:seconds".

| | | | iteration | $\eta_{qsdp}$ | $\eta_{gap}$ | time |
|---|---|---|---|---|---|---|
| problem | $m_E; n_s$ | rank(B) | sgs\|admm\|gb | sgs\|admm\|gb | sgs\|admm\|gb | sgs\|admm\|gb |
| bqp500-8 | 501 ; 501 | 10 | 6300 \| 8832 \| 16496 | 9.9-7 \| 9.9-7 \| 9.9-7 | 1.3-6 \| -1.6-6 \| 5.8-6 | 6:24 \| 7:17 \| 16:20 |
| bqp500-9 | 501 ; 501 | 10 | 6532 \| 9015 \| 18065 | 9.9-7 \| 9.9-7 \| 9.9-7 | 9.9-7 \| -6.5-7 \| -3.5-6 | 6:39 \| 7:37 \| 18:10 |
| bqp500-10 | 501 ; 501 | 10 | 7199 \| 9787 \| 24119 | 9.9-7 \| 9.9-7 \| 9.9-7 | -1.9-6 \| 2.1-6 \| -2.3-6 | 7:09 \| 8:12 \| 24:15 |
| gka1d | 101 ; 101 | 10 | 1600 \| 2266 \| 4068 | 9.8-7 \| 9.9-7 \| 9.7-7 | -4.2-7 \| -8.8-7 \| 7.4-7 | 06 \| 06 \| 13 |
| gka2d | 101 ; 101 | 10 | 1903 \| 3097 \| 5601 | 9.9-7 \| 9.9-7 \| 9.3-7 | -5.9-7 \| -2.4-7 \| -3.8-8 | 07 \| 09 \| 21 |
| gka3d | 101 ; 101 | 10 | 2431 \| 3101 \| 5618 | 9.9-7 \| 9.9-7 \| 9.9-7 | -2.6-7 \| -3.8-7 \| 1.7-8 | 08 \| 09 \| 19 |
| gka4d | 101 ; 101 | 10 | 2266 \| 2787 \| 6632 | 9.9-7 \| 9.9-7 \| 9.9-7 | 2.3-7 \| -4.4-7 \| -1.9-8 | 08 \| 09 \| 22 |
| soybean-large-2 | 308 ; 307 | 10 | 1267 \| 1717 \| 11208 | 9.9-7 \| 9.9-7 \| 9.9-7 | -5.8-8 \| -6.5-8 \| -7.9-8 | 20 \| 23 \| 2:55 |
| soybean-large-3 | 308 ; 307 | 10 | 936 \| 1362 \| 9261 | 8.3-7 \| 9.1-7 \| 9.8-7 | -5.1-8 \| -5.7-8 \| -1.7-8 | 17 \| 17 \| 2:29 |
| soybean-large-4 | 308 ; 307 | 10 | 1681 \| 2132 \| 13401 | 9.9-7 \| 9.9-7 \| 9.9-7 | -1.0-7 \| -1.0-7 \| -4.3-8 | 29 \| 28 \| 3:49 |
| soybean-large-5 | 308 ; 307 | 10 | 834 \| 1229 \| 3937 | 9.9-7 \| 9.9-7 \| 9.9-7 | -3.2-8 \| -1.9-8 \| -2.3-8 | 14 \| 18 \| 1:08 |
| soybean-large-6 | 308 ; 307 | 10 | 310 \| 475 \| 707 | 9.4-7 \| 8.9-7 \| 8.3-7 | -8.1-8 \| -5.8-8 \| -1.5-7 | 05 \| 06 \| 12 |
| soybean-large-7 | 308 ; 307 | 10 | 1028 \| 1327 \| 3970 | 9.9-7 \| 9.9-7 \| 9.9-7 | -3.6-8 \| -6.3-8 \| -1.8-8 | 19 \| 20 \| 1:12 |
| soybean-large-8 | 308 ; 307 | 10 | 782 \| 1091 \| 2901 | 9.8-7 \| 9.9-7 \| 8.9-7 | -3.7-8 \| -4.5-8 \| -1.0-8 | 14 \| 15 \| 51 |
| soybean-large-9 | 308 ; 307 | 10 | 928 \| 1187 \| 4901 | 9.8-7 \| 9.8-7 \| 9.9-7 | 1.1-7 \| -6.0-8 \| -1.7-8 | 17 \| 19 \| 1:26 |
| soybean-large-10 | 308 ; 307 | 10 | 309 \| 489 \| 518 | 9.9-7 \| 9.9-7 \| 9.7-7 | 2.0-7 \| 3.1-7 \| 1.4-7 | 06 \| 07 \| 09 |
| soybean-large-11 | 308 ; 307 | 10 | 877 \| 1605 \| 1755 | 9.9-7 \| 8.6-7 \| 9.5-7 | -2.2-7 \| 3.5-7 \| -2.6-7 | 17 \| 23 \| 32 |
| spambase-small-2 | 301 ; 300 | 10 | 409 \| 610 \| 2792 | 8.8-7 \| 9.5-7 \| 9.0-7 | -3.1-7 \| -3.9-7 \| -1.1-6 | 06 \| 07 \| 40 |
| spambase-small-3 | 301 ; 300 | 10 | 476 \| 665 \| 1201 | 9.6-7 \| 9.9-7 \| 9.6-7 | 7.8-9 \| -3.7-8 \| -3.3-8 | 09 \| 08 \| 17 |
| spambase-small-4 | 301 ; 300 | 10 | 1305 \| 1983 \| 6073 | 9.9-7 \| 9.9-7 \| 9.9-7 | -4.5-9 \| 6.6-9 \| -1.7-8 | 20 \| 28 \| 1:36 |

Table 3.1: The performance of sGS-PADMM, ADMM, ADMMGB on QSDP-$\theta_+$, QSDP-BIQ and QSDP-RCP problems (accuracy = $10^{-6}$). In the table, "sgs" stands for sGS-PADMM and "gb" stands for ADMMGB, respectively. The computation time is in the format of "hours:minutes:seconds".

| | | | iteration | $\eta_{\text{qsdp}}$ | $\eta_{\text{gap}}$ | time |
|---|---|---|---|---|---|---|
| problem | $m_E; n_s$ | rank(B) | sgs\|admm\|gb | sgs\|admm\|gb | sgs\|admm\|gb | sgs\|admm\|gb |
| spambase-small-5 | 301 ; 300 | 10 | 608 \| 819 \| 868 | 8.5-7 \| 9.8-7 \| 9.9-7 | -7.3-7 \| -2.7-7 \| -1.4-7 | 11 \| 11 \| 14 |
| spambase-small-6 | 301 ; 300 | 10 | 811 \| 1198 \| 1334 | 9.9-7 \| 9.9-7 \| 9.9-7 | -1.5-7 \| -2.0-7 \| -1.3-7 | 14 \| 17 \| 23 |
| spambase-small-7 | 301 ; 300 | 10 | 849 \| 1240 \| 1359 | 9.9-7 \| 9.9-7 \| 9.9-7 | 4.0-7 \| 2.8-7 \| 1.8-7 | 15 \| 18 \| 25 |
| spambase-small-8 | 301 ; 300 | 10 | 1109 \| 1244 \| 1501 | 9.9-7 \| 9.9-7 \| 8.8-7 | 7.1-8 \| 9.3-8 \| 7.6-8 | 20 \| 18 \| 27 |
| spambase-small-9 | 301 ; 300 | 10 | 1090 \| 1415 \| 1440 | 9.9-7 \| 9.7-7 \| 9.9-7 | -1.7-7 \| 2.9-8 \| -1.3-8 | 20 \| 21 \| 27 |
| spambase-small-10 | 301 ; 300 | 10 | 1081 \| 1341 \| 1500 | 9.9-7 \| 9.9-7 \| 9.9-7 | 1.7-7 \| 1.5-7 \| -1.5-7 | 20 \| 22 \| 27 |
| spambase-small-11 | 301 ; 300 | 10 | 1319 \| 1482 \| 1653 | 9.9-7 \| 9.9-7 \| 9.9-7 | -3.6-7 \| -8.3-7 \| -5.8-7 | 25 \| 25 \| 31 |
| spambase-medium-2 | 901 ; 900 | 10 | 471 \| 596 \| 1201 | 9.9-7 \| 9.9-7 \| 8.9-7 | -1.6-6 \| -1.3-6 \| -1.9-6 | 1:42 \| 1:37 \| 4:01 |
| spambase-medium-3 | 901 ; 900 | 10 | 1205 \| 1582 \| 11000 | 9.9-7 \| 9.9-7 \| 9.9-7 | -2.0-7 \| -1.8-7 \| -2.2-7 | 4:18 \| 4:16 \| 36:54 |
| spambase-medium-4 | 901 ; 900 | 10 | 2560 \| 2990 \| 4045 | 9.7-7 \| 9.8-7 \| 9.9-7 | -2.3-6 \| 2.5-6 \| 1.1-6 | 9:06 \| 8:04 \| 13:37 |
| spambase-medium-5 | 901 ; 900 | 10 | 1414 \| 1900 \| 2901 | 9.9-7 \| 9.9-7 \| 9.0-7 | 7.4-8 \| 3.8-8 \| -1.1-6 | 5:06 \| 5:17 \| 9:58 |
| spambase-medium-6 | 901 ; 900 | 10 | 1607 \| 2107 \| 2698 | 9.9-7 \| 9.9-7 \| 9.9-7 | -1.0-8 \| 3.7-8 \| -1.3-6 | 6:01 \| 6:16 \| 9:25 |
| spambase-medium-7 | 901 ; 900 | 10 | 1805 \| 2508 \| 2846 | 9.9-7 \| 9.9-7 \| 9.9-7 | -8.7-8 \| -4.5-8 \| -1.4-6 | 6:55 \| 7:36 \| 10:00 |
| spambase-medium-8 | 901 ; 900 | 10 | 1655 \| 2309 \| 2489 | 9.9-7 \| 9.9-7 \| 9.9-7 | -2.6-8 \| -6.7-8 \| 4.6-7 | 6:19 \| 6:54 \| 8:47 |
| spambase-medium-9 | 901 ; 900 | 10 | 1683 \| 2330 \| 2687 | 9.9-7 \| 9.9-7 \| 9.9-7 | 2.6-8 \| -5.9-8 \| 2.2-8 | 6:23 \| 6:56 \| 9:38 |
| spambase-medium-10 | 901 ; 900 | 10 | 1641 \| 2030 \| 2617 | 9.9-7 \| 9.9-7 \| 9.8-7 | -6.5-7 \| -4.7-7 \| 1.9-6 | 6:11 \| 5:59 \| 9:22 |
| spambase-medium-11 | 901 ; 900 | 10 | 1608 \| 1838 \| 3210 | 9.9-7 \| 9.9-7 \| 9.9-7 | -5.0-7 \| 5.4-7 \| 9.0-7 | 6:06 \| 5:20 \| 11:21 |
| abalone-medium-2 | 401 ; 400 | 10 | 500 \| 682 \| 1301 | 9.9-7 \| 9.9-7 \| 8.5-7 | -7.4-8 \| 5.8-8 \| 3.4-8 | 16 \| 17 \| 40 |
| abalone-medium-3 | 401 ; 400 | 10 | 715 \| 1011 \| 1679 | 9.9-7 \| 9.9-7 \| 9.9-7 | -2.5-9 \| 1.3-8 \| -1.1-8 | 24 \| 28 \| 56 |
| abalone-medium-4 | 401 ; 400 | 10 | 372 \| 626 \| 684 | 9.9-7 \| 9.9-7 \| 9.9-7 | -5.3-8 \| 3.6-9 \| 6.3-9 | 12 \| 16 \| 24 |

Table 3.1: The performance of sGS-PADMM, ADMM, ADMMGB on QSDP-$\theta_+$, QSDP-BIQ and QSDP-RCP problems (accuracy = $10^{-6}$). In the table, "sgs" stands for sGS-PADMM and "gb" stands for ADMMGB, respectively. The computation time is in the format of "hours:minutes:seconds".

| | | | iteration | $\eta_{qsdp}$ | $\eta_{gap}$ | time |
|---|---|---|---|---|---|---|
| problem | $m_E; n_s$ | rank(B) | sgs\|admm\|gb | sgs\|admm\|gb | sgs\|admm\|gb | sgs\|admm\|gb |
| abalone-medium-5 | 401 ; 400 | 10 | 524 \| 779 \| 942 | 9.9-7 \| 9.9-7 \| 9.9-7 | -3.8-8 \| -1.4-7 \| -9.6-8 | 18 \| 21 \| 32 |
| abalone-medium-6 | 401 ; 400 | 10 | 536 \| 946 \| 1162 | 9.7-7 \| 9.9-7 \| 9.9-7 | -1.3-7 \| -2.3-7 \| -1.8-7 | 22 \| 27 \| 38 |
| abalone-medium-7 | 401 ; 400 | 10 | 1046 \| 1676 \| 2013 | 9.9-7 \| 9.9-7 \| 9.9-7 | -8.9-8 \| -4.2-8 \| -3.3-8 | 37 \| 47 \| 1:09 |
| abalone-medium-8 | 401 ; 400 | 10 | 745 \| 1123 \| 1641 | 9.6-7 \| 9.7-7 \| 9.9-7 | -3.9-8 \| -2.2-7 \| -9.1-8 | 27 \| 32 \| 55 |
| abalone-medium-9 | 401 ; 400 | 10 | 1035 \| 1504 \| 1709 | 9.9-7 \| 9.5-7 \| 9.9-7 | -8.3-8 \| 7.1-8 \| -1.2-8 | 38 \| 43 \| 1:02 |
| abalone-medium-10 | 401 ; 400 | 10 | 1349 \| 1803 \| 1904 | 9.9-7 \| 9.4-7 \| 9.8-7 | -1.7-7 \| -2.0-7 \| -2.2-7 | 49 \| 51 \| 1:07 |
| abalone-medium-11 | 401 ; 400 | 10 | 1066 \| 1504 \| 1704 | 9.9-7 \| 9.7-7 \| 9.5-7 | -1.1-7 \| -1.6-7 \| -1.6-7 | 40 \| 45 \| 1:02 |
| abalone-large-2 | 1001 ; 1000 | 10 | 594 \| 734 \| 909 | 9.9-7 \| 9.8-7 \| 9.9-7 | 4.6-7 \| 4.5-7 \| 1.3-7 | 3:16 \| 2:35 \| 3:54 |
| abalone-large-3 | 1001 ; 1000 | 10 | 656 \| 1014 \| 1901 | 9.9-7 \| 9.9-7 \| 9.9-7 | -1.4-8 \| -7.2-8 \| -4.4-8 | 3:03 \| 3:37 \| 8:20 |
| abalone-large-4 | 1001 ; 1000 | 10 | 505 \| 749 \| 995 | 9.9-7 \| 9.9-7 \| 9.8-7 | -1.3-9 \| -1.6-8 \| -6.6-8 | 2:42 \| 2:39 \| 4:24 |
| abalone-large-5 | 1001 ; 1000 | 10 | 752 \| 1187 \| 1550 | 9.8-7 \| 9.9-7 \| 9.9-7 | -6.8-8 \| -1.8-7 \| -1.2-7 | 4:11 \| 4:16 \| 6:53 |
| abalone-large-6 | 1001 ; 1000 | 10 | 886 \| 1364 \| 1670 | 9.9-7 \| 9.9-7 \| 9.9-7 | -9.5-8 \| -1.1-7 \| -1.2-7 | 4:09 \| 4:56 \| 7:27 |
| abalone-large-7 | 1001 ; 1000 | 10 | 1206 \| 1614 \| 2251 | 9.9-7 \| 9.9-7 \| 9.9-7 | -1.1-7 \| 1.8-8 \| -7.5-8 | 5:40 \| 5:47 \| 9:59 |
| abalone-large-8 | 1001 ; 1000 | 10 | 1092 \| 1721 \| 2046 | 9.9-7 \| 9.9-7 \| 9.9-7 | -3.1-7 \| -1.8-7 \| -2.9-7 | 5:08 \| 6:14 \| 9:07 |
| abalone-large-9 | 1001 ; 1000 | 10 | 1557 \| 2407 \| 2746 | 9.8-7 \| 9.9-7 \| 9.9-7 | -3.8-7 \| -3.5-7 \| -2.8-7 | 8:30 \| 8:47 \| 12:15 |
| abalone-large-10 | 1001 ; 1000 | 10 | 1682 \| 2488 \| 2821 | 9.9-7 \| 9.9-7 \| 9.9-7 | -1.6-7 \| -2.6-7 \| -2.5-7 | 8:00 \| 9:06 \| 12:39 |
| abalone-large-11 | 1001 ; 1000 | 10 | 1923 \| 3005 \| 3723 | 9.8-7 \| 9.9-7 \| 9.9-7 | 1.3-7 \| 3.6-8 \| -3.5-8 | 9:17 \| 11:00 \| 16:39 |
| segment-medium-2 | 701 ; 700 | 10 | 1016 \| 1541 \| 1880 | 9.7-7 \| 9.8-7 \| 9.9-7 | 1.3-6 \| -1.1-6 \| 2.5-7 | 2:07 \| 2:13 \| 3:26 |
| segment-medium-3 | 701 ; 700 | 10 | 713 \| 714 \| 1801 | 9.4-7 \| 9.5-7 \| 9.2-7 | -4.0-7 \| -9.7-7 \| -8.7-7 | 1:24 \| 1:03 \| 3:20 |
| segment-medium-4 | 701 ; 700 | 10 | 2282 \| 2710 \| 17881 | 9.9-7 \| 9.9-7 \| 9.9-7 | -7.1-8 \| -6.5-8 \| -6.5-8 | 4:30 \| 4:25 \| 34:11 |

Table 3.1: The performance of sGS-PADMM, ADMM, ADMMGB on QSDP-$\theta_+$, QSDP-BIQ and QSDP-RCP problems (accuracy = $10^{-6}$). In the table, "sgs" stands for sGS-PADMM and "gb" stands for ADMMGB, respectively. The computation time is in the format of "hours:minutes:seconds".

| | | | iteration | $\eta_{\text{qsdp}}$ | $\eta_{\text{gap}}$ | time |
|---|---|---|---|---|---|---|
| problem | $m_E; n_s$ | rank(B) | sgs\|admm\|gb | sgs\|admm\|gb | sgs\|admm\|gb | sgs\|admm\|gb |
| segment-medium-5 | 701 ; 700 | 10 | 2322 \| 3100 \| 18701 | 9.9-7 \| 9.9-7 \| 9.9-7 | -1.2-7 \| -9.5-8 \| -7.3-8 | 4:40 \| 5:02 \| 35:56 |
| segment-medium-6 | 701 ; 700 | 10 | 2966 \| 3916 \| 25000 | 9.9-7 \| 9.9-7 \| 1.4-6 | -1.7-7 \| -1.4-7 \| -1.3-7 | 6:12 \| 6:29 \| 51:26 |
| segment-medium-7 | 701 ; 700 | 10 | 3185 \| 4268 \| 25000 | 9.9-7 \| 9.9-7 \| 1.6-6 | -1.7-7 \| -1.7-7 \| -1.6-7 | 7:03 \| 7:34 \| 53:28 |
| segment-medium-8 | 701 ; 700 | 10 | 2998 \| 4140 \| 25000 | 9.9-7 \| 9.9-7 \| 1.1-6 | -1.6-7 \| -1.7-7 \| -6.7-8 | 6:28 \| 7:09 \| 52:54 |
| segment-medium-9 | 701 ; 700 | 10 | 2123 \| 2635 \| 8801 | 9.9-7 \| 9.9-7 \| 9.9-7 | -1.9-7 \| -3.0-8 \| -4.3-8 | 4:32 \| 4:25 \| 18:04 |
| segment-medium-10 | 701 ; 700 | 10 | 1695 \| 2414 \| 6101 | 9.9-7 \| 9.9-7 \| 9.8-7 | -2.4-7 \| -1.2-7 \| -2.2-8 | 3:35 \| 4:07 \| 12:27 |
| segment-medium-11 | 701 ; 700 | 10 | 1454 \| 2437 \| 2101 | 9.4-7 \| 9.7-7 \| 8.6-7 | 6.4-8 \| -6.3-7 \| -1.5-7 | 3:01 \| 4:00 \| 4:13 |
| segment-large-2 | 1001 ; 1000 | 10 | 1348 \| 1823 \| 2038 | 9.6-7 \| 9.9-7 \| 9.9-7 | -1.3-6 \| -1.3-6 \| -1.4-6 | 6:30 \| 6:15 \| 8:40 |
| segment-large-3 | 1001 ; 1000 | 10 | 479 \| 533 \| 1601 | 9.9-7 \| 9.9-7 \| 8.7-7 | -4.0-7 \| -1.0-6 \| -4.4-7 | 2:10 \| 1:53 \| 6:49 |
| segment-large-4 | 1001 ; 1000 | 10 | 2157 \| 2802 \| 20226 | 9.9-7 \| 9.9-7 \| 9.9-7 | -9.1-8 \| -9.5-8 \| -7.1-8 | 9:57 \| 9:57 \| 1:27:58 |
| segment-large-5 | 1001 ; 1000 | 10 | 2618 \| 3404 \| 25000 | 9.9-7 \| 9.9-7 \| 1.0-6 | -1.1-7 \| -9.3-8 \| -8.3-8 | 12:13 \| 12:12 \| 1:50:29 |
| segment-large-6 | 1001 ; 1000 | 10 | 3236 \| 4143 \| 25000 | 9.9-7 \| 9.9-7 \| 1.4-6 | -1.8-7 \| -1.8-7 \| -1.2-7 | 15:28 \| 15:20 \| 1:52:58 |
| segment-large-7 | 1001 ; 1000 | 10 | 3505 \| 4318 \| 25000 | 9.9-7 \| 9.9-7 \| 1.8-6 | -1.8-7 \| -1.7-7 \| -1.9-7 | 17:07 \| 16:39 \| 1:56:00 |
| segment-large-8 | 1001 ; 1000 | 10 | 3063 \| 3749 \| 25000 | 9.9-7 \| 9.9-7 \| 1.2-6 | -9.3-8 \| -7.8-8 \| -1.0-7 | 14:55 \| 14:18 \| 1:56:05 |
| segment-large-9 | 1001 ; 1000 | 10 | 2497 \| 3248 \| 15649 | 9.9-7 \| 9.9-7 \| 9.9-7 | -1.4-7 \| -1.2-7 \| -5.1-8 | 12:05 \| 13:16 \| 1:11:25 |
| segment-large-10 | 1001 ; 1000 | 10 | 1723 \| 2226 \| 4901 | 9.9-7 \| 9.9-7 \| 9.9-7 | 7.4-9 \| 1.4-8 \| -2.1-8 | 8:00 \| 8:12 \| 21:45 |
| segment-large-11 | 1001 ; 1000 | 10 | 1571 \| 2331 \| 3417 | 9.9-7 \| 9.7-7 \| 9.9-7 | 1.9-7 \| -5.1-7 \| -1.7-8 | 7:20 \| 8:30 \| 15:23 |
| housing-2 | 507 ; 506 | 10 | 3183 \| 5358 \| 4689 | 9.4-7 \| 9.7-7 \| 9.7-7 | -1.9-7 \| 1.8-7 \| 2.0-7 | 2:54 \| 3:22 \| 3:48 |
| housing-3 | 507 ; 506 | 10 | 845 \| 1970 \| 1714 | 9.9-7 \| 9.9-7 \| 9.9-7 | -1.5-7 \| 1.2-7 \| -2.2-8 | 48 \| 1:16 \| 1:24 |
| housing-4 | 507 ; 506 | 10 | 805 \| 1742 \| 2057 | 9.4-7 \| 9.9-7 \| 9.9-7 | -2.5-8 \| -4.8-8 \| -3.4-8 | 45 \| 1:09 \| 1:45 |

Table 3.1: The performance of sGS-PADMM, ADMM, ADMMGB on QSDP-$\theta_+$, QSDP-BIQ and QSDP-RCP problems (accuracy = $10^{-6}$). In the table, "sgs" stands for sGS-PADMM and "gb" stands for ADMMGB, respectively. The computation time is in the format of "hours:minutes:seconds".

| | | | iteration | $\eta_{\text{qsdp}}$ | $\eta_{\text{gap}}$ | time |
|---|---|---|---|---|---|---|
| problem | $m_E; n_s$ | rank(B) | sgs\|admm\|gb | sgs\|admm\|gb | sgs\|admm\|gb | sgs\|admm\|gb |
| housing-5 | 507 ; 506 | 10 | 874 \| 1262 \| 1774 | 9.9-7 \| 9.9-7 \| 9.9-7 | 2.4-7 \| -2.3-7 \| -2.6-7 | 1:10 \| 1:14 \| 3:08 |
| housing-6 | 507 ; 506 | 10 | 586 \| 826 \| 1005 | 9.9-7 \| 9.9-7 \| 9.9-7 | -1.9-8 \| 2.9-9 \| -8.6-8 | 1:41 \| 1:26 \| 1:39 |
| housing-7 | 507 ; 506 | 10 | 583 \| 906 \| 1069 | 9.9-7 \| 9.9-7 \| 9.9-7 | -1.3-7 \| -2.7-7 \| -1.7-7 | 32 \| 37 \| 56 |
| housing-8 | 507 ; 506 | 10 | 682 \| 904 \| 1074 | 9.9-7 \| 9.3-7 \| 9.9-7 | -1.1-7 \| -6.9-9 \| -6.6-8 | 39 \| 38 \| 59 |
| housing-9 | 507 ; 506 | 10 | 765 \| 1208 \| 1590 | 8.5-7 \| 9.9-7 \| 9.8-7 | -1.5-7 \| -1.3-8 \| 8.5-8 | 44 \| 53 \| 1:26 |
| housing-10 | 507 ; 506 | 10 | 1027 \| 1381 \| 1541 | 9.9-7 \| 9.9-7 \| 9.9-7 | -6.4-8 \| -1.6-7 \| -1.0-7 | 58 \| 1:02 \| 1:27 |
| housing-11 | 507 ; 506 | 10 | 867 \| 1327 \| 1359 | 9.9-7 \| 9.9-7 \| 9.9-7 | -1.0-7 \| -9.0-8 \| -9.2-8 | 49 \| 1:01 \| 1:19 |

### 3.4.2 Nearest correlation matrix (NCM) approximations

In this subsection, we first consider the problem of finding the nearest correlation matrix (NCM) to a given matrix $G \in \mathcal{S}^n$:

$$\begin{array}{ll} \min & \frac{1}{2}\|H \circ (X - G)\|_F^2 + \langle C,\, X\rangle \\ \text{s.t.} & \mathcal{A}_E X \;=\; b_E, \quad X \in \mathcal{S}^n_+ \cap \mathcal{K}\,, \end{array} \tag{3.73}$$

where $H \in \mathcal{S}^n$ is a nonnegative weight matrix, $\mathcal{A}_E : \mathcal{S}^n \to \Re^{m_E}$ is a linear map, $G \in \mathcal{S}^n$, $C \in \mathcal{S}^n$ and $b_E \in \Re^{m_E}$ are given data, $\mathcal{K}$ is a nonempty simple closed convex set, e.g., $\mathcal{K} = \{W \in \mathcal{S}^n \mid L \le W \le U\}$ with $L, U \in \mathcal{S}^n$ being given matrices. In fact, this is also an instance of the general model of problem (3.64) with no inequality constraints, $\mathcal{Q}X = H \circ H \circ X$ and $\mathcal{B}X = H \circ X$. We place this special example of QSDP here since an extension will be considered next.

Now, let's consider an interesting variant of the above NCM problem:

$$\begin{array}{ll} \min & \|H \circ (X - G)\|_2 + \langle C,\, X\rangle \\ \text{s.t.} & \mathcal{A}_E X \;=\; b_E, \quad X \in \mathcal{S}^n_+ \cap \mathcal{K}\,. \end{array} \tag{3.74}$$

Note, in (3.74), instead of the Frobenius norm, we use the spectral norm. By introducing a slack variable $Y$, we can reformulate problem (3.74) as

$$\begin{array}{ll} \min & \|Y\|_2 + \langle C,\, X\rangle \\ \text{s.t.} & H \circ (X - G) = Y, \quad \mathcal{A}_E X = b_E, \quad X \in \mathcal{S}^n_+ \cap \mathcal{K}\,. \end{array} \tag{3.75}$$

The dual of problem (3.75) is given by

$$\begin{array}{ll} \max & -\delta^*_{\mathcal{K}}(-Z) + \langle H \circ G,\, \Xi\rangle + \langle b_E,\, y_E\rangle \\ \text{s.t.} & Z + H \circ \Xi + S + \mathcal{A}^*_E y_E = C, \quad \|\Xi\|_* \le 1, \quad S \in \mathcal{S}^n_+\,, \end{array} \tag{3.76}$$

which is obviously equivalent to the following problem

$$\begin{array}{ll} \max & -\delta^*_{\mathcal{K}}(-Z) + \langle H \circ G,\, \Xi\rangle + \langle b_E,\, y_E\rangle \\ \text{s.t.} & Z + H \circ \Xi + S + \mathcal{A}^*_E y_E = C, \quad \|\Gamma\|_* \le 1, \quad S \in \mathcal{S}^n_+\,, \\ & \mathcal{D}^*\Gamma - \mathcal{D}^*\Xi = 0, \end{array} \tag{3.77}$$

where $\mathcal{D} : \mathcal{S}^n \to \mathcal{S}^n$ is a nonsingular linear operator. Note that sGS-PADMM can not be directly applied to solve the problem (3.76) while the equivalent reformulation (3.77) fits our model nicely.

In our numerical test, matrix $\widehat{G}$ is the gene correlation matrix from [33]. For testing purpose we perturb $\widehat{G}$ to

$$G := (1-\alpha)\widehat{G} + \alpha E,$$

where $\alpha \in (0,1)$ and $E$ is a randomly generated symmetric matrix with entries in $[-1,1]$. We also set $G_{ii} = 1,\ i = 1,\ldots,n$. The weight matrix $H$ is generated from a weight matrix $H_0$ used by a hedge fund company. The matrix $H_0$ is a $93 \times 93$ symmetric matrix with all positive entries. It has about 24% of the entries equal to $10^{-5}$ and the rest are distributed in the interval $[2, 1.28 \times 10^3]$. It has 28 eigenvalues in the interval $[-520, -0.04]$, 11 eigenvalues in the interval $[-5 \times 10^{-13}, 2 \times 10^{-13}]$, and the rest of 54 eigenvalues in the interval $[10^{-4}, 2 \times 10^4]$. The Matlab code for generating the matrix $H$ is given by

```
tmp = kron(ones(25,25),H0); H = tmp(1:n,1:n); H = (H'+H)/2.
```

The reason for using such a weight matrix is because the resulting problems generated are more challenging to solve as opposed to a randomly generated weight matrix. Note that the matrices $G$ and $H$ are generated in the same way as in [29]. For simplicity, we further set $C = 0$ and $\mathcal{K} = \{X \in \mathcal{S}^n :\ X \geq -0.5\}$.

Generally speaking, there is no widely accepted stopping criterion for spectral norm H-weighted NCM problem (3.75). Here, with reference to the general relative residue (3.63), we measure the accuracy of an approximate optimal solution $(X, Z, \Xi, S, y_E)$ for spectral norm H-weighted NCM problem problem (3.74) (equivalently (3.75)) and its dual (3.76) (equivalently (3.77)) by using the following relative residual derived from the general optimality condition (3.62):

$$\eta_{\rm sncm} = \max\{\eta_P, \eta_D, \eta_Z, \eta_{S_1}, \eta_{S_2}, \eta_\Xi\}, \tag{3.78}$$

Table 3.2: The performance of sGS-PADMM, ADMM, ADMMGB on Frobenius norm H-weighted NCM problems (dual of (3.73)) (accuracy = $10^{-6}$). In the table, "sgs" stands for sGS-PADMM and "gb" stands for ADMMGB, respectively. The computation time is in the format of "hours:minutes:seconds".

| | | | iteration | $\eta_{\rm qsdp}$ | $\eta_{\rm gap}$ | time |
|---|---|---|---|---|---|---|
| problem | $n_s$ | $\alpha$ | sgs\|admm\|gb | sgs\|admm\|gb | sgs\|admm\|gb | sgs\|admm\|gb |
| Lymph | 587 | 0.10 | 263 \| 522 \| 696 | 9.9-7 \| 9.9-7 \| 9.9-7 | -4.4-7 \| -4.5-7 \| -4.0-7 | 30 \| 53 \| 1:23 |
| | 587 | 0.05 | 264 \| 356 \| 592 | 9.9-7 \| 9.9-7 \| 9.9-7 | -3.9-7 \| -3.4-7 \| -3.0-7 | 29 \| 35 \| 1:08 |
| ER | 692 | 0.10 | 268 \| 355 \| 711 | 9.9-7 \| 9.9-7 \| 9.9-7 | -5.1-7 \| -4.7-7 \| -4.2-7 | 43 \| 51 \| 1:58 |
| | 692 | 0.05 | 226 \| 293 \| 603 | 9.9-7 \| 9.9-7 \| 9.9-7 | -4.2-7 \| -3.8-7 \| -3.3-7 | 37 \| 43 \| 1:54 |
| Arabidopsis | 834 | 0.10 | 510 \| 528 \| 725 | 9.9-7 \| 9.9-7 \| 9.9-7 | -5.9-7 \| -5.3-7 \| -3.9-7 | 2:11 \| 2:02 \| 3:03 |
| | 834 | 0.05 | 444 \| 470 \| 650 | 9.9-7 \| 9.9-7 \| 9.9-7 | -5.8-7 \| -5.2-7 \| -4.8-7 | 1:51 \| 1:43 \| 2:44 |
| Leukemia | 1255 | 0.10 | 292 \| 420 \| 826 | 9.9-7 \| 9.9-7 \| 9.9-7 | -5.4-7 \| -5.3-7 \| -4.4-7 | 3:13 \| 4:11 \| 9:13 |
| | 1255 | 0.05 | 251 \| 408 \| 670 | 9.9-7 \| 9.7-7 \| 9.6-7 | -5.4-7 \| -4.9-7 \| -4.0-7 | 2:48 \| 4:03 \| 7:35 |
| hereditarybc | 1869 | 0.10 | 555 \| 634 \| 871 | 9.9-7 \| 9.9-7 \| 9.9-7 | -9.1-7 \| -9.1-7 \| -7.0-7 | 17:39 \| 18:38 \| 28:01 |
| | 1869 | 0.05 | 530 \| 626 \| 839 | 9.9-7 \| 9.9-7 \| 9.9-7 | -8.7-7 \| -8.7-7 \| -5.2-7 | 16:50 \| 18:15 \| 26:34 |

where

$$\eta_P = \frac{\|\mathcal{A}_E X - b_E\|}{1+\|b_E\|}, \quad \eta_D = \frac{\|Z + H\circ\Xi + S + \mathcal{A}_E^* y_E\|}{1+\|Z\|+\|S\|}, \quad \eta_Z = \frac{\|X - \Pi_{\mathcal{K}}(X-Z)\|}{1+\|X\|+\|Z\|},$$

$$\eta_{S_1} = \frac{|\langle S,\, X\rangle|}{1+\|S\|+\|X\|}, \quad \eta_{S_2} = \frac{\|X - \Pi_{\mathcal{S}^n_+}(X)\|}{1+\|X\|},$$

$$\eta_\Xi = \frac{\|\Xi - \Pi_{\{X\in\Re^{n\times n}:\|X\|_*\le 1\}}(\Xi - H\circ(X-G))\|}{1+\|\Xi\|+\|H\circ(X-G)\|}.$$

Firstly, numerical results for solving F-norm H-weighted NCM problems (3.74) are reported. We compare all three algorithms, namely sGS-PADMM, ADMM, ADMMGB using the relative residue (3.72). We terminate the solvers when $\eta_{\rm qsdp} < 10^{-6}$ with the maximum number of iterations set at 25000.

In Table 3.2, we report detailed numerical results for sGS-PADMM, ADMM and ADMMGB in solving various instances of F-norm H-weighted NCM problem. As we can see from Table 3.2, our sGS-PADMM is certainly more efficient than the other two algorithms on most of the problems tested.

Table 3.3: The performance of sGS-PADMM, ADMM, ADMMGB on spectral norm H-weighted NCM problem (3.77) (accuracy $= 10^{-5}$). In the table, "sgs" stands for sGS-PADMM and "gb" stands for ADMMGB, respectively. The computation time is in the format of "hours:minutes:seconds".

| | | | iteration | $\eta_{\text{sncm}}$ | $\eta_{\text{gap}}$ | time |
|---|---|---|---|---|---|---|
| problem | $n_s$ | $\alpha$ | sgs\|admm\|gb | sgs\|admm\|gb | sgs\|admm\|gb | sgs\|admm\|gb |
| Lymph | 587 | 0.10 | 4110\|6048\|7131 | 9.9-6\|9.9-6\|1.0-5 | -3.4-5\|-2.8-5\|-2.7-5 | 13:21\|17:10\|21:43 |
| | 587 | 0.05 | 5001\|7401\|8101 | 9.8-6\|9.9-6\|9.9-6 | -2.0-5\|-2.3-5\|-8.1-6 | 19:41\|21:25\|25:13 |
| ER | 692 | 0.10 | 3251\|4844\|6478 | 9.9-6\|9.9-6\|1.0-5 | -3.1-5\|-2.6-5\|-6.0-6 | 15:06\|19:30\|28:03 |
| | 692 | 0.05 | 4201\|5851\|7548 | 9.3-6\|9.8-6\|1.0-5 | -3.5-5\|-2.9-5\|-3.4-5 | 18:44\|23:46\|32:57 |
| Arabid. | 834 | 0.10 | 3344\|6251\|7965 | 9.9-6\|9.7-6\|1.0-5 | -3.8-5\|-2.0-5\|-3.7-5 | 23:20\|40:12\|54:31 |
| | 834 | 0.05 | 2496\|3101\|3231 | 9.9-6\|9.9-6\|1.0-5 | -9.1-5\|-4.3-5\|-5.3-5 | 17:03\|19:53\|21:56 |
| Leukemia | 1255 | 0.10 | 4351\|6102\|7301 | 9.9-6\|9.9-6\|1.0-5 | -3.7-5\|-3.3-5\|-3.0-5 | 1:22:42\|1:49:02\|2:16:52 |
| | 1255 | 0.05 | 3957\|5851\|10151 | 9.9-6\|9.7-6\|9.5-6 | -7.2-5\|-5.7-5\|-1.1-5 | 1:18:19\|1:44:47\|3:26:08 |

The rest of this subsection is devoted to the numerical results of the spectral norm H-weighted NCM problem (3.74). As mentioned before, sGS-PADMM is applied to solve the problem (3.77) rather than (3.76). We implemented all the algorithms for solving problem (3.77) using the relative residue (3.78). We terminate the solvers when $\eta_{\text{sncm}} < 10^{-5}$ with the maximum number of iterations set at 25000. In Table 3.3, we report detailed numerical results for sGS-PADMM, ADMM and ADMMGB in solving various instances of spectral norm H-weighted NCM problem. As we can see from Table 3.3, our sGS-PADMM is much more efficient than the other two algorithms.

Observe that although there is no convergence guarantee, one may still apply the directly extended ADMM with 4 blocks to the original dual problem (3.76) by adding a proximal term for the $\Xi$ part. We call this method LADMM. Moreover, by using the same proximal strategy for $\Xi$, a convergent linearized alternating direction method with a Gausssian back substitution proposed in [25] (we call the method LADMMGB here and use the parameter $\alpha = 0.99$ in the Gasussian back substitution step) can also be applied to the original problem (3.76). We have also implemented

Table 3.4: The performance of LADMM, LADMMGB on spectral norm H-weighted NCM problem(3.76) (accuracy = $10^{-5}$). In the table, "lgb" stands for LADMMGB. The computation time is in the format of "hours:minutes:seconds".

| | | | iteration | $\eta_{\text{sncm}}$ | $\eta_{\text{gap}}$ | time |
|---|---|---|---|---|---|---|
| problem | $n_s$ | $\alpha$ | ladmm\|lgb | ladmm\|lgb | ladmm\|lgb | ladmm\|lgb |
| Lymph | 587 | 0.10 | 8401 \| 25000 | 9.9-6 \| 1.4-5 | -1.6-5 \| -2.1-5 | 23:59 \| 1:22:58 |
| Lymph | 587 | 0.05 | 13609 \| 25000 | 9.9-6 \| 2.3-5 | -1.6-5 \| -4.2-5 | 39:29 \| 1:18:50 |

LADMM and LADMMGB in MATLAB. Our experiments show that solving the problem (3.76) directly is much slower than solving the equivalent problem (3.77). Thus, the reformulation of (3.76) to (3.77) is in fact advantageous for both ADMM and ADMMGB. In Table 3.4, for the purpose of illustration we list a couple of detailed numerical results on the performance of LADMM and LADMMGB.

### 3.4.3 Convex quadratic programming (QP)

In this subsection, we consider the following convex quadratic programming problems

$$\min\left\{\frac{1}{2}\langle x,\, Qx\rangle + \langle c,\, x\rangle \mid Ax = b,\ \bar{b} - Bx \in \mathcal{C},\ x \in \mathcal{K}\right\}, \tag{3.79}$$

where vector $c \in \Re^n$ and positive semidefinite matrix $Q \in \mathcal{S}^n_+$ define the linear and quadratic costs for decision variable $x \in \Re^n$, matrices $A \in \Re^{m_E \times n}$ and $B \in \Re^{m_I \times n}$ respectively define the equality and inequality constraints, $\mathcal{C} \subseteq \Re^{m_I}$ is a closed convex cone, e.g., the nonnegative orthant $\mathcal{C} = \{\bar{x} \in \Re^{m_I} \mid \bar{x} \geq 0\}$, $\mathcal{K} \subseteq \Re^n$ is a nonempty simple closed convex set, e.g., $\mathcal{K} = \{x \in \Re^n \mid l \leq x \leq u\}$ with $l, u \in \Re^n$ being given vectors. The dual of (3.79) takes the following form

$$\begin{array}{ll} \max & -\delta^*_{\mathcal{K}}(-z) - \frac{1}{2}\langle x',\, Qx'\rangle + \langle b,\, y\rangle + \langle \bar{b},\, \bar{y}\rangle \\ \text{s.t.} & z - Qx' + A^*y + B^*\bar{y} = c, \quad x' \in \Re^n, \quad \bar{y} \in \mathcal{C}^\circ, \end{array} \tag{3.80}$$

where $\mathcal{C}^\circ$ is the polar cone [53, Section 14] of $\mathcal{C}$. We are interesting in the case when the dimensions $n$ and $m_E + m_I$ are extremely large. In order to handle the equality and inequality constraints simultaneously, as well as to use Algorithm sGS-padmm, we propose to add a slack variable $\bar{x}$ to get the following problem:

$$\begin{array}{ll} \min & \frac{1}{2}\langle x,\, Qx\rangle + \langle c,\, x\rangle \\ \text{s.t.} & \begin{bmatrix} A & \\ B & I \end{bmatrix} \begin{bmatrix} x \\ \bar{x} \end{bmatrix} = \begin{bmatrix} b \\ \bar{b} \end{bmatrix}, \quad x \in \mathcal{K}, \quad \bar{x} \in \mathcal{C}. \end{array} \tag{3.81}$$

The dual of problem (3.81) is given by

$$\begin{array}{ll} \max & (-\delta^*_{\mathcal{K}}(-z) - \delta^*_{\mathcal{C}}(-\bar{z})) - \frac{1}{2}\langle x',\, Qx'\rangle + \langle b,\, y\rangle + \langle \bar{b},\, \bar{y}\rangle \\ \text{s.t.} & \begin{bmatrix} z \\ \bar{z} \end{bmatrix} - \begin{bmatrix} Qx' \\ 0 \end{bmatrix} + \begin{bmatrix} A^* & B^* \\ & I \end{bmatrix} \begin{bmatrix} y \\ \bar{y} \end{bmatrix} = \begin{bmatrix} c \\ 0 \end{bmatrix}. \end{array} \tag{3.82}$$

When we apply our Algorithm sGS-padmm for solving (3.82), if the linear map $B$ is large scale and dense, we can decompose the linear system into several small pieces. More specifically, for the constraints $Bx + \bar{x} = \bar{b}$ and given positive integer $N$, we propose the following decomposition scheme

$$Bx + \bar{x} = \bar{b} \Rightarrow \begin{bmatrix} B_1 & I_1 & & \\ \vdots & & \ddots & \\ B_N & & & I_N \end{bmatrix} \begin{bmatrix} x \\ \bar{x}_1 \\ \vdots \\ \bar{x}_N \end{bmatrix} = \begin{bmatrix} \bar{b}_1 \\ \vdots \\ \bar{b}_N \end{bmatrix}.$$

Note that our Algorithm sGS-padmm also allow us to decompose the linear map $Q$ in the following way:

$$Qx' = [Q_1 \ldots Q_p] \begin{bmatrix} x'_1 \\ \vdots \\ x'_p \end{bmatrix} = Q_1 x'_1 + \ldots + Q_p x'_p.$$

In our numerical experiments, we test our Algorithm sGS-padmm on the convex quadratic programming problems generated from the following binary integer nonconvex quadratic (BIQ) programming:

$$\left\{ \frac{1}{2}\langle x,\, Q_0 x\rangle + \langle c,\, x\rangle \mid x \in \{0,1\}^{n_0} \right\} \tag{3.83}$$

with $Q_0 \in \mathcal{S}^{n_0}$. Let $Y = xx^T$, we have $\langle x,\, Q_0 x\rangle = \langle Y,\, Q_0\rangle$. By relaxing the binary constraint, we can add the following valid inequalities

$$x_i(1-x_j) \geq 0, x_j(1-x_i) \geq 0, (1-x_i)(1-x_j) \geq 0.$$

Since $x \in \{0,1\}^{n_0}$, we know that $\langle x,\, x\rangle = \langle e,\, x\rangle$, where $e := \text{ones}(n_0, 1)$. Hence

$$\langle x,\, Q_0 x\rangle = \langle x,\, (Q + \lambda I)x\rangle - \lambda\langle e,\, x\rangle.$$

Choose $\lambda = \lambda_{\min}(Q_0)$ such that $Q_0 + \lambda I \succeq 0$. Then, we obtain the following convex quadratic programming relaxation:

$$\begin{array}{rl} \min & \frac{1}{2}\langle x,\, (Q_0 + \lambda I)x\rangle + \langle c - \lambda e,\, x\rangle \\ \text{s.t.} & \text{Diag}(Y) - x = 0, \\ & -Y_{ij} + x_i \geq 0,\ -Y_{ij} + x_j \geq 0, \\ & Y_{ij} - x_i - x_j \geq -1,\ \forall i < j, j = 2, \ldots, n_0, \\ & Y \in \mathcal{S}^{n_0},\ Y \geq 0,\ x \geq 0. \end{array} \tag{3.84}$$

Denote $\tilde{n} = (n_0^2 + 3n_0)/2$ and $\tilde{x} := [\mathbf{svec}(Y); x] \in \Re^{\tilde{n}}$. Since the equality constraint in (3.84) is relatively easy, we further add valid equations $A\tilde{x} = b$, where $A \in \Re^{n_0 \times \tilde{n}}$ and $b \in \Re^{n_0}$ are randomly generated. Thus, we can construct the following convex quadratic programming problem:

$$\begin{array}{rl} \min & \frac{1}{2}\langle x,\, (Q_0 + \lambda I)x\rangle + \langle c - \lambda e,\, x\rangle \\ \text{s.t.} & A\tilde{x} = b,\ \text{Diag}(Y) - x = 0, \\ & -Y_{ij} + x_i \geq 0,\ -Y_{ij} + x_j \geq 0, \\ & Y_{ij} - x_i - x_j \geq -1,\ \forall i < j, j = 2, \ldots, n_0, \\ & \tilde{x} := [\mathbf{svec}(Y); x],\ Y \in \mathcal{S}^{n_0},\ Y \geq 0,\ x \geq 0. \end{array} \tag{3.85}$$

We need to emphasis that in problem (3.85), the matrix which defines the quadratic cost is given by $\text{Diag}(0, Q_0 + \lambda I)$. It is in fact a low rank sparse positive semidefinite matrix. In addition, compared with the problem size $\tilde{n}$, matrix $Q_0 \in \Re^{n_0 \times n_0}$ is still quite small. To test our idea of the decomposition of large and dense quadratic term $Q$, we replace the quadratic term in (3.85) by randomly generated instances, i.e.,

$$\begin{array}{ll} \min & \frac{1}{2}\langle \tilde{x}, \widetilde{Q}\tilde{x}\rangle + \langle c - \lambda e, x\rangle \\ \text{s.t.} & A\tilde{x} = b, \ \mathrm{Diag}(Y) - x = 0, \\ & -Y_{ij} + x_i \geq 0, \ -Y_{ij} + x_j \geq 0, \\ & Y_{ij} - x_i - x_j \geq -1, \ \forall i < j, j = 2, \ldots, n_0, \\ & \tilde{x} := [\mathbf{svec}(Y); x], \ Y \in \mathcal{S}^{n_0}, \ Y \geq 0, \ x \geq 0, \end{array} \tag{3.86}$$

where, for simplicity, $\widetilde{Q} \in \Re^{\tilde{n}\times\tilde{n}}$ is a randomly generated positive definite matrix.

Here we compare our algorithm sGS-padmm with Gurobi 6.0 [22] (the state-of-the-art solver for large scale quadratic programming). We have implemented the algorithms sGS-padmm, in Matlab version 7.13. The numerical results reported later are obtained from a workstation running on 64-bit Windows Operating System having 16 cores with 32 Intel Xeon E5-2650 processors at 2.60GHz and 64 GB memory. When we test our sGS-padmm algorithm, we restrict the number of threads used by Matlab to be 1. On the other hand, since Gurobi was built to fully exploit parallelism, we test Gurobi by setting its threads parameter to be 1, 4, 8, 16 and 32, respectively. We also emphasis that for large scale quadratic programming problems, Gurobi need a very large RAM to meet the memory requirement of the Cholesky decomposition, while sGS-padmm is scalable with respect to the memory used to store the problem.

We measure the accuracy of an approximate optimal solution $(x, z, x', s, y, \bar{y})$ for convex quadratic programming (3.79) and its dual (3.80) by using the following relative residual obtained from the general optimality condition (3.63):

$$\eta_{\text{qp}} = \max\{\eta_P, \eta_D, \eta_Q, \eta_z, \eta_{\bar{y}}\}, \tag{3.87}$$

where

$$\eta_P = \frac{\|AX - b\|}{1+\|b\|}, \quad \eta_D = \frac{\|z - Qx' + s + A^*y + B^*\bar{y} - C\|}{1+\|c\|},$$
$$\eta_Z = \frac{\|x - \Pi_{\mathcal{K}}(x-z)\|}{1+\|x\|+\|z\|}, \quad \eta_{\bar{y}} = \frac{\|\bar{y} - \Pi_{\mathcal{C}^\circ}(\bar{y} - Bx + \bar{b})\|}{1+\|\bar{y}\|+\|Bx\|},$$
$$\eta_Q = \frac{\|Qx - Qx'\|}{1+\|Qx\|}.$$

We terminate the sGS-PADMM when $\eta_{\rm qp} < 10^{-5}$ with the maximum number of iterations set at 25000. For Gurobi, we also set the error tolerance to be $10^{-5}$. However, due to the natural of the interior algorithm, Gurobi generally will achieve higher accuracy than $10^{-5}$.

Table 3.5 reports detailed numerical results for sGS-PADMM and Gurobi for solving convex quadratic programming problems (3.85). The first three columns of the table give the problem name, the dimension of the variable, the number of linear equality constraints and inequality constraints, respectively. Then, we list in the fourth column the block numbers of our decomposition with respect to the linear equality, inequality constraints and quadratic term. We list the total number of iterations and the running time for sGS-PADMM using only one thread for computation. Meanwhile, for comparison purpose, we list all the running times of Gurobi using 1, 4, 8, 16 and 32 threads, respectively. The memory used by Gurobi during computation is listed in the last column. As can be observed, in term of running time, sGS-PADMM is comparable with Gurobi on the medium size problems. In fact, sGS-PADMM is much faster when Gurobi use only 1 thread. When the problem size grows, sGS-PADMM turns out to be faster than Gurobi, even Gurobi use all 32 threads for computation. One can see that our Algorithm sGS-PADMM is scalable with respect to the problem dimension.

Table 3.5: The performance of sGS-PADMM on BIQ-QP problems (dual of (3.85)) (accuracy $= 10^{-5}$). In the table, "sGS" stands for sGS-PADMM. The computation time is in the format of "hours:minutes:seconds".

<table>
<tr><th></th><th>$(\mathrm{A},\mathrm{B},\mathrm{Q})_{\mathrm{blk}}$</th><th>iters</th><th colspan="2">time</th><th>memory</th></tr>
<tr><th>problem| n | $m_E, m_I$</th><th>sGS</th><th>sGS</th><th>sGS(1)</th><th>Gurobi(1|4|8|16|32)</th><th>Gurobi</th></tr>
<tr><td>be100.1 |5150 |200,14850</td><td>(2,25,1)</td><td>2143</td><td>58</td><td>2:37|58|35|26|25</td><td>0.3 GB</td></tr>
<tr><td></td><td>(2,50,1)</td><td>2925</td><td>1:42</td><td></td><td></td></tr>
<tr><td></td><td>(2,100,1)</td><td>2770</td><td>2:17</td><td></td><td></td></tr>
<tr><td>be120.3.1 |7380 |240,21420</td><td>(2,25,1)</td><td>2216</td><td>1:32</td><td>6:37|2:44|1:31|1:01|1:08</td><td>0.6 GB</td></tr>
<tr><td></td><td>(2,50,1)</td><td>2492</td><td>2:23</td><td></td><td></td></tr>
<tr><td></td><td>(2,100,1)</td><td>2864</td><td>3:57</td><td></td><td></td></tr>
<tr><td>be150.3.1 |11475 |300,33525</td><td>(2,25)</td><td>2500</td><td>3:56</td><td>26:16|8:46|5:02|3:11|3:49</td><td>1.5 GB</td></tr>
<tr><td></td><td>(2,50,1)</td><td>2918</td><td>4:33</td><td></td><td></td></tr>
<tr><td></td><td>(2,100,1)</td><td>3324</td><td>6:41</td><td></td><td></td></tr>
<tr><td>be200.3.1 |20300|400,59700</td><td>(2,25)</td><td>3310</td><td>13:09</td><td>2:07:52|45:58|25:50|14:19 |13:32</td><td>5.0 GB</td></tr>
<tr><td></td><td>(2,50,1)</td><td>3596</td><td>11:37</td><td></td><td></td></tr>
<tr><td></td><td>(2,100,1)</td><td>4145</td><td>15:33</td><td></td><td></td></tr>
<tr><td>be250.1 |31625 |500,93375</td><td>(2,25)</td><td>2899</td><td>24:21</td><td>8:12:36|2:21:13|1:46:45|53:58 |40:51</td><td>10.0 GB</td></tr>
<tr><td></td><td>(2,50,1)</td><td>3625</td><td>22:41</td><td></td><td></td></tr>
<tr><td></td><td>(2,100,1)</td><td>4440</td><td>29:11</td><td></td><td></td></tr>
</table>

In figure 3.2, we present the performance profile in terms of the number of iterations and computing time for sGS-PADMM on (3.85) by decomposing the inequality constraints into different number of blocks. More specifically, for problem be100.1, we test our Algorithm sGS-PADMM with the decomposition parameters chosen as $(\mathrm{A},\mathrm{Q})_{\mathrm{blk}} = (2,1)$ and $\mathrm{B}_{\mathrm{blk}} = 1, 2, \ldots, 50$. It is interesting to note the running time at $\mathrm{B}_{\mathrm{blk}} = 1$ is approximately 7 times of the running time at $\mathrm{B}_{\mathrm{blk}} = 1$. Moreover, although the decomposition brings more iterations, the largest iterations number (reached at $\mathrm{B}_{\mathrm{blk}} = 47$) is only 2 times of the smallest iterations number (reached at $\mathrm{B}_{\mathrm{blk}} = 1$). These observations clearly state that it is in fact good to do sGS-PADMM style decomposition for convex quadratic decomposition problems with many linear equality and inequality constraints.

Table 3.6 reports detailed numerical results for sGS-PADMM and Gurobi in solving convex quadratic programming problems (3.86). As can be observed, for these

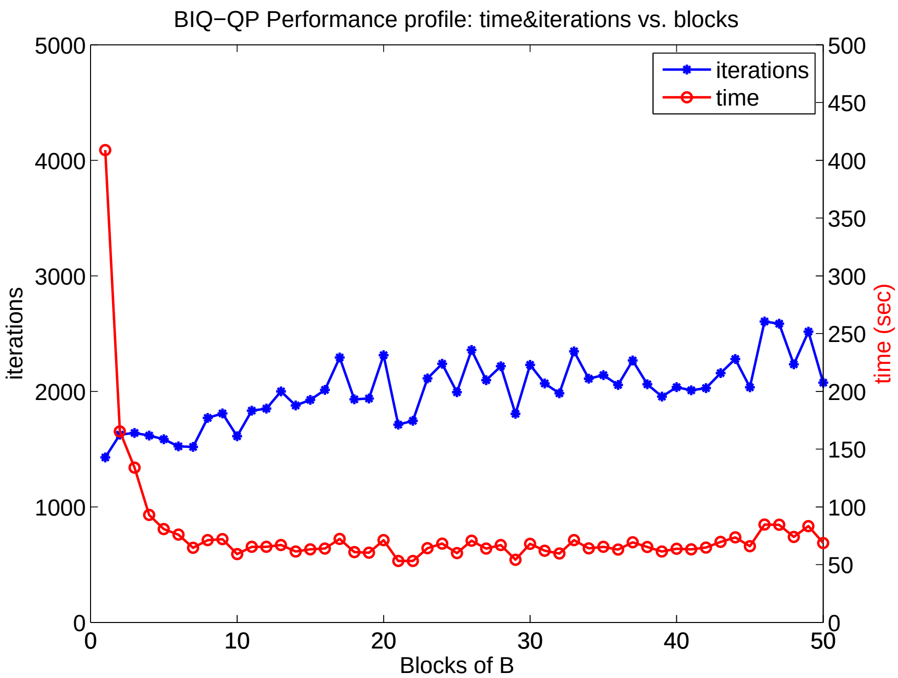


Figure 3.2: Performance profile of sGS-PADMM for solving (3.85) in terms of iter. and time with different number of $B_{blk}$

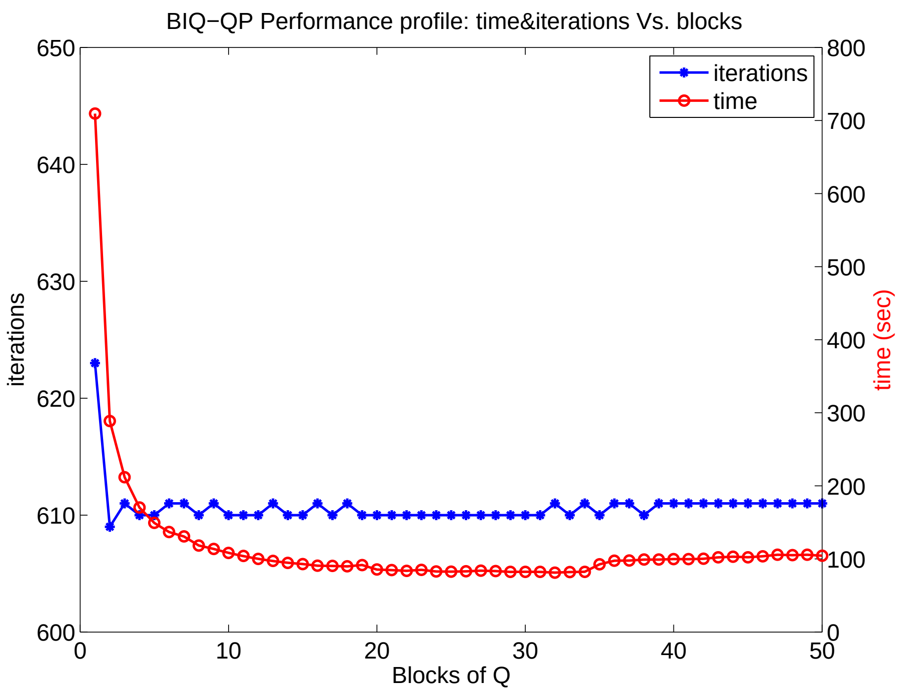


Figure 3.3: Performance profiles of sGS-PADMM for solving (3.86) in terms of iter. and time with different number of $Q_{blk}$.

large scale problems with large and dense quadratic term $Q$, sGS-PADMM can be significantly faster than Gurobi. In addition, sGS-PADMM, free from the large memory requirements as for Gurobi, can solve these problems on a normal PC without large RAM. Above facts indicate that as a Phase I algorithm, sGS-PADMM can quickly generated a good initial point.

Table 3.6: The performance of sGS-PADMM on randomly generated BIQ-QP problems (dual of (3.86)) (accuracy = $10^{-5}$). In the table, "sGS" stands for sGS-PADMM. The computation time is in the format of "hours:minutes:seconds".

| | $(A,B,Q)_{blk}$ | iters | time | | memory |
|---|---|---|---|---|---|
| problem \| n \| $m_E, m_I$ | sGS | sGS | sGS(1) | Gurobi(1\|4\|8\|16\|32) | Gurobi |
| be100.1 \|5150 \|200,14850 | (2,25,25) | 789 | 47 | 27:57\|7:52\|4:23\|3:31\|3:36 | 1.4 GB |
| | (2,50,50) | 1057 | 1:34 | | |
| | (2,100,100) | 1134 | 2:58 | | |
| be120.3.1 \|7380 \|240,21420 | (2,25,25) | 528 | 40 | 1:34:46\|26:58\|14:46\|11:43\|9:37 | 3.0 GB |
| | (2,50,50) | 625 | 1:15 | | |
| | (2,100,100) | 810 | 2:48 | | |
| be150.3.1 \|11475 \|300,33525 | (2,25,25) | 515 | 1:19 | 6:21:43\|1:45:21\|54:39\|39:46\|32:52 | 8.0 GB |
| | (2,50,50) | 611 | 1:38 | | |
| | (2,100,100) | 715 | 3:26 | | |
| be200.3.1 \|20300 \|400,59700 | (2,25,25) | 1139 | 6:45 | 36:30:08\|8:32:49\|5:14:24\|3:29:43\|3:07:01 | 25.0 GB |
| | (2,50,50) | 783 | 4:28 | | |
| | (2,100,100) | 839 | 6:30 | | |
| be250.1 \|31625 \|500,93375 | (2,25,25) | 644 | 10:04 | -:-:-\|-:-:-\|-:-:-\|-:-:-\| over 24:00:00* | 62.0† GB |
| | (2,50,50) | 718 | 9:29 | | |
| | (2,100,100) | 874 | 11:38 | | |

In Figure 3.3, we present the performance profiles in terms of the number of iterations and computing time for sGS-PADMM for solving (3.86) by decomposing the quadratic term $Q$ into different number of blocks. More specifically, for problem be150.3.1, we test our Algorithm sGS-PADMM with the decomposition parameters

*Even we use all the 32 threads, Gurobi is still in the pre-solving step after 24 hours.

† In fact, for this problem, Gurobi runs out of memory, although our work station has 64GB RAM.

chosen as $(\mathrm{A}, \mathrm{B})_{\mathrm{blk}} = (2, 50)$ and $\mathrm{Q}_{\mathrm{blk}} = 1, 2, \ldots, 50$. One can obtain similar conclusion as before, i.e., for these problems, it is in fact good to do sGS-padmm style decomposition on quadratic term $Q$.

In this Chapter, we have proposed a symmetric Gauss-Seidel based convergent yet efficient proximal ADMM for solving convex composite quadratic programming problems, with a coupling linear equality constraint. The ability of dealing with non-separable convex quadratic functions in the objective function makes the proposed algorithm very flexible in solving various convex optimization problems. By conducting numerical experiments on large scale convex quadratic programming with many equality and inequality constraints, QSDP and its extensions, we have presented convincing numerical results to demonstrate the superior performance of our proposed sGS-padmm. As is mentioned before, our primary motivation of introducing this sGS-padmm is to quickly generate a good initial point so as to warm-start methods which have fast local convergence properties. For standard linear SDP and linear SDP with doubly nonnegative constraints, this has already been done by Zhao, Sun and Toh in [73] and Yang, Sun and Toh in [69], respectively. Naturally, our next target is to extend the approach of [73, 69] to solve convex composite quadratic programming problems with an initial point generated by sGS-padmm.

Chapter 4

# Phase II: An inexact proximal augmented Lagrangian method for convex composite quadratic programming

In this Chapter, we discuss our Phase II framework for solving the convex composite optimization problem. The purpose of this phase is to obtain high accurate solutions efficiently after warm-started by our Phase I algorithm.

Consider the compact form of our general convex composite quadratic optimization model

$$\begin{aligned} \min \quad & \theta(y_1) + f(y) + \varphi(z_1) + g(z) \\ \text{s.t.} \quad & \mathcal{A}^* y + \mathcal{B}^* z = c, \end{aligned} \tag{4.1}$$

where $\theta : \mathcal{Y}_1 \to (-\infty, +\infty]$ and $\varphi : \mathcal{Z}_1 \to (-\infty, +\infty]$ are simple closed proper convex functions, $f : \mathcal{Y}_1 \times \mathcal{Y}_2 \times \ldots \times \mathcal{Y}_p \to \Re$ and $g : \mathcal{Z}_1 \times \mathcal{Z}_2 \times \ldots \times \mathcal{Z}_q \to \Re$ are convex quadratic functions with $\mathcal{Y} = \mathcal{Y}_1 \times \mathcal{Y}_2 \times \ldots \times \mathcal{Y}_p$ and $\mathcal{Z} = \mathcal{Z}_1 \times \mathcal{Z}_2 \times \ldots \times \mathcal{Z}_q$. For notational convenience, we write

$$\theta_f(y) := \theta(y_1) + f(y) \quad \forall y \in \mathcal{Y} \quad \text{and} \quad \varphi_g(z) := \varphi(z_1) + g(z) \quad \forall z \in \mathcal{Z}. \tag{4.2}$$

Given $\sigma > 0$, we denote by $l$ the Lagrangian function for (4.1):

$$l(y, z; x) = \theta_f(y) + \varphi_g(z) + \langle x, \mathcal{A}^* y + \mathcal{B}^* z - c \rangle, \tag{4.3}$$

and by $\mathcal{L}_\sigma$ the augmented Lagrangian function associated with problem (4.1):

$$\mathcal{L}_\sigma(y, z; x) = \theta_f(y) + \varphi_g(z) + \langle x, \mathcal{A}^* y + \mathcal{B}^* z - c \rangle + \frac{\sigma}{2}\|\mathcal{A}^* y + \mathcal{B}^* z - c\|^2. \tag{4.4}$$

## 4.1 A proximal augmented Lagrangian method of multipliers

For our Phase II algorithm for solving (4.1), we propose the following proximal minimization framework for given positive parameter $\sigma_k$:

$$\begin{aligned}&(y^{k+1}, z^{k+1}, x^{k+1})\\ &= \arg\max_{x}\min_{y,z}\{l(y, z; x) + \frac{1}{2\sigma_k}\|y - y^k\|^2_{\Lambda_1} + \frac{1}{2\sigma_k}\|z - z^k\|^2_{\Lambda_2} - \frac{1}{2\sigma_k}\|x - x^k\|^2\},\end{aligned} \tag{4.5}$$

where $\Lambda_1 : \mathcal{Y} \to \mathcal{Y}$ and $\Lambda_2 : \mathcal{Z} \to \mathcal{Z}$ are two self-adjoint, positive definite linear operators. An inexact form of the implementation works as follows:

**Algorithm pALM: A proximal augmented Lagrangian method of multipliers for solving** (4.1)

Let $\sigma_0, \sigma_\infty > 0$ be given parameters. Choose $(y^0, z^0, x^0) \in \text{dom}(\theta_f) \times \text{dom}(\varphi_g) \times \mathcal{X}$. For $k = 0, 1, 2, ...$, generate $(y^{k+1}, z^{k+1})$ and $x^{k+1}$ according to the following iteration.

**Step 1.** Compute

$$(y^{k+1}, z^{k+1}) \approx \text{argmin}_{y,z}\{\mathcal{L}_{\sigma_k}(y, z; x^k) + \frac{1}{2\sigma_k}\|y - y^k\|^2_{\Lambda_1} + \frac{1}{2\sigma_k}\|z - z^k\|^2_{\Lambda_2}\}. \tag{4.6}$$

**Step 2.** Compute

$$x^{k+1} = x^k + \sigma_k(\mathcal{A}^* y^{k+1} + \mathcal{B}^* z^{k+1} - c).$$

**Step 3.** Update $\sigma_{k+1} \uparrow \sigma_\infty \leq \infty$ .

Note that the only difference between our pALM and the classical proximal augmented Lagrangian method is that we put more general positive definite terms $\frac{1}{2\sigma_k}\|y-y^k\|^2_{\Lambda_1}$ and $\frac{1}{2\sigma_k}\|z-z^k\|^2_{\Lambda_2}$ in (4.5) instead of multiples of identity operators. In the subsequent discussions readers will find that this modification not only necessary but also may generate easier subproblems. Before that, we first show that our pALM in fact can be regarded as a primal-dual proximal point algorithm (PPA) so that the nice convergence properties still hold.

Define an operator $\mathcal{T}_l$ by

$$\mathcal{T}_l(y,z,x) := \{(y',z',x') \mid (y',z',-x') \in \partial l(y,z;x)\},$$

whose corresponding inverse operator is given by

$$\mathcal{T}_l^{-1}(y',z',x') := \arg\min_{y,z}\max_{x}\{l(y,z;x) - \langle y',y\rangle - \langle z',z\rangle + \langle x',x\rangle\}. \tag{4.7}$$

Let $\Lambda = \mathrm{Diag}\,(\Lambda_1,\Lambda_2,I) \succ 0$ and define function

$$\widetilde{l}(y,z,x) \equiv l(\Lambda^{-\frac{1}{2}}(y,z,x)) \quad \forall (y,z,x) \in \mathcal{Y}\times\mathcal{Z}\times\mathcal{X}.$$

Similarly, we define an operator $\mathcal{T}_{\widetilde{l}}$ associated with $\widetilde{l}$, by

$$\mathcal{T}_{\widetilde{l}}(y,z,x) := \{(y',z',x') \mid (y',z',-x') \in \partial\widetilde{l}(y,z;x)\}.$$

We know by simple calculations that

$$\mathcal{T}_{\widetilde{l}}(y,z,x) \equiv \Lambda^{-\frac{1}{2}}\mathcal{T}_l(\Lambda^{-\frac{1}{2}}(y,z,x)) \quad \forall (y,z,x) \in \mathcal{Y}\times\mathcal{Z}\times\mathcal{X}$$

and $\mathcal{T}_{\widetilde{l}}^{-1}(0) = \Lambda^{\frac{1}{2}}\mathcal{T}_l^{-1}(0)$. Since $\mathcal{T}_l$ is a maximal monotone operator [53, Corollary 37.5.2], we know that $\mathcal{T}_{\widetilde{l}}$ is also a maximal monotone operator.

**Proposition 4.1.** *Let $\{(y^k,z^k,x^k)\}$ be the sequence generated by (4.5). Then,*

$$(y^{k+1},z^{k+1},x^{k+1}) = \Lambda^{-\frac{1}{2}}(\mathcal{I}+\sigma_k\mathcal{T}_{\widetilde{l}})^{-1}(\Lambda^{\frac{1}{2}}(y^k,z^k,x^k)). \tag{4.8}$$

*Thus pALM can be viewed as a generalized PPA algorithm for solving $0 \in \mathcal{T}_{\widetilde{l}}(y,z,x)$.*

**Proof.** By combine [55, Theorem 5] and Proposition 2.2, we can easily prove the required results. □

Next, we discuss the stopping criteria for the subproblem (4.6) in Algorithm pALM. Assume that $\lambda_{\min}$ and $\lambda_{\max}$ ($\lambda_{\max} \geq \lambda_{\min} > 0$) are the smallest and largest eigenvalues of the self-adjoint positive definite operator $\Lambda$, respecitvely. Denote $w = (y, z, x)$ and $\widetilde{w} = \Lambda^{\frac{1}{2}} w$. Let $\mathcal{S}_k(w) = \mathcal{T}_l(w) + \sigma_k^{-1}\Lambda(w - w^k)$ and $\widetilde{\mathcal{S}}_k(\widetilde{w}) = \mathcal{T}_{\widetilde{l}}(\widetilde{\omega}) + \sigma_k^{-1}(\widetilde{w} - \widetilde{w}^k)$. We use the following stopping criteria proposed in [55, 54] to terminate the subproblem in pALM:

$$\begin{array}{lll} (A) & \operatorname{dist}(0, \mathcal{S}_k(w^{k+1})) \leq \dfrac{\varepsilon_k\sqrt{\lambda_{\min}}}{\sigma_k}, & \displaystyle\sum_{k=0}^{\infty} \varepsilon_k < +\infty, \\ (B) & \operatorname{dist}(0, \mathcal{S}_k(w^{k+1})) \leq \dfrac{\delta_k \lambda_{\min}}{\sigma_k}\|w^{k+1} - w^k\|, & \displaystyle\sum_{k=0}^{\infty} \delta_k < +\infty. \end{array} \tag{4.9}$$

The following proposition gives the relation between $\operatorname{dist}(0, \mathcal{S}(w))$ and $\operatorname{dist}(0, \widetilde{\mathcal{S}}_k(\widetilde{w}))$.

**Proposition 4.2.** *It holds that*

$$\sqrt{\lambda_{\min}}\operatorname{dist}(0, \widetilde{\mathcal{S}}_k(\widetilde{w}^{k+1})) \leq \operatorname{dist}(0, \mathcal{S}_k(w^{k+1})). \tag{4.10}$$

*Therefore, (A) implies*

$$(A') \qquad \operatorname{dist}(0, \widetilde{\mathcal{S}}_k(\widetilde{w}^{k+1})) \leq \frac{\varepsilon_k}{\sigma_k}, \quad \sum_{k=0}^{\infty} \varepsilon_k < +\infty$$

*and (B) implies*

$$(B') \qquad \operatorname{dist}(0, \widetilde{\mathcal{S}}_k(\widetilde{w}^{k+1})) \leq \frac{\delta_k}{\sigma_k}\|\widetilde{w}^{k+1} - \widetilde{w}^k\|, \quad \sum_{k=0}^{\infty} \delta_k < +\infty,$$

*respectively.*

**Proof.** Since $\mathcal{T}_l(w^{k+1})$ is a closed and convex set, there exists $u^{k+1} \in \mathcal{T}_l(w^{k+1})$, such that $\operatorname{dist}(0, \mathcal{S}_k(w^{k+1})) = \|u^{k+1} + \sigma_k^{-1}\Lambda(w - w^k)\|$. Let $\widetilde{u}^{k+1} = \Lambda^{-\frac{1}{2}} u^{k+1}$, we have that $\widetilde{u}^{k+1} \in \mathcal{T}_{\widetilde{l}}(\widetilde{w}^{k+1})$. Therefore,

$$\begin{aligned} \|u^{k+1} + \sigma_k^{-1}\Lambda(w - w^k)\| &= \|\Lambda^{\frac{1}{2}}(\widetilde{u}^{k+1} + \sigma_k^{-1}(\widetilde{w}^{k+1} - \widetilde{w}^k))\| \\ &\geq \sqrt{\lambda_{\min}}\|\widetilde{u}^{k+1} + \sigma_k^{-1}(\widetilde{w}^{k+1} - \widetilde{w}^k)\| \\ &\geq \sqrt{\lambda_{\min}}\operatorname{dist}(0, \widetilde{\mathcal{S}}_k(\widetilde{u}^{k+1})). \end{aligned}$$

That is $\sqrt{\lambda_{\min}}\text{dist}(0, \widetilde{\mathcal{S}}_k(\widetilde{w}^{k+1})) \leq \text{dist}(0, \mathcal{S}_k(w^{k+1}))$.

Criterion (B′) can be obtained by observing the fact that

$$\|w^{k+1} - w^k\| = \|\Lambda^{-\frac{1}{2}}(\widetilde{w}^{k+1} - \widetilde{w}^k)\| \leq \frac{\|\widetilde{w}^{k+1} - \widetilde{w}^k\|}{\sqrt{\lambda_{\min}}}.$$

The proof of the proposition is completed. □

The global convergence of the pALM algorithm follows from Rockafellar [55, 54] without much difficulty.

**Theorem 4.3.** *Suppose that Assumption 4 holds and the solutions set of problem* (4.1) *is nonempty. Then the sequence* $\{(y^k, z^k, x^k)\}$ *generated by pALM with stopping criterion* (A) *is bounded and* $(y^k, z^k)$ *converges to the optimal solution of (4.1),* $x^k$ *converges to the optimal solution of the dual problem.*

To study the local convergence rate of our proposed Algorithm pALM, we need the following error bound assumption proposed in [38].

**Assumption 6** (Error bound assumption)**.** *For a maximal monotone operator* $\mathcal{T}(\xi)$ *with* $\mathcal{T}^{-1}(0) := \Xi$ *is nonempty, there exist* $\varepsilon > 0$ *and* $a > 0$ *such that*

$$\forall \eta \in \mathcal{B}(0, \varepsilon) \quad \text{and} \quad \forall \xi \in \mathcal{T}^{-1}(\eta), \quad \text{dist}(\xi, \Xi) \leq a\|\eta\|. \tag{4.11}$$

**Remark 4.4.** The above assumption contains the case that $\mathcal{T}^{-1}$ is locally Lipschitz at 0, which was used extensively in [55, 54] for deriving the convergence rate of proximal point algorithms.

**Remark 4.5.** The error bound assumption (4.11) holds automatically when $\mathcal{T}_l$ is a polyhedral multifunction [52]. Specifically, for the convex quadratic programming (3.80), if the simple convex set $\mathcal{K}$ is a polyhedra, then Assumption 6 holds for the corresponding $\mathcal{T}_l$.

In the next proposition, we discuss the relation between error bound assumptions on $\mathcal{T}_l$ and $\mathcal{T}_{\widetilde{l}}$.

**Proposition 4.6.** *Assume that* $\Omega := \mathcal{T}_l^{-1}(0)$ *is nonempty and that there exist* $\varepsilon > 0$ *and* $a > 0$ *such that*

$$\forall u \in \mathcal{B}(0, \varepsilon) \quad \text{and} \quad \forall w \in \mathcal{T}_l^{-1}(u), \quad \text{dist}(w, \Omega) \leq a\|u\|.$$

*Then, we have* $\widetilde{\Omega} := \mathcal{T}_{\widetilde{l}}^{-1}(0) = \Lambda^{\frac{1}{2}}\Omega$ *is nonempty and*

$$\forall \widetilde{u} \in \mathcal{B}(0, \frac{\varepsilon}{\sqrt{\lambda_{\max}}}) \quad \text{and} \quad \forall \widetilde{w} \in \mathcal{T}_{\widetilde{l}}^{-1}(\widetilde{u}), \quad \text{dist}(\widetilde{w}, \widetilde{\Omega}) \leq a\lambda_{\max}\|\widetilde{u}\|,$$

*i.e., the error bound assumption also holds for* $\mathcal{T}_{\widetilde{l}}$.

**Proof.** For any given $\widetilde{u} \in \mathcal{B}(0, \frac{\varepsilon}{\sqrt{\lambda_{\max}}})$ and $\widetilde{w} \in \mathcal{T}_{\widetilde{l}}^{-1}(\widetilde{u})$, let

$$u = \Lambda^{\frac{1}{2}}\widetilde{u} \quad \text{and} \quad w = \Lambda^{-\frac{1}{2}}\widetilde{w}.$$

We have that $\|u\| = \|\Lambda^{\frac{1}{2}}\widetilde{u}\| \leq \sqrt{\lambda_{\max}}\|\widetilde{u}\| \leq \varepsilon$ and $w \in \mathcal{T}_l^{-1}(u)$. Thus, $\text{dist}(w, \Omega) \leq a\|u\|$. Since $\Omega$ is closed and convex, there exist $\omega \in \Omega$ such that $\text{dist}(w, \Omega) = \|w - \omega\|$. Let $\widetilde{\omega} = \Lambda^{\frac{1}{2}}\omega$, then we know that $\widetilde{\omega} \in \widetilde{\Omega}$ and

$$\begin{aligned}\text{dist}(w, \Omega) &= \|w - \omega\| = \|\Lambda^{-\frac{1}{2}}(\widetilde{w} - \widetilde{\omega})\| \\ &\geq \frac{\|\widetilde{w} - \widetilde{\omega}\|}{\sqrt{\lambda_{\max}}} \geq \frac{\text{dist}(\widetilde{w}, \widetilde{\Omega})}{\sqrt{\lambda_{\max}}}.\end{aligned}$$

Therefore,

$$\frac{\text{dist}(\widetilde{w}, \widetilde{\Omega})}{\sqrt{\lambda_{\max}}} \leq a\|u\| \leq a\sqrt{\lambda_{\max}}\|\widetilde{u}\|.$$

This completes the proof of the proposition. □

After all these preparations, we are now ready to present the local linear convergence of the Algorithm pALM.

**Theorem 4.7.** *Suppose Assumption 6 holds for* $\mathcal{T}_l$, *i.e.,* $\Omega = \mathcal{T}_l^{-1}(0)$ *is nonempty and there exist* $\varepsilon > 0$ *and* $a > 0$ *such that*

$$\forall u \in \mathcal{B}(0, \varepsilon) \quad \text{and} \quad \forall w \in \mathcal{T}_l^{-1}(u), \quad \text{dist}(w, \Omega) \leq a\|u\|.$$

*Let* $\{w^k\} = \{(y^k, z^k; x^k)\}$ *be the sequence generated by pALM with stopping criterion* $(\mathrm{B}')$. *Recall that* $\widetilde{w}^k = \Lambda^{\frac{1}{2}} w^k$ *and* $\widetilde{\Omega} = \Lambda^{\frac{1}{2}}\Omega$. *Then, for all* $k$ *sufficiently large,*

$$\mathrm{dist}(\widetilde{w}^{k+1}, \widetilde{\Omega}) \leq \theta_k \mathrm{dist}(\widetilde{w}^k, \widetilde{\Omega}), \tag{4.12}$$

*where* $\theta_k = (\frac{a\sqrt{\lambda_{\max}}}{\sqrt{a^2\lambda_{\max} + \sigma_k^2}} + 2\delta_k)(1-\delta_k)^{-1} \to \frac{a\sqrt{\lambda_{\max}}}{\sqrt{a^2\lambda_{\max} + \sigma_\infty^2}}$ *as* $k \to +\infty$.

**Proof.** By combining Proposition 4.6 and Theorem 2.1 in [38], we can readily obtain the desired results. □

Note that in practice it is difficult to compute $\mathrm{dist}(0, S_k(w^{k+1}))$ in criteria (A) and (B) for terminating Algorithm pALM. Hence, we need implementable criteria for terminating Algorithm pALM. Denote

$$\hat{y}^{k+1} = \mathrm{Prox}_{\hat{\theta}}(y^{k+1} - \nabla_y h_k(y^{k+1}, z^{k+1})) \quad \text{and} \quad \hat{z}^{k+1} = \mathrm{Prox}_{\hat{\varphi}}(z^{k+1} - \nabla_z h_k(y^{k+1}, z^{k+1})).$$

Thus

$$0 \in \partial\hat{\theta}(\hat{y}^{k+1}) + \hat{y}^{k+1} - y^{k+1} + \nabla_y h_k(y^{k+1}, z^{k+1}), \tag{4.13}$$

which implies

$$y^{k+1} - \hat{y}^{k+1} + \nabla_y h_k(\hat{y}^{k+1}, \hat{z}^{k+1}) - \nabla_y h_k(y^{k+1}, z^{k+1}) \in \partial\hat{\theta}(\hat{y}^{k+1}) + \nabla_y h_k(\hat{y}^{k+1}, \hat{z}^{k+1}). \tag{4.14}$$

Similarly we can also get

$$z^{k+1} - \hat{z}^{k+1} + \nabla_z h_k(\hat{y}^{k+1}, \hat{z}^{k+1}) - \nabla_z h_k(y^{k+1}, z^{k+1}) \in \partial\hat{\varphi}(\hat{z}^{k+1}) + \nabla_z h_k(\tilde{y}^{k+1}, \hat{z}^{k+1}). \tag{4.15}$$

Let $\hat{x}^{k+1} = x^k + \sigma_k(\mathcal{A}^*\hat{y}^{k+1} + \mathcal{B}^*\hat{z}^{k+1} - c)$ and $\hat{w}^{k+1} = (\hat{y}^{k+1}, \hat{z}^{k+1}, \hat{x}^{k+1})$. By [54, Propositon 7], we have

$$(\partial_y \mathcal{L}_{\sigma_k}(\hat{y}^{k+1}, \hat{z}^{k+1}, x^k), \partial_z \mathcal{L}_{\sigma_k}(\hat{y}^{k+1}, \hat{z}^{k+1}, x^k), \sigma_k^{-1}(x^k - \hat{x}^{k+1})) \in \mathcal{T}_l(\hat{w}^{k+1}).$$

Recall that $\mathcal{S}_k(w) = \mathcal{T}_l(w) + \sigma_k^{-1}\Lambda(w - w^k)$. Thus, we know that

$$\begin{aligned}
\mathrm{dist}(0, \mathcal{S}_k(\hat{w}^{k+1})) &\leq \mathrm{dist}(0, \mathcal{T}_l(\hat{w}^{k+1})) + \|\sigma_k^{-1}\Lambda(\hat{w}^{k+1} - w^k)\| \\
&\leq \mathrm{dist}(0, \partial_y \mathcal{L}_{\sigma_k}(\hat{y}^{k+1}, \hat{z}^{k+1}, x^k)) + \mathrm{dist}(0, \partial_z \mathcal{L}_{\sigma_k}(\hat{y}^{k+1}, \hat{z}^{k+1}, x^k)) \\
&\quad + \sigma_k^{-1}\|x^k - \hat{x}^{k+1}\| + \lambda_{\max}\sigma_k^{-1}\|w^k - \hat{w}^{k+1}\| \\
&\leq \|y^{k+1} - \hat{y}^{k+1} + \nabla_y h_k(\hat{y}^{k+1}, \hat{z}^{k+1}) - \nabla_y h_k(y^{k+1}, z^{k+1})\| \\
&\quad + \|z^{k+1} - \hat{z}^{k+1} + \nabla_z h_k(\hat{y}^{k+1}, \hat{z}^{k+1}) - \nabla_z h_k(y^{k+1}, z^{k+1})\| \\
&\quad + \sigma_k^{-1}\|x^k - \hat{x}^{k+1}\| + \lambda_{\max}\sigma_k^{-1}\|w^k - \hat{w}^{k+1}\| \\
&\leq (1 + L_{h_k})(\|y^{k+1} - \hat{y}^{k+1}\| + \|z^{k+1} - \hat{z}^{k+1}\|) + \sigma_k^{-1}\|x^k - \hat{x}^{k+1}\| \\
&\quad + \lambda_{\max}\sigma_k^{-1}\|w^k - \hat{w}^{k+1}\|,
\end{aligned}$$

where $L_{h_k}$ is the the Lipschitz constant of $\nabla h_k$. Therefore, we obtain a computable upper bound for $\mathrm{dist}(0, \mathcal{S}_k(\hat{w}^{k+1}))$. Then, the implementable criteria for terminating Algorithm pALM can be easily constructed.

### 4.1.1 An inexact alternating minimization method for inner subproblems

In this subsection, we will introduce an inexact alternating minimization method for solving the inner subproblem (4.6). Consider the following problem:

$$\min_{u \in \mathcal{U}, v \in \mathcal{V}} H(u, v) := p(u) + q(v) + h(u, v), \tag{4.16}$$

where $\mathcal{U}$ and $\mathcal{V}$ are two real finite dimensional Euclidean spaces, $p : \mathcal{U} \to (-\infty, +\infty]$ and $q : \mathcal{V} \to (-\infty, +\infty]$ are two closed proper convex functions and $h : \mathcal{U} \times \mathcal{V} \to (-\infty, +\infty]$ is a closed proper convex function and is continuous differentiable on some open neighborhoods of $\mathrm{dom}(p) \times \mathrm{dom}(q)$. We propose the following inexact alternating minimization method:

$$\begin{cases} u^{k+1} \approx \operatorname{argmin}_u \{p(u) + h(u, v^k)\}, \\ v^{k+1} \approx \operatorname{argmin}_v \{q(v) + h(u^{k+1}, v)\}. \end{cases} \tag{4.17}$$

Given $\varepsilon_1 > 0, \varepsilon_2 > 0$, the following criteria are used to terminate the above subproblems:

$$\begin{cases} H(u^{k+1}, v^k) \leq H(u^k, v^k) - \varepsilon_1 \|r_1^{k+1}\|, \\ H(u^{k+1}, v^{k+1}) \leq H(u^{k+1}, v^k) - \varepsilon_2 \|r_2^{k+1}\|, \end{cases} \tag{4.18}$$

where

$$\begin{cases} r_1^{k+1} := \operatorname{prox}_p(u^{k+1} - \nabla_u h(u^{k+1}, v^k)) - u^{k+1}, \\ r_2^{k+1} := \operatorname{prox}_q(v^{k+1} - \nabla_v h(u^{k+1}, v^{k+1})) - v^{k+1}. \end{cases}$$

We make the following assumption:

**Assumption 7.** *For a given* $(u^0, v^0) \in \mathcal{U} \times \mathcal{V}$*, the set* $S := \{(u, v) \in \mathcal{U} \times \mathcal{V} \mid H(u, v) \leq H(u^0, v^0)\}$ *is compact and* $H(\cdot)$ *is continuous on* $S$*.*

**Assumption 8.** *For arbitrary* $u^k \in \operatorname{dom}(p)$ *and* $v^k \in \operatorname{dom}(q)$*, each of the optimization problems in* (4.17) *admits a solution.*

Next, we establish the convergence of the proposed inexact alternating minimization method.

**Lemma 4.8.** *Given* $(u^k, v^k) \in \operatorname{int}(\operatorname{dom}(p) \times \operatorname{dom}(q))$*,* $u^{k+1}$ *and* $v^{k+1}$ *are well-defined.*

**Proof.** If $u^k$ is an optimal solution for the first subproblem in (4.17), then

$$\operatorname{prox}_p(u^k - \nabla_u h(u^k, v^k)) - u^k = 0,$$

which implies that the first inequality in (4.18) is satisfied. Otherwise, denote one of the solutions to the first subproblem as $\hat{u}^{k+1}$. We have

$$\operatorname{prox}_p(\hat{u}^{k+1} - \nabla_u h(\hat{u}^{k+1}, v^k)) - \hat{u}^{k+1} = 0.$$

By the continuity of proximal residual and the fact $H(u^k, v^k) > H(\hat{u}^{k+1}, v^k)$, we know that there is a neighborhood of $\hat{u}^{k+1}$ such that for any point in this neighborhood, the first inequality in (4.18) is satisfied. Similarly the second inequality is also achievable. Thus, $u^{k+1}$ and $v^{k+1}$ are well-defined. □

**Proposition 4.9.** *Suppose Assumptions 7 and 8 hold, then the sequences* $\{u^{k+1}, v^k\}$ *and* $\{u^k, v^k\}$ *are bounded and every cluster point of each of these sequences is an optimal solution to problem* (4.16).

**Proof.** From Assumption 7, we know that the sequences $\{u^{k+1}, v^k\}$ and $\{u^k, v^k\}$ generated by the inexact alternating minimization procedure are bounded. Thus, the sequence $\{u^{k+1}, v^k\}$ must admit at least one cluster point. Then, for any cluster point of the sequence $\{u^{k+1}, v^k\}$, say $(\bar{u}, \bar{v})$, there exists a subsequence $\{u^{k_l+1}, v^{k_l}\}$ such that $\lim_{l\to\infty}(u^{k_l+1}, v^{k_l}) = (\bar{u}, \bar{v})$.

Note that the sequence $\{u^{k_l+1}, v^{k_l+1}\}$ is also bounded, then there is a subset of $\{k_l\}$, denoted as $\{k_n\}_{n=1,2,...}$ such that

$$\lim_{n\to\infty}(u^{k_n+1}, v^{k_n}) = (\bar{u}, \bar{v}) \quad \text{and} \quad \lim_{n\to\infty}(u^{k_n+1}, v^{k_n+1}) = (\bar{u}, \hat{v}).$$

From Assumption 7 and (4.18), we have $\|r_1^k\| \to 0$ and $\|r_2^k\| \to 0$ as $k \to \infty$. By the continuity of proximal mapping we have

$$\text{prox}_p(\bar{u} - \nabla_u h(\bar{u}, \bar{v})) = \bar{u}. \tag{4.19}$$

Similarly, we have

$$\text{prox}_q(\hat{v} - \nabla_v h(\bar{u}, \hat{v})) = \hat{v},$$

which means $\hat{v} = \text{argmin}_v H(\bar{u}, v)$. Since $H(u, v)$ is continuous on $S$ and the function value is monotonically decreasing in the inexact alternating minimization method, we know that

$$H(\bar{u}, \hat{v}) = H(\bar{u}, \bar{v}).$$

Thus, we have $\bar{v} = \text{argmin}_v H(\bar{u}, v)$, which can be equivalently reformulated as

$$\text{prox}_q(\bar{v} - \nabla_v h(\bar{u}, \bar{v})) = \bar{v}. \tag{4.20}$$

By combining (4.19) and (4.20), we know that $(\bar{u}, \bar{v})$ is an optimal solution to (4.16). Thus, any cluster point of the sequence $\{u^{k+1}, v^k\}$ is an optimal solution to problem (4.16). The desired results for the sequence $\{u^k, v^k\}$ can be obtained similarly. □

Let

$$\Phi_k(y, z) := \mathcal{L}_{\sigma_k}(y, z; x^k) + \frac{1}{2\sigma_k}\|y - y^k\|^2_{\Lambda_1} + \frac{1}{2\sigma_k}\|z - z^k\|^2_{\Lambda_2}.$$

The aforementioned inexact alternating minimization method, when applied to (4.6), has the following template.

**Algorithm iAMM: An inexact alternating minimization method for the inner subproblem** (4.6)

Choose tolerance $\varepsilon > 0$. Choose $(y^{k,0}, z^{k,0}) \in \operatorname{dom}(\theta_f) \times \operatorname{dom}(\varphi_g)$. For $l = 0, 1, 2, ...,$ generate $(y^{k,l+1}, z^{k,l+1})$ according to the following iteration.

**Step 1.** Compute

$$y^{k,l+1} \approx \arg\min_y \Phi_k(y, z^{k,l}). \tag{4.21}$$

**Step 2.** Compute

$$z^{k,l+1} \approx \arg\min_z \Phi_k(y^{k,l+1}, z). \tag{4.22}$$

Based on (4.18), we discuss the stopping criteria for the subproblems (4.21) and (4.22). In order to simplify the subsequent discussions, denote

$$\Phi_k(y, z) = \hat{\theta}(y) + \hat{\varphi}(z) + h_k(y, z),$$

where $\hat{\theta}(y) \equiv \theta(y_1)\ \forall y \in \mathcal{Y}$, $\hat{\varphi}(z) \equiv \varphi(z_1)\ \forall z \in \mathcal{Z}$ are the nonsmooth functions, and $h_k$ is the smooth function given as follows:

$$\begin{aligned} h_k(y, z) \quad = \quad & f(y) + g(z) + \langle x^k, \mathcal{A}^* y + \mathcal{B}^* z - c\rangle + \frac{\sigma_k}{2}\|\mathcal{A}^* y + \mathcal{B}^* z - c\|^2 \\ & + \frac{1}{2\sigma_k}\|y - y^k\|^2_{\Sigma_1} + \frac{1}{2\sigma_k}\|z - z^k\|^2_{\Sigma_2}, \end{aligned} \tag{4.23}$$

i.e., we split $\Phi_k$ into the summation of nonsmooth part and smooth part. For the $l$-th iteration in Algorithm iAMM, define the following residue functions

$$\begin{cases} R_1^{k,l+1} = y^{k,l+1} - \text{Prox}_{\hat{\theta}}(y^{k,l+1} - \nabla_y h_k(y^{k,l+1}, z^{k,l})), \\ R_2^{k,l+1} = z^{k,l+1} - \text{Prox}_{\hat{\varphi}}(z^{k,l+1} - \nabla_z h_k(y^{k,l+1}, z^{k,l+1})). \end{cases} \tag{4.24}$$

Given the tolerance $\varepsilon > 0$, we propose the following stopping criteria:

$$\begin{cases} \Phi_k(y^{k,l+1}, z^{k,l}) - \Phi_k(y^{k,l}, z^{k,l}) \leq -\varepsilon \|R_1^{k,l+1}\|, \\ \Phi_k(y^{k,l+1}, z^{k,l+1}) - \Phi_k(y^{k,l+1}, z^{k,l}) \leq -\varepsilon \|R_2^{k,l+1}\|. \end{cases} \tag{4.25}$$

In the next theorem, we establish the convergence of Algorithm iAMM.

**Theorem 4.10.** *Suppose the sequence* $\{(y^{k,l}, z^{k,l})\}$ *generated by iAMM with stopping criteria* (4.25)*. Then it converges to the unique optimal solution of problem* (4.6)*.*

**Proof.** Due to the strong convexity of $\Phi_k(y, z)$, we know that the Assumption 7 and 8 hold for function $\Phi_k$. Therefore, by Proposition 4.9, we have that any cluster point of the sequence $\{(y^{k,l}, z^{k,l})\}$ is an optimal solution of problem (4.6). The result then follows by noting that the inner subproblem (4.6) has an unique optimal solution. □

## 4.2 The second stage of solving convex QSDP

As a prominent example of the convex composite quadratic optimization problems, in this section, we focus on applying our Phase II algorithm on the following convex quadratic semidefinite programming problem:

$$\begin{array}{ll} \min & \frac{1}{2}\langle X, \mathcal{Q}X \rangle + \langle C, X \rangle \\ \text{s.t.} & \mathcal{A}_E X = b_E, \quad \mathcal{A}_I X \geq b_I, \quad X \in \mathcal{S}^n_+ \cap \mathcal{K}, \end{array} \tag{4.26}$$

where $\mathcal{Q}$ is a self-adjoint positive semidefinite linear operator from $\mathcal{S}^n$ to $\mathcal{S}^n$, $\mathcal{A}_E : \mathcal{S}^n \to \Re^{m_E}$ and $\mathcal{A}_I : \mathcal{S}^n \to \Re^{m_I}$ are two linear maps, $C \in \mathcal{S}^n$, $b_E \in \Re^{m_E}$ and $b_I \in \Re^{m_I}$ are given data, $\mathcal{K}$ is a nonempty simple closed convex set, e.g., $\mathcal{K} = \{X \in$

$\mathcal{S}^n \mid L \leq X \leq U\}$ with $L, U \in \mathcal{S}^n$ being given matrices. Carefully examine shows that the dual problem associated with (4.26) can be written as following:

$$\begin{array}{rl} \max & -\delta_{\mathcal{K}}^*(-Z) - \frac{1}{2}\langle W,\, \mathcal{Q}W\rangle + \langle b_E,\, y_E\rangle + \langle b_I,\, y_I\rangle \\ \text{s.t.} & Z - \mathcal{Q}W + S + \mathcal{A}_E^* y_E + \mathcal{A}_I^* y_I = C, \\ & y_I \geq 0, \quad S \in \mathcal{S}_+^n, \quad W \in \mathcal{W}, \end{array} \tag{4.27}$$

where $\mathcal{W} \subseteq \mathcal{S}^n$ is any subspace such that $\text{Range}(\mathcal{Q}) \subseteq \mathcal{W}$. In fact, when $\mathcal{Q}$ is singular, we have infinite many dual problems corresponding to the primal problem (4.26). While in Phase I, we consider the case $\mathcal{W} = \mathcal{S}^n$ in the dual problem (4.27), in the second phase, we must restrict $\mathcal{W} = \text{Range}(\mathcal{Q})$ to avoid the unboundedness of the dual solution $W$, i.e.,

$$\begin{array}{rl} \max & -\delta_{\mathcal{K}}^*(-Z) - \frac{1}{2}\langle W,\, \mathcal{Q}W\rangle + \langle b_E,\, y_E\rangle + \langle b_I,\, y_I\rangle \\ \text{s.t.} & Z - \mathcal{Q}W + S + \mathcal{A}_E^* y_E + \mathcal{A}_I^* y_I = C, \\ & y_I \geq 0, \quad S \in \mathcal{S}_+^n, \quad W \in \mathcal{W} = \text{Range}(\mathcal{Q}). \end{array} \tag{4.28}$$

The reason for this special choice will be revealed in the subsequent analysis. Problem (4.28) can be equivalently recast as

$$\begin{array}{rl} \min & \delta_{\mathcal{K}}^*(-Z) + \frac{1}{2}\langle W,\, \mathcal{Q}W\rangle - \langle b_E,\, y_E\rangle - \langle b_I,\, y_I\rangle \\ \text{s.t.} & Z - \mathcal{Q}W + S + \mathcal{A}_E^* y_E + \mathcal{A}_I^* y_I = C, \\ & u + y_I = 0, \quad u \leq 0, \quad S \in \mathcal{S}_+^n, \quad W \in \mathcal{W}. \end{array} \tag{4.29}$$

Define the affine function $\Gamma : \mathcal{S}^n \times \mathcal{W} \times \mathcal{S}^n \times \Re^{m_E} \times \Re^{m_I} \to \mathcal{S}^n$ by

$$\Gamma(Z, W, S, y_E, y_I) := Z - \mathcal{Q}W + \mathcal{S} + \mathcal{A}_E^* y_E + \mathcal{A}_I^* y_I - C.$$

Similarly, define the linear function $\gamma : \Re^{m_I} \times \Re^{m_I} \to \Re^{m_I}$ by

$$\gamma(u, y_I) := u + y_I.$$

Let $\sigma > 0$, the augmented Lagrangian function associated with (4.29) is given as follows:

$$\mathcal{L}_\sigma(Z,W,u,S,y_E,y_I;X,x) = \left\{ \begin{array}{l} \delta^*_{\mathcal{K}}(-Z) + \frac{1}{2}\langle W,\, \mathcal{Q}W\rangle - \langle b_E,\, y_E\rangle - \langle b_I,\, y_I\rangle \\ +\frac{\sigma}{2}\|\Gamma(Z,W,S,y_E,y_I) + \sigma^{-1}X\|^2 \\ +\frac{\sigma}{2}\|\gamma(u,y_I) + \sigma^{-1}x\|^2 - \frac{1}{2\sigma}\|X\|^2 - \frac{1}{2\sigma}\|x\|^2 \end{array} \right\} \quad (4.30)$$

for all $(Z,W,u,S,y_E,y_I,X,x) \in \mathcal{S}^n \times \mathcal{W} \times \Re^{m_I} \times \mathcal{S}^n \times \Re^{m_E} \times \Re^{m_I} \times \mathcal{S}^n \times \Re^{m_I}$. When we apply Algorithm pALM to solve (4.29), in the $k$th iteration, we propose to add the following proximal term:

$$\begin{aligned} \Lambda_k(Z,W,u,S,y_E,y_I) \;:=\; & \frac{1}{2\sigma_k}(\|Z - Z^k\|^2 + \|W - W^k\|^2_{\mathcal{Q}} + \|u - u^k\|^2 + \|S - S^k\|^2 \\ & + \|y_E - y_E^k\|^2 + \|y_I - y_I^k\|^2). \end{aligned} \quad (4.31)$$

Being regarded as a self-adjoint linear operator defined on $\mathcal{W} = \text{Range}(\mathcal{Q})$, $\mathcal{Q}$ is in fact positive definite. Thus, the above proximal term satisfies the requirement of Algorithm pALM. Then, the inner subproblem (4.6) takes the form of

$$\begin{aligned} & (Z^{k+1}, W^{k+1}, u^{k+1}, S^{k+1}, y_E^{k+1}, y_I^{k+1}) \\ & \approx \text{argmin} \left\{ \begin{array}{l} \mathcal{L}_{\sigma_k}(Z,W,u,S,y_E,y_I;X^k,x^k) + \Lambda_k(Z,W,u,S,y_E,y_I) \mid Z \in \mathcal{S}^n, \\ W \in \mathcal{W}, u \in \Re^{m_I}_-, S \in \mathcal{S}^n_+, y_E \in \Re^{m_E}, y_I \in \Re^{m_I} \end{array} \right\}. \end{aligned} \quad (4.32)$$

By adding proximal terms and choosing $\mathcal{W} = \text{Range}(\mathcal{Q})$, we are actually dealing with a strongly convex function in (4.32). This is in fact a key idea in the designing of our second stage algorithm. Here, we propose to apply Algorithm iAMM to solve subproblem (4.32), i.e., we solve optimization problems with respect to $(Z,W,u)$ and $(S,y_E,y_I)$ alternatively. Therefore, we only need to focus on solving the inner subproblems (4.21) and (4.22).

For our QSDP problem (4.29), the inner subproblem (4.21) takes the following form:

$$\min \left\{ \begin{array}{l} \Psi(Z,W,u) := \delta^*_{\mathcal{K}}(-Z) + \frac{1}{2}\langle W,\, \mathcal{Q}W\rangle + \frac{\sigma}{2}(\|Z - \mathcal{Q}W - \widehat{C}\|^2 + \|u - \hat{c}\|^2) \\ +\frac{1}{2\sigma}(\|Z - \widehat{Z}\|^2 + \|W - \widehat{W}\|^2_{\mathcal{Q}} + \|u - \hat{u}\|^2) \mid Z \in \mathcal{S}^n, W \in \mathcal{W}, u \in \Re^{m_I}_- \end{array} \right\},$$

where $(\widehat{C}, \widehat{c}, \widehat{Z}, \widehat{W}, \widehat{u}) \in \mathcal{S}^n \times \Re^{m_I} \times \mathcal{S}^n \times \mathcal{W} \times \Re^{m_I}$ are given data. Given $\sigma > 0$ and $(\widehat{C}, \widehat{Z}) \in \mathcal{S}^n \times \mathcal{S}^n$, denote

$$Z(W) := \sigma(\mathcal{Q}W + \widehat{C}) + \sigma^{-1}\widehat{Z} \quad \forall\, W \in \mathcal{W} \quad \text{and} \quad \hat{\sigma} = \sigma + \sigma^{-1}.$$

By Proposition 2.6, we know that if $(Z^*, W^*, u^*) = \operatorname{argmin}\{\Psi(Z, W, u) \,|\, Z \in \mathcal{S}^n, W \in \mathcal{W}, u \in \Re^{m_I}_-\}$, then

$$\left\{\begin{array}{l} W^* = \operatorname{argmin}\left\{\begin{array}{l} \varphi(W) := -\hat{\sigma}^{-1}\langle Z(W),\, \Pi_{\mathcal{K}}(-Z(W))\rangle \\ -\dfrac{1}{2\hat{\sigma}}(\|\Pi_{\mathcal{K}}(-Z(W))\|^2 - \|\mathcal{Q}W + \widehat{C} - \widehat{Z}\|^2) \\ +\dfrac{1}{2}\langle W,\, \mathcal{Q}W\rangle + \dfrac{1}{2\sigma}\|W - \widehat{W}\|^2_{\mathcal{Q}} \,|\, W \in \mathcal{W} \end{array}\right\}, \\ Z^* = \hat{\sigma}^{-1}(Z(W^*) + \Pi_{\mathcal{K}}(-Z(W^*))), \\ u^* = \min\left\{\hat{\sigma}^{-1}(\sigma\hat{c} + \sigma^{-1}\hat{u}), 0\right\}. \end{array}\right. \tag{4.33}$$

Hence, we need to solve the following problem

$$W^* = \operatorname{argmin}\{\varphi(W) \,|\, W \in \mathcal{W}\}. \tag{4.34}$$

The objective function in (4.34) is continuously differentiable with the gradient given as follows:

$$\nabla\varphi(W) = (1 + \sigma^{-1})\mathcal{Q}W + \hat{\sigma}^{-1}(\mathcal{Q}(\mathcal{Q}W + \widehat{C} - \widehat{Z}) - \sigma\mathcal{Q}\Pi_{\mathcal{K}}(-Z(W))) - \sigma^{-1}\mathcal{Q}\widehat{W}.$$

Hence, solving (4.34) is equivalent to solving the following nonsmooth equation:

$$\nabla\varphi(W) = 0, \quad W \in \mathcal{W}. \tag{4.35}$$

Note that, if $\mathcal{K}$ is a polyhedral set, then $\nabla\varphi$ is piecewise smooth. For any $W \in \mathcal{W}$, define

$$\hat{\partial}^2\varphi(W) := (1 + \sigma^{-1})\mathcal{Q} + \hat{\sigma}^{-1}\mathcal{Q}(\mathcal{I} + \sigma^2\partial\Pi_{\mathcal{K}}(-Z(W)))\mathcal{Q},$$

where $\partial\Pi_{\mathcal{K}}(-Z(W))$ is the Clarke subdifferential [6] of $\Pi_{\mathcal{K}}(\cdot)$ at $-Z(W)$, $\mathcal{I} : \mathcal{W} \to \mathcal{W}$ is the identity map. Note that from [27], we know that

$$\hat{\partial}^2\varphi(W)\, D = \partial^2\varphi(W)\, D \quad \forall\, D \in \mathcal{W}, \tag{4.36}$$

where $\partial^2\varphi(W)$ denotes the generalized Hessian of $\varphi$ at $W$, i.e., the Clarke subdifferential of $\nabla\varphi$ at $W$. Given $W \in \mathcal{W}$, let $\mathcal{U}_W^0 \in \partial\Pi_{\mathcal{K}}(-Z(W))$ be given , we know that

$$\mathcal{V}_W^0 = (1+\sigma^{-1})\mathcal{Q} + \hat{\sigma}^{-1}\mathcal{Q}(\mathcal{I} + \sigma^2\mathcal{U}_W^0)\mathcal{Q} \in \hat{\partial}^2\varphi(W). \tag{4.37}$$

In fact if $\mathcal{K} = \{X \in \mathcal{S}^n \mid L \leq X \leq U\}$ with given $L, U \in \mathcal{S}^n$, we can easily find an element $\mathcal{U}_W^0 \in \partial\Pi_{\mathcal{K}}(-Z(W))$ by using (2.5). After all the perparation, we can design a semismooth Newton-CG method as in [73] to solve (4.35).

**Algorithm SNCG: A semismooth Newton-CG algorithm.**

Given $\mu \in (0, 1/2)$, $\bar{\eta} \in (0, 1)$, $\tau \in (0, 1]$, and $\delta \in (0, 1)$. Perform the $j$th iteration as follows.

**Step 1.** Compute

$$\eta_j := \min(\bar{\eta}, \|\nabla\varphi(y^j)\|^{1+\tau}).$$

Apply the conjugate gradient (CG) algorithm to find an approximation solution $D^j \in \mathcal{W}$ to

$$\mathcal{V}_j\, D = -\nabla\varphi(W^j), \tag{4.38}$$

where $\mathcal{V}_j \in \hat{\partial}^2\varphi(W^j)$ is defined as in (4.37).

**Step 2.** Set $\alpha_j = \delta^{m_j}$, where $m_j$ is the first nonnegative integer $m$ for which

$$\varphi(W^j + \delta^m D^j) \leq \varphi(W^j) + \mu\delta^m\langle\nabla\varphi(W^j), D^j\rangle. \tag{4.39}$$

**Step 3.** Set $W^{j+1} = W^j + \alpha_j\, D^j$.

The convergence results for the above SNCG algorithm are stated in Theorem 4.11.

**Theorem 4.11.** *Suppose that at each step $j \geq 0$, when the CG algorithm terminates, the tolerance $\eta_j$ is achieved, i.e.,*

$$\|\nabla\varphi(W^j) + \mathcal{V}_j\, D^j\| \leq \eta_j. \tag{4.40}$$

*Then the sequence $\{W^j\}$ converges to the unique optimal solution, say $\overline{W}$, of the optimization problem in* (4.34) *and*

$$\|W^{j+1} - \overline{W}\| = O(\|W^j - \overline{W}\|^{1+\tau}). \tag{4.41}$$

**Proof.** Since $\varphi(W)$ is a strongly convex function defined on $\mathcal{W} = \text{Range}(\mathcal{Q})$, problem (4.34) then has a unique solution $\overline{W}$ and the level set $\{W \in \mathcal{W} \,|\, \varphi(W) \leq \varphi(W^0)\}$ is compact. Therefore, the sequence generated by SNCG is bounded as $D^j$ is a descent direction [73, Propsition 3.3]. Note that for all $W \in \text{Range}(\mathcal{Q})$, every $\mathcal{V} \in \hat{\partial}^2\varphi(W)$ is self-adjoint and positive definite on Range($\mathcal{Q}$), the desired results thus can be easily obtained by combining [73, Theorem 3.4 and 3.5]. □

**Remark 4.12.** Note that in above algorithm, the approximate solution of (4.38), i.e., the obtained direction $D_j$, need to be maintained within the subspace Range($\mathcal{Q}$). Fortunately, when Algorithm CG is applied to solve (4.38), the requirement $D_j \in \text{Range}(\mathcal{Q})$ will always be satisfied if the starting point of Algorithm CG is chosen to be in Range($\mathcal{Q}$) [67]. In fact, one can always choose 0 as a starting point in Algorithm CG.

Next we focus on the subproblem corresponding to $(S, y_E, y_I)$. The discussion presented here is in fact similar to the aforementioned discussion about solving the subproblem corresponding to $(Z, W, u)$. The inner subproblem (4.22) now takes the following form:

$$\min \left\{ \begin{array}{l} \Phi(S, y_E, y_I) := -\langle b_E,\, y_E\rangle - \langle b_I,\, y_I\rangle + \dfrac{\sigma}{2}\|S + \mathcal{A}_E^* y_E + \mathcal{A}_I^* y_I - \widehat{C}\|^2 \\ +\dfrac{\sigma}{2}\|y_I - \hat{c}\|^2 + \dfrac{1}{2\sigma}(\|S - \widehat{S}\|^2 + \|y_E - \hat{y}_E\|^2 + \|y_I - \hat{y}_I\|^2) \,|\, S \in \mathcal{S}_+^n, \\ y_E \in \Re^{m_E}, y_I \in \Re^{m_I} \end{array} \right\}, \tag{4.42}$$

where $(\widehat{C}, \widehat{S}, \hat{c}, \hat{y}_E, \hat{y}_I) \in \mathcal{S}^n \times \mathcal{S}^n_+ \times \Re^{m_I} \times \Re^{m_E} \times \Re^{m_I}$ are given data. Given $\sigma > 0$ and $(\widehat{C}, \widehat{S}) \in \mathcal{S}^n \times \mathcal{S}^n$, denote

$$S(y_E, y_I) := \sigma(\widehat{C} - \mathcal{A}_E^* y_E - \mathcal{A}_I^* y_I) + \sigma^{-1}\widehat{S} \quad \forall\, (y_E, y_I) \in \Re^{m_E} \times \Re^{m_I}.$$

Again by Proposition 2.7, we know that if $(S^*, y_E^*, y_I^*) = \operatorname{argmin}\{\Phi(S, y_E, y_I) \,|\, S \in \mathcal{S}^n_+, y_E \in \Re^{m_E}, y_I \in \Re^{m_I}\}$, then

$$\begin{cases} (y_E^*, y_I^*) = \operatorname{argmin} \left\{ \begin{array}{l} \phi(y_E, y_I) := -\langle b_E,\, y_E\rangle - \langle b_I,\, y_I\rangle + \dfrac{1}{2\hat{\sigma}}\|\Pi_{\mathcal{S}^n_+}(-S(y_E, y_I))\|^2 \\ +\dfrac{1}{2\hat{\sigma}}\|\widehat{C} - \mathcal{A}_E^* y_E - \mathcal{A}_I^* y_I - \widehat{S}\|^2 + \dfrac{\sigma}{2}\|y_I - \hat{c}\|^2 \\ +\dfrac{1}{2\sigma}(\|y_E - \hat{y}_E\|^2 + \|y_I - \hat{y}_I\|^2) \,|\, y_E \in \Re^{m_E}, y_I \in \Re^{m_I} \end{array} \right\}, \\ S^* = \hat{\sigma}^{-1}\Pi_{\mathcal{S}^n_+}(S(y_E^*, y_I^*)), \end{cases} \tag{4.43}$$

where $\hat{\sigma} = \sigma + \sigma^{-1}$. Then, we need to solve the following problem

$$(y_E^*, y_I^*) = \operatorname{argmin}\{\phi(y_E, y_I) \,|\, (y_E, y_I) \in \Re^{m_E} \times \Re^{m_I}\}. \tag{4.44}$$

The objective function in (4.44) is continuously differentiable with the gradient given as follows:

$$\begin{aligned} \nabla\phi(y_E, y_I) \;&=\; \hat{\sigma}^{-1}\begin{pmatrix} \mathcal{A}_E \\ \mathcal{A}_I \end{pmatrix}\big(\sigma\Pi_{\mathcal{S}^n_+}(-S(y_E, y_I)) + \mathcal{A}_E^* y_E + \mathcal{A}_I^* y_I + \widehat{S} - \widehat{C}\big) \\ &\quad +\sigma\begin{pmatrix} 0 \\ y_I - \hat{c} \end{pmatrix} + \sigma^{-1}\begin{pmatrix} y_E - \hat{y}_E \\ y_I - \hat{y}_I \end{pmatrix} - \begin{pmatrix} b_E \\ b_I \end{pmatrix}. \end{aligned}$$

Hence, solving (4.34) is equivalent to solving the following nonsmooth equation:

$$\nabla\phi(y_E, y_I) = 0, \quad (y_E, y_I) \in \Re^{m_E} \times \Re^{m_I}. \tag{4.45}$$

Given $(y_E, y_I) \in \Re^{m_E} \times \Re^{m_I}$, define

$$\hat{\partial}^2\phi(y_E, y_I) := \hat{\sigma}^{-1}\begin{pmatrix} \mathcal{A}_E \\ \mathcal{A}_I \end{pmatrix}\big(\mathcal{I} + \sigma^2\partial\Pi_{\mathcal{S}^n_+}(-S(y_E, y_I))\big)(\mathcal{A}_E^*, \mathcal{A}_I^*) + \begin{pmatrix} \sigma^{-1}I_1 & \\ & \hat{\sigma}I_2 \end{pmatrix},$$

where $\mathcal{I} : \mathcal{S}^n \to \mathcal{S}^n$ is the identity map, $I_1 \in \Re^{m_E \times m_E}$ and $I_2 \in \Re^{m_I \times m_I}$ are identity matrices, $\partial\Pi_{\mathcal{S}^n_+}(-S(y_E, y_I))$ is the Clark subdifferential of $\Pi_{\mathcal{S}^n_+}$ at $-S(y_E, y_I)$. Note that one can find an element in $\partial\Pi_{\mathcal{S}^n_+}(-S(y_E, y_I))$ by using (2.6) based on the results obtained in [47]. Then, equation (4.34) can be efficiently solved by the semismooth Newton-CG method presented above. The convergence analysis can be similarly derived as in Theorem 4.11.

### 4.2.1 The second stage of solving convex QP

Although convex quadratic programming can be viewed as a special case of QSDP, we study in this subsection, as an application of the idea of using our symmetric Gauss-Seidel technique in Phase II algorithm, the second phase of solving convex quadratic programming problem. Consider the following convex quadratic programming problem

$$\min \left\{ \frac{1}{2}\langle x,\, Qx\rangle + \langle c,\, x\rangle \mid Ax = b,\; \bar{b} - Bx \in \mathcal{C},\; x \in \mathcal{K} \right\}, \tag{4.46}$$

where matrices $Q \in \mathcal{S}^n_+$, $A \in \Re^{m_E \times n}$ and $B \in \Re^{m_I \times n}$, vectors $b$, $c$ and $\bar{b}$ are given data, $\mathcal{C} \subseteq \Re^{m_I}$ is a closed convex cone, e.g., the nonnegative orthant $\mathcal{C} = \{\bar{x} \in \Re^{m_I} \mid \bar{x} \geq 0\}$, $\mathcal{K} \subseteq \Re^n$ is a nonempty simple closed convex set, e.g., $\mathcal{K} = \{x \in \Re^n \mid l \leq x \leq u\}$ with $l, u \in \Re^n$ being given vectors. The dual problem of (4.46) we consider here is

$$\begin{array}{ll} \max & -\delta^*_{\mathcal{K}}(-z) - \frac{1}{2}\langle w,\, Qw\rangle + \langle \bar{b},\, \bar{y}\rangle + \langle b,\, y\rangle \\ \text{s.t.} & z - Qw + B^*\bar{y} + A^*y = c, \quad \bar{y} \in \mathcal{C}^\circ, \quad w \in \text{Range}(Q). \end{array} \tag{4.47}$$

Similar as in (4.28), we further require $w \in \text{Range}(Q)$ comparing to the dual problem (3.80) considered in Phase I. Note that (4.47) can be equivalently recast as

$$\begin{array}{ll} \min & \delta^*_{\mathcal{K}}(-z) + \frac{1}{2}\langle w,\, Qw\rangle - \langle b,\, y\rangle - \langle \bar{b},\, \bar{y}\rangle \\ \text{s.t.} & \begin{bmatrix} z \\ \bar{z} \end{bmatrix} - \begin{bmatrix} Qw \\ 0 \end{bmatrix} + \begin{bmatrix} A^* & B^* \\ & I \end{bmatrix} \begin{bmatrix} y \\ \bar{y} \end{bmatrix} = \begin{bmatrix} c \\ 0 \end{bmatrix}, \\ & \bar{z} \in \mathcal{C}, \quad w \in \text{Range}(Q). \end{array} \tag{4.48}$$

Below, we focus on applying pALM, i.e., our algorithm in Phase II, to solve problem (4.48). Note that, by Remark 4.5, if $\mathcal{K}$ in problem (4.48) is assumed to be polyhedral, the error bound assumption (Assumption 6) holds automatically for the corresponding $\mathcal{T}_l$. Given $\sigma > 0$, the augmented Lagrangian function associated with (4.48) is given as follows:

$$\begin{aligned}\mathcal{L}_\sigma(z,\bar{z},w,y,\bar{y};x,\bar{x}) &= \delta_{\mathcal{K}}^*(-z) + \frac{1}{2}\langle w,\, Qw\rangle - \langle b,\, y\rangle - \langle \bar{b},\, \bar{y}\rangle + \frac{\sigma}{2}\|\bar{z}+\bar{y}+\sigma^{-1}\bar{x}\|^2 \\ &\quad + \frac{\sigma}{2}\|z - Qw + A^*y + B^*y + \sigma^{-1}x - c\|^2 - \frac{1}{2\sigma}(\|x\|^2 + \|\bar{x}\|^2).\end{aligned}$$

In the $k$th iteration of Algorithm pALM, we propose to add the following proximal term:

$$\Lambda_k(z,\bar{z},w,y,\bar{y}) = \frac{1}{2\sigma_k}(\|z-z^k\|^2 + \|\bar{z}-\bar{z}^k\|^2 + \|w-w^k\|_Q^2 + \|y-y^k\|^2 + \|\bar{y}-\bar{y}^k\|^2).$$

By restricting $w \in \text{Range}(Q)$, the positive definiteness of the added proximal term is guaranteed. Then, the inner subproblem (4.6) takes the form of

$$\begin{aligned}&(z^{k+1},\bar{z}^{k+1},w^{k+1},y^{k+1},\bar{y}^{k+1}) \\ &\approx \text{argmin}\left\{\begin{array}{l}\Psi_k(z,\bar{z},w,y,\bar{y}) := \mathcal{L}_{\sigma^k}(z,\bar{z},w,y,\bar{y};x^k,\bar{x}^k) + \Lambda_k(z,\bar{z},w,y,\bar{y}) \\ \mid z\in\Re^n, \bar{z}\in\mathcal{C}, w\in\text{Range}(Q), y\in\Re^{m_E}, \bar{y}\in\Re^{m_I}\end{array}\right\}.\end{aligned} \tag{4.49}$$

To solve (4.49), we can follow the same idea discussed in (4.33). Specifically, in each iteration of pLAM, we solve the following unconstrained minimization problem

$$\min\{\varphi(w,y,\bar{y}) := \min_{z\in\Re^n,\bar{z}\in\mathcal{C}} \Psi(z,\bar{z},w,y,\bar{y}) \mid w\in\text{Range}(Q), y\in\Re^{m_E}, \bar{y}\in\Re^{m_I}\}. \tag{4.50}$$

Instead of using the semismooth Newton-CG algorithm to solve (4.50), one can solve this subproblem with an inexact accelerated proximal gradient (APG) algorithm proposed in [29]. The quadratic model used by the inexact APG can be constructed as follows. By adopting the majorization technique proposed in [69], we can obtain a convex quadratic function $\hat{\varphi}_k$ as a majorization function of $\varphi$ at $(w^k,y^k,\bar{y}^k)$, i.e., we have that $\hat{\varphi}_k(w^k,y^k,\bar{y}^k) = \varphi(w^k,y^k,\bar{y}^k)$ and $\hat{\varphi}_k(w,y,\bar{y}) \geq \varphi(w,y,\bar{y}),\ \forall (w,y,\bar{y}) \in \text{Range}(Q)\times\Re^{m_E}\times\Re^{m_I}$. Thus, in each iteration of Algorithm iAPG, the following unconstrained convex quadratic programming problem needs to be solved

$$\min\{\hat{\varphi}_k(w,y,\bar{y}) \mid w\in\text{Range}(Q), y\in\Re^{m_E}, \bar{y}\in\Re^{m_I}\}. \tag{4.51}$$

Note that solving (4.51) is equivalent to solving a large scale linear system corresponding to $(w, y, \bar{y})$. It can be efficiently solved via a preconditioned CG (PCG) algorithm provided a suitable preconditioner can be found. If such a preconditioner is not available, then we can use the one cycle symmetric block Gauss-Seidel (sGS) technique developed in Chapter 3 to manipulate problem (4.51). In this way, we can decompose the large scale linear system into three small pieces with each of them corresponding to only one variable of $(w, y, \bar{y})$ and then solve these three linear systems separately by the PCG algorithm. Now, it should be easy to find a suitable preconditioner for each smaller linear system. By Theorem 3.3, our sGS technique used to manipulate problem (4.51) can be regarded as taking a scaled gradient step for solving (4.51). Thus, the whole process we discussed here can still be viewed as an inexact APG algorithm for solving (4.50) with one more proximal term corresponding to sGS technique needs to be added to $\hat{\varphi}_k$ in (4.51). Then, the global and local convergence results follow from [29, Theorem 2.1], Theorem (4.3) and Theorem (4.7).

In fact, as a simple but not that fast approach, we can also directly apply our (inexact) sGS technique to problem (4.49). The procedure can be described as follows: given $(z^k, \bar{z}^k, w^k, y^k, \bar{y}^k, x^k, \bar{x}^k) \in \Re^n \times \mathcal{C} \times \text{Range}(Q) \times \Re^{m_E} \times \Re^{m_I} \times \Re^n \times \Re^{m_I}$, $(z^{k+1}, \bar{z}^{k+1}, w^{k+1}, y^{k+1}, \bar{y}^{k+1})$ is obtained via

$$\left\{\begin{array}{l}
\bar{y}^{k+\frac{1}{2}} \approx \text{argmin}_{\bar{y}\in\Re^{m_I}} \Psi_k(z^k, \bar{z}^k, w^k, y^k, \bar{y}), \\
y^{k+\frac{1}{2}} \approx \text{argmin}_{y\in\Re^{m_E}} \Psi_k(z^k, \bar{z}^k, w^k, y, \bar{y}^{k+\frac{1}{2}}), \\
w^{k+\frac{1}{2}} \approx \text{argmin}_{w\in\text{Range}(Q)} \Psi_k(z^k, \bar{z}^k, w, y^{k+\frac{1}{2}}, \bar{y}^{k+\frac{1}{2}}), \\
(z^{k+1}, \bar{z}^{k+1}) = \text{argmin}_{z\in\Re^n, \bar{z}\in\mathcal{C}} \Psi_k(z^k, \bar{z}^k, w^{k+\frac{1}{2}}, y^{k+\frac{1}{2}}, \bar{y}^{k+\frac{1}{2}}), \\
w^{k+1} \approx \text{argmin}_{w\in\text{Range}(Q)} \Psi_k(z^{k+1}, \bar{z}^{k+1}, w, y^{k+\frac{1}{2}}, \bar{y}^{k+\frac{1}{2}}), \\
y^{k+1} \approx \text{argmin}_{y\in\Re^{m_E}} \Psi_k(z^{k+1}, \bar{z}^{k+1}, w^{k+1}, y, \bar{y}^{k+\frac{1}{2}}), \\
\bar{y}^{k+1} \approx \text{argmin}_{\bar{y}\in\Re^{m_I}} \Psi_k(z^{k+1}, \bar{z}^{k+1}, w^{k+1}, y^{k+1}, \bar{y}).
\end{array}\right. \tag{4.52}$$

Note that the joint minimization of $(z, \bar{z})$ in (4.52) can be carried out analytically.

Instead of further decomposing $w$,$y$ and $\bar{y}$ into smaller pieces as we have done in Phase I algorithm, we allow inexact minimizations in (4.52). In this way, Algorithm PCG can be applied to obtain high-accuracy solutions for these linear systems. By Theorem 3.3, procedure (4.52) is equivalent to solving (4.49) with an additional proximal term corresponding to sGS technique and an error term corresponding to inexact minimizations of $w$,$y$ and $\bar{y}$ added to $\Psi_k$. Since this extra error term can be arbitrarily small when the PCG algorithm is applied to solve the resulted linear systems in (4.52), the above procedure can be regarded as a special implementation of solving subproblem (4.6) in Algorithm pALM. In addition, the stopping criteria (A) and (B) for this special case are achievable. Thus, the convergence results still hold. Due to the appearance of the inexact minimizations in the one cycle symmetric block Gauss-Seidel procedure (4.52), we refer the resulted algorithm as inexact symmetric Gauss-Seidel based proximal augmented Lagrangian algorithm (inexact sGS-Aug). One remarkable property of our proposed inexact sGS-Aug algorithm here is that we can still enjoy the linear convergence rate of Algorithm pALM by only doing one cycle symmetric Gauss-Seidel procedure (4.52). More specifically, under the same setting of Theorem 4.7, by using the discussions in Section 3.1.2 on the structure of $\widehat{\mathcal{O}}$ in (3.11), it is not difficult to derive that the convergence rate $\theta_k$ in (4.12) satisfies

$$\theta_k \to \bar{\theta} \leq \frac{1}{\sqrt{1+\bar{c}}} \quad \text{as } k \to \infty, \tag{4.53}$$

where $\bar{c} = \dfrac{1}{a^2(3 + 2\|Q\|^2 + \|A\|^2)}$. Note that the constant number $\bar{\theta}$ in (4.53) is independent of $\sigma$ and if $a$ is not large, it can be a decent number smaller than 1.

Observing that in our proposed algorithms, it is important that the resulted large scale linear systems can be solved by the PCG efficiently. For this purpose, we discuss a novel approach to construct suitable preconditioners for given symmetric positive definite linear systems. Consider the following symmetric positive definite linear system

$$Ax = b,$$

where matrix $A \in \mathcal{S}^n$ is symmetric positive definite, vector $b \in \Re^n$ is given data. Suppose that $A$ has the following spectral decomposition

$$A = P\Lambda P^T,$$

where $\Lambda$ is the diagonal matrix with diagonal entries consisting of the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > 0$ of $A$ and $P$ is a corresponding orthogonal matrix of eigenvectors. Then, for given integer $1 \leq r \leq n$, we propose the following preconditioner:

$$\begin{aligned} \widetilde{A} &:= \sum_{i=1}^{r} \lambda_i P_i P_i^T + \frac{\lambda_r}{2} \sum_{i=r+1}^{n} P_i P_i^T \\ &= \sum_{i=1}^{r} \lambda_i P_i P_i^T + \frac{\lambda_r}{2}(I - \sum_{i=1}^{r} P_i P_i^T) \qquad (4.54) \\ &= \frac{\lambda_r}{2} I + \sum_{i=1}^{r} (\lambda_i - \frac{\lambda_r}{2}) P_i P_i^T, \end{aligned}$$

where $I \in \Re^{n\times n}$ is the identity matrix, $P_i$ is the $i$th column of matrix $P$. Note that $\widetilde{A}^{-1}$ can be easily obtained as follows:

$$\widetilde{A}^{-1} = \frac{2}{\lambda_r} I + \sum_{i=1}^{r} (\frac{1}{\lambda_i} - \frac{2}{\lambda_r}) P_i P_i^T.$$

Following the same idea in (4.54), we can also design a practically useful morjorization for $A$ as follows:

$$A \preceq \widehat{A} := \sum_{i=1}^{r} \lambda_i P_i P_i^T + \lambda_r \sum_{i=r+1}^{n} P_i P_i^T = \lambda_r I + \sum_{i=1}^{r} (\lambda_i - \lambda_r) P_i P_i^T.$$

In practice, Matlab built in function "eigs" can be used to find the first $r$ eigenvalues and their corresponding eigenvectors.

## 4.3 Numerical results

In this section, we conduct a variety of large scale QSDP problems and convex quadratic programming problems to evaluate the performance of our proposed Phase II algorithm.

Firstly, we focus on the QSDP problems. Apart from the QSDP-BIQ problems (3.69) and QSDP-$\theta_+$ problems (3.70), we also test here the following QSDP-QAP problems. The QSDP-QAP problem is given by:

$$\begin{array}{ll} \min & \frac{1}{2}\langle X,\, \mathcal{Q}X\rangle + \langle A_2 \otimes A_1,\, X\rangle \\ \text{s.t.} & \sum_{i=1}^{n} X^{ii} = I,\ \langle I,\, X^{ij}\rangle = \delta_{ij} \quad \forall\, 1 \le i \le j \le n, \\ & \langle E,\, X^{ij}\rangle = 1 \quad \forall\, 1 \le i \le j \le n, \quad X \in \mathcal{S}^{n^2}_+,\ X \in \mathcal{K}, \end{array} \tag{4.55}$$

where $E$ is the matrix of ones, and $\delta_{ij} = 1$ if $i = j$, and 0 otherwise, $\mathcal{K} = \{X \in \mathcal{S}^{n^2} \mid X \ge 0\}$. In our numerical experiments, the test instances $(A_1, A_2)$ are taken from the QAP Library [3]. Note that the linear operator $\mathcal{Q}$ used here is the same as been generated in (3.68) and used in the test of Phase I algorithm. For simplicity, we still don't include the general inequality constraints here, i.e., $\mathcal{A}_I$ and $b_I$ are vacuous.

In Phase II, when our inexact proximal augmented Lagrangian algorithm is applied to solve QSDP problems, it is in fact a generalization of SDPNAL [73] and SDPNAL+ [69]. Hence, we would like to call this special implementation of our Phase II algorithm as QSDPNAL. Since we use the Phase I algorithm sGS-PADMM to warm start our QSDPNAL, we also list the numerical results obtained by running sGS-PADMM alone for the purpose of demonstrating the power and the importance of the proposed inexact proximal augmented Lagrangian algorithm for solving difficult QSDP problems. All our computational results for the tested QSDP problems are obtained from a workstation running on 64-bit Windows Operating System having 16 cores with 32 Intel Xeon E5-2650 processors at 2.60GHz and 64 GB memory.

We measure the accuracy of an approximate optimal solution $(X, Z, \Xi, S, y_E)$ for QSDP (4.26) and its dual (4.28) by using the following relative residual:

$$\eta_{\rm qsdp} = \max\{\eta_P, \eta_D, \eta_Z, \eta_{S_1}, \eta_{S_2}\}, \tag{4.56}$$

where

$$\eta_P = \frac{\|\mathcal{A}_E X - b_E\|}{1 + \|b_E\|}, \quad \eta_D = \frac{\|Z + \mathcal{B}^*\Xi + S + \mathcal{A}_E^* y_E - C\|}{1 + \|C\|}, \quad \eta_Z = \frac{\|X - \Pi_{\mathcal{K}}(X - Z)\|}{1 + \|X\| + \|Z\|},$$

$$\eta_{S_1} = \frac{|\langle S,\, X\rangle|}{1 + \|S\| + \|X\|}, \quad \eta_{S_2} = \frac{\|X - \Pi_{\mathcal{S}^n_+}(X)\|}{1 + \|X\|}.$$

We terminate the solvers sGS-PADMMand QSDPNAL when $\eta_{\text{qsdp}} < 10^{-6}$ with the maximum number of iterations set at 25000.

In table 4.1, we present the detailed numerical results for QSDPNAL and sGS-PADMM in solving some large scale QSDP problems. In the table, "it" and "itersub" stand for the number of outer iterations and the total number of inner iterations of QSDPNAL, respectively. "itersGS" stands for the total number of iterations of sGS-PADMM used to warm start QSDPNAL. It is interesting to note that QSDPNAL can solve all the 49 difficult QSDP-QAP problems to an accuracy of $10^{-6}$ efficiently, while the Phase I algorithm sGS-PADMM can only solve 5 QSDP-QAP problems to required accuracy. Besides, QSDPNAL generally outperform sGS-PADMM in terms of the computing time, especially when the problem size is large. The superior numerical performance of QSDPNAL over sGS-PADMM demonstrate the power and the necessity of our proposed two phase framework.

Table 4.1: The performance of QSDPNAL (a) and sGS-PADMM(b) on QSDP-$\theta_+$, QSDP-QAP and QSDP-BIQ problems (accuracy = $10^{-6}$). The computation time is in the format of "hours:minutes:seconds".

| | | iter.a | iter.b | $\eta_{\text{qsdp}}$ | $\eta_{\text{gap}}$ | time |
|---|---|---|---|---|---|---|
| problem | $m_E; n_s$ | it\|itsub\|itsGS | | a\|b | a\|b | a\|b |
| theta6 | 4375 ; 300 | 0 \| 0 \| 311 | 311 | 7.9-7 \| 7.9-7 | 2.1-6 \| 2.1-6 | 09 \| 08 |
| theta62 | 13390 ; 300 | 0 \| 0 \| 153 | 153 | 9.6-7 \| 9.6-7 | -1.1-7 \| -1.1-7 | 04 \| 04 |
| theta8 | 7905 ; 400 | 0 \| 0 \| 314 | 314 | 9.5-7 \| 9.5-7 | 2.7-6 \| 2.7-6 | 19 \| 19 |
| theta82 | 23872 ; 400 | 0 \| 0 \| 158 | 158 | 9.5-7 \| 9.5-7 | -3.7-8 \| -3.7-8 | 10 \| 10 |
| theta83 | 39862 ; 400 | 0 \| 0 \| 156 | 156 | 9.5-7 \| 9.5-7 | 3.3-8 \| 3.3-8 | 10 \| 10 |
| theta10 | 12470 ; 500 | 0 \| 0 \| 340 | 340 | 9.8-7 \| 9.8-7 | 3.2-6 \| 3.2-6 | 32 \| 31 |
| theta102 | 37467 ; 500 | 0 \| 0 \| 150 | 150 | 8.7-7 \| 8.7-7 | 6.4-7 \| 6.4-7 | 15 \| 14 |
| theta103 | 62516 ; 500 | 0 \| 0 \| 202 | 202 | 9.8-7 \| 9.8-7 | -4.2-8 \| -4.2-8 | 20 \| 20 |
| theta104 | 87245 ; 500 | 0 \| 0 \| 162 | 162 | 9.8-7 \| 9.8-7 | 5.9-8 \| 5.9-8 | 16 \| 16 |
| theta12 | 17979 ; 600 | 0 \| 0 \| 354 | 354 | 9.5-7 \| 9.5-7 | -3.9-6 \| -3.9-6 | 48 \| 47 |
| theta123 | 90020 ; 600 | 0 \| 0 \| 204 | 204 | 9.7-7 \| 9.7-7 | -9.2-8 \| -9.2-8 | 30 \| 28 |
| san200-0.7-1 | 5971 ; 200 | 4 \| 5 \| 500 | 2197 | 3.2-7 \| 9.3-7 | 6.3-9 \| 6.1-6 | 06 \| 21 |
| sanr200-0.7 | 6033 ; 200 | 0 \| 0 \| 177 | 177 | 9.5-7 \| 9.5-7 | 1.9-7 \| 1.9-7 | 03 \| 02 |
| c-fat200-1 | 18367 ; 200 | 8 \| 8 \| 1050 | 1972 | 9.6-7 \| 9.9-7 | -7.7-6 \| -2.6-6 | 15 \| 23 |

Table 4.1: The performance of QSDPNAL (a) and sGS-PADMM(b) on QSDP-$\theta_+$, QSDP-QAP and QSDP-BIQ problems (accuracy = $10^{-6}$). The computation time is in the format of "hours:minutes:seconds".

| | | iter.a | iter.b | $\eta_{\rm qsdp}$ | $\eta_{\rm gap}$ | time |
|---|---|---|---|---|---|---|
| problem | $m_E; n_s$ | it\|itsub\|itsGS | | a\|b | a\|b | a\|b |
| hamming-8-4 | 11777 ; 256 | 0 \| 0 \| 2493 | 2493 | 9.9-7 \| 9.9-7 | -6.0-7 \| -6.0-7 | 51 \| 48 |
| hamming-9-8 | 2305 ; 512 | 249 \| 249 \| 600 | 4120 | 9.9-7 \| 9.9-7 | -2.6-8 \| -4.4-6 | 1:56 \| 5:34 |
| hamming-8-3-4 | 16129 ; 256 | 0 \| 0 \| 202 | 202 | 6.9-7 \| 6.9-7 | 5.4-6 \| 5.4-6 | 05 \| 04 |
| hamming-9-5-6 | 53761 ; 512 | 0 \| 0 \| 446 | 446 | 8.2-7 \| 8.2-7 | -1.1-5 \| -1.1-5 | 47 \| 42 |
| brock200-1 | 5067 ; 200 | 0 \| 0 \| 198 | 198 | 9.7-7 \| 9.7-7 | 9.9-8 \| 9.9-8 | 03 \| 02 |
| brock200-4 | 6812 ; 200 | 0 \| 0 \| 201 | 201 | 9.3-7 \| 9.3-7 | 1.1-7 \| 1.1-7 | 03 \| 03 |
| brock400-1 | 20078 ; 400 | 0 \| 0 \| 168 | 168 | 9.0-7 \| 9.0-7 | 8.6-7 \| 8.6-7 | 11 \| 10 |
| keller4 | 5101 ; 171 | 0 \| 0 \| 669 | 669 | 9.9-7 \| 9.9-7 | -1.3-8 \| -1.3-8 | 08 \| 07 |
| p-hat300-1 | 33918 ; 300 | 0 \| 0 \| 452 | 452 | 9.9-7 \| 9.9-7 | -1.0-6 \| -1.0-6 | 13 \| 12 |
| G43 | 9991 ; 1000 | 4 \| 4 \| 700 | 982 | 8.8-7 \| 9.5-7 | 7.1-7 \| -5.0-6 | 4:39 \| 5:38 |
| G44 | 9991 ; 1000 | 4 \| 4 \| 700 | 955 | 6.2-7 \| 8.8-7 | 5.4-7 \| 4.6-6 | 4:39 \| 5:31 |
| G45 | 9991 ; 1000 | 4 \| 4 \| 700 | 954 | 5.5-7 \| 9.0-7 | 4.2-7 \| 4.8-6 | 4:41 \| 5:29 |
| G46 | 9991 ; 1000 | 4 \| 4 \| 700 | 1000 | 8.6-7 \| 8.8-7 | -1.8-7 \| 6.6-6 | 4:36 \| 6:19 |
| G47 | 9991 ; 1000 | 4 \| 4 \| 702 | 985 | 5.9-7 \| 9.2-7 | 4.0-6 \| -4.8-6 | 4:40 \| 7:49 |
| 1dc.256 | 3840 ; 256 | 5 \| 7 \| 600 | 2312 | 6.5-7 \| 9.4-7 | 1.1-6 \| -1.6-5 | 12 \| 38 |
| 1et.256 | 1665 ; 256 | 0 \| 0 \| 4972 | 4972 | 9.9-7 \| 9.9-7 | -4.9-7 \| -4.9-7 | 1:36 \| 1:48 |
| 1tc.256 | 1313 ; 256 | 2 \| 4 \| 9512 | 12051 | 9.9-7 \| 9.9-7 | -4.0-6 \| -3.2-6 | 3:05 \| 4:25 |
| 1zc.256 | 2817 ; 256 | 0 \| 0 \| 3147 | 3147 | 9.9-7 \| 9.9-7 | -3.7-7 \| -3.7-7 | 1:02 \| 1:00 |
| 1dc.512 | 9728 ; 512 | 0 \| 0 \| 2032 | 2032 | 9.9-7 \| 9.9-7 | -4.4-7 \| -4.4-7 | 3:25 \| 3:12 |
| 1et.512 | 4033 ; 512 | 8 \| 8 \| 4297 | 4440 | 9.7-7 \| 9.8-7 | -1.8-6 \| -2.9-6 | 7:13 \| 7:50 |
| 1tc.512 | 3265 ; 512 | 1 \| 7 \| 12591 | 11801 | 9.9-7 \| 9.9-7 | -4.4-6 \| -4.4-6 | 20:58 \| 25:35 |
| 2dc.512 | 54896 ; 512 | 0 \| 0 \| 2368 | 2368 | 9.9-7 \| 9.9-7 | -5.0-6 \| -5.0-6 | 3:52 \| 5:42 |
| 1zc.512 | 6913 ; 512 | 0 \| 0 \| 2719 | 2719 | 9.9-7 \| 9.9-7 | -3.4-6 \| -3.4-6 | 4:38 \| 6:40 |
| 1dc.1024 | 24064 ; 1024 | 0 \| 0 \| 2418 | 2418 | 9.9-7 \| 9.9-7 | -8.5-7 \| -8.5-7 | 18:38 \| 22:41 |
| 1et.1024 | 9601 ; 1024 | 0 \| 0 \| 3186 | 3186 | 9.9-7 \| 9.9-7 | -5.1-7 \| -5.1-7 | 25:31 \| 21:28 |
| 1tc.1024 | 7937 ; 1024 | 5 \| 6 \| 5199 | 5922 | 9.8-7 \| 9.9-7 | -7.5-6 \| -1.0-5 | 39:22 \| 39:25 |
| 1zc.1024 | 16641 ; 1024 | 8 \| 8 \| 1938 | 3113 | 9.9-7 \| 9.9-7 | 6.9-6 \| 7.8-6 | 14:48 \| 21:07 |
| 2dc.1024 | 169163 ; 1024 | 0 \| 0 \| 3460 | 3460 | 9.7-7 \| 9.7-7 | -3.0-5 \| -3.0-5 | 28:11 \| 23:24 |
| be250.1 | 251 ; 251 | 88 \| 108 \| 1589 | 4120 | 9.9-7 \| 9.9-7 | 7.0-7 \| -6.4-7 | 38 \| 1:07 |
| be250.2 | 251 ; 251 | 143 \| 213 \| 1980 | 3555 | 8.6-7 \| 9.9-7 | 1.8-7 \| -7.5-7 | 51 \| 58 |
| be250.3 | 251 ; 251 | 120 \| 152 \| 1680 | 3558 | 9.4-7 \| 9.9-7 | -9.7-8 \| -9.6-7 | 43 \| 58 |
| be250.4 | 251 ; 251 | 93 \| 124 \| 1650 | 4072 | 9.9-7 \| 9.9-7 | 8.5-7 \| -2.1-6 | 40 \| 1:05 |

Table 4.1: The performance of Qsdpnal (a) and sGS-padmm(b) on QSDP-$\theta_+$, QSDP-QAP and QSDP-BIQ problems (accuracy = $10^{-6}$). The computation time is in the format of "hours:minutes:seconds".

| | | iter.a | iter.b | $\eta_{qsdp}$ | $\eta_{gap}$ | time |
|---|---|---|---|---|---|---|
| problem | $m_E; n_s$ | it\|itsub\|itsGS | | a\|b | a\|b | a\|b |
| be250.5 | 251 ; 251 | 91 \| 124 \| 1639 | 3204 | 9.5-7 \| 9.9-7 | -3.8-8 \| -9.1-7 | 39 \| 52 |
| be250.6 | 251 ; 251 | 77 \| 99 \| 1394 | 3250 | 9.7-7 \| 9.9-7 | 1.5-6 \| -2.8-7 | 33 \| 51 |
| be250.7 | 251 ; 251 | 97 \| 133 \| 1728 | 3699 | 9.2-7 \| 9.9-7 | 1.2-7 \| -6.5-7 | 42 \| 59 |
| be250.8 | 251 ; 251 | 116 \| 149 \| 1516 | 3516 | 8.2-7 \| 9.9-7 | -1.8-7 \| -9.1-7 | 37 \| 56 |
| be250.9 | 251 ; 251 | 104 \| 128 \| 2139 | 3586 | 9.0-7 \| 9.9-7 | -5.8-7 \| -3.4-7 | 46 \| 59 |
| be250.10 | 251 ; 251 | 98 \| 131 \| 1750 | 3302 | 6.3-7 \| 9.9-7 | -2.7-7 \| -1.1-6 | 38 \| 52 |
| bqp100-1 | 101 ; 101 | 24 \| 26 \| 1134 | 1339 | 9.6-7 \| 9.9-7 | -9.0-7 \| -2.2-7 | 07 \| 07 |
| bqp100-2 | 101 ; 101 | 47 \| 52 \| 1717 | 2493 | 9.6-7 \| 9.9-7 | 2.8-7 \| 2.6-8 | 11 \| 13 |
| bqp100-3 | 101 ; 101 | 2 \| 2 \| 1661 | 1751 | 7.8-7 \| 9.9-7 | -6.6-9 \| -2.7-6 | 09 \| 09 |
| bqp100-4 | 101 ; 101 | 16 \| 16 \| 1478 | 2910 | 9.9-7 \| 9.7-7 | -6.7-7 \| -10.0-8 | 09 \| 16 |
| bqp100-5 | 101 ; 101 | 13 \| 14 \| 1746 | 1911 | 9.9-7 \| 9.9-7 | -3.3-7 \| -5.7-8 | 10 \| 10 |
| bqp100-6 | 101 ; 101 | 8 \| 8 \| 1383 | 1405 | 9.9-7 \| 9.9-7 | 5.0-7 \| 3.3-7 | 08 \| 08 |
| bqp100-7 | 101 ; 101 | 40 \| 44 \| 1322 | 1770 | 9.9-7 \| 9.9-7 | -9.8-7 \| -5.7-7 | 09 \| 10 |
| bqp100-8 | 101 ; 101 | 19 \| 21 \| 1454 | 1820 | 8.7-7 \| 9.9-7 | 5.6-7 \| 7.3-7 | 09 \| 10 |
| bqp100-9 | 101 ; 101 | 28 \| 28 \| 1371 | 2038 | 8.2-7 \| 9.9-7 | -6.7-7 \| 2.0-6 | 09 \| 11 |
| bqp100-10 | 101 ; 101 | 38 \| 52 \| 2331 | 2904 | 9.7-7 \| 9.7-7 | 1.6-7 \| 2.8-7 | 14 \| 15 |
| bqp250-1 | 251 ; 251 | 97 \| 119 \| 1864 | 3899 | 9.8-7 \| 9.9-7 | -3.1-7 \| -8.0-7 | 40 \| 1:02 |
| bqp250-2 | 251 ; 251 | 80 \| 107 \| 1712 | 4120 | 9.2-7 \| 9.9-7 | -1.9-8 \| -4.9-7 | 37 \| 1:06 |
| bqp250-3 | 251 ; 251 | 95 \| 133 \| 2103 | 4102 | 9.9-7 \| 9.9-7 | 7.2-7 \| -3.9-6 | 45 \| 1:04 |
| bqp250-4 | 251 ; 251 | 93 \| 105 \| 1611 | 3103 | 9.3-7 \| 9.9-7 | -1.9-7 \| -4.2-7 | 35 \| 50 |
| bqp250-5 | 251 ; 251 | 85 \| 111 \| 1664 | 4419 | 9.5-7 \| 9.9-7 | 4.2-7 \| -2.0-6 | 37 \| 1:10 |
| bqp250-6 | 251 ; 251 | 80 \| 100 \| 1470 | 2952 | 9.9-7 \| 9.9-7 | 1.3-6 \| -1.0-6 | 32 \| 47 |
| bqp250-7 | 251 ; 251 | 106 \| 131 \| 1469 | 3844 | 6.7-7 \| 9.9-7 | -8.7-8 \| -1.5-6 | 34 \| 1:01 |
| bqp250-8 | 251 ; 251 | 91 \| 113 \| 1605 | 2716 | 9.9-7 \| 9.9-7 | 7.3-8 \| -8.8-7 | 35 \| 43 |
| bqp250-9 | 251 ; 251 | 91 \| 130 \| 1674 | 4200 | 9.7-7 \| 9.8-7 | 4.2-7 \| -6.7-7 | 37 \| 1:06 |
| bqp250-10 | 251 ; 251 | 86 \| 107 \| 1396 | 3027 | 9.9-7 \| 9.9-7 | 9.4-7 \| -7.7-7 | 31 \| 47 |
| bqp500-1 | 501 ; 501 | 175 \| 250 \| 2508 | 6003 | 9.9-7 \| 9.9-7 | 1.7-7 \| -3.9-7 | 4:43 \| 7:58 |
| bqp500-2 | 501 ; 501 | 164 \| 253 \| 2186 | 6609 | 9.8-7 \| 9.8-7 | 2.8-7 \| -4.7-7 | 4:33 \| 8:52 |
| bqp500-3 | 501 ; 501 | 144 \| 213 \| 2205 | 7443 | 9.9-7 \| 9.8-7 | 4.4-7 \| 8.4-7 | 4:22 \| 9:53 |
| bqp500-4 | 501 ; 501 | 125 \| 161 \| 1574 | 6962 | 9.9-7 \| 9.9-7 | -1.0-6 \| -1.5-6 | 3:09 \| 9:10 |
| bqp500-5 | 501 ; 501 | 145 \| 194 \| 1676 | 5801 | 9.8-7 \| 8.9-7 | 1.2-7 \| 1.7-6 | 3:21 \| 7:44 |
| bqp500-6 | 501 ; 501 | 174 \| 245 \| 2104 | 6894 | 9.0-7 \| 9.9-7 | -4.3-7 \| -4.7-7 | 4:03 \| 9:22 |

Table 4.1: The performance of Qsdpnal (a) and sGS-padmm(b) on QSDP-$\theta_+$, QSDP-QAP and QSDP-BIQ problems (accuracy = $10^{-6}$). The computation time is in the format of "hours:minutes:seconds".

| | | iter.a | iter.b | $\eta_{\rm qsdp}$ | $\eta_{\rm gap}$ | time |
|---|---|---|---|---|---|---|
| problem | $m_E; n_s$ | it\|itsub\|itsGS | | a\|b | a\|b | a\|b |
| bqp500-7 | 501 ; 501 | 165 \| 232 \| 2373 | 6528 | 9.9-7 \| 9.9-7 | -3.7-7 \| -7.8-7 | 4:20 \| 8:45 |
| bqp500-8 | 501 ; 501 | 167 \| 244 \| 2609 | 6261 | 9.9-7 \| 9.9-7 | -4.9-7 \| -4.6-7 | 4:42 \| 8:15 |
| bqp500-9 | 501 ; 501 | 178 \| 270 \| 2904 | 6532 | 9.6-7 \| 9.9-7 | -5.2-7 \| 9.9-7 | 5:15 \| 8:44 |
| bqp500-10 | 501 ; 501 | 154 \| 218 \| 1924 | 6434 | 9.9-7 \| 9.9-7 | 2.2-7 \| 9.9-7 | 3:40 \| 8:33 |
| gka1d | 101 ; 101 | 13 \| 13 \| 1364 | 1600 | 8.9-7 \| 9.8-7 | -4.6-7 \| -4.2-7 | 08 \| 09 |
| gka2d | 101 ; 101 | 30 \| 41 \| 1550 | 1927 | 9.2-7 \| 9.9-7 | -7.1-8 \| -5.0-7 | 10 \| 11 |
| gka3d | 101 ; 101 | 11 \| 11 \| 1970 | 2292 | 9.9-7 \| 9.9-7 | -4.1-7 \| -3.7-7 | 12 \| 12 |
| gka4d | 101 ; 101 | 2 \| 2 \| 2038 | 2157 | 9.9-7 \| 9.6-7 | 3.5-7 \| 3.4-7 | 12 \| 12 |
| chr12a | 232 ; 144 | 46 \| 88 \| 3490 | 25000 | 9.9-7 \| 1.0-5 | -1.4-5 \| **-1.4-4** | 36 \| 3:03 |
| chr12b | 232 ; 144 | 33 \| 86 \| 4224 | 25000 | 9.9-7 \| 9.1-6 | -2.8-5 \| **-1.4-4** | 45 \| 3:03 |
| chr12c | 232 ; 144 | 70 \| 130 \| 4718 | 25000 | 9.9-7 \| 1.5-5 | -2.3-5 \| **-2.2-4** | 51 \| 3:03 |
| chr15a | 358 ; 225 | 45 \| 99 \| 4010 | 25000 | 9.8-7 \| 1.1-5 | -2.6-5 \| **-1.4-4** | 1:24 \| 5:39 |
| chr15b | 358 ; 225 | 75 \| 103 \| 4462 | 25000 | 9.9-7 \| 1.3-5 | -2.7-5 \| **-1.7-4** | 1:27 \| 5:40 |
| chr15c | 358 ; 225 | 47 \| 75 \| 3601 | 25000 | 9.9-7 \| 1.2-5 | -3.4-5 \| **-1.9-4** | 1:10 \| 5:41 |
| chr18a | 511 ; 324 | 61 \| 111 \| 4297 | 25000 | 9.9-7 \| 1.3-5 | -2.5-5 \| **-2.1-4** | 2:40 \| 11:26 |
| chr18b | 511 ; 324 | 764 \| 1083 \| 8210 | 25000 | 9.9-7 \| 1.4-6 | -1.1-6 \| -5.0-6 | 6:54 \| 10:48 |
| chr20a | 628 ; 400 | 72 \| 111 \| 5101 | 25000 | 9.9-7 \| 8.3-6 | -1.8-5 \| -9.9-5 | 6:12 \| 23:45 |
| chr20b | 628 ; 400 | 57 \| 103 \| 4544 | 25000 | 9.9-7 \| 8.1-6 | -1.4-5 \| -7.5-5 | 5:50 \| 23:47 |
| chr20c | 628 ; 400 | 101 \| 154 \| 6940 | 25000 | 9.9-7 \| 1.6-5 | -2.9-5 \| **-2.3-4** | 8:26 \| 23:41 |
| chr22a | 757 ; 484 | 44 \| 171 \| 5975 | 25000 | 9.9-7 \| 4.1-6 | -1.8-5 \| -6.5-5 | 12:13 \| 33:39 |
| chr22b | 757 ; 484 | 51 \| 180 \| 6284 | 25000 | 9.9-7 \| 3.4-6 | -1.7-5 \| -5.3-5 | 12:43 \| 33:39 |
| els19 | 568 ; 361 | 81 \| 281 \| 10293 | 25000 | 9.9-7 \| 2.5-6 | -1.5-5 \| -3.3-5 | 15:10 \| 22:51 |
| esc16a | 406 ; 256 | 39 \| 134 \| 3938 | 25000 | 9.9-7 \| 7.3-6 | -9.4-6 \| -7.2-5 | 1:42 \| 7:48 |
| esc16b | 406 ; 256 | 130 \| 469 \| 9020 | 25000 | 9.9-7 \| 9.0-6 | -1.6-5 \| **-2.0-4** | 4:20 \| 7:47 |
| esc16c | 406 ; 256 | 140 \| 465 \| 10483 | 25000 | 9.9-7 \| 7.4-6 | -5.6-5 \| **-1.4-4** | 4:54 \| 7:45 |
| esc16d | 406 ; 256 | 16 \| 16 \| 915 | 812 | 9.9-7 \| 9.9-7 | -3.5-7 \| -5.6-7 | 19 \| 15 |
| esc16e | 406 ; 256 | 21 \| 21 \| 930 | 983 | 9.8-7 \| 9.9-7 | 8.5-7 \| 7.4-7 | 19 \| 18 |
| esc16g | 406 ; 256 | 32 \| 33 \| 1339 | 1700 | 9.9-7 \| 9.8-7 | -9.9-7 \| -1.2-6 | 28 \| 31 |
| esc16h | 406 ; 256 | 26 \| 58 \| 2020 | 25000 | 8.5-7 \| 2.9-6 | -3.0-6 \| -1.7-5 | 47 \| 7:46 |
| esc16i | 406 ; 256 | 42 \| 67 \| 1718 | 1811 | 9.9-7 \| 9.9-7 | -5.1-7 \| -8.0-7 | 39 \| 33 |
| esc16j | 406 ; 256 | 46 \| 49 \| 1290 | 2363 | 9.7-7 \| 9.9-7 | 8.3-7 \| -2.4-6 | 28 \| 44 |
| had12 | 232 ; 144 | 43 \| 78 \| 3083 | 25000 | 9.8-7 \| 1.3-5 | -1.7-5 \| -9.4-5 | 31 \| 3:04 |

Table 4.1: The performance of QSDPNAL (a) and sGS-PADMM(b) on QSDP-$\theta_+$, QSDP-QAP and QSDP-BIQ problems (accuracy = $10^{-6}$). The computation time is in the format of "hours:minutes:seconds".

| | | iter.a | iter.b | $\eta_{\rm qsdp}$ | $\eta_{\rm gap}$ | time |
|---|---|---|---|---|---|---|
| problem | $m_E; n_s$ | it\|itsub\|itsGS | | a\|b | a\|b | a\|b |
| had14 | 313 ; 196 | 58 \| 90 \| 5427 | 25000 | 9.9-7 \| 1.0-5 | -1.4-5 \| -9.3-5 | 1:22 \| 4:38 |
| had16 | 406 ; 256 | 80 \| 143 \| 6286 | 25000 | 9.9-7 \| 1.3-5 | -1.5-5 \| -9.7-5 | 2:32 \| 7:30 |
| had18 | 511 ; 324 | 54 \| 120 \| 4387 | 25000 | 9.9-7 \| 1.1-5 | -1.1-5 \| -6.6-5 | 2:47 \| 11:48 |
| had20 | 628 ; 400 | 105 \| 146 \| 7808 | 25000 | 9.9-7 \| 1.2-5 | -1.5-5 \| **-1.1-4** | 9:21 \| 23:33 |
| nug12 | 232 ; 144 | 35 \| 51 \| 1786 | 25000 | 9.9-7 \| 7.3-6 | -2.1-5 \| -8.5-5 | 19 \| 3:11 |
| nug14 | 313 ; 196 | 29 \| 51 \| 2082 | 25000 | 9.9-7 \| 9.7-6 | -2.4-5 \| -9.8-5 | 32 \| 4:44 |
| nug15 | 358 ; 225 | 29 \| 52 \| 2056 | 25000 | 9.9-7 \| 9.2-6 | -1.7-5 \| -9.4-5 | 41 \| 5:43 |
| nug16a | 406 ; 256 | 40 \| 63 \| 2260 | 25000 | 9.9-7 \| 1.1-5 | -2.3-5 \| **-1.1-4** | 56 \| 7:51 |
| nug16b | 406 ; 256 | 41 \| 62 \| 2130 | 25000 | 9.7-7 \| 9.2-6 | -2.5-5 \| **-1.0-4** | 53 \| 7:48 |
| nug17 | 457 ; 289 | 32 \| 60 \| 2119 | 25000 | 9.9-7 \| 1.1-5 | -2.8-5 \| **-1.1-4** | 1:03 \| 9:21 |
| nug18 | 511 ; 324 | 34 \| 60 \| 2179 | 25000 | 9.9-7 \| 9.8-6 | -2.5-5 \| -9.8-5 | 1:19 \| 12:14 |
| nug20 | 628 ; 400 | 42 \| 70 \| 2269 | 25000 | 9.5-7 \| 9.4-6 | -2.1-5 \| -9.0-5 | 2:51 \| 24:40 |
| nug21 | 691 ; 441 | 43 \| 67 \| 2785 | 25000 | 9.8-7 \| 1.1-5 | -2.4-5 \| **-1.1-4** | 4:07 \| 30:05 |
| rou12 | 232 ; 144 | 41 \| 50 \| 1770 | 25000 | 9.8-7 \| 8.0-6 | -3.1-5 \| -8.9-5 | 17 \| 3:15 |
| rou15 | 358 ; 225 | 33 \| 45 \| 1640 | 25000 | 8.7-7 \| 7.2-6 | -1.9-5 \| -7.6-5 | 30 \| 6:01 |
| rou20 | 628 ; 400 | 31 \| 41 \| 1650 | 25000 | 9.9-7 \| 6.1-6 | -1.9-5 \| -5.6-5 | 1:51 \| 24:25 |
| scr12 | 232 ; 144 | 66 \| 93 \| 3190 | 25000 | 9.9-7 \| 7.4-6 | -7.4-6 \| -7.3-5 | 32 \| 3:14 |
| scr15 | 358 ; 225 | 62 \| 89 \| 3422 | 25000 | 9.9-7 \| 1.1-5 | -1.7-5 \| **-1.1-4** | 1:06 \| 5:51 |
| scr20 | 628 ; 400 | 52 \| 81 \| 3700 | 25000 | 9.9-7 \| 9.7-6 | -1.5-5 \| **-1.0-4** | 4:27 \| 24:12 |
| tai12a | 232 ; 144 | 40 \| 54 \| 2086 | 25000 | 9.6-7 \| 9.5-6 | -3.4-5 \| **-1.2-4** | 21 \| 3:15 |
| tai12b | 232 ; 144 | 56 \| 91 \| 4635 | 25000 | 9.9-7 \| 1.7-5 | -3.2-5 \| **-2.4-4** | 47 \| 3:11 |
| tai15a | 358 ; 225 | 36 \| 47 \| 1597 | 25000 | 9.4-7 \| 6.5-6 | -1.8-5 \| -6.1-5 | 30 \| 6:05 |
| tai15b | 358 ; 225 | 61 \| 165 \| 4330 | 4088 | 9.9-7 \| 9.9-7 | -2.7-6 \| -2.5-6 | 1:36 \| 58 |
| tai17a | 457 ; 289 | 34 \| 43 \| 1509 | 25000 | 9.8-7 \| 6.3-6 | -1.6-5 \| -5.6-5 | 43 \| 9:29 |
| tai20a | 628 ; 400 | 41 \| 51 \| 1627 | 25000 | 8.9-7 \| 5.5-6 | -1.6-5 \| -5.1-5 | 1:52 \| 24:26 |

In the second part of this section, we focus on the large scale convex quadratic programming problems. We test convex quadratic programming problems constructed in (3.86) which have been used in the test of Phase I algorithm (sGS-PADMM). We measure the accuracy of an approximate optimal solution $(x, z, x', s, y, \bar{y})$ for convex quadratic programming (4.46) and its dual (4.47) by using the following relative residual :

$$\eta_{\rm qp} = \max\{\eta_P, \eta_D, \eta_Q, \eta_z, \eta_{\bar y}\}, \tag{4.57}$$

where

$$\eta_P = \frac{\|AX - b\|}{1+\|b\|}, \quad \eta_D = \frac{\|z - Qx' + s + A^*y + B^*\bar y - C\|}{1+\|c\|},$$
$$\eta_Z = \frac{\|x - \Pi_{\mathcal{K}}(x-z)\|}{1+\|x\|+\|z\|}, \quad \eta_{\bar y} = \frac{\|\bar y - \Pi_{\mathcal{C}^\circ}(\bar y - Bx + \bar b)\|}{1+\|\bar y\|+\|Bx\|},$$
$$\eta_Q = \frac{\|Qx - Qx'\|}{1+\|Qx\|}.$$

Note that in Phase I, we terminate the sGS-padmm when $\eta_{\rm qp} < 10^{-5}$. Now, with the help of Phase II algorithm, we hope to obtain high accuracy solutions efficiently with $\eta_{\rm qp} < 10^{-6}$. Here, we test the very special implementation of our Phase II algorithm, the inexact symmetric Gauss-Seidel based proximal augmented Lagrangian algorithm (inexact sGS-Aug), for solving convex quadratic programming problems. We will switch the solver from sGS-padmm to inexact sGS-Aug when $\eta_{\rm qp} < 10^{-5}$ and stop the whole process when $\eta_{\rm qp} < 10^{-6}$.

Table 4.2: The performance of inexact sGS-Aug on randomly generated BIQ-QP problems (accuracy $= 10^{-6}$). The computation time is in the format of "hours:minutes:seconds".

| problem \| n \| $m_E, m_I$ | $(\mathrm{A,B,Q})_{\rm blk}$ | it\|itsGS | $\eta_{\rm qp}$ | $\eta_{\rm gap}$ | time |
|---|---|---|---|---|---|
| be100.1 \|5150 \|200,14850 | (2,25,25) | 24 \| 901 | 6.1-7 | 1.4-8 | 58 |
| be120.3.1 \|7380 \|240,21420 | (2,25,25) | 42 \| 694 | 7.7-7 | 6.2-8 | 56 |
| be150.3.1 \|11475 \|300,33525 | (2,25,25) | 17 \| 703 | 8.2-7 | 7.1-8 | 1:51 |
| be200.3.1 \|20300 \|400,59700 | (2,50,50) | 25 \| 860 | 9.5-7 | -3.2-8 | 5:31 |
| be250.1 \|31625 \|500,93375 | (2,50,50) | 20 \| 1495 | 7.1-7 | 3.3-8 | 18:10 |

Table 4.2 reports the detailed numerical results for inexact sGS-Aug for solving convex quadratic programming problems (3.86). In the table, "it" stands for the number of iterations of inexact sGS-Aug. "itersGS" stands for the total number

of iterations of sGS-padmm used to warm start sGS-Aug with its decomposition parameters set to be $(\mathrm{A}, \mathrm{B}, \mathrm{Q})_{\mathrm{blk}}$. As can be observed, our Phase II algorithm can obtain high accuracy solutions efficiently. This fact again demonstrates the power and the necessity of our proposed two phase framework.

Chapter 5

# Conclusions

In this thesis, we designed algorithms for solving high dimensional convex composite quadratic programming problems with large numbers of linear equality and inequality constraints. In order to solve the targeted problems to desired accuracy efficiently, we introduced a two phase augmented Lagrangian method, with Phase I to generate a reasonably good initial point and Phase II to obtain accurate solutions fast.

In Phase I, by carefully examining a class of convex composite quadratic programming problems, we introduced the one cycle symmetric block Gauss-Seidel technique. This technique enabled us to deal with the nonseparable structure in the objective function even when a coupled nonsmooth term was involving. Based on this technique, we were able to design a novel symmetric Gauss-Seidel based proximal ADMM (sGS-PADMM) for solving convex composite quadratic programming. The ability of dealing with coupling quadratic terms in the objective function made the proposed algorithm very flexible in solving various multi-block convex optimization problems. By conducting numerical experiments including large scale convex quadratic programming (QP) problems and convex quadratic semidefinite programming (QSDP) problems, we presented convincing numerical results to demonstrate the superior performance of our proposed sGS-PADMM.

In Phase II, in order to obtain more accurate solutions efficiently, we studied the inexact proximal augmented Lagrangian method (pALM). We establish the global convergence of our proposed algorithm based on the classic results of proximal point algorithms. Under the error bound assumption, the local linear convergence of Algorithm pALM was also analyzed. The inner subproblems were solved by an inexact alternating minimization method. Then, we specialized the proposed pALM algorithm to QSDP problems and convex QP problems. We discussed in detail the implementation issues of solving the resulted inner subproblems. The aforementioned symmetric Gauss-Seidel technique was also shown can be wisely incorporated into our Phase II algorithm. Numerical experiments conducted on a variety of large scale difficult convex QSDP problems and high dimensional convex QP problems demonstrated that our proposed algorithms can efficiently solve these problems to high accuracy.

There are still many interesting problems that will lead to further development of algorithms for solving convex composite quadratic optimization problems. Below we briefly list some research directions that deserve more explorations.

- Is it possible to extend our one cycle symmetric block Gauss-Seidel technique to more general cases with more than one nonsmooth terms involved?
- In Phase I, can one find a simpler and better algorithm than sGS-PADMM for general convex problems?
- In Phase II, is it possible to provide some reasonably weak and manageable sufficient conditions to guarantee the error bound assumption for QSDP problems?

# Bibliography

[1] A. Y. Alfakih, A. Khandani, and H. Wolkowicz, *Solving Euclidean distance matrix completion problems via semidefinite programming*, Computational optimization and applications, 12 (1999), pp. 13–30.

[2] S. Bi, S. Pan, and J.-S. Chen, *Nonsingularity conditions for the fischer-burmeister system of nonlinear SDPs*, SIAM Journal on Optimization, 21 (2011), pp. 1392–1417.

[3] R. E. Burkard, S. E. Karisch, and F. Rendl, *QAPLIB – a quadratic assignment problem library*, Journal of Global optimization, 10 (1997), pp. 391–403.

[4] 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, (2014), pp. 1–23.

[5] 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 Publishing Corporation, 2013.

[6] F. Clarke, *Optimization and nonsmooth analysis*, John Wiley and Sons, New York, 1983.

[7] R. W. Cottle, *Symmetric dual quadratic programs*, tech. report, DTIC Document, 1962.

[8] ——, *Note on a fundamental theorem in quadratic programming*, Journal of the Society for Industrial & Applied Mathematics, 12 (1964), pp. 663–665.

[9] G. B. Dantzig, *Quadratic programming: A variant of the wolfe-markowitz algorithms*, tech. report, DTIC Document, 1961.

[10] ——, *Quadratic programming*, in Linear Programming and Extensions, Princeton Uiversity Press, Princeton, USA, 1963, ch. 12-4, pp. 490–498.

[11] 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.

[12] F. Facchinei and J.-S. Pang, *Finite-dimensional variational inequalities and complementarity problems*, vol. 1, Springer, 2003.

[13] M. Fazel, T. K. Pong, D. F. 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.

[14] M. Fortin and R. Glowinski, *Augmented Lagrangian methods*, vol. 15 of Studies in Mathematics and its Applications, North-Holland Publishing Co., Amsterdam, 1983. Applications to the numerical solution of boundary value problems, Translated from the French by B. Hunt and D. C. Spicer.

[15] D. Gabay, *Applications of the method of multipliers to variational inequalities*, in Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems, M. Fortin and R. Glowinski, eds., vol. 15 of Studies in Mathematics and Its Applications, Elsevier, 1983, pp. 299–331.

[16] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite element approximation*, Computers and Mathematics with Applications, 2 (1976), pp. 17–40.

[17] R. Glowinski, *Lectures on numerical methods for nonlinear variational problems*, vol. 65 of Tata Institute of Fundamental Research Lectures on Mathematics and Physics, Tata Institute of Fundamental Research, Bombay, 1980. Notes by M. G. Vijayasundaram and M. Adimurthi.

[18] R. Glowinski and A. Marrocco, *Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité, d'une classe de problèmes de dirichlet non linéares*, Revue Francaise d'Automatique, Informatique et Recherche Opérationelle, 9 (1975), pp. 41–76.

[19] N. I. Gould, *On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem*, Mathematical Programming, 32 (1985), pp. 90–99.

[20] N. I. Gould, M. E. Hribar, and J. Nocedal, *On the solution of equality constrained quadratic programming problems arising in optimization*, SIAM Journal on Scientific Computing, 23 (2001), pp. 1376–1395.

[21] N. I. Gould and P. L. Toint, *A quadratic programming bibliography*, Numerical Analysis Group Internal Report, 1 (2000).

[22] I. Gurobi Optimization, *Gurobi optimizer reference manual*, 2015.

[23] 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.

[24] 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.

[25] B. He and X. Yuan, *Linearized alternating direction method of multipliers with Gaussian back substitution for separable convex programming*, Numerical Algebra, Control and Optimization, 3 (2013), pp. 247–260.

[26] N. J. Higham, *Computing the nearest correlation matrix – a problem from finance*, IMA journal of Numerical Analysis, 22 (2002), pp. 329–343.

[27] J.-B. Hiriart-Urruty, J.-J. Strodiot, and V. H. Nguyen, *Generalized hessian matrix and second-order optimality conditions for problems with $C^{1,1}$ data*, Applied mathematics and optimization, 11 (1984), pp. 43–56.

[28] M. Hong and Z.-Q. Luo, *On the linear convergence of the alternating direction method of multipliers*, arXiv preprint arXiv:1208.3922, (2012).

[29] K. Jiang, D. F. Sun, and K.-C. Toh, *An inexact accelerated proximal gradient method for large scale linearly constrained convex SDP*, SIAM Journal on Optimization, 22 (2012), pp. 1042–1064.

[30] O. Klopp, *Noisy low-rank matrix completion with general sampling distribution*, Bernoulli, 20 (2014), pp. 282–303.

[31] N. Krislock, J. Lang, J. Varah, D. K. Pai, and H.-P. Seidel, *Local compliance estimation via positive semidefinite constrained least squares*, IEEE Transactions on Robotics, 20 (2004), pp. 1007–1011.

[32] C. Lemaréchal and J.-B. Hiriart-Urruty, *Convex analysis and minimization algorithms* II, vol. 306 of Grundlehren der mathematischen Wissenschaften, Springer-Verlag Berlin Heidelberg, 1993.

[33] L. Li and K.-C. Toh, *An inexact interior point method for l1-regularized sparse covariance selection*, Mathematical Programming Computation, 2 (2010), pp. 291–315.

[34] M. Li, D. Sun, and K.-C. Toh, *A convergent 3-block semi-proximal ADMM for convex minimization problems with one strongly convex block*, arXiv preprint arXiv:1410.7933, (2014).

[35] ——, *A majorized ADMM with indefinite proximal terms for linearly constrained convex composite optimization*, arXiv preprint arXiv:1412.1911, (2014).

[36] T. Lin, S. Ma, and S. Zhang, *On the convergence rate of multi-block ADMM*, arXiv preprint arXiv:1408.4265, (2014).

[37] ——, *On the global linear convergence of the ADMM with multi-block variables*, arXiv preprint arXiv:1408.4266, (2014).

[38] F. J. Luque, *Asymptotic convergence analysis of the proximal point algorithm*, SIAM Journal on Control and Optimization, 22 (1984), pp. 277–293.

[39] B. C. D. J. C. M. M. Trick, V. Chvatal and R. Tarjan, *The second dimacs implementation challenge – NP hard problems: Maximum clique, graph coloring, and satisfiability*, (1992).

[40] F. Meng, D. F. Sun, and G. Zhao, *Semismoothness of solutions to generalized equations and the Moreau-Yosida regularization*, Mathematical programming, 104 (2005), pp. 561–581.

[41] W. Miao, *Matrix Completion Models with Fixed Basis Coefficients and Rank Regularized Prbolems with Hard Constraints*, PhD thesis, Department of Mathematics, Nationla University of Singapore, 2013.

[42] W. Miao, S. Pan, and D. F. Sun, *A rank-corrected procedure for matrix completion with fixed basis coefficients*, Technical Report, National University of Singapore, (2014).

[43] R. Mifflin, *Semismooth and semiconvex functions in constrained optimization*, SIAM Journal on Control and Optimization, 15 (1977), pp. 959–972.

[44] J. J. Moreau, *Proximité et dualité dans un espace hilbertien*, Bulletin de la Societe Mathematique de France, 93 (1965), pp. 273–299.

[45] S. Negahban and M. J. Wainwright, *Restricted strong convexity and weighted matrix completion: Optimal bounds with noise*, The Journal of Machine Learning Research, 13 (2012), pp. 1665–1697.

[46] Y. Nesterov, *Gradient methods for minimizing composite functions*, Mathematical Programming, 140 (2013), pp. 125–161.

[47] J.-S. Pang, D. F. Sun, and J. Sun, *Semismooth homeomorphisms and strong stability of semidefinite and Lorentz complementarity problems*, Mathematics of Operations Research, 28 (2003), pp. 39–63.

[48] J. Peng and Y. Wei, *Approximating k-means-type clustering via semidefinite programming*, SIAM Journal on Optimization, 18 (2007), pp. 186–205.

[49] H. Qi, *Local duality of nonlinear semidefinite programming*, Mathematics of Operations Research, 34 (2009), pp. 124–141.

[50] H. Qi and D. F. Sun, *A quadratically convergent Newton method for computing the nearest correlation matrix*, SIAM journal on matrix analysis and applications, 28 (2006), pp. 360–385.

[51] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Mathematical Programming, 58 (1993), pp. 353–367.

[52] S. M. Robinson, *Some continuity properties of polyhedral multifunctions*, in Mathematical Programming at Oberwolfach, vol. 14 of Mathematical Programming Studies, Springer Berlin Heidelberg, 1981, pp. 206–214.

[53] R. T. Rockafellar, *Convex analysis*, Princeton Mathematical Series, No. 28, Princeton University Press, Princeton, N.J., 1970.

[54] R. T. Rockafellar, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Mathematics of operations research, 1 (1976), pp. 97–116.

[55] ——, *Monotone operators and the proximal point algorithm*, SIAM Journal on Control and Optimization, 14 (1976), pp. 877–898.

[56] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*, vol. 317 of Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Springer-Verlag, Berlin, 1998.

[57] N. Sloane, *Challenge problems: Independent sets in graphs*, 2000.

[58] D. F. Sun and J. Sun, *Semismooth matrix-valued functions*, Mathematics of Operations Research, 27 (2002), pp. 150–169.

[59] D. F. Sun, K.-C. Toh, and L. Yang, *A convergent 3-block semi-proximal alternating direction method of multipliers for conic programming with* 4*-type of constraints*, arXiv preprint arXiv:1404.5378, (2014).

[60] J. Sun, *A convergence proof for an affine-scaling algorithm for convex quadratic programming without nondegeneracy assumptions*, Mathematical Programming, 60 (1993), pp. 69–79.

[61] J. Sun and S. Zhang, *A modified alternating direction method for convex quadratically constrained quadratic semidefinite programs*, European Journal of Operational Research, 207 (2010), pp. 1210–1220.

[62] 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.

[63] K. Toh, R. Tütüncü, and M. Todd, *Inexact primal-dual path-following algorithms for a special class of convex quadratic SDP and related problems*, Pacific Journal of optimization, 3 (2007).

[64] K.-C. Toh, *Solving large scale semidefinite programs via an iterative solver on the augmented systems*, SIAM Journal on Optimization, 14 (2004), pp. 670–698.

[65] ——, *An inexact primal–dual path following algorithm for convex quadratic SDP*, Mathematical programming, 112 (2008), pp. 221–254.

[66] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, *Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization*, in Proc. of Neural Information Processing Systems, vol. 3, 2009.

[67] S. Wright and J. Nocedal, *Numerical optimization*, vol. 2, Springer New York, 1999.

[68] B. Wu, *High-Dimensional Analysis on Matrix Decomposition with Application to Correlation Matrix Estimation in Factor Models*, PhD thesis, Department of Mathematics, Nationla University of Singapore, 2014.

[69] L. Yang, D. F. Sun, and K.-C. Toh, *SDPNAL +: A majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints*, arXiv preprint arXiv:1406.0942, (2014).

[70] Y. Ye, *On the complexity of approximating a KKT point of quadratic programming*, Mathematical programming, 80 (1998), pp. 195–211.

[71] K. Yosida, *Functional analysis*, vol. 11, 1995.

[72] X. Y. Zhao, *A semismooth Newton-CG augmented Lagrangian method for large scale linear and convex quadratic SDPs*, PhD thesis, Department of Mathematics, National University of Singapore, 2009.

[73] X.-Y. Zhao, D. F. Sun, and K.-C. Toh, *A Newton-CG augmented Lagrangian method for semidefinite programming*, SIAM Journal on Optimization, 20 (2010), pp. 1737–1765.

# A TWO-PHASE AUGMENTED LAGRANGIAN METHOD FOR CONVEX COMPOSITE QUADRATIC PROGRAMMING

**LI XUDONG**

**NATIONAL UNIVERSITY OF SINGAPORE**

**2015**

**A two-phase augmented Lagrangian method for convex composite quadratic programming** **Li Xudong** **2015**