A SEMISMOOTH NEWTON-CG AUGMENTED LAGRANGIAN METHOD FOR LARGE SCALE LINEAR AND CONVEX QUADRATIC SDPS

ZHAO XINYUAN

(M.Sc., NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF MATHEMATICS

NATIONAL UNIVERSITY OF SINGAPORE

2009

Acknowledgements

I hope to express my gratitude towards a number of people who have supported and encouraged me in the work of this thesis.

First and foremost, I would like to express my deepest respect and most sincere gratitude to my advisor Professor Toh Kim Chuan. He has done a lot for me in the last six years, since I signed up with him as a masters student in my first year, curious to learn more about optimization. With his endless supply of fresh ideas and openness to looking at new problems in different areas, his guidance has proved to be indispensable to my research. I will always remember the patient guidance, encouragement and advice he has provided throughout my time as his student.

My deepest thanks to my co-advisor Professor Defeng Sun, for his patient introducing me into the field of convex optimization, for his enthusiasm about discussing mathematical issues and for the large amount of time he devoted to my concerns. This work would not have been possible without his amazing depth of knowledge and tireless enthusiasm. I am very fortunate to have had the opportunity to work with him.

My grateful thanks also go to Professor Gongyun Zhao for his courses on numerical optimization. It was his unique style of teaching that enrich my knowledge in optimization algorithms and software. I am much obliged to my group members of optimization in mathematical department. Their enlightening suggestions and encouragements made me feel I was not isolated in my research. I feel very lucky to have been a part of this group, and I will cherish the memories of my time with them.

I would like to thank the Mathematics Department at the National University of Singapore provided excellent working conditions and support that providing the Scholarship which allowed me to undertake this research and complete my thesis.

I am as ever, especially indebted to my parents, for their unquestioning love and support encouraged me to complete this work. Last but not least, I am also greatly indebted to my husband not only for his constant encouragement but also for his patience and understanding throughout the years of my research.

Contents

Summary

This thesis presents a semismooth Newton-CG augmented Lagrangian method for solving linear and convex quadratic semidefinite programming problems from the perspective of approximate Newton methods. We study, under the framework of Euclidean Jordan algebras, the properties of minimization problems of linear and convex objective functions subject to linear, second-order, and positive semidefinite cone constraints simultaneously.

We exploit classical results of proximal point methods and recent advances on sensitivity and perturbation analysis of nonlinear conic programming to analyze the convergence of our proposed method. For the inner problems developed in our method, we show that the positive definiteness of the generalized Hessian of the objective function in these inner problems, a key property for ensuring the efficiency of using an inexact semismooth Newton-CG method to solve the inner problems, is equivalent to an interesting condition corresponding to the dual problems.

As a special case, linear symmetric cone programming is thoroughly examined under this framework. Based on the the nice and simple structure of linear symmetric cone programming and its dual, we characterize the Lipschitz continuity of the solution mapping for the dual problem at the origin.

Numerical experiments on a variety of large scale convex linear and quadratic semidefinite programming show that the proposed method is very efficient. In particular, two classes of convex quadratic semidefinite programming problems – the nearest correlation matrix problem and the Euclidean distance matrix completion problem are discussed in details. Extensive numerical results for large scale SDPs show that the proposed method is very powerful in solving the SDP relaxations arising from combinatorial optimization or binary integer quadratic programming.

Chapter 1

Introduction

In the recent years convex quadratic semidefinite programming (QSDP) problems have received more and more attention. The importance of convex quadratic semidefinite programming problems is steadily increasing thanks to the many important application areas of engineering, mathematical, physical, management sciences and financial economics. More recently, from the development of the theory in nonlinear and convex programming [114, 117, 24], in this thesis we are strongly spurred by the study of the theory and algorithm for solving large scale convex quadratic programming over special symmetric cones. Because of the inefficiency of interior point methods for large scale SDPs, we introduce a semismooth Newton-CG augmented Lagrangian method to solve the large scale convex quadratic programming problems.

The important family of linear programs enters the framework of convex quadratic programming with a zero quadratic term in their objective functions. For linear semidefinite programming, there are many applications in combinatorial optimization, control theory, structural optimization and statistics, see the book by Wolkowicz, Saigal and Vandenberghe [133]. Because of the simple structure of linear SDP and its dual, we extend the theory and algorithm to linear conic programming and investigate the conditions of the convergence for the semismooth Newton-CG augmented Lagrangian algorithm.

1.1 Motivation and related approaches

Since the 1990s, semidefinite programming has been one of the most exciting and active research areas in optimization. There are tremendous research achievement on the theory, algorithms and applications of semidefinite programming. The standard convex quadratic semidefinite programming (SDP) is defined to be

$$(QSDP) \qquad \begin{array}{ll} \min & \frac{1}{2}\langle X,\, \mathcal{Q}(X)\rangle + \langle C,\, X\rangle \\ \text{s.t.} & \mathcal{A}(X) = b, \\ & X \succeq 0, \end{array}$$

where $\mathcal{Q} : \mathcal{S}^n \to \mathcal{S}^n$ is a given self-adjoint and positive semidefinite linear operator, $\mathcal{A} : \mathcal{S}^n \to \Re^m$ is a linear mapping, $b \in \Re^m$, and $\mathcal{S}^n$ is the space of $n \times n$ symmetric matrices endowed with the standard trace inner product. The notation $X \succeq 0$ means that X is positive semidefinite. Of course, convex quadratic SDP includes linear SDP as a special case, by taking $\mathcal{Q} = 0$ in the problem $(QSDP)$ (see [19] and [133] for example). When we use sequential quadratic programming techniques to solve nonlinear semidefinite optimization problems, we naturally encounter $(QSDP)$.

Since $\mathcal{Q}$ is self-adjoint and positive semidefinite, it has a self-adjoint and positive semidefinite square root $\mathcal{Q}^{1/2}$. Then the $(QSDP)$ can be equivalently written as the following linear conic programming

$$\begin{array}{ll} \min & t + \langle C,\, X\rangle \\ \text{s.t.} & \mathcal{A}(X) = b, \\ & \sqrt{(t-1)^2 + 2\|\mathcal{Q}^{1/2}(X)\|_F^2} \leq (t+1), \\ & X \succeq 0, \end{array} \tag{1.1}$$

where $\|\cdot\|_F$ denotes Frobenius norm. This suggests that one may then use those well developed and publicly available softwares, based on interior point methods (IPMs), such as SeDuMi [113] and SDPT3 [128], and a few others to solve (1.1), and so the problem $(QSDP)$, directly. For convex optimization problems, interior-point methods

(IPMs) have been well developed since they have strong theoretical convergence [82, 134]. However, since at each iteration these solvers require to formulate and solve a dense Schur complement matrix (cf. [17]), which for the problem $(QSDP)$ amounts to a linear system of dimension $(m + 2 + n^2) \times (m + 2 + n^2)$. Because of the very large size and ill-conditioning of the linear system of equations, direct solvers are difficult to solve it. Thus interior point methods with direct solvers, efficient and robust for solving small and medium sized SDP problems, face tremendous difficulties in solving large scale problems. By appealing to specialized preconditioners, interior point methods can be implemented based on iterative solvers to overcome the ill-conditioning (see [44, 8]). In [81], the authors consider an interior-point algorithm based on reducing a primal-dual potential function. For the large scale linear system, the authors suggested using the conjugate gradient (CG) method to compute an approximate direction. Toh et al [123] and Toh [122] proposed inexact primal-dual path-following methods to solve a class of convex quadratic SDPs and related problems.

There also exist a number of non-interior point methods for solving large scale convex QSDP problems. Kočvara and Stingl [60] used a modified barrier method (a variant of the Lagrangian method) combined with iterative solvers for convex nonlinear and semidefinite programming problems having only inequality constraints and reported computational results for the code PENNON [59] with the number of equality constraints up to $125,000$. Malick, Povh, Rendl, and Wiegele [73] applied the Moreau-Yosida regularization approach to solve linear SDPs. As shown in the computational experiments, their regularization methods are efficient on several classes of large-scale SDP problems (n not too large, say $n \leq 1000$, but with a large number of constraints). Related to the boundary point method [88] and the regularization methods presented in [73], the approach of Jarre and Rendl [55] is to reformulate the linear conic problem as the minimization of a convex differentiable function in the primal-dual space.

Before we talk more about other numerical methods, let us first introduce some applications of convex QSDP problems arising from financial economics, combinatorial optimizaiton, second-order cone programming, and etc.

1.1.1 Nearest correlation matrix problems

As an important statistical application of convex quadratic SDP problem, the nearest correlation matrix (NCM) problem arises in marketing and financial economics. For example, in the finance industry, compute stock data is often not available over a given period and currently used techniques for dealing with missing data can result in computed correlation matrices having nonpositive eigenvalues. Again in finance, an investor may wish to explore the effect on a portfolio of assigning correlations between certain assets differently from the historical values, but this again can destroy the semidefiniteness of the matrix. The use of approximate correlation matrices in these applications can render the methodology invalid and lead to negative variances and volatilities being computed, see [33], [91], and [127].

For finding a valid nearest correlation matrix (NCM) to a given symmetric matrix G, Higham [51] considered the following convex QSDP problem

$$
\begin{array}{lll}
(NCM) \qquad & \min & \frac{1}{2}\|X - G\|^2 \\
& \text{s.t.} & \operatorname{diag}(X) = e, \\
& & X \in \mathcal{S}^n_+.
\end{array}
$$

where $e \in \Re^n$ is the vector of all ones. The norm in the (NCM) problem can be Frobenius norm, the H-weighted norm and the W-weighted norm, which will be given in details in the later chapter. In [51], Higham developed an alternating projection method for solving the NCM problems with a weighted Frobenius norm. However, due to the linear convergence of the projection approach used by Higham [51], its convergence can be very slow when solving large scale problems. Anjos et al [4] formulated the nearest correlation matrix problem as an optimization problem with a quadratic objective function and semidefinite programming constraints. Using such a formulation they derived and tested a primal-dual interior-exterior-point algorithm designed especially for robustness and handling the case where $\mathcal{Q}$ is sparse. However the number of variables is $O(n^2)$ and this approach is presented as impractical for large n SDP problems. With three classes of preconditioners for the augmented equation being employed, Toh [122] applied inexact primal-dual path-following methods to solve the weighted NCM problems. Numerical results in [122] show that inexact IPMs are efficient and robust for convex QSDPs with the dimension of matrix variable up to 1600.

Realizing the difficulties in using IPMs, many researchers study other methods to solve the NCM problems and related problems. Malick [72] and Boyd and and Xiao [18] proposed, respectively, a quasi-Newton method and a projected gradient method to the Lagrangian dual problem of the problem (NCM) with the continuously differentiable dual objective function. Since the dimension of the variables in the dual problem is only equal to the number of equality constraints in the primal problem, these two dual based approaches are relatively inexpensive at each iteration and can solve some of these problems with size up to serval thousands. Based on recent developments on the strongly semismoothness of matrix valued functions, Qi and Sun developed a nonsmooth Newton method with quadratic convergence for the NCM problem in [90]. Numerical experiments in [90] showed that the proposed nonsmooth Newton method is highly effective. By using an analytic formula for the metric projection onto the positive semidefinite cone, Qi and Sun also applied an augmented Lagrangian dual based approach to solve the H-norm nearest correlation matrix problems in [92]. The inexact smoothing Newton method designed by Gao and Sun [43] to calibrate least squares semidefinite programming with equality and inequality constraints is not only fast but also robust. More recently, a penalized likelihood approach in [41] was proposed to estimate a positive semidefinite correlation matrix from incomplete data, using information on the uncertainties of the correlation coefficients. As stated in [41], the penalized likelihood approach can effectively estimate the correlation matrices in the predictive sense when the dimension of the matrix is less than 2000.

1.1.2 Euclidean distance matrix problems

An $n \times n$ symmetric matrix $D = (d_{ij})$ with nonnegative elements and zero diagonal is called a pre-distance matrix (or dissimilarity matrix). In addition, if there exist points

$x_1, x_2, \ldots, x_n$ in $\Re^r$ such that

$$d_{ij} = \|x^i - x^j\|^2, \quad i, j = 1, 2, \ldots, n, \tag{1.2}$$

then D is called a Euclidean distance matrix (EDM). The smallest value of r is called the embedding dimension of D. The Euclidean distance matrix completion problem consists in finding the missing elements (squared distances) of a partial Euclidean distance matrix D. It is known that the EDM problem is NP-hard [6, 79, 105]. For solving a wide range of Euclidean distance geometry problems, semidefinite programming (SDP) relaxation techniques can be used in many of which are concerning Euclidean distance, such as data compression, metric-space embedding, covering and packing, chain folding and machine learning problems [25, 53, 67, 136, 130]. Second-order cone programming (SOCP) relaxation was proposed in [35, 125]. In recent years, sensor network localization and molecule structure prediction [13, 34, 80] have received a lot of attention as the important applications of Euclidean distance matrices.

The sensor network localization problem consists of locating the positions of wireless sensors, given only the distances between sensors that are within radio range and the positions of a subset of the sensors (called anchors). Although it is possible to find the position of each sensor in a wireless sensor network with the aid of Global Positioning System (GPS) [131] installed in all sensors, it is not practical to use GPS due to its high power consumption, expensive price and line of sight conditions for a large number of sensors which are densely deployed in a geographical area.

There have been many algorithms published recently that solve the sensor network localization problem involving SDP relaxations and using SDP solvers. The semidefinite programming (SDP) approach to localization was first described by Doherty et al [35]. In this algorithm, geometric constraints between nodes are represented by ignoring the non-convex inequality constraints but keep the convex ones, resulting in a convex second-order cone optimization problem. A drawback of their technique is that all position estimations will lie in the convex hull of the known points. A gradient-descent minimization method, first reported in [66], is based on the SDP relaxation to solve the distance geometry problem.

Unfortunately, in the SDP sensor localization model the number of constraints is in the order of $O(n^2)$, where n is the number of sensors. The difficulty is that each iteration of interior-point algorithm SDP solvers needs to factorize and solve a dense matrix linear system whose dimension is the number of constraints. The existing SDP solvers have very poor scalability since they can only handle SDP problems with the dimension and the number of constraints up to few thousands. To overcome this difficulty, Biswas and Ye [12] provided a distributed or decomposed SDP method for solving Euclidean metric localization problems that arise from ad hoc wireless sensor networks. By only using noisy distance information, the distributed SDP method was extended to the large 3D graphs by Biswas, Toh and Ye [13], using only noisy distance information, and with out any prior knowledge of the positions of any of the vertices.

Another instance of the Euclidean distance geometry problem arises in molecular conformation, specifically, protein structure determination. It is well known that protein structure determination is of great importance for studying the functions and properties of proteins. In order to determine the structure of protein molecules, KurtWüuthrich and his co-researchers started a revolution in this field by introducing nuclear magnetic resonance (NMR) experiments to estimate lower and upper bounds on interatomic distances for proteins in solution [135]. The book by Crippen and Havel [34] provided a comprehensive background to the links between molecule conformation and distance geometry.

Many approaches have been developed for the molecular distance geometry problem, see a survey in [137]. In practice, the EMBED algorithm, developed by Crippen and Havel [34], can be used for dealing with the distance geometry problems arising in NMR molecular modeling and structure determination by performing some bound smoothing techniques. Based on graph reduction, Hendrickesom [49] developed a software package, ABBIE, to determinate the molecular structure with a given set of distances. More and Wu [80] showed in the DGSOL algorithm that global solutions of the molecular distance geometry problems can be determined reliably and efficiently by using global smoothing techniques and a continuation approach for global optimization. The distance geometry program APA, based on an alternating projections algorithm proposed by Glunt et al [94], is designed to determine the three-dimensional structure of proteins using distance geometry. Biswas, Toh and Ye also applied the distributed algorithm in [13] to reconstruct reliably and efficiently the configurations of large 3D protein molecules from a limited number of given pairwise distances corrupted by noise.

1.1.3 SDP relaxations of nonconvex quadratic programming

Numerous combinatorial optimization problems can be cast as the following quadratic programming in ± 1 variables,

$$\max \quad \langle x, Lx \rangle \quad \text{such that} \quad x \in \{-1, 1\}^n, \tag{1.3}$$

where L is a symmetric matrix. Although problem (1.3) is NP-hard, semidefinite relaxation technique can be applied to solve the problem (1.3) for obtaining a solvable problem by relaxing the constraints and perturbing the objective function. Let $X = xx^T$, we get the following relaxation problem:

$$\max \quad \langle L, X \rangle \quad \text{such that} \quad \mathrm{diag}(X) = e, \; X \succeq 0, \tag{1.4}$$

where e is the vector of ones in $\Re^n$. Of course, a binary quadratic integer quadratic programming problem takes the form as follows

$$\max \quad \langle y, Qy \rangle \quad \text{such that} \quad y \in \{0, 1\}^n, \tag{1.5}$$

where Q is a symmetric (non positive semidefinite) matrix of order n. The problem (1.4) is equivalent to (1.3) via $x = 2y - e$, where $y \in \{0, 1\}^n$. In 1991, Lovász and Schrijver [71] introduced the matrix-cut operators for $0 - 1$ integer programs. The problem (1.5) can be used to model some specific combinatorial optimization problems where the special structure of the problem yields SDP models [36, 133, 120]. However, this SDP relaxation enables the solution of the problem (1.4) that are too large for conventional methods to handle efficiently.

Many graph theoretic optimization problems can be stated in this way: to find a maximum cardinality stable set (MSS) of a given graph. The maximum stable set problem is a classical NP-Hard optimization problem which has been studied extensively. Numerous approaches for solving or approximating the MSS problem have been proposed. A survey paper [14] by Bomze et al. gives a broad overview of progress made on the maximum clique problem, or equivalently the MSS problem, in the last four decades. Semidefinite relaxations have also been widely considered for the stable set problem, introduced by Grötschel, Lovász and Schrijver [47]. More work on this problem includes Mannino and Sassano [74], Sewell [107], Pardalos and Xue [86], and Burer, Monteiro, and Zhang [20]. For the subset of large scale SDPs from the collection of random graphs, the relaxation of MSS problems can be solved by the iterative solvers based on the primal-dual interior-point method [121], the boundary-point method [88], and the modified barrier method [60]. Recently, low-rank approximations of such relaxations have recently been used by Burer, Monteiro and Zhang (see [21]) to get fast algorithms for the stable set problem and the maximum cut problem.

Due to the fast implementation of wireless telephone networks, semidefinite relaxations for frequency assignment problems (FAP) has grown quickly over the past years. Even though all variants of FAP are theoretically hard, instances arising in practice might be either small or highly structured such that enumerative techniques, such as the spectral bundle (SB) method [48], the BMZ method [21], and inexact interior-point method [121] are able to handle these instances efficiently. This is typically not the case. Frequency assignment problems are also hard in practice in the sense that practically relevant instances are too large to be solved to optimality with a good quality guarantee.

The quadratic assignment problem (QAP) is a well known problem from the category of the facilities location problems. Since it is NP-complete [104], QAP is one of the most difficult combinatorial optimization problems. Many well known NP-complete problems, such as traveling salesman problem and the graph partitioning problem, can be easily formulated as a special case of QAP. A comprehensive summary on QAP is given in [5, 23, 84]. Since it is unlikely that these relaxations can be solved using direct algorithms, Burer and Vandenbussche [22] proposed a augmented Lagrangian method for optimizing the lift-and-project relaxations of QAP and binary integer programs introduced by Lovász and Schrijver [71]. In [95], Rendl and Sotirov discussed a variant of the bundle method to solve the relaxations of QAP at least approximately with reasonable computational effort.

1.1.4 Convex quadratic SOCP problems

Let $\mathbf{X}$ and $\mathbf{Y}$ be finite dimensional real Hilbert spaces each equipped with a scalar product $\langle \cdot, \cdot \rangle$ and its induced norm $\| \cdot \|$. The second-order cone programming (SOCP) problem with a convex quadratic objective function

$$(QSOCP) \qquad \begin{array}{ll} \min & \frac{1}{2}\langle x, \mathcal{Q}x\rangle + \langle c_0, x\rangle \\ \text{s.t.} & \|\mathcal{A}_i(x) + b_i\| \leq \langle c_i, x\rangle + d_i, \ i = 1, \ldots, p, \end{array}$$

where $\mathcal{Q}$ is a self-adjoint and positive semidefinite linear operator in $\mathbf{X}$, $c_0 \in \mathbf{X}$, $\mathcal{A} : \mathbf{X} \to \mathbf{Y}$ is a linear mapping, $c_i \in \mathbf{X}$, $b_i \in \mathbf{Y}$, and $d_i \in \Re$, for $i = 1, \ldots, p$. Thus the inequality constraint in $(QSOCP)$ can be written as an affine mapping:

$$\|\mathcal{A}_i(x) + b_i\| \leq \langle c_i, x\rangle + d_i \quad \Leftrightarrow \quad \begin{bmatrix} c_i^T \\ \mathcal{A}_i \end{bmatrix} x + \begin{bmatrix} d_i \\ b_i \end{bmatrix} \in \mathbf{K}^{q_i},$$

where $\mathbf{K}^{q_i}$ denotes the second-order cone of dimension q_i defined as

$$\mathcal{K}^{q_i} := \{x = (x_0, \tilde{x}) \in \Re \times \Re^{q_i - 1} \mid \|\tilde{x}\| \leq x_0\}. \tag{1.6}$$

Since the objective is a convex quadratic function and the constraints define a convex set, the problem (QSOCP) is a convex quadratic programming problem. Without the quadratic term in the objective function, the problem $(QSOCP)$ becomes the standard SOCP problem which is a linear optimization problem over a cross product of second-order cones.

A wide range of problems can be formulated as SOCP problems; they include linear programming (LP) problems, convex quadratically constrained quadratic programming problems, filter design problems [30, 126], antenna array weight design [62, 63, 64], and problems arising from limit analysis of collapses of solid bodies [29]. In [69], Lobo et al.

introduced an extensive list of applications problems that can be formulated as SOCPs. For a comprehensive introduction to SOCP, we refer the reader to the paper by Alizadeh and Goldfarb [2].

As a special case of SDP, SOCP problems can be solved as SDP problems in polynomial time by interior point methods. However, it is far more efficient computationally to solve SOCP problems directly because of numerical grounds and computational complexity concerns. There are various solvers available for solving SOCP. SeDuMi is a widely available package [113] that is based on the Nesterov-Todd method and presents a theoretical basis for his computational work in [112]. SDPT3 [128] implements an infeasible path-following algorithm for solving conic optimization problems involving semidefinite, second-order and linear cone constraints. Sparsity in the data is exploited whenever possible. But these IPMs sometimes fail to deliver solutions with satisfactory accuracy. Then Toh et al. [123] improved SDPT3 by using inexact primal-dual path-following algorithms for a special class of linear, SOCP and convex quadratic SDP problems. However, restricted by the fact that interior point algorithms need to store and factorize a large (and often dense) matrix, we try to solve large scale convex quadratic $SOCP$ problems by the augmented Lagrangian method as a special case of convex QSDPs.

1.2 Organization of the thesis

In this thesis, we study a semismooth Newton-CG augmented Lagrangian dual approach to solve large scale linear and convex quadratic programming with linear, SDP and SOC conic constraints. Our principal objective in this thesis is twofold:

- to undertake a comprehensive introduction of a semismooth Newton-CG augmented Lagrangian method for solving large scale linear and convex quadratic programs over symmetric cones; and
- to design efficient practical variant of the theoretical algorithm and perform extensive numerical experiments to show the robustness and efficiency of our proposed method.

In the recent years, taking the benefit of the great development of theories for nonlinear programming, large scale convex quadratic programming over symmetric cones have received more and more attention in combinatorial optimization, optimal control problems, structural analysis and portfolio optimization. Chapter 1 contains an overview on the development and related work in the area of large scale convex quadratic programming. From the view of the theory and application of convex quadratic programs, we present the motivation to develop the method proposed in this thesis.

Under the framework of Euclidean Jordan algebras over symmetric cones in Faraut and Korányi [38], many optimization-related classical results can be generalized to symmetric cones [118, 129]. For nonsmooth analysis of vector valued functions over the Euclidean Jordan algebra associated with symmetric matrices, see [27, 28, 109] and associated with the second order cone, see [26, 40]. Moreover, [116] and [57] study the analyticity, differentiability, and semismoothness of Löwner's operator and spectral functions associated with the space of symmetric matrices. All these development is the theoretical basis of the augmented Lagrangian methods for solving convex quadratic programming over symmetric cones. In Chapter 2, we introduce the concepts and notations of (directional) derivative of semismooth functions. Based on the Euclidean Jordan algebras, we discussed the properties of metric projector over symmetric cones.

The Lagrangian dual method was initiated by Hestenes [50] and Powell [89] for solving equality constrained problems and was extended by Rockafellar [102, 103] to deal with inequality constraints for convex programming problems. Many authors have made contributions of global convergence and local superlinear convergence (see, e.g., Tretyakov [119] and Bertsekas [10, 11]). However, it has long been known that the augmented Lagrangian method for convex problems is a gradient ascent method applied to the corresponding dual problems [100]. This inevitably leads to the impression that the augmented Lagrangian method for solving SDPs may converge slowly for the outer iteration.

In spite of that, Sun, Sun, and Zhang [117] revealed that under the strong second order sufficient condition and constraint nondegeneracy proposed and studied by [114], the augmented Lagrangian method for nonlinear semidefinite programming can be locally regarded as an approximate generalized Newton method applied to solve a semismooth equation. Moreover, Liu and Zhang [68] extended the results in [117] to nonlinear optimization problems over the second-order cone. The good convergence for nonlinear SDPs and SOCPs inspired us to investigate the augmented Lagrangian method for convex quadratic programming over symmetric cones.

Based on the convergence analysis for convex programming [102, 103], under the strong second order sufficient condition and constraint nondegeneracy studied by [114], we design the semismooth Newton-CG augmented Lagrangian method and analyze its convergence for solving convex quadratic programming over symmetric cones in Chapter 3. Since the projection operators over symmetric cones are strongly semismooth [115], in the second part of this chapter we introduce a semismooth Newton-CG method (SNCG) for solving inner problems and analyze its global and local superlinear (quadratic) convergence.

Due to the special structure of linear SDP and its dual, the constraint nondegeneracy condition and the strong second order sufficient condition developed by Chan and Sun [24] provided a theoretical foundation for the analysis of the convergence rate of the augmented Lagrangian method for linear SDPs. In Chapter 4, motivated by [102, 103], [114], and [24], under the uniqueness of Lagrange multipliers, we establish the equivalence among the Lipschitz continuity of the solution mapping at the origin, the second order sufficient condition, and the strict primal-dual constraint qualification. For inner problems, we show that the constraint nondeneracy for the corresponding dual problems is equivalent to the positive definiteness of the generalized Hessian of the objective functions in inner problems. This is important for the success of applying an iterative solver to the generalized Newton equations in solving these inner problems.

The fifth chapter and sixth chapter are on numerical issues of the semismooth Newton-CG augmented Lagrangian algorithm for linear and convex quadratic semidefinite programming respectively. We report numerical results in these two chapters for a variety of large scale linear and convex quadratic SDPs and SOCPs. Numerical experiments show that the semismooth Newton-CG augmented Lagrangian method is a robust and effective iterative procedure for solving large scale linear and convex quadratic symmetric cone programming and related problems.

The final chapter of this thesis, seventh Chapter, states conclusions and lists directions for future research about the semismooth Newton-CG augmented Lagrangian method.

Chapter 2

Preliminaries

To analyze the convex quadratic programming problems over symmetric cones, we use results from semismooth matrix functions and the metric projector onto the symmetric cones. This chapter will cite some definitions and properties that are essential to our discussion.

2.1 Notations and Basics

2.1.1 Notations

Let $\mathbf{X}$ and $\mathbf{Y}$ be two finite-dimensional real Hilbert spaces. Let $\mathcal{O}$ be an open set in $\mathbf{X}$ and $\Phi : \mathcal{O} \subseteq \mathbf{X} \to \mathbf{Y}$ be a locally Lipschitz continuous function on the open set $\mathcal{O}$. Then Φ is almost everywhere F(réchet)-differentiable by Rademacher's theorem. Let $\mathcal{D}_\Phi$ denote the set of F(réchet)-differentiable points of Φ in $\mathcal{O}$. Then, the Bouligand subdifferential of Φ at $x \in \mathcal{O}$, denoted by $\partial_B \Phi(x)$, is

$$\partial_B \Phi(x) := \left\{ \lim_{k \to \infty} \mathcal{J}\Phi(x^k) \mid x^k \in \mathcal{D}_\Phi, x^k \to x \right\},$$

where $\mathcal{J}\Phi(x)$ denotes the F-derivative of Φ at x. Clarke's generalized Jacobian of Φ at x [32] is the convex hull of $\partial_B \Phi(x)$, i.e.,

$$\partial \Phi(x) = \operatorname{conv}\{\partial_B \Phi(x)\}. \tag{2.1}$$

Mifflin first introduced the semismoothness of functionals in [77] and then Qi and Sun [93] extended the concept to vector valued functions. Suppose that $\mathbf{X}$, $\mathbf{X}'$ and $\mathbf{Y}$ are finite-dimensional real Hilbert spaces with each equipped with a scalar product $\langle \cdot, \cdot \rangle$ and its induced norm $\| \cdot \|$.

Definition 2.1. *Let $\Phi : \mathcal{O} \subseteq \mathbf{X} \to \mathbf{Y}$ be a locally Lipschitz continuous function on the open set $\mathcal{O}$. We say that Φ is semismooth at a point $x \in \mathcal{O}$ if*

(i) Φ is directionally differentiable at x; and

(ii) for any $\Delta x \in \mathbf{X}$ and $V \in \partial \Phi(x + \Delta x)$ with $\Delta x \to 0$,

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

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

$$\Phi(x + \Delta x) - \Phi(x) - V(\Delta x) = O(\|\Delta x\|^2). \tag{2.2}$$

The Bouligand-subdifferential of composite functions proved in [114, Lemma 2.1] will be given here.

Lemma 2.1. *Let $F : \mathbf{X} \to \mathbf{Y}$ be a continuously differentiable function on an open neighborhood $\mathcal{O}$ of $\bar{x} \in \mathbf{X}$ and $\Phi : \mathcal{O}_Y \subseteq \mathbf{X}'$ be a locally Lipschitz continuous function on an open set $\mathcal{O}_Y$ containing $\bar{y} := F(\bar{x})$. Suppose that Φ is directionally differentiable at every point in $\mathcal{O}_Y$ and that $\mathcal{J}F(\bar{x})$ is onto. Then it holds that*

$$\partial_B(\Phi \circ F)(\bar{x}) = \partial_B \Phi(\bar{y}) \mathcal{J}F(\bar{x}),$$

where $\circ$ stands for the composite operation.

For a closed set $\mathbf{D} \subseteq \mathbf{X}$, let $\operatorname{dist}(x, \mathbf{D})$ denote the distance from a point $x \in \mathbf{X}$ to $\mathbf{D}$, that is,

$$\operatorname{dist}(x, \mathbf{D}) := \inf_{z \in \mathbf{D}} \|x - z\|.$$

For any closed set $D \subseteq X$, the contingent and inner tangent cones of D at x, denoted by $\mathcal{T}_{\mathbf{D}}(x)$ and $\mathcal{T}^i_{\mathbf{D}}(x)$ respectively, can be written in the form

$$\begin{aligned} \mathcal{T}_{\mathbf{D}}(x) &= \{h \in \mathbf{X} \mid \exists t_n \downarrow 0, \operatorname{dist}(x + t_n h, \mathbf{D}) = o(t_n)\}, \\ \mathcal{T}^i_{\mathbf{D}}(x) &= \{h \in \mathbf{X} \mid \operatorname{dist}(x + th, \mathbf{D}) = o(t), t \geq 0\}. \end{aligned}$$

In general, these two cones can be different and inner tangent cone can be nonconvex. However, for convex closed sets, the contingent and inner tangent cones are equal to each other and convex [16, Proposition 2.55].

Proposition 2.2. *If $\mathbf{D}$ is a convex closed set and $x \in \mathbf{D}$, then*

$$\mathcal{T}_{\mathbf{D}}(x) = \mathcal{T}^i_{\mathbf{D}}(x).$$

It just follows from the above proposition that for convex sets, since the contingent and inner tangent cones are equal, or equivalently that

$$\mathcal{T}_{\mathbf{D}}(x) = \{h \in \mathbf{X} \mid \operatorname{dist}(x + th, \mathbf{D}) = o(t), t \geq 0\}. \tag{2.3}$$

So in this thesis, for convex closed set we will speak of tangent cones rather than contingent or inner tangent cones.

2.1.2 Euclidean Jordan algebra

In this subsection, we briefly describe some concepts, properties, and results from Euclidean Jordan algebras that are needed in this thesis. All these can be found in the articles [39, 106] and the book [38] by Faraut and Korányi.

A Euclidean Jordan algebra is a vector space with the following property:

Definition 2.2. *A Euclidean Jordan algebra is a triple $(\mathbf{V}, \circ, \langle \cdot, \cdot \rangle)$ where $(\mathbf{V}, \langle \cdot, \cdot \rangle)$ is a finite dimensional real inner product space and a bilinear mapping (Jordan product) $(x, y) \to x \circ y$ from $\mathbf{V} \times \mathbf{V}$ into $\mathbf{V}$ is defined with the following properties*

(i) $x \circ y = y \circ x$ for all $x, y \in \mathbf{V}$,

(ii) $x^2 \circ (x \circ y) = x \circ (x^2 \circ y)$ *for all* $x, y \in \mathbf{V}$*, where* $x^2 := x \circ x$*, and*

(iii) $\langle x \circ y, z \rangle = \langle y, x \circ z \rangle$ *for all* $x, y, z \in \mathbf{V}$.

In addition, we assume that there is an element $e \in \mathbf{V}$ (called the *unit* element) such that $x \circ e = x$ for all $x \in \mathbf{V}$.

Henceforth, let $\mathbf{V}$ be a Euclidean Jordan algebra and call $x \circ y$ the Jordan product of x and y. For an element $x \in \mathbf{V}$, let $m(x)$ be the degree of the minimal polynomial of x. We have

$$m(x) = \min\{k > 0 \mid (e, x, x^2, \ldots, x^k) \text{ are linearly dependent}\},$$

and define the rank of $\mathbf{V}$ as $r = \max\{m(x) \mid x \in \mathbf{V}\}$. An element $c \in \mathbf{V}$ is an *idempotent* if $c^2 = c$. Two idempotents c and d are said to be *orthogonal* if $c \circ d = 0$. We say that an idempotent is *primitive* if it is nonzero and cannot be written as a sum of two nonzero idempotents. We say that a finite set $\{c_1, \ldots, c_r\}$ is a *Jordan frame* in $\mathbf{V}$ if each c_j is a primitive idempotent (i.e., $c_i^2 = c_i$) and if

$$c_i \circ c_j = 0 \text{ if } i \neq j \quad \text{and} \quad \sum_{k=1}^{r} c_k = e.$$

Theorem 2.3. *(Spectral theorem, second version [38]). Let* $\mathbf{V}$ *be a Euclidean Jordan algebra with rank* r*. Then for every* $x \in \mathbf{V}$*, there exists a Jordan frame* $\{c_1, \ldots, c_r\}$ *and real numbers* $\lambda_1, \ldots, \lambda_r$ *such that the following spectral decomposition of* x *satisfied,*

$$x = \lambda_1 c_1 + \cdots + \lambda_r c_r. \tag{2.4}$$

The numbers λ_j *are uniquely determined by* x *and called the eigenvalues of* x*. Furthermore, the determinant and trace of* x *are given by*

$$det(x) = \prod_{j=1}^{r} \lambda_j, \quad tr(x) = \sum_{j=1}^{r} \lambda_j.$$

In a Euclidean Jordan algebra $\mathbf{V}$, for an element $x \in \mathbf{V}$, we define the corresponding linear transformation (Lyapunov transformation) $\mathcal{L}(x) : \mathbf{V} \to \mathbf{V}$ by

$$\mathcal{L}(x)y = x \circ y.$$

Note that for each $x \in \mathbf{V}$, $\mathcal{L}(x)$ is a self-adjoint linear transformation with respect to the inner product in the sense that

$$\langle \mathcal{L}(x)y,\, z\rangle = \langle y,\, \mathcal{L}(x)z\rangle, \quad \forall\, y, z \in \mathbf{V}.$$

Let $\|\cdot\|$ be the norm on $\mathbf{V}$ induced by inner product

$$\|x\| := \sqrt{\langle x,\, x\rangle} = \left(\sum_{j=1}^{r} \lambda_j^2(x)\right)^{1/2}, \quad x \in \mathbf{V}.$$

And we say that x and y *operator commute* if $\mathcal{L}(x)$ and $\mathcal{L}(y)$ commute, i.e., $\mathcal{L}(x)\mathcal{L}(y) = \mathcal{L}(y)\mathcal{L}(x)$. It is well known that x and y operator commute if and only if x and y have their spectral decompositions with respect to a common Jordan frame ([106, Theorem 27]). For examples, if $\mathbf{V} = \mathcal{S}^n$, matrices X and Y operator commute if and only if $XY = YX$; if $\mathbf{V} = \mathcal{K}^q$, vectors x and y operator commute if and only if either $\tilde{y}$ is a multiple of $\tilde{x}$ or $\tilde{x}$ is a multiple of $\tilde{y}$.

A symmetric cone [38] is the set of all squares

$$\mathbf{K} = \{x^2 \mid x \in \mathbf{V}\}. \tag{2.5}$$

When $\mathbf{V} = \mathcal{S}^n, \Re^q$ or $\Re^n$, we have the following results:

- Case $\mathbf{V} = \Re^n$. Consider $\Re^n$ with the (usual) inner product and Jordan product defined respectively by

$$\langle\, x, y\,\rangle = \sum_{i=1}^{n} x_i y_i \quad \text{and} \quad x \circ y = x * y,$$

  where x_i denotes the ith component of x, and $x * y = (x_i y_i)$ denotes the componentwise product of vectors x and y. Then $\Re^n$ is a Euclidean Jordan algebra with $\Re^n_+$ as its cone of squares.

- Case $\mathbf{V} = \mathcal{S}^n$. Let $\mathcal{S}^n$ be the set of all $n \times n$ real symmetric matrices with the inner and Jordan product given by

$$\langle\, X, Y\,\rangle := \text{trace}(XY) \quad \text{and} \quad X \circ Y := \frac{1}{2}(XY + YX).$$

In this setting, the cone of squares $\mathcal{S}^n_+$ is the set of all positive semidefinite matrices in $\mathcal{S}^n$. The identity matrix is the unit element. The set $\{E_1, E_2, \ldots, E_n\}$ is a Jordan frame in $\mathcal{S}^n$ where E_i is the diagonal matrix with 1 in the (i,i)-slot and zeros elsewhere. Note that the rank of $\mathcal{S}^n$ is n. Given any $X \in \mathcal{S}^n$, there exists an orthogonal matrix P with columns of eigenvectors $p_1, p_2, \ldots, p_n$ and a real diagonal matrix $D = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ such that $X = PDP^T$. Clearly,

$$X = \lambda_1 p_1 p_1^T + \cdots + \lambda_n p_n p_n^T$$

is the spectral decomposition of X.

- Case $\mathbf{V} = \Re^q$. Consider $\Re^q (q > 1)$ where any element x is written as $x = (x_0; \tilde{x})$ with $x_0 \in \Re$ and $\hat{x} \in \Re^{q-1}$. The inner product in $\Re^q$ is the usual inner product. The Jordan product $x \circ y$ in R^q is defined by

$$x \circ y = \begin{pmatrix} x^T y \\ y_0 \tilde{x} + x_0 \tilde{y} \end{pmatrix}$$

In this Euclidean Jordan algebra $(\Re^q, \circ, \langle \cdot, \cdot \rangle)$, the cone of squares, denoted by $\mathcal{K}^q$ is called the Lorentz cone (or the second-order cone). It is given by

$$\mathcal{K}^q = \{x : \|\tilde{x}\| \le x_0\}.$$

The unit element in $\mathcal{K}^q$ is $e = (1; 0)$. We note the spectral decomposition of any $x \in \Re^q$:

$$x = \lambda_1 u_1 + \lambda_2 u_2,$$

where for $i = 1, 2$,

$$\lambda_i = x_0 + (-1)^i \|\tilde{x}\| \quad \text{and} \quad u_i = \frac{1}{2}(1; (-1)^i w),$$

where $w = \tilde{x}/\|\tilde{x}\|$ if $\tilde{x} \neq 0$; otherwise w can be any vector in $\Re^{q-1}$ with $\|w\| = 1$.

Let c be an idempotent element (if $c^2 = c$) in a Jordan algebra $\mathbf{V}$ satisfying $2\mathcal{L}^3(c) - 3\mathcal{L}^2(c) + \mathcal{L}(c) = 0$. Then $\mathcal{L}(c)$ has three distinct eigenvalues $1, \frac{1}{2}$, and 0 with the corresponding eigenspace $\mathbf{V}(c, 1)$, $\mathbf{V}(c, \frac{1}{2})$, and $\mathbf{V}(c, 0)$, where

$$\mathbf{V}(c, i) := \{x \in \mathbf{V} \mid \mathcal{L}(c)x = ix, \ i = 1, \frac{1}{2}, 0\}.$$

Then $\mathbf{V}$ is the direct sum of those eigenspaces

$$\mathbf{V} = \mathbf{V}(c,1) \oplus \mathbf{V}(c,\frac{1}{2}) \oplus \mathbf{V}(c,0) \tag{2.6}$$

is called the *Peirce decomposition* of $\mathbf{V}$ with respect to the idempotent c.

A Euclidean Jordan algebra is said to be *simple* if it is not the direct sum of two Euclidean Jordan algebras. In the sequel we assume that $\mathbf{V}$ is a simple Euclidean Jordan algebra of rank r and $\dim(\mathbf{V}) = n$. Then, we know that from the spectral decomposition theorem that an idempotent c is primitive if and only if $\dim(\mathbf{V}(c,1)) = 1$ [38, Page 65].

Let $\{c_1, c_2, \ldots, c_r\}$ be a Jordan frame in a Euclidean Jordan algebra $\mathbf{V}$. Since the operators $\mathcal{L}(c_i)$ commute [38, Lemma IV.1.3], for $i, j \in \{1, 2, \ldots, r\}$, we consider the eigenspaces

$$\begin{aligned} \mathbf{V}_{ii} &:= \mathbf{V}(c_i;1) = \Re c_i \quad \text{and} \\ \mathbf{V}_{ij} &:= \mathbf{V}(c_i,\tfrac{1}{2}) \cap \mathbf{V}(c_j,\tfrac{1}{2}) \quad \text{when } i \neq j. \end{aligned} \tag{2.7}$$

Then we have the following important results from [38, Theorem IV.2.1, Lemma IV.2.2].

Theorem 2.4. *(i) The space $\mathbf{V}$ decomposes in the following direct sum:*

$$\mathbf{V} = \bigoplus_{i \leq j} \mathbf{V}_{ij}.$$

(ii) If we denote by $\mathcal{P}_{ij}$ the orthogonal projection onto $\mathbf{V}_{ij}$, then

$$\mathcal{P}_{ii} = \mathcal{P}(c_i) \quad \textit{and} \quad \mathcal{P}_{ij} = 4\mathcal{L}(c_i)\mathcal{L}(c_j), \tag{2.8}$$

where $\mathcal{P}(c)$ is the projection in the Peirce decomposition onto $\mathbf{V}(c,1)$, given by $\mathcal{P}(c) = \mathcal{L}(c)(2\mathcal{L}(c) - 1)$.

Let d denote the dimension of $\mathbf{V}_{ij}$. Since $\dim(\mathbf{V}_{ij}) = \dim(\mathbf{V}_{kl})$ ([38, Corollary IV.2.6]), then

$$n = r + \frac{d}{2} r(r-1). \tag{2.9}$$

2.2 Metric projectors

Let $\mathbf{X}$ be a finite dimensional real Hilbert space each equipped with a scalar product $\langle \cdot, \cdot \rangle$ and its induced norm $\| \cdot \|$ and $\mathbf{K}$ be a closed convex set in $\mathbf{X}$. Let $\Pi_{\mathbf{K}} : \mathbf{X} \to \mathbf{X}$ denote the metric projection over $\mathbf{K}$, i.e., for any $x \in \mathbf{X}$, $\Pi_{\mathbf{K}}(x)$ is the unique optimal solution to the convex programming problem:

$$\begin{array}{ll} \min & \frac{1}{2}\langle z - x, z - x \rangle \\ \text{s.t.} & z \in \mathbf{K}. \end{array} \tag{2.10}$$

For any $x \in \mathbf{X}$, let $x_+ := \Pi_{\mathbf{K}}(x)$ and $x_- := \Pi_{\mathbf{K}^*}(-x)$, where $\mathbf{K}^*$ is the dual cone of $\mathbf{K}$, i.e.,

$$\mathbf{K}^* := \{v \in \mathbf{X} \mid \langle v, z \rangle \geq 0 \quad \forall\, z \in \mathbf{K}\}.$$

We then have the Moreau decomposition [78],

$$x = x_+ - x_- \quad \text{and} \quad \langle x_+, x_- \rangle = 0 \quad \forall\, x \in \mathbf{X},$$

It is well known [138] that the metric projector $\Pi_{\mathbf{K}}(\cdot)$ is Lipschitz continuous with the Lipschitz constant 1, that is, for any two vectors $y, z \in \mathbf{K}$,

$$\|\Pi_{\mathbf{K}}(y) - \Pi_{\mathbf{K}}(z)\| \leq \|y - z\|.$$

Hence, $\Pi_{\mathbf{K}}(\cdot)$ is F-differentiable almost everywhere in $\mathbf{X}$ and for any $x \in \mathbf{X}$, $\partial \Pi_{\mathbf{K}}(x)$ is well defined. The following lemma is the general properties of $\partial \Pi_{\mathbf{K}}(\cdot)$ from [76, Proposition 1].

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

(i) *V is self-adjoint.*

(ii) $\langle d, Vd \rangle \geq 0, \ \forall\, d \in \mathbf{X}$.

(iii) $\langle Vd, d - Vd \rangle \geq 0, \ \forall\, d \in \mathbf{X}$.

In this thesis, we assume that $\mathbf{K}$ is a closed convex cone with $\mathbf{K}^* = \mathbf{K}$. For the study of the later chapters, $\mathbf{K}$ contains $\Re^n_+$, $\mathcal{S}^n_+$ and K^q, which is a symmetric cone satisfying (2.5).

In the following discussion, we represent the properties of the metric projectors over symmetric cones defined in a Euclidean Jordan algebra results from Euclidean Jordan algebras given by [116, 118, 129].

Under a simple Euclidean Jordan algebra $\mathbf{V}$ with rank r, we can define a *Löwner's operator* [58] associated with $\mathbf{V}$ by

$$\phi_{\mathbf{V}}(x) := \sum_{i=1}^{r} \phi(\lambda_i(x))c_i, \tag{2.11}$$

where $\phi : \Re \to \Re$ is a scalar valued function and $x \in \mathbf{V}$ has the spectral decomposition as in (2.4). In particular, let $\phi(t) = t_+ := \max(0, t)$, $t \in \Re$, Löwner's operator becomes the metric projection operator x over the symmetric cone $\mathbf{K}$, i.e.,

$$x_+ = \sum_{i=1}^{r} (\lambda_i(x))_+ c_i.$$

Let $\tau = (\tau_1, \tau_2, \ldots, \tau_r) \in \Re^r$. Suppose that ϕ is differentiable at τ_i, for $i = 1, 2, \ldots, r$. Define the *first divided difference* of ϕ at τ, denoted by $\phi^{[1]}(\tau)$, as the $r \times r$ symmetric matrix with its ijth entry $(\phi^{[1]}(\tau))_{ij}$ given by $[\tau_i, \tau_j]$, where

$$[\tau_i, \tau_j] := \begin{cases} \frac{\phi(\tau_i) - \phi(\tau_j)}{\tau_i - \tau_j} & \text{if } \tau_i \neq \tau_j \\ \phi'(\tau_i) & \text{if } \tau_i = \tau_j \end{cases}, \quad i, j = 1, 2, \ldots, r.$$

From Koarányi [58, Page 74] and [116, Theorem 3.2], the following proposition shows that $\phi_{\mathbf{V}}$ is (continuously) differentiable at x if and only if $\phi(\cdot)$ is (continuously) differentiable at $\lambda_i(x)$, for $i = 1, 2, \ldots, r$.

Theorem 2.6. *Suppose that $x = \sum_{i=1}^{r} \lambda_i(x) c_i$ defined by (2.4). The $\phi_{\mathbf{V}}$ is (continuously) differentiable at x if and only if for each $i = 1, 2, \ldots, r$, $\phi(\cdot)$ is (continuously) differentiable at $\lambda_i(x)$ and for any $h \in \mathbf{V}$, the derivative of $\phi_{\mathbf{V}}(x)$ is given by*

$$(\phi'_{\mathbf{V}})(x)h = \sum_{i=1}^{r} (\phi^{[1]}(\tau))_{ii} \langle c_i, h\rangle c_i + \sum_{1 \le i < l \le r} 4(\phi^{[1]}(\tau))_{il} c_i \circ (c_i \circ h). \tag{2.12}$$

When $\phi(t) = t_+$, $\phi(\cdot)$ is differentiable almost everywhere except $t = 0$. Therefore, we will next introduce the Bouligand-subdifferential of $\phi_{\mathbf{V}}(x)$ when x has zero eigenvalues, which is based on the report [118] on which the thesis [129] is based.

Suppose there exists two integers s_1 and s_2 such that the eigenvalues of x are arranged in the decreasing order

$$\lambda_1(x) \geq \cdots \geq \lambda_{s_1}(x) > 0 = \lambda_{s_1+1}(x) = \cdots = \lambda_{s_2}(x) > \lambda_{s_2+1}(x) \geq \cdots \geq \lambda_r(x). \quad (2.13)$$

Let $0 < \tau \leq \min\{\lambda_{s_1}(x)/2, -\lambda_{s_2+1}(x)\}$. Define a function $\hat{\phi}^\tau : \Re \to \Re_+$ as

$$\hat{\phi}^\tau(t) = \begin{cases} t & \text{if } t > \tau \\ 2t - \tau & \text{if } t \in [\tau/2, \tau] \\ 0 & \text{if } t < \tau/2. \end{cases}$$

For $t \in \Re$, let $\tilde{\phi}^\tau(t) := \phi(t) - \hat{\phi}^\tau(t)$. Define

$$\begin{aligned} \Re^{|\beta|}_{>} &:= \{z \in \Re^{|\beta|} \mid z_1 \geq z_2 \geq \cdots \geq z_{|\beta|}, z_i \neq 0 \; \forall\, i\}, \\ \mathcal{U}_{|\beta|} &:= \{\Omega \mid \Omega = \lim_{k\to\infty} \phi^{[1]}(z^k), z^k \to 0, z^k \in R^{|\beta|}_{>}\}. \end{aligned}$$

Proposition 2.7. *Suppose that $x \in \mathbf{V}$ has eigenvalues satisfying (2.13). Then $W \in \partial_B(\tilde{\phi}^\tau_{\mathbf{V}})(x)$ if and only if there exist a $\Omega \in \mathcal{U}_{|\beta|}$ and a Jordan frame $\{\tilde{c}_{s_1+1}, \ldots, \tilde{c}_{s_2+1}\}$ satisfying $\tilde{c}_{s_1+1} + \cdots + \tilde{c}_{s_2} = c_{s_1+1} + \cdots + \cdots + c_{s_2}$, such that*

$$W(h) = \sum_{i=s_1+1}^{s_2} \Omega_{ii} \langle \tilde{c}_i, h\rangle \tilde{c}_i + \sum_{s_1+1 \leq i < l \leq s_2} 4\Omega_{(i-s_1)(l-s_1)} \tilde{c}_i \circ (\tilde{c}_i \circ h), \quad \forall\, h \in \mathbf{V}.$$

Furthermore, if $W \in \partial(\tilde{\phi}^\tau_{\mathbf{V}})(x)$, we have that $W - W^2$ is positive semidefinite.

In particular, for any $h \in \mathbf{V}$, define

$$W^I(h) = \sum_{i=s_1+1}^{s_2} \langle \tilde{c}_i, h\rangle \tilde{c}_i + \sum_{s_1+1 \leq i < l \leq s_2} 4\tilde{c}_i \circ (\tilde{c}_i \circ h).$$

Then, from Proposition 2.7, we know that

$$W^I \in \partial_B(\tilde{\phi}^\tau_{\mathbf{V}})(x)$$

Under a simple Euclidean Jordan algebra $\mathbf{V}$ with rank r, the Bouligand-subdifferential of the metric projection $\Pi_{\mathbf{K}}(\cdot)$ is given by

$$\partial_B \Pi_{\mathbf{K}}(x) = (\hat{\phi}^\tau_{\mathbf{V}})'(x) + \partial_B(\tilde{\phi}^\tau_{\mathbf{V}})(x). \tag{2.14}$$

In particular, there are two interesting elements V^0 and V^I in $\partial_B \Pi_{\mathbf{K}}(x)$, given by

$$V^0 = (\hat{\phi}^\tau_{\mathbf{V}})'(x) \quad \text{and} \quad V^I = (\hat{\phi}^\tau_{\mathbf{V}})'(x) + W^I. \tag{2.15}$$

Next based on the matrix representations of elements in the symmetric cones, we introduce some definitions about x_+ which will be used later.

For $1 \le i < l \le r$, there exist d mutually orthonormal vectors $\{v_{il}^{(1)}(x), v_{il}^{(2)}(x), \ldots, v_{il}^{(d)}(x)\}$ in $\mathbf{V}$ such that

$$\mathcal{P}_{il}(x) = \sum_{j=1}^{d} \langle v_{il}^{(j)}(x), \cdot \rangle v_{il}^{(j)}(x),$$

where $d = \dim(\mathbf{V}_{il})$ satisfies (2.9). Then

$$\left\{ c_1(x), c_2(x), \ldots, c_r(x), v_{il}^{(1)}(x), v_{il}^{(2)}(x), \ldots, v_{il}^{(d)}(x), 1 \le i < l \le r \right\}$$

is an orthonormal basis of $\mathbf{V}$. Define three index sets

$$\alpha := \{1, \ldots, s_1\}, \quad \beta := \{s_1 + 1, \ldots, s_2\}, \quad \gamma := \{s_2 + 1, \ldots, r\}. \tag{2.16}$$

For the simplicity of the notation, define $h_{ii} := \mathcal{P}_{ii} h$ and $h_{il} := \mathcal{P}_{il} h$, for $1 \le i \le l \le r$. Then, corresponding to three index sets, we can denote that

$$\begin{aligned}
h_{\alpha\alpha} &= \sum_{i=1}^{s_1} h_{ii} + \sum_{1 \le i < l \le s_1} h_{jl}, \quad h_{\alpha\beta} = \sum_{i=1}^{s_1} \sum_{l=s_1+1}^{s_2} h_{il}, \\
h_{\alpha\gamma} &= \sum_{i=1}^{s_1} \sum_{l=s_2+1}^{r} h_{il}, \quad h_{\beta\beta} = \sum_{i=s_1+1}^{s_2} h_{ii} + \sum_{s_1+1 \le i < l \le s_2} h_{jl}, \\
h_{\beta\gamma} &= \sum_{i=s_1+1}^{s_2} \sum_{l=s_2+1}^{r} h_{il}, h_{\gamma\gamma} = \sum_{i=s_2+1}^{r} h_{ii} + \sum_{s_2+1 \le i < l \le r} h_{jl}.
\end{aligned} \tag{2.17}$$

And define

$$U_\alpha := [c_1, c_2, \ldots, v_{il}^{(j)}, (j = 1, \ldots, d, i = 1, \ldots, s_1, l = i+1, \ldots, r)],$$

$$U_\beta := [c_{s_1+1}, \ldots, c_{s_2}, v_{il}^{(j)}, (j = 1, \ldots, d, i = s_1 + 1, \ldots, s_2, l = i+1, \ldots, r)]$$

$$U_\gamma := [c_{s_2+1}, \ldots, c_r, v_{il}^{(j)}, (j = 1, \ldots, d, i = s_1 + 1, \ldots, s_2, l = i+1, \ldots, r)]$$

Let $U = [U_\alpha, U_\beta, U_\gamma]$ and $\{u_1, u_2, \ldots, u_n\}$ be the columns of U. For any $z \in \mathbf{V}$, let $L(z)$, $P(z)$, $P_{il}(z)$ be the corresponding (matrix) representations of $\mathcal{L}(z)$, $\mathcal{P}(z)$ and $\mathcal{P}_{il}(z)$ with respect to the basis $\{u_1, u_2, \ldots, u_n\}$. Let $\tilde{e}$ denote the coefficients of e with respect to the basis $\{u_1, u_2, \ldots, u_n\}$, i.e.,

$$e = \sum_{i=1}^{n} \langle e, u_i \rangle u_i = U\tilde{e}.$$

And $\widetilde{U}_\alpha, \widetilde{U}_\beta$, and $\widetilde{U}_\gamma$ denote the coefficients of e with respect to the basis $\{u_1, u_2, \ldots, u_n\}$. Then, the projector $x_+ \in \mathbf{K}$ can be rewritten as

$$x_+ = U(L(x))_+\tilde{e}$$

Definition 2.3. *For any $x \in \mathbf{V}$, suppose that the eigenvalues of x satisfy (2.13), the tangent cone of $\mathbf{K}$ at x_+ is given by*

$$\mathcal{T}_{\mathbf{K}}(x_+) = \{h \in \mathbf{V} \mid h_{\beta\beta} + h_{\beta\gamma} + h_{\gamma\gamma} \succeq 0\}. \tag{2.18}$$

The lineality space of $\mathcal{T}_{\mathbf{K}}(x_+)$, i.e., the largest linear space in $\mathcal{T}_{\mathbf{K}}(x_+)$, denoted by lin($\mathcal{T}_{\mathbf{K}}(x_+)$), takes the following form:

$$\mathrm{lin}(\mathcal{T}_{\mathbf{K}}(x_+)) = \{h \in \mathbf{V} \mid h_{\beta\beta} = 0, h_{\beta\gamma} = 0, h_{\gamma\gamma} = 0\} \tag{2.19}$$

The critical cone of $\mathbf{K}$ at x_+ is defined as

$$\mathcal{C}(x_+) := \mathcal{T}_{\mathbf{K}}(x_+) \cap (x_+ - x)^\perp = \{h \in \mathbf{V} \mid h_{\beta\beta} \succeq 0, h_{\beta\gamma} = 0, h_{\gamma\gamma} = 0\}. \tag{2.20}$$

The affine hull of $\mathcal{C}(x_+)$, denoted by $\mathrm{aff}(\mathcal{C}(x_+))$, can thus be written as

$$\mathrm{aff}(\mathcal{C}(x_+)) = \{h \in \mathbf{V} \mid h_{\beta\gamma} = 0, h_{\gamma\gamma} = 0\}. \tag{2.21}$$

Motivated by Shapiro [108] and Bonnans and Shapiro [16], the authors in [118] introduce a linear-quadratic function $\Upsilon_v : \mathbf{V} \times \mathbf{V} \to \Re$ in the next definition, which will help us to define the strong second order sufficient condition for the proposed problems.

Definition 2.4. *For any $v \in \mathbf{V}$, a linear-quadratic function $\Upsilon_v : \mathbf{V} \times \mathbf{V} \to \Re$, which is linear in the first variable and quadratic in the second variable, is defined by*

$$\Upsilon_v(s, h) := 2\, \langle s \cdot h, v^\dagger \cdot h \rangle, \quad (s, h) \in \mathbf{V} \times \mathbf{V}, \tag{2.22}$$

where $v^\dagger$ is the Moore-Penrose pseudo-inverse of v.

The following property, given by [118, 129], of the above linear-quadratic function Υ_v defined in (3.27) about the metric projector $x_+ = \Pi_{\mathbf{K}}(\cdot)$ over $\mathbf{K}$.

Proposition 2.8. *If* $h \in \mathrm{aff}(x_+)$, *since* $x_+ = \sum_{j=1}^{s_1} \lambda_j c_j$, *then*

$$\Upsilon_{x_+}(x_+ - x, h) = \sum_{j=1}^{s_1} \sum_{l=s_2+1}^{r} \frac{-\lambda_l}{\lambda_j} \|h_{jl}\|^2. \tag{2.23}$$

Chapter 3

Convex quadratic programming over symmetric cones

3.1 Convex quadratic symmetric cone programming

Let $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ be three finite dimensional real Hilbert spaces each equipped with a scalar product $\langle \cdot, \cdot \rangle$ and its induced norm $\|\cdot\|$. We consider the following convex quadratic symmetric cone programming (QSCP),

$$
(P) \qquad \begin{array}{ll} \min\limits_{x \in \mathbf{X}} & f_0(x) := \frac{1}{2}\langle x, \mathcal{Q}(x) \rangle + \langle c, x \rangle \\ \text{s.t.} & \mathcal{A}(x) = b, \\ & \mathcal{B}(x) \succeq d, \end{array}
$$

where $\mathcal{Q} : \mathbf{X} \to \mathbf{X}$ is a self-adjoint positive semidefinite linear operator in $\mathbf{X}$, $\mathcal{A} : \mathbf{X} \to \mathbf{Y}$ and $\mathcal{B} : \mathbf{X} \to \mathbf{Z}$ are linear mappings, $b \in \mathbf{Y}$, $d \in \mathbf{Z}$, $c \in \mathbf{X}$ and $\mathbf{K}$ is a symmetric cone in $\mathbf{Z}$, defined in (2.5). The symbol "$\succeq$" denotes that $\mathcal{B}(x) - d \in \mathbf{K}$. In this thesis, we consider the symmetric cone consisting of the linear cone $\Re^l_+$, the second order cone $\mathcal{K}^q$ or the positive semidefinite cone $\mathcal{S}^n_+$.

The Lagrangian dual problem associated with (P) is

$$(D) \qquad \begin{array}{ll} \max & g_0(y,z) := \inf_{x\in\mathbf{X}} L_0(x,y,z) \\ \text{s.t.} & y \in \mathbf{Y},\ z \succeq 0. \end{array}$$

where the Lagrangian function $L_0 : \mathbf{X}\times\mathbf{Y}\times\mathbf{Z} \to \Re$ of (P) is defined as

$$L_0(x,y,z) := f_0(x) - \langle y, \mathcal{A}(x) - b\rangle - \langle\, z, \mathcal{B}(x) - d\rangle.$$

Given a penalty parameter $\sigma > 0$, the *augmented Lagrangian* function for the convex quadratic programming problem (P) is defined as

$$\begin{aligned} L_\sigma(x,y,z) \;=\; & \frac{1}{2}\langle\, x, \mathcal{Q}(x)\rangle + \langle c, x\rangle - \langle\, y, \mathcal{A}(x) - b\rangle + \frac{\sigma}{2}\|\mathcal{A}(x) - b\|^2 \\ & + \frac{1}{2\sigma}\Big[\|\Pi_{\mathbf{K}}[z - \sigma(\mathcal{B}(x) - d)]\|^2 - \|z\|^2\Big], \end{aligned} \tag{3.1}$$

where $(x,y,z) \in \mathbf{X}\times\mathbf{Y}\times\mathbf{Z}$ and for any $z \in \mathbf{Z}$, $\Pi_{\mathbf{K}}(z)$ is the metric projection onto $\mathbf{K}$ at z. For any $\sigma \geq 0$, $L_\sigma(x,y,z)$ is convex in $x \in \mathbf{X}$ and concave in $(y,z) \in \mathbf{Y}\times\mathbf{Z}$, and

$$\lim_{\sigma\downarrow 0} L_\sigma(x,y,z) = \begin{cases} L_0(x,y,z) & \text{if } z \succeq 0, \\ -\infty & \text{otherwise.} \end{cases}$$

Note that $\|\Pi_{\mathbf{K}}(\cdot)\|^2$ is continuously differentiable [138], then the augmented Lagrangian function defined in (3.1) is continuously differentiable.

For a given nondecreasing sequence of numbers σ_k,

$$0 < \sigma_k \ \uparrow\ \sigma_\infty \leq +\infty \tag{3.2}$$

and an initial multiplier $(y^0, z^0) \in \mathbf{Y}\times\mathbf{Z}$, the augmented Lagrangian method for solving problem (P) and its dual (D) generates sequences $x^k \subset \mathbf{X}$, $y^k \subset \mathbf{Y}$, and $z^k \subset \mathbf{Z}$ as follows

$$\begin{cases} x^{k+1} \approx \arg\min_{x\in\mathbf{X}} L_{\sigma_k}(x, y^k, z^k) \\ y^{k+1} = y^k - \sigma_k(\mathcal{A}(x^{k+1}) - b) \\ z^{k+1} = \Pi_{\mathbf{K}}[z^k - \sigma_k(\mathcal{B}(x^{k+1}) - d)] \\ \sigma_{k+1} = \rho\,\sigma_k \text{ or } \sigma_{k+1} = \sigma_k. \end{cases} \tag{3.3}$$

From the augmented Lagrangian algorithm (3.3), we need to find an optimal solution to the inner problem $\min_{x\in\mathbf{X}} L_{\sigma_k}(x, y^k, z^k)$. Because of the computational cost and time, here, we only solve the inner minimization problem in (3.3) inexactly. Under some stopping criteria shown in the later section, the algorithm still converges to a dual optimal solution.

From [102, 103], we know that the augmented Lagrangian method can be expressed in terms of the method of multipliers for (D). For the sake of subsequent developments, we introduce related concepts to this.

Let $l(x, y, z) : \mathbf{X} \times \mathbf{Y} \times \mathbf{Z} \to \Re$ be the ordinary Lagrangian function for (P) in the extended form:

$$l(x, y, z) = \begin{cases} L_0(x, y, z) & \text{if} \quad x \in \mathbf{X} \text{ and } (y, z) \in \mathbf{Y} \times \mathbf{K}, \\ -\infty & \text{if} \quad x \in \mathbf{X} \text{ and } (y, z) \notin \mathbf{Y} \times \mathbf{K}, \end{cases} \tag{3.4}$$

The essential objective function in (P) is

$$f(x) = \inf_{(y,z)\in\mathbf{Y}\times\mathbf{Z}} l(x, y, z) = \begin{cases} f_0(x) & \text{if } x \in F(P), \\ -\infty & \text{otherwise}, \end{cases} \tag{3.5}$$

where $F(P) := \{x \in \mathbf{X} \mid \mathcal{A}(x) = b,\, \mathcal{B}(x) \succeq d\}$ denotes the feasible set of problem (P), while the essential objective function in (D) is defined as

$$g(y, z) = \inf_{x\in\mathbf{X}} l(x, y, z) = \begin{cases} g_0(y, z) & \text{if } y \in \mathbf{Y}, z \in \mathbf{K}, \\ -\infty & \text{if } y \in \mathbf{Y}, z \notin \mathbf{K}. \end{cases} \tag{3.6}$$

Assume that $F(P) \neq \emptyset$ and $g(y, z) \not\equiv -\infty$. As in Rockafellar [102], we can define the following maximal monotone operators

$$T_l(x, y, z) = \{(v, u_1, u_2) \in \mathbf{X} \times \mathbf{Y} \times \mathbf{Z} \mid (v, -u_1, -u_2) \in \partial l(x, y, z)\},$$

for $(x, y, z) \in \mathbf{X} \times \mathbf{Y} \times \mathbf{Z}$, and

$$\begin{aligned} T_f(x) &= \{v \in \mathbf{X} \mid v \in \partial f(x)\}, \quad x \in \mathbf{X}, \\ T_g(y, z) &= \{(u_1, u_2) \in \mathbf{Y} \times \mathbf{Z} \mid (-u_1, -u_2) \in \partial g(y, z)\}, \quad (y, z) \in \mathbf{Y} \times \mathbf{Z} \end{aligned}$$

For each $(v, u_1, u_2) \in \mathbf{X} \times \mathbf{Y} \times \mathbf{Z}$, consider the following parameterized problem:

$$(P(v, u_1, u_2)) \qquad \begin{array}{ll} \min & f_0(x) + \langle v,\, x\rangle \\ \text{s.t.} & \mathcal{A}(x) - u_1 = b, \\ & \mathcal{B}(x) - u_2 \succeq d, \end{array}$$

By using the fact that f is convex and $F(P)$ is nonempty, we know from Rockafellar [99, Theorem 23.5] that for each $v \in \mathbf{X}$,

$$\begin{aligned} T_f^{-1}(v) &= \arg\min_{x\in\mathbf{X}}\{f(x) + \langle v,\, x\rangle\} \\ &= \text{set of all optimal solutions to } (P(v,0,0)). \end{aligned} \tag{3.7}$$

By the same token, since $g \not\equiv -\infty$, we have that

$$\begin{aligned} T_g^{-1}(u_1, u_2) &= \arg\max_{(y,z)\in\mathbf{Y}\times\mathbf{Z}}\{g(y,z) + \langle\, u_1, y\,\rangle + \langle\, u_2, z\,\rangle\} \\ &= \text{set of all optimal solutions to } (D(0, u_1, u_2)), \end{aligned} \tag{3.8}$$

where $(D(v, u_1, u_2))$ is the ordinary dual problem of $P(v, u_1, u_2)$ and $(D(0, u_1, u_2))$ takes the form as follows

$$(D(0, u_1, u_2)) \qquad \begin{array}{ll} \min & g_0(y,z) + \langle\, u_1, y\,\rangle + \langle\, u_2, z\,\rangle \\ \text{s.t.} & y \in \mathbf{Y}, z \succeq 0. \end{array}$$

As an application of [101, Theorems 17' & 18'], $\min(P(0, u_1, u_2)) = \sup(D(0, u_1, u_2))$ if the level set of (P) is nonempty and bounded, i.e.

Assumption 3.1. *For the problem (P), there exists an $\alpha \in \Re$ such that the level sets $\{x \in \mathbf{X} \mid f_0(x) \le \alpha,\, x \in F(P)\}$ is nonempty and bounded.*

Then

$$T_g^{-1}(u_1, u_2) = \partial p(u_1, u_2), \quad \text{where } p(u_1, u_2) = \inf(P(0, u_1, u_2)).$$

Finally,

$$\begin{aligned} T_l^{-1}(v, u_1, u_2) &= \arg\min_{x\in\mathbf{X}}\max_{(y,z)\in\mathbf{Y}\times\mathbf{Z}}\{l(x,y,z) - \langle v,\, x\rangle + \langle\, u_1, y\,\rangle + \langle\, u_2, z\,\rangle\} \\ &= \text{set of all } (x,y,z) \text{ satisfying the KKT conditions (3.12)} \\ &\quad\ \text{for } (P(v, u_1, u_2)). \end{aligned} \tag{3.9}$$

Definition 3.1. *[103] For a maximal monotone operator T from a finite dimensional linear vector space $\mathcal{X}$ to itself, we say that its inverse T^{-1} is Lipschitz continuous at the origin (with modulus $a \geq 0$) if there is a unique solution $\bar{z}$ to $z = T^{-1}(0)$, and for some $\tau > 0$ we have*

$$\|z - \bar{z}\| \leq a\|w\| \quad \textit{whenever} \quad z \in T^{-1}(w) \quad \textit{and} \quad \|w\| \leq \tau. \tag{3.10}$$

We have the direction condition for the Lipschitz continuity of T_g^{-1} which is in Rockafellar [102, Proposition 3].

Proposition 3.2. *T_g^{-1} is Lipschitz continuous at the origin, i.e., $T_g^{-1}(0,0) = \{(\bar{y}, \bar{z})\}$, and for some $\delta > 0$ we have*

$$\|(y,z) - (\bar{y}, \bar{z})\| \leq a_g \|(u_1, u_2)\|,$$

whenever $(y,z) \in T_g^{-1}(u_1, u_2)$ and $\|(u_1, u_2)\| \leq \delta$, if and only if the convex function $p(u_1, u_2)$ is finite and differentiable at $(u_1, u_2) = (0,0)$, and there exist $\lambda > 0$ and $\varepsilon > 0$ such that

$$p(u_1, u_2) \leq p(0,0) + \langle\, u_1, \nabla_y p(0,0) \rangle + \langle\, u_2, \nabla_z p(0,0) \rangle + \lambda \|(u_1, u_2)\|^2, \tag{3.11}$$

for all (u_1, u_2) satisfying $\|(u_1, u_2)\| \leq \varepsilon$.

3.2 Primal SSOSC and constraint nondegeneracy

The first order optimality condition, namely the Karush-Kuhn-Tucker (KKT) condition, for (P) is

$$\begin{cases} \mathcal{Q}(x) + c - \mathcal{A}^* y - \mathcal{B}^* z = 0, \\ \mathcal{A}(x) = b, \ \mathcal{B}(x) \succeq d, \langle\, z, \mathcal{B}(x) - d \,\rangle = 0, \\ x \in \mathbf{X}, \ y \in \mathbf{Y} \ z \succeq 0, \end{cases} \tag{3.12}$$

where $\mathcal{A}^* : \mathbf{Y} \to \mathbf{X}$ and $\mathcal{B}^* : \mathbf{Z} \to \mathbf{X}$ are the adjoins of the linear mappings $\mathcal{A}$ and $\mathcal{B}$ respectively. For any KKT triple $(x, y, z) \in \mathbf{X} \times \mathbf{Y} \times \mathbf{Z}$ satisfying (3.12), we call $x \in \mathbf{X}$

a stationary point and (y, z) a Lagrange multiplier with respect to x. Let $\mathcal{M}(x)$ be the set of all Lagrange multipliers at x.

If the following Robinson's constraint qualification holds at $\bar{x}$, then $\mathcal{M}(\bar{x})$ is nonempty and bounded [16, Theorem 3.9 and Proposition 3.17].

Assumption 3.3. *Let $\bar{x}$ be a feasible solution to the convex QSCP problem (P). Robinson's constraint qualification (CQ) [98] is said to hold at $\bar{x}$ if*

$$\begin{pmatrix} \mathcal{A} \\ \mathcal{B} \end{pmatrix} \mathbf{X} + \begin{pmatrix} \{0\} \\ \mathcal{T}_{\mathbf{K}}(\mathcal{B}(\bar{x}) - d) \end{pmatrix} = \begin{pmatrix} \mathbf{Y} \\ \mathbf{Z} \end{pmatrix}, \tag{3.13}$$

where $\mathcal{T}_{\mathbf{K}}(s)$ is the tangent cone of $\mathbf{K}$ at s.

For any $(\bar{y}, \bar{z}) \in \mathcal{M}(\bar{x})$, suppose that $A := \bar{z} - (\mathcal{B}(\bar{x}) - d)$. Since $\bar{z} \succeq 0$, $\mathcal{B}(\bar{x}) \succeq d$ and $\langle \bar{z}, \mathcal{B}(x) - d \rangle = 0$, we can assume that A has the spectral decomposition as in (2.4), i.e.,

$$A = \lambda_1 c_1 + \lambda_2 c_2 + \cdots + \lambda_n c_n \tag{3.14}$$

where $\{\lambda_1, \ldots, \lambda_r\}$ are the eigenvalues of A being arranged in the nondecreasing order, satisfying

$$\lambda_1 \geq \cdots \geq \lambda_{s_1} > 0 = \lambda_{s_1+1} = \cdots = \lambda_{s_2} > \lambda_{s_2+1} \geq \cdots \geq \lambda_n. \tag{3.15}$$

Then

$$\bar{z} = \sum_{j=1}^{s_1} \lambda_j c_j, \quad (\mathcal{B}(\bar{x}) - d) = - \sum_{j=s_2+1}^{n} \lambda_j c_j. \tag{3.16}$$

According to the definition in (2.16), we denote three index sets

$$\alpha := \{j \mid \lambda_j > 0\}, \quad \gamma := \{j \mid \lambda_j < 0\}, \quad \beta := \{1, \ldots, n\} \backslash (\alpha \cup \gamma). \tag{3.17}$$

From Definition 2.3, we know that the tangent cone of $\mathbf{K}$ at $(\mathcal{B}(\bar{x}) - d)$ is

$$\mathcal{T}_{\mathbf{K}}(\mathcal{B}(\bar{x}) - d) = \{H \in \mathbf{Z} \mid H_{\alpha\alpha} + H_{\alpha\beta} + H_{\beta\beta} \succeq 0\}, \tag{3.18}$$

the critical cone of $\mathbf{K}$ at $(\mathcal{B}(\bar{x}) - d)$ is defined by

$$\mathcal{C}(\mathcal{B}(\bar{x}) - d) := \mathcal{T}_{\mathbf{K}}(\mathcal{B}(\bar{x}) - d) \cap \bar{z}^{\perp} = \{H \in \mathbf{Z} \mid H_{\alpha\alpha} = 0, H_{\alpha\beta} = 0, H_{\beta\beta} \succeq 0\}, \tag{3.19}$$

and the affine hull of $\mathcal{C}(\mathcal{B}(\bar{x}) - d)$ can be written as

$$\mathrm{aff}(\mathcal{C}(\mathcal{B}(\bar{x}) - d)) = \{H \in \mathbf{Z} \mid H_{\alpha\alpha} = 0, H_{\alpha\beta} = 0\}. \tag{3.20}$$

Then, the critical cone $\mathcal{C}(\bar{x})$ of the problem (P) at $\bar{x}$ is given by

$$\begin{aligned}\mathcal{C}(\bar{x}) &= \{h \in \mathbf{X} \mid \mathcal{A}h = 0,\, \mathcal{B}h \in \mathcal{T}_{\mathbf{K}}(\mathcal{B}(\bar{x}) - d), \langle \mathcal{Q}(\bar{x}) + c, h\rangle = 0\} \\ &= \{h \in \mathbf{X} \mid \mathcal{A}h = 0,\, \mathcal{B}h \in \mathcal{C}(\mathcal{B}(\bar{x}) - d)\} \\ &= \{h \in \mathbf{X} \mid \mathcal{A}h = 0,\, (\mathcal{B}h)_{\alpha\alpha} = 0, (\mathcal{B}h)_{\alpha\beta} = 0, (\mathcal{B}h)_{\beta\beta} \succeq 0\}. \end{aligned} \tag{3.21}$$

However, it is difficult to give an explicit formula to the affine hull of $\mathcal{C}(\bar{x})$, denoted by $\mathrm{aff}(\mathcal{C}(\bar{x}))$. We define the following outer approximation set instead of $\mathrm{aff}(\mathcal{C}(\bar{x}))$ with respect to $(\bar{y}, \bar{z}) \in \mathcal{M}(\bar{x})$ by

$$\begin{aligned}\mathrm{app}(\bar{y}, \bar{z}) &= \{h \in \mathbf{X} \mid \mathcal{A}h = 0, \mathcal{B}h \in \mathrm{aff}(\mathcal{C}(\mathcal{B}(\bar{x}) - d))\} \\ &= \{h \in \mathbf{X} \mid \mathcal{A}h = 0, (\mathcal{B}h)_{\alpha\alpha} = 0, (\mathcal{B}h)_{\alpha\beta} = 0\}\end{aligned} \tag{3.22}$$

Then for any $(\bar{y}, \bar{z}) \in \mathcal{M}(\bar{x})$, we have that

$$\mathrm{aff}(\mathcal{C}(\bar{x})) \subseteq \mathrm{app}(\bar{y}, \bar{z}). \tag{3.23}$$

The next proposition shows that the equality in (3.23) holds if $(\bar{y}, \bar{z}) \in \mathcal{M}(\bar{x})$ satisfies a constraint qualification stronger than Robinson's CQ (3.13) at $\bar{x}$.

Proposition 3.4. *[114, Proposition 3.1] Let $\bar{x}$ be a feasible solution to the convex quadratic SDP problem (P) and $(\bar{y}, \bar{z}) \in \mathcal{M}(\bar{x})$. We say that $(\bar{y}, \bar{z})$ satisfies the strict constraint qualification (CQ) [16]*

$$\begin{pmatrix} \mathcal{A} \\ \mathcal{B} \end{pmatrix} \mathbf{X} + \begin{pmatrix} \{0\} \\ \mathcal{T}_{\mathbf{K}}(\mathcal{B}(\bar{x}) - d) \cap \bar{z}^{\perp} \end{pmatrix} = \begin{pmatrix} \mathbf{Y} \\ \mathbf{Z} \end{pmatrix}, \tag{3.24}$$

Then $\mathcal{M}(\bar{x})$ is a singleton and $\mathrm{aff}(\mathcal{C}(\bar{x})) = \mathrm{app}(\bar{y}, \bar{z})$.

By the introduction of the constraint nondegeneracy for sensitivity and stability in optimization and variational inequalities in [15, 110], we have the following formula for the problem (P).

Assumption 3.5. *Let $\bar{x}$ be a feasible solution to the convex quadratic SDP problem (P) and $(\bar{y}, \bar{z}) \in \mathcal{M}(\bar{x})$. We say that the primal constraint nondegeneracy holds at $\bar{x}$ to the problem (P) if*

$$\begin{pmatrix} \mathcal{A} \\ \mathcal{B} \end{pmatrix} \mathbf{X} + \begin{pmatrix} \{0\} \\ \mathrm{lin}[\mathcal{T}_{\mathbf{K}}(\mathcal{B}(\bar{x}) - d)] \end{pmatrix} = \begin{pmatrix} \mathbf{Y} \\ \mathbf{Z} \end{pmatrix}. \tag{3.25}$$

To discuss the rate of convergence, we introduce a strong form of the strong second order sufficient condition for nonlinear programming over symmetric cones given by [118], which is an extension from nonlinear semidefinite programming introduced by Sun [114].

Assumption 3.6. *Let $\bar{x}$ be a feasible solution to (P) and $(\bar{y}, \bar{z}) \in \mathcal{M}(\bar{x})$. If the primal constraint nondegeneracy (3.25) holds at $\bar{x}$, we say that the strong second order sufficient condition holds at $\bar{x}$ if*

$$\langle h, \nabla^2_{xx} L_0(\bar{x}, \bar{y}, \bar{z})\, h \rangle + \Upsilon_{(\mathcal{B}(\bar{x})-d)}\,(\bar{z}, \mathcal{B}h) > 0, \quad \forall\, h \in \mathrm{aff}(\mathcal{C}(\bar{x})) \setminus \{0\}, \tag{3.26}$$

where the linear-quadratic function $\Upsilon_B : \mathbf{X} \times \mathbf{X} \to \Re$ is defined by

$$\Upsilon_B(S, H) := 2\, \langle S \cdot H, B^{\dagger} \cdot H \rangle, \quad (S, H) \in \mathbf{X} \times \mathbf{X}, \tag{3.27}$$

where $B^{\dagger}$ is the Moore-Penrose pseudo-inverse of B.

Remark 3.7. *The primal constraint nondegeneracy (3.25) holds at $\bar{x}$ implies that $(\bar{y}, \bar{z})$ satisfies the strict constraint qualification (3.24), from Proposition 3.4, we know that $\mathcal{M}(\bar{x}) = \{(\bar{y}, \bar{z})\}$ and $\mathrm{app}(\bar{y}, \bar{z}) = \mathrm{aff}(\mathcal{C}(\bar{x}))$.*

3.3 A semismooth Newton-CG method for inner problems

In this section we introduce a semismooth Newton-CG method for solving the inner problems involved in the augmented Lagrangian method (3.3). For this purpose, we need the practical CG method described in [45, Algorithm 10.2.1] for solving the symmetric positive definite linear system. Since our convergence analysis of the semismooth Newton-CG method heavily depends on this practical CG method and its convergence property (Lemma 3.8), we shall give it a brief description here.

3.3.1 A practical CG method

In this subsection, we consider a practical conjugate gradient (CG) method to solve the following linear equation

$$\mathcal{W}(x) = R\,, \tag{3.28}$$

where the linear operator $\mathcal{W} : \mathbf{X} \to \mathbf{X}$ is a self-adjoint and positive definite operator, x and $R \in \mathbf{X}$. The practical conjugate gradient algorithm [45, Algorithm 10.2.1] depends on two parameters: a maximum number of CG iterations $i_{max} > 0$ and a tolerance $\eta \in (0, \|R\|)$.

Algorithm 1. A Practical CG Algorithm: $[CG(\eta, i_{max})]$

Step 0. Given $x^0 = 0$ and $r^0 = R - \mathcal{W}x^0 = R$.

Step 1. While $(\|r^i\| > \eta)$ or $(i < i_{max})$

Step 1.1. $i = i + 1$

Step 1.2. If $i = 1$; $p^1 = r^0$; else; $\beta_i = \|r^{i-1}\|^2/\|r^{i-2}\|^2$, $p^i = r^{i-1} + \beta_i\, p^{i-1}$; end

Step 1.3. $\alpha_i = \|r^{i-1}\|^2/\langle p^i, \mathcal{W}p^i\rangle$

Step 1.4. $x^i = x^{i-1} + \alpha_i p^i$

Step 1.5. $r^i = r^{i-1} - \alpha_i \mathcal{W}p^i$

Lemma 3.8. *Let $0 < \bar{i} \leq i_{\max}$ be the number of iterations when the practical CG Algorithm 1 terminates. For all $i = 1, 2, \cdots, \bar{i}$, the iterates $\{x^i\}$ generated by Algorithm 1 satisfies*

$$\frac{1}{\lambda_{\max}(\mathcal{W})} \leq \frac{\langle x^i, R\rangle}{\|R\|^2} \leq \frac{1}{\lambda_{\min}(\mathcal{W})}, \tag{3.29}$$

where $\lambda_{\min}(\mathcal{W})$ and $\lambda_{\max}(\mathcal{W})$ are the smallest and largest eigenvalues of the matrix representation of $\mathcal{W}$, respectively.

Proof. Let x^* be the exact solution to (3.28) and $e^i = x^* - x^i$ be the error in the ith iteration for $i \geq 0$. From [124, Theorem 38.1], we know that

$$\langle r^i, r^j \rangle = 0 \quad \text{for } j = 1, 2, \ldots, i-1, \tag{3.30}$$

where $r^i = b - \mathcal{W}x^i$. By using (3.30), the fact that in Algorithm 1, $r^0 = R$, and the definition of β_i, we have that

$$\begin{array}{rcl} \langle p^1, R \rangle &=& \|r^0\|^2, \\ \langle p^i, R \rangle &=& \langle r^{i-1}, R \rangle + \beta_i \langle p^{i-1}, R \rangle = 0 + \prod_{j=2}^{i} \beta_j \langle p^1, R \rangle = \|r^{i-1}\|^2 \quad \forall\, i > 1. \end{array} \tag{3.31}$$

From [124, Theorem 38.2], we know that for $i \geq 1$,

$$\|e^{i-1}\|_{\mathcal{W}}^2 = \|e^i\|_{\mathcal{W}}^2 + \langle \alpha_i p^i, \mathcal{W}(\alpha_i p^i) \rangle, \tag{3.32}$$

which, together with $\alpha_i \|r^{i-1}\|^2 = \langle \alpha_i p^i, \mathcal{W}(\alpha_i p^i) \rangle$ (see Step 1.3), implies that

$$\alpha_i \|r^{i-1}\|^2 = \|e^{i-1}\|_{\mathcal{W}}^2 - \|e^i\|_{\mathcal{W}}^2. \tag{3.33}$$

Here for any $x \in \mathbf{X}$, $\|x\|_{\mathcal{W}} := \sqrt{\langle x, \mathcal{W}x \rangle}$. For any $i \geq 1$, by using (3.31), (3.33), and the fact that $x^0 = 0$, we have that

$$\begin{array}{rcl} \langle x^i, R \rangle &=& \langle x^{i-1}, R \rangle + \alpha_i \langle p^i, R \rangle = \langle x^0, R \rangle + \sum_{j=1}^{i} \alpha_j \langle p^j, R \rangle = \sum_{j=1}^{i} \alpha_j \|r^{j-1}\|^2 \\ &=& \sum_{j=1}^{i} \left[\|e^{j-1}\|_{\mathcal{W}}^2 - \|e^j\|_{\mathcal{W}}^2 \right] = \|e^0\|_{\mathcal{W}}^2 - \|e^i\|_{\mathcal{W}}^2, \end{array} \tag{3.34}$$

which, together with (3.32), implies that

$$\langle x^i,\, R \rangle \geq \langle x^{i-1},\, R \rangle, \quad i = 1, 2, \ldots, \bar{i}.$$

Thus

$$\frac{1}{\lambda_{\max}(\mathcal{W})} \leq \alpha_1 = \frac{\langle x^1, R \rangle}{\|R\|^2} \leq \frac{\langle x^i,\, R \rangle}{\|R\|^2}. \tag{3.35}$$

Since $e^0 = x^* - x^0 = \mathcal{W}^{-1}R$, by (3.34), we obtain that for $1 \leq i \leq \bar{i}$,

$$\frac{\langle x^i, R \rangle}{\|R\|^2} \leq \frac{\|e^0\|_{\mathcal{W}}^2}{\|R\|^2} = \frac{\|\mathcal{W}^{-1}R\|_{\mathcal{W}}^2}{\|R\|^2} \leq \frac{1}{\lambda_{\min}(\mathcal{W})}. \tag{3.36}$$

By combining (3.35) and (3.36), we complete the proof. □

3.3.2 Inner problems

To apply the augmented Lagrangian method (3.3) to solve problems (P) and (D), for some fixed $(y, z) \in \mathbf{Y} \times \mathbf{Z}$ and $\sigma > 0$, we need determine the optimal solution to the following convex problem

$$\min \quad \{\varphi_\sigma(x) := L_\sigma(x, y, z) \mid x \in \mathbf{X}\}. \tag{3.37}$$

It is known from [138] that the augmented Lagrangian function $L_\sigma(\cdot)$ is continuously differentiable and for any $(x, y, z) \in \mathbf{X} \times \mathbf{Y} \times \mathbf{Z}$,

$$\nabla\varphi_\sigma(x) = \mathcal{Q}(x) + c - \mathcal{A}^*(y - \sigma(\mathcal{A}(x) - b)) - \mathcal{B}^*\Pi_{\mathbf{K}}(z - \sigma(\mathcal{B}(x) - d)).$$

To find the minimizer of the unconstrained problem (3.37), since $\Pi_{\mathbf{K}}(\cdot)$ is strongly semismooth everywhere, we can use the semismooth Newton-CG method to solve the following nonlinear equation

$$\nabla\varphi_\sigma(x) = 0, \quad \text{for any } (y, z) \in \mathcal{M}(x). \tag{3.38}$$

By the Lipschitz continuity of $\Pi_{\mathbf{K}}(\cdot)$, according to Rademacher's Theorem, $\nabla\varphi_\sigma$ is almost everywhere Fréchet-differentiable in $\mathbf{X}$. For $x \in \mathbf{X}$, the generalized Hessian of φ_σ at x is defined as

$$\partial^2\varphi_\sigma(x) := \partial(\nabla\varphi_\sigma)(x),$$

where $\partial(\nabla\varphi_\sigma)(x)$ is defined in (2.1). Since it is difficult to express $\partial^2\varphi_\sigma(x)$ exactly, we define the following alternative for $\partial^2\varphi_\sigma(x)$ with

$$\hat{\partial}^2\varphi_\sigma(x) := \mathcal{Q} + \sigma\mathcal{A}^*\mathcal{A} + \sigma\mathcal{B}^*\,\partial\Pi_{\mathbf{K}}(z - \sigma(\mathcal{B}(x) - d))\mathcal{B}.$$

From [32, p.75], for $h \in \mathbf{X}$,

$$\partial^2\varphi_\sigma(x)h \subseteq \hat{\partial}^2\varphi_\sigma(x)h,$$

which means that if every element in $\hat{\partial}^2\varphi_\sigma(x)$ is positive definite, so is every element in $\partial^2\varphi_\sigma(x)$.

To apply the semismooth Newton-CG method which will be presented later, we need to choose an element $\widehat{V}_\sigma(x) \in \hat{\partial}^2 \varphi_\sigma(x)$. In the algorithm, we can construct $\widehat{V}_\sigma(x)$ as

$$\widehat{V}_\sigma^0(x) := \mathcal{Q} + \sigma \mathcal{A}^* \mathcal{A} + \sigma \mathcal{B}^* V_\sigma^0 \mathcal{B} \in \hat{\partial}^2 \varphi_\sigma(x), \tag{3.39}$$

where $V_\sigma^0 \in \partial \Pi_{\mathbf{K}}(z - \sigma(\mathcal{B}(x) - d))$ is given by (2.15).

Next we shall characterize the property that $\widehat{V}_\sigma(\hat{x})$ is positive definite. From the discussion in [102, section 4], we know that for any $(x, y, z) \in \mathbf{X} \times \mathbf{Y} \times \mathbf{Z}$,

$$L_\sigma(x, y, z) = \max_{(\xi,\zeta) \in \mathbf{Y} \times \mathbf{Z}} \left\{ l(x, \xi, \zeta) - \frac{1}{2\sigma} \|(\xi, \zeta) - (y, z)\|^2 \right\}.$$

For the existence of the optimal solutions to inner problem (3.37), we need the following condition:

Assumption 3.9. *For inner problem (3.37), there exists an* $\alpha_0 \in \Re$ *such that the level sets* $\{x \in \mathbf{X} \mid \varphi_\sigma(x) \leq \alpha_0\}$ *is nonempty and bounded.*

Under Assumption 3.9, by the definition of g in (3.6), we can deduce from [101, Theoremss 17' and 18'] that

$$\begin{aligned} \min_{x \in \mathbf{X}} \varphi_\sigma(x) &= \min_{x \in \mathbf{X}} \max_{(\xi,\zeta) \in \mathbf{Y} \times \mathbf{Z}} \left\{ l(x, \xi, \zeta) - \frac{1}{2\sigma} \|(\xi, \zeta) - (y, z)\|^2 \right\} \\ &= \max_{\xi \in \mathbf{Y}, \zeta \in \mathbf{Z}} \left\{ g(\xi, \zeta) - \frac{1}{2\sigma} \|(\xi, \zeta) - (y, z)\|^2 \right\} \\ &= \max_{\xi \in \mathbf{Y}, \zeta \succeq 0} \left\{ g_0(\xi, \zeta) - \frac{1}{2\sigma} \|(\xi, \zeta) - (y, z)\|^2 \right\} \end{aligned} \tag{3.40}$$

Hence, the dual of inner problem (3.37) is

$$\begin{array}{ll} \max & g_0(\xi, \zeta) - \frac{1}{2\sigma} \|(\xi, \zeta) - (y, z)\|^2 \\ \text{s.t.} & \xi \in \mathbf{Y}, \zeta \succeq 0 \end{array} \tag{3.41}$$

For any $(\hat{x}, \hat{\xi}, \hat{\zeta}) \in \mathbf{X} \times \mathbf{Y} \times \mathbf{Z}$, we say that $(\hat{x}, \hat{\xi}, \hat{\zeta})$ is a saddle point of the RHS function in (3.40) if it satisfies the following saddle-point conditions for (3.37) and (3.41)

$$\begin{cases} \mathcal{Q}(\hat{x}) + c - \mathcal{A}^*\hat{\xi} - \mathcal{B}^*\hat{\zeta} = 0, \\ b - \mathcal{A}(\hat{x}) - \frac{1}{\sigma}(\hat{\xi} - y) = 0, \\ \frac{1}{\sigma}(\hat{\zeta} - z) + (\mathcal{B}(\hat{x}) - d) \succeq 0, \\ \langle \hat{\zeta}, \frac{1}{\sigma}(\hat{\zeta} - z) + (\mathcal{B}(\hat{x}) - d) \rangle = 0, \\ \hat{x} \in \mathbf{X}, \hat{\xi} \in \mathbf{Y}, \hat{\zeta} \succeq 0. \end{cases} \tag{3.42}$$

Then for any saddle point $(\hat{x}, \hat{\xi}, \hat{\zeta})$ satisfying (3.42), we have that

$$\langle \hat{\zeta}, \hat{\zeta} - (z - \sigma(\mathcal{B}(\hat{x}) - d)) \rangle = 0,$$

with $\hat{\zeta} = \Pi_{\mathbf{K}}(z - \sigma(\mathcal{B}(\hat{x}) - d))$.

Proposition 3.10. *Suppose that Assumption 3.9 is satisfied. Let $(\hat{x}, \hat{\xi}, \hat{\zeta}) \in \mathbf{X} \times \mathbf{Y} \times \mathbf{Z}$ be a saddle point satisfying (3.42). Let $\hat{\zeta}$ and $\hat{\zeta} - (z - \sigma(\mathcal{B}(\hat{x}) - d))$ have the spectral decomposition as in (3.16). Then the following conditions are equivalent:*

(i) The constraint nondegeneracy condition holds at $(\hat{\xi}, \hat{\zeta})$, i.e.,

$$\begin{pmatrix} \mathcal{Q} & \mathcal{A}^* & \mathcal{B}^* \\ 0 & 0 & \mathcal{I} \end{pmatrix} \begin{pmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{pmatrix} + \begin{pmatrix} \{0\} \\ lin[\mathcal{T}_{\mathbf{K}}(\hat{\zeta})] \end{pmatrix} = \begin{pmatrix} \mathbf{X} \\ \mathbf{Z} \end{pmatrix}$$

or, equivalently,

$$\mathcal{Q}(\mathbf{X}) + \mathcal{A}^*(\mathbf{Y}) + \mathcal{B}^*(lin[\mathcal{T}_{\mathbf{K}}(\hat{\zeta})]) = \mathbf{X} \tag{3.43}$$

where $lin[\mathcal{T}_{\mathbf{K}}(\hat{\zeta})]$ denotes the lineality space of $\mathcal{T}_{\mathbf{K}}(\hat{\zeta})$ as in (2.19), i.e.,

$$lin[\mathcal{T}_{\mathbf{K}}(\hat{\zeta})] = \{h \in \mathbf{Z} \mid h_{\beta\beta} = 0, h_{\beta\gamma} = 0, h_{\gamma\gamma} = 0\}, \tag{3.44}$$

where the index sets β and γ are defined in (2.16).

(ii) Every element $\widehat{V}_\sigma(\hat{x}) \in \hat{\partial}^2\varphi_\sigma(\hat{x})$ is self-adjoint and positive definite.

(iii) $\widehat{V}^0_\sigma(\hat{x}) \in \hat{\partial}^2\varphi_\sigma(\hat{x})$ is self-adjoint and positive definite.

Proof. "(i) $\Rightarrow$ (ii)"Let $\widehat{V}_\sigma(\hat{x})$ be an arbitrary element in $\hat{\partial}^2\varphi_\sigma(\hat{x})$ defined by (3.39). Then, there exists an element $V_\sigma \in \partial\Pi_{\mathbf{K}}(z - \sigma(\mathcal{B}(\hat{x}) - d))$ such that

$$\widehat{V}_\sigma(\hat{x}) = \mathcal{Q} + \sigma(\mathcal{A}^*\mathcal{A} + \mathcal{B}^* V_\sigma \mathcal{B}).$$

Since, by Lemma 2.5, V_σ is self-adjoint and positive semidefinite, we know that $\widehat{V}_\sigma(\hat{x})$ is also self-adjoint and positive semidefinite.

Next, we show the positive definiteness of $\widehat{V}_\sigma(\hat{x})$. Let $h \in \mathbf{X}$ be such that $\widehat{V}_\sigma(\hat{x})h = 0$. Then, by (iii) of Lemma 2.5, we obtain that

$$\begin{aligned} 0 = \langle h,\, \widehat{V}_\sigma(\hat{x})h\rangle &= \langle h,\, \mathcal{Q}h\rangle + \sigma(\langle h,\, \mathcal{A}^*\mathcal{A}h\rangle + \langle h,\, \mathcal{B}^* V_\sigma \mathcal{B}h\rangle) \\ &\geq \langle h,\, \mathcal{Q}h\rangle + \sigma(\|\mathcal{A}h\|^2 + \|V_\sigma \mathcal{B}h\|^2), \end{aligned}$$

which, together with the positive semidefiniteness of $\mathcal{Q}$, implies that

$$\mathcal{Q}h = 0, \quad \mathcal{A}h = 0, \quad \text{and} \quad V_\sigma \mathcal{B}h = 0.$$

For any $V_\sigma \in \partial\Pi_{\mathbf{K}}(z - \sigma(\mathcal{B}(\hat{x}) - d))$ and $h \in X$ such that $V_\sigma \mathcal{B}h = 0$, we can obtain that

$$\mathcal{B}h \in [\mathrm{lin}(\mathcal{T}_{\mathbf{K}}(\hat{\zeta}))]^\perp.$$

Since the constraint nondegeneracy condition (3.43) holds at $(\hat{\xi}, \hat{\zeta})$, there exist $h_x \in \mathbf{X}$, $h_y \in \mathbf{Y}$ and $h_z \in \mathrm{lin}(\mathcal{T}_{\mathbf{K}}(\hat{\zeta}))$ such that

$$\mathcal{Q}h_x + \mathcal{A}^* h_y + \mathcal{B}^* h_z = h.$$

Hence, since $\mathcal{B}h \in [\mathrm{lin}(\mathcal{T}_{\mathbf{K}}(\hat{\zeta}))]^\perp$ and $h_z \in \mathrm{lin}(\mathcal{T}_{\mathbf{K}}(\hat{\zeta}))$, it holds that

$$\begin{aligned} \langle h,\, h\rangle &= \langle h,\, \mathcal{Q}h_x + \mathcal{A}^* h_y + \mathcal{B}^* h_z\rangle = \langle h,\, \mathcal{Q}h_x\rangle + \langle h,\, \mathcal{A}^* h_y\rangle + \langle h,\, \mathcal{B}^* h_z\rangle \\ &= \langle \mathcal{Q}h,\, h_x\rangle + \langle \mathcal{A}h,\, h_y\rangle + \langle \mathcal{B}h,\, h_z\rangle = 0. \end{aligned}$$

Thus $h = 0$. This, together with the fact that V_σ is self-adjoint and positive semidefinite, shows that $\widehat{V}_\sigma(\hat{x})$ is self-adjoint and positive definite.

"(ii) $\Rightarrow$ (iii)". This is obviously true since $\widehat{V}^0_\sigma(\hat{x}) \in \hat{\partial}^2\varphi_\sigma(\hat{x})$.

" (iii) $\Rightarrow$ (i) ". Assume on the contrary that the constraint nondegenerate condition (3.43) does not hold at $(\hat{\xi}, \hat{\zeta})$. Then, we have

$$[\mathcal{Q}(\mathbf{X})]^{\perp} \cap [\mathcal{A}^*(\mathbf{Y})]^{\perp} \cap \left[\mathcal{B}^*\mathrm{lin}(\mathcal{T}_{\mathbf{K}}(\hat{\zeta}))\right]^{\perp} \neq \{0\}. \tag{3.45}$$

Take an arbitrary $0 \neq h \in [\mathcal{Q}(\mathbf{X})]^{\perp} \cap [\mathcal{A}^*(\mathbf{Y})]^{\perp} \cap \left[\mathcal{B}^*\mathrm{lin}(\mathcal{T}_{\mathbf{K}}(\hat{\zeta}))\right]^{\perp}$. Then for any $x \in \mathbf{X}$,

$$\langle h,\, \mathcal{Q}x\rangle = \langle \mathcal{Q}h,\, x\rangle = 0 \quad \Rightarrow \quad \mathcal{Q}h = 0.$$

For any $y \in \mathbf{Y}$,

$$\langle h,\, \mathcal{A}^*y\rangle = \langle \mathcal{A}h,\, y\rangle = 0 \quad \Rightarrow \quad \mathcal{A}h = 0.$$

And for any $z \in \mathrm{lin}(\mathcal{T}_{\mathbf{K}}(\hat{\zeta}))$,

$$\langle h,\, \mathcal{B}^*z\rangle = \langle \mathcal{B}h,\, z\rangle = 0 \quad \Rightarrow \quad (\mathcal{B}h)_{\alpha\alpha} = 0, (\mathcal{B}h)_{\alpha\beta} = 0, (\mathcal{B}h)_{\alpha\gamma} = 0.$$

From the definition of V_σ^0 in (2.15), it follows that $V_\sigma^0(\mathcal{B}h) = 0$. Therefore, for the corresponding $\widehat{V}_\sigma^0(\hat{x})$ given by (3.39), we can obtain that

$$\langle h,\, \widehat{V}_\sigma^0(\hat{x})h\rangle = \langle h,\, \mathcal{Q}h\rangle + \sigma(\langle h,\, \mathcal{A}h\rangle + \langle \mathcal{B}h,\, V_\sigma^0(\mathcal{B}h)\rangle) = 0,$$

which contradicts (iii) since $h \neq 0$. This contradiction shows that (i) holds. □

Remark 3.11. *The condition (3.43) is actually the constraint nondegeneracy condition for the following problem*

$$\max \quad -\frac{1}{2}\langle x,\, \mathcal{Q}(x)\rangle + \langle b,\, \xi\rangle + \langle d,\, \zeta\rangle \tag{3.46}$$

$$\text{s.t.} \quad -\mathcal{Q}(x) + \mathcal{A}^*\xi + \mathcal{B}^*\zeta = c, \tag{3.47}$$

$$\xi \in \mathbf{Y},\ \zeta \succeq 0. \tag{3.48}$$

If the constraint nondegeneracy condition (3.43) holds at $(\hat{\xi}, \hat{\zeta})$, $\mathcal{M}(\hat{\xi}, \hat{\zeta})$ *is a singleton, i.e.,* $\mathcal{M}(\hat{\xi}, \hat{\zeta}) = \{\hat{x}\}$.

Since $\widehat{V}_\sigma^I(\hat{x})$ is an element in $\hat{\partial}^2\varphi(\hat{x})$, given by

$$\widehat{V}_\sigma^I(x) := \mathcal{Q} + \sigma\mathcal{A}^*\mathcal{A} + \sigma\mathcal{B}^*V_\sigma^I\mathcal{B} \in \hat{\partial}^2\varphi_\sigma(x), \tag{3.49}$$

where $V_\sigma^I \in \partial\Pi_{\mathbf{K}}(z - \sigma(\mathcal{B}(x) - d))$ given by (2.15). Similar to Proposition 3.10, we give a weaker condition for the positive definiteness of $\widehat{V}_\sigma^I(\hat{x})$ based on its particular structure.

Corollary 3.12. *Suppose that Assumption 3.9 is satisfied. Let* $(\hat{x}, \hat{\xi}, \hat{\zeta}) \in \mathbf{X} \times \mathbf{Y} \times \mathbf{Z}$ *be a saddle point satisfying (3.42) and* $\hat{\zeta}$ *and* $\hat{\zeta} - (z - \sigma(\mathcal{B}(\hat{x}) - d))$ *have the spectral decomposition as in (3.16). Then* $\widehat{V}_\sigma^I(\hat{x})$ *is self-adjoint and positive definite if the point* $\hat{x}$ *satisfies the following strict constraint qualification (CQ)*

$$\begin{pmatrix} \mathcal{Q} & \mathcal{A}^* & \mathcal{B}^* \\ 0 & 0 & \mathcal{I} \end{pmatrix} \begin{pmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{pmatrix} + \begin{pmatrix} \{0\} \\ \mathcal{T}_{\mathbf{K}}(\hat{\zeta}) \cap (\mathcal{B}(\hat{x}) - d)^\perp \end{pmatrix} = \begin{pmatrix} \mathbf{X} \\ \mathbf{Z} \end{pmatrix}.$$

or, equivalently,

$$\mathcal{Q}(\mathbf{X}) + \mathcal{A}^*\mathbf{Y} + \mathcal{B}^* \left[\mathcal{T}_{\mathbf{K}}(\hat{\zeta}) \cap (\mathcal{B}(\hat{x}) - d)^\perp\right] = \mathbf{X}, \tag{3.50}$$

where $\mathcal{I}$ *is the identity mapping from* $\mathbf{Z}$ *to* $\mathbf{Z}$ *and* $\mathcal{T}_{\mathbf{K}}(\hat{\zeta}) \cap (\mathcal{B}(\hat{x}) - d)^\perp$ *i.e.,*

$$\mathcal{T}_{\mathbf{K}}(\hat{\zeta}) \cap (\mathcal{B}(\hat{x}) - d)^\perp = \{h \in \mathbf{Z} \mid h_{\beta\beta} \succeq 0, h_{\beta\gamma} = 0, h_{\gamma\gamma} = 0\}, \tag{3.51}$$

where index sets β *and* γ *are defined in (2.16).*

Proof. From the definition of $V_\sigma^I(\hat{x})$ in (3.49) and Lemma 2.5, since $V^I(\hat{x}) \in \partial\Pi_{\mathbf{K}}(z - \sigma(\mathcal{B}(x) - d))$ is self-adjoint and positive semidefinite, we know that $V_\sigma^I(\hat{x})$ is also self-adjoint and positive semidefinite. Next, we show the positive definiteness of $\widehat{V}_\sigma^I(\hat{x})$.

Let $h \in \mathbf{X}$ be such that $\widehat{V}_\sigma^I(\hat{x})h = 0$. Then, by (iii) of Lemma 2.5, we obtain that

$$\begin{aligned} 0 = \langle h, \widehat{V}_\sigma^I(\hat{x})h\rangle &= \langle h, \mathcal{Q}h\rangle + \sigma(\langle h, \mathcal{A}^*\mathcal{A}h\rangle + \langle h, \mathcal{B}^* V_\sigma^I(\hat{x})\mathcal{B}h\rangle) \\ &\geq \langle h, \mathcal{Q}h\rangle + \sigma(\|\mathcal{A}h\|^2 + \|V_\sigma^I(\hat{x})\mathcal{B}h\|^2), \end{aligned}$$

which implies that

$$\mathcal{Q}h = 0, \quad \mathcal{A}h = 0, \quad \text{and} \quad V_\sigma^I(\hat{x})\mathcal{B}h = 0.$$

From the definition of $V_\sigma^I(\hat{x})$ in (2.15) and $h \in X$, we have that

$$V_\sigma^I(\hat{x})\mathcal{B}h = 0 \quad \Rightarrow \quad (\mathcal{B}h)_{\alpha\alpha} = 0, (\mathcal{B}h)_{\alpha\beta} = 0, (\mathcal{B}h)_{\alpha\gamma} = 0, \text{ and } (\mathcal{B}h)_{\beta\beta} = 0.$$

Thus

$$\mathcal{B}h \in [\mathcal{T}_{\mathbf{K}}(\hat{\zeta}) \cap (\mathcal{B}(\hat{x}) - d)^\perp]^\perp.$$

Since $\hat{x}$ satisfies the strict CQ (3.50), there exist $h_x \in \mathbf{X}$, $h_y \in \mathbf{Y}$ and $h_z \in \mathcal{T}_{\mathbf{K}}(\hat{\zeta}) \cap (\mathcal{B}(\hat{x}) - d)^{\perp}$ such that $\mathcal{Q}h_x + \mathcal{A}^* h_y + \mathcal{B}^* h_z = h$. Hence, since $\mathcal{B}h \in [\mathcal{T}_{\mathbf{K}}(\hat{\zeta}) \cap (\mathcal{B}(\hat{x}) - d)^{\perp}]^{\perp}$, it holds that

$$\begin{aligned} \langle h,\, h\rangle &= \langle h,\, \mathcal{Q}h_x + \mathcal{A}^* h_y + \mathcal{B}^* h_z\rangle \\ &= \langle h,\, \mathcal{Q}h_x\rangle + \langle h,\, \mathcal{A}^* h_y\rangle + \langle h,\, \mathcal{B}^* h_z\rangle \\ &= \langle \mathcal{B}h,\, h_z\rangle = 0. \end{aligned}$$

Thus $h = 0$. This, together with the fact that $V_\sigma^I(\hat{x})$ is self-adjoint and positive semidefinite, shows that $\widehat{V}_\sigma^I(\hat{x})$ is self-adjoint and positive definite. □

3.3.3 A semismooth Newton-CG method

Next we shall introduce a promised semismooth Newton-CG (SNCG) algorithm to solve (3.37). Choose $x^0 \in \mathbf{X}$. Then the algorithm can be stated as follows.

A SNCG algorithm $[SNCG(x^0, y, z, \sigma)]$

Step 0. Given $\mu \in (0, 1/2)$, $\bar{\eta} \in (0, 1)$, $\tau \in (0, 1]$, $\tau_1, \tau_2 \in (0, 1)$, and $\delta \in (0, 1)$.

Step 1. For $j = 0, 1, 2, \ldots$

Step 1.1. Given a maximum number of CG iterations $n_j > 0$ and compute

$$\eta_j := \min(\bar{\eta}, \|\nabla\varphi_\sigma(x^j)\|^{1+\tau}).$$

Apply the practical CG Algorithm $[CG(\eta_j, n_j)]$ to find an approximation solution d^j to

$$(\widehat{V}_\sigma(x^j) + \varepsilon_j I)\, d = -\nabla\varphi_\sigma(x^j), \tag{3.52}$$

where $\widehat{V}_\sigma(x^j) \in \hat{\partial}^2\varphi_\sigma(x^j)$ defined in (3.39) and

$$\varepsilon_j := \tau_1 \min\{\tau_2, \|\nabla\varphi_\sigma(x^j)\|\}.$$

Step 1.2. Set $\alpha_j = \delta^{m_j}$, where m_j is the first nonnegative integer m for which

$$\varphi_\sigma(x^j + \alpha_j d^j) \le \varphi_\sigma(x^j) + \mu\alpha_j \langle \nabla\varphi_\sigma(x^j), d^j\rangle. \tag{3.53}$$

Step 1.3. Set $x^{j+1} = x^j + \alpha_j\, d^j$.

Remark 3.13. *In the SNCG algorithm, since $\widehat{V}_\sigma(x^j)$ is always positive semidefinite, the matrix $\widehat{V}_\sigma(x^j) + \varepsilon_j I$ is positive definite as long as $\nabla\varphi_\sigma(x^j) \neq 0$. So we can always apply Algorithm 1 to solve the equation (3.52).*

Now we can analyze the global convergence of the SNCG algorithm with the assumption that $\nabla\varphi_\sigma(x^j) \neq 0$ for any $j \ge 0$. From Lemma 3.8, we know that the search direction d^j generated by the SNCG algorithm is always a descent direction. This is stated in the following proposition.

Proposition 3.14. *For every $j \ge 0$, the search direction d^j generated in Step 1.2 of the SNCG algorithm satisfies*

$$\frac{1}{\lambda_{\max}(\widetilde{V}_\sigma^j)} \le \frac{\langle -\nabla\varphi_\sigma(x^j), d^j\rangle}{\|\nabla\varphi_\sigma(x^j)\|^2} \le \frac{1}{\lambda_{\min}(\widetilde{V}_\sigma^j)}, \tag{3.54}$$

where $\widetilde{V}_\sigma^j := \widehat{V}_\sigma(x^j) + \varepsilon_j I$ and $\lambda_{\max}(\widetilde{V}_\sigma^j)$ and $\lambda_{\min}(\widetilde{V}_\sigma^j)$ are the largest and smallest eigenvalues of $\widetilde{V}_\sigma^j$ respectively.

Theorem 3.15. *Suppose that Assumption 3.9 holds for problem (3.37). Then the SNCG algorithm is well defined and any accumulation point $\hat{x}$ of $\{x^j\}$ generated by the SNCG algorithm is an optimal solution to the inner problem (3.37).*

Proof. By Step 1.1 in the SNCG algorithm, for any $j \ge 0$, since, by (3.54), d^j is a descent direction, the SNCG algorithm is well defined. Under Assumption 3.9, since the level set $\{x \in \mathbf{X} \mid \varphi_\sigma(x) \le \varphi_\sigma(x^0)\}$ is a closed and bounded convex set, the sequence $\{x^j\}$ is bounded. Let $\hat{x}$ be any accumulation point of $\{x^j\}$. Then, by making use of Proposition 3.14 and the Lipschitz continuity of $\Pi_{\mathbf{K}}(\cdot)$, we can easily derive that $\nabla\varphi_\sigma(\hat{x}) = 0$. By the convexity of $\varphi_\sigma(\cdot)$, $\hat{x}$ is an optimal solution of (3.37). □

Since the SNCG algorithm is well defined, we next shall discuss its rate of convergence for solving inner problems (3.37).

Theorem 3.16. *Suppose that Assumption 3.9 holds for problem (3.37). Let* $\bar{x}$ *be an accumulation point of the infinite sequence* $\{x^j\}$ *generated by the SNCG algorithm for solving the inner problems (3.37). Suppose that for* $j \geq 0$*, the practical CG algorithm terminates when the tolerance* η_j *is achieved, i.e.,*

$$\|\nabla\varphi_\sigma(x^j) + (\widehat{V}_\sigma(x^j) + \varepsilon_j I)\, d^j\| \leq \eta_j. \tag{3.55}$$

If the constraint nondegeneracy condition (3.43) holds at $(\hat{\xi}, \hat{\zeta})$*, then the whole sequence* $\{x^j\}$ *converges to* $\hat{x}$ *and*

$$\|x^{j+1} - \hat{x}\| = O(\|x^j - \hat{x}\|^{1+\tau}). \tag{3.56}$$

Proof. By Theorem 3.15, we know that the infinite sequence $\{x^j\}$ is bounded and $\hat{x}$ is an optimal solution to (3.37) with

$$\nabla\varphi_\sigma(\hat{x}) = 0.$$

Since the constraint nondegenerate condition (3.43) is assumed to hold at $(\hat{\xi}, \hat{\zeta})$, $\hat{x}$ is the unique optimal solution to (3.37). It then follows from Theorem 3.15 that $\{x^j\}$ converges to $\hat{x}$. From Proposition 4.6, we know that for any $V_\sigma(\hat{x}) \in \hat{\partial}^2\varphi_\sigma(\hat{x})$ defined in (3.39), there exists a $V_\sigma \in \partial\Pi_{\mathbf{K}}(z - \sigma(\mathcal{B}(\hat{x}) - d))$ such that

$$\widehat{V}_\sigma(\hat{x}) = \mathcal{Q} + \sigma(\mathcal{A}^*\mathcal{A} + \mathcal{B}^* V_\sigma \mathcal{B}) \succ \mathbf{0}.$$

Then, for all j sufficiently large, $\{\|(\widehat{V}_\sigma(x^j) + \varepsilon_j I)^{-1}\|\}$ is uniformly bounded.

For any $\widehat{V}_\sigma(x^j)$, $j \geq 0$, there exists a $V_\sigma(x^j) \in \partial\Pi_{\mathbf{K}}(z - \sigma(\mathcal{B}(x^j) - d))$ such that

$$\widehat{V}_\sigma(x^j) = \mathcal{Q} + \sigma(\mathcal{A}^*\mathcal{A} + \mathcal{B}^* V_\sigma(x^j)\mathcal{B}). \tag{3.57}$$

Since $\Pi_{\mathbf{K}}(\cdot)$ is strongly semismooth [115], it holds that for all j sufficiently large,

$$\begin{aligned}
&\|x^j + d^j - \hat{x}\| \;=\; \|x^j + (\widehat{V}_\sigma(x^j) + \varepsilon_j I)^{-1}((\nabla\varphi_\sigma(x^j) + (\widehat{V}_\sigma(x^j) + \varepsilon_j I)\, d^j) - \nabla\varphi(x^j)) - \hat{x}\| \\
\leq\;& \|x^j - \hat{x} - (\widehat{V}_\sigma(x^j) + \varepsilon_j I)^{-1}\nabla\varphi_\sigma(x^j)\| + \|(\widehat{V}_\sigma(x^j) + \varepsilon_j I)^{-1}\|\, \|\nabla\varphi_\sigma(x^j) + (\widehat{V}_\sigma(x^j) + \varepsilon_j I)\, d^j\| \\
\leq\;& \|(\widehat{V}_\sigma(x^j) + \varepsilon_j I)^{-1}\| \Big(\|\nabla\varphi_\sigma(x^j) - \nabla\varphi_\sigma(\hat{x}) - \widehat{V}_\sigma(x^j)(x^j - \hat{x})\| + (\varepsilon_j \|x^j - \hat{x}\| + \eta_j) \Big) \\
\leq\;& O(\|\mathcal{B}^*\| \|\Pi_{\mathbf{K}}(z - \sigma(\mathcal{B}(x^j) - d)) - \Pi_{\mathbf{K}}(z - \sigma(\mathcal{B}(\hat{x}) - d)) - V_\sigma(x^j)(-\sigma\mathcal{B}(x^j - \hat{x}))\| \\
& + \tau_1 \|\nabla\varphi_\sigma(x^j)\| \|x^j - \hat{x}\| + \|\nabla\varphi_\sigma(x^j)\|^{1+\tau}) \\
\leq\;& O\left(\|\sigma\mathcal{B}(x^j - \hat{x})\|^2 + \tau_1 \|\nabla\varphi_\sigma(x^j) - \nabla\varphi_\sigma(\hat{x})\| \|x^j - \hat{x}\| + \|\nabla\varphi(x^j) - \nabla\varphi(\hat{x})\|^{1+\tau} \right) \\
\leq\;& O(\|x^j - \hat{x}\|^2 + \tau_1 \sigma \|\mathcal{B}^*\| \|\mathcal{B}\| \|x^j - \hat{x}\|^2 + (\sigma \|\mathcal{B}^*\| \|\mathcal{B}\| \|x^j - \hat{x}\|)^{1+\tau}) \\
=\;& O(\|x^j - \hat{x}\|^{1+\tau}),
\end{aligned} \tag{3.58}$$

which implies that for all j sufficiently large,

$$x^j - \hat{x} = -d^j + O(\|d^j\|^{1+\tau}) \quad \text{and} \quad \|d^j\| \to 0. \tag{3.59}$$

For each $j \geq 0$, let $R^j := \nabla\varphi_\sigma(x^j) + (\widehat{V}_\sigma(x^j) + \varepsilon_j I)\, d^j$. Then, for all j sufficiently large,

$$\begin{aligned}
\langle R^j, d^j \rangle \;&\leq\; \eta_j \|d^j\| \leq \|d^j\| \|\nabla\varphi_\sigma(x^j)\|^{1+\tau} \leq \|\nabla\varphi_\sigma(x^j) - \nabla\varphi_\sigma(\hat{x})\|^{1+\tau} \|d^j\| \\
&\leq\; (\sigma \|\mathcal{B}^*\| \|\mathcal{B}\| \|x^j - \hat{x}\|)^{1+\tau} \|d^j\| \leq O(\|d^j\|^{2+\tau}),
\end{aligned}$$

that is,

$$-\langle \nabla\varphi_\sigma(x^j),\, d^j \rangle \geq \langle d^j,\, (\widehat{V}_\sigma(x^j) + \varepsilon_j I)\, d^j \rangle + O(\|d^j\|^{2+\tau}),$$

which, together with (3.59) and the fact that $\|(\widehat{V}_\sigma(x^j) + \varepsilon_j I)^{-1}\|$ is uniformly bounded, implies that there exists a constant $\hat{\delta} > 0$ such that

$$-\langle \nabla\varphi_\sigma(x^j), d^j \rangle \geq \hat{\delta} \|d^j\|^2 \qquad \text{for all } j \text{ sufficiently large.}$$

Since $\nabla\varphi_\sigma(\cdot)$ is (strongly) semismooth at $\hat{x}$ (because $\Pi_{\mathbf{K}}(\cdot)$ is strongly semismooth everywhere), from [37, Theorem 3.3 & Remark 3.4] or [83], we know that for $\mu \in (0, 1/2)$, there exists an integer j_0 such that for any $j \geq j_0$,

$$\varphi_\sigma(x^j + d^j) \leq \varphi(x^j) + \mu \langle \nabla\varphi_\sigma(x^j), d^j \rangle,$$

which means that for all $j \geq j_0$,

$$x^{j+1} = x^j + d^j.$$

This, together with (3.58), completes the proof. $\square$

Theorem 3.16 shows that the rate of convergence for the SNCG algorithm is of order $(1+\tau)$. If $\tau = 1$, this corresponds to quadratic convergence. However, this will need more iterations in the practical CG method. To save computational time, in practice we choose $\tau = 0.1 \sim 0.2$, which still ensures the SNCG algorithm achieves superlinear convergence.

3.4 A NAL method for convex QSCP

In this section, for any $k \geq 0$, let $\varphi_{\sigma_k}(\cdot) \equiv L_{\sigma_k}(\cdot, y^k, z^k)$. Since the inner problems can not be solved exactly, we will use the following stopping criteria considered by Rockafellar [102, 103] for terminating the SNCG algorithm:

(A) $\varphi_{\sigma_k}(x^{k+1}) - \inf \varphi_{\sigma_k} \leq \mu_k^2/2\sigma_k, \qquad \mu_k \geq 0,\ \sum_{k=0}^{\infty} \mu_k < \infty.$

(B) $\varphi_{\sigma_k}(x^{k+1}) - \inf \varphi_{\sigma_k} \leq (\delta_k^2/2\sigma_k)\|(y^{k+1}, z^{k+1}) - (y^k, z^k)\|^2, \qquad \delta_k \geq 0,\ \sum_{k=0}^{\infty} \delta_k < \infty.$

(B') $\|\nabla \varphi_{\sigma_k}(x^{k+1})\| \leq (\delta_k'/\sigma_k)\|(y^{k+1}, z^{k+1}) - (y^k, z^k)\|, \qquad 0 \leq \delta_k' \to 0.$

We shall introduce a semismooth Newton-CG augmented Lagrangian algorithm for solving the convex quadratic problems (P) and (D).

A NAL Algorithm

Step 0. Given $(x^0, y^0, z^0) \in \mathbf{X} \times \mathbf{Y} \times \mathbf{K}$, $\sigma_0 > 0$, a threshold $\bar{\sigma} \geq \sigma_0 > 0$ and $\rho > 1$.

Step 1. For $k = 0, 1, 2, \ldots$

Step 1.1. Starting with x^k as the initial point, apply the SNCG algorithm to $\varphi_{\sigma_k}(\cdot)$ to find $x^{k+1} = \text{SNCG}(x^k, y^k, z^k, \sigma_k)$.

Step 1.2. Updating $y^{k+1} = y^k - \sigma_k(\mathcal{A}(x^{k+1}) - b)$ and $z^{k+1} = \Pi_{\mathbf{K}}(z^k - \sigma_k(\mathcal{B}^* x^{k+1} - d))$ satisfying (A), (B) or (B').

Step 1.3. If $\sigma_k \leq \bar{\sigma}$, $\sigma_{k+1} = \rho\,\sigma_k$ or $\sigma_{k+1} = \sigma_k$.

The global convergence of the NAL algorithm follows from Rockafellar [103, Theorem 1] and [102, Theorem 4] without much difficulty.

Theorem 3.17. *Let the NAL algorithm be executed with stopping criterion (A). Assume that (P) satisfies Robinson's CQ (3.13), then the sequence $\{(y^k, z^k)\} \subset \mathbf{Y} \times \mathbf{K}$ generated by the NAL algorithm is bounded and $\{(y^k, z^k)\}$ converges to $(\bar{y}, \bar{z})$, where $(\bar{y}, \bar{z})$ is some optimal solution to (D), and $\{x^k\}$ is asymptotically minimizing for (P) with $\max(D) = \inf(P)$.*

If $\{(y^k, z^k)\}$ is bounded and Assumption 3.1 is satisfied, then the sequence $\{x^k\}$ is also bounded, and all of its accumulation points of the sequence $\{x^k\}$ are optimal solutions to (P).

By using the result from [118] and [129] on the extension of Theorem 4.1 in [114], we can obtain the following corollary for the Lipschitz continuity of T_l^{-1}.

Corollary 3.18. *Let $\bar{x}$ be a feasible solution to (P). Suppose that Robinson's CQ (3.13) holds at $\bar{x}$ and $(\bar{x}, \bar{y}, \bar{z}) \in \mathbf{X} \times \mathbf{Y} \times \mathbf{Z}$ be a KKT point satisfying the KKT conditions (3.12). If the strong second order sufficient condition (3.26) holds at $\bar{x}$ and $\bar{x}$ is constraint nondegenerate satisfying (3.25), T_l^{-1} is Lipschitz continuous at the origin with modulus a_l.*

Next we state the local linear convergence of the Newton-CG augmented Lagrangian algorithm.

Theorem 3.19. *Let the NAL algorithm be executed with stopping criteria (A) and (B). Suppose that (P) satisfies Robinson's CQ (3.13). If T_g^{-1} is Lipschitz continuous at the origin with modulus a_g, then the generated sequence $\{(y^k, z^k)\} \subset \mathbf{Y} \times \mathbf{K}$ is bounded and $\{(y^k, z^k)\}$ converges to the unique solution $(\bar{y}, \bar{z})$ with $\max(D) = \min(P)$, and for all k*

sufficiently large,

$$\|(y^{k+1}, z^{k+1}) - (\bar{y}, \bar{z})\| \leq \theta_k \|(y^k, z^k) - (\bar{y}, \bar{z})\|,$$

where

$$\theta_k = \left[a_g(a_g^2 + \sigma_k^2)^{-1/2} + \delta_k\right](1 - \delta_k)^{-1} \to \theta_\infty = a_g(a_g^2 + \sigma_\infty^2)^{-1/2} < 1,\ \sigma_k \to \sigma_\infty\,,$$

and a_g *is a Lipschitz constant of* T_g^{-1} *at the origin (cf. Proposition 3.2). The conclusions of Theorem 3.17 about* $\{(y^k, z^k)\}$ *are valid.*

Moreover, if the stopping criterion (B') *is also used and Assumption 3.5 and 3.6 are satisfied, then in addition to the above conclusions the sequence* $\{x^k\} \to \bar{x}$*, where* $\bar{x}$ *is the unique optimal solution to* (P)*, and one has for all* k *sufficiently large,*

$$\|x^{k+1} - \bar{x}\| \leq \theta'_k \|(y^{k+1}, z^{k+1}) - (y^k, z^k)\|,$$

where $\theta'_k = a_l(1 + \delta'_k)/\sigma_k \to \delta_\infty = a_l/\sigma_\infty$ *and* $a_l (\geq a_g)$ *is a Lipschitz constant of* T_l^{-1} *at the origin.*

Remark 3.20. *Note that in (3.3) we can also add the term* $\frac{1}{2\sigma_k}\|x - x^k\|^2$ *to* $L_{\sigma_k}(x, y^k, z^k)$ *such that* $L_{\sigma_k}(x, y^k, z^k) + \frac{1}{2\sigma_k}\|x - x^k\|^2$ *is a strongly convex function. This actually corresponds to the proximal method of multipliers considered in [102, Section 5] for which the* k*-th iteration is given by*

$$\begin{cases} x^{k+1} \approx \arg\min_{x \in \mathbf{X}} \{L_{\sigma_k}(x, y^k, z^k) + \frac{1}{2\sigma_k}\|x - x^k\|^2\} \\ y^{k+1} = y^k - \sigma(\mathcal{A}(x^{k+1}) - b) \\ z^{k+1} = \Pi_{\mathbf{K}}[z^k - \sigma(\mathcal{B}(x^{k+1}) - d)] \\ \sigma_{k+1} = \rho\,\sigma_k \text{ or } \sigma_{k+1} = \sigma_k. \end{cases} \tag{3.60}$$

Convergence analysis for (3.60) can be conducted in a parallel way as for (3.3).

Chapter 4

Linear programming over symmetric cones

In this chapter we will study in details the semismooth Newton-CG augmented Lagrangian method for solving linear symmetric cone programming. Due to the explicit form of the dual problem, we can characterize the Lipschitiz continuity of the corresponding solution mapping at the origin for the analysis of the rate of convergence of our proposed method. For the inner problems, we will give the condition that equivalent to the positive definiteness of the generalized Hessian of the objective function in those inner problems.

4.1 Linear symmetric cone programming

According to the convex QSCP problems (P) and (D), we consider the following linear programming over symmetric cones,

$$(LP) \qquad \begin{array}{rl} \min\limits_{x\in\mathbf{X}} & \langle c,\, x\rangle \\ \text{s.t.} & \mathcal{A}(x) = b, \\ & \mathcal{B}(x) \succeq d. \end{array}$$

Let $\mathcal{F}_{LP} := \{x \in \mathbf{X} \mid \mathcal{A}(x) = b, \mathcal{B}(x) \succeq d\}$ be the feasible set of (LP). Thus the dual of (LP) takes the form

$$\begin{array}{llll} (LD) & \max & \langle b,\, y\rangle + \langle z,\, d\rangle \\ & \text{s.t.} & \mathcal{A}^* y + \mathcal{B}^* z = c, \\ & & y \in \mathbf{Y},\ z \succeq 0. \end{array}$$

Let $\mathcal{F}_{LD} := \{(y, z) \in \mathbf{Y} \times \mathbf{K} \mid \mathcal{A}^* y + \mathcal{B}^* z = c\}$ be the feasible set of (LD). The KKT conditions of (LP) and (LD) are as follows:

$$\begin{cases} \mathcal{A}^* y + \mathcal{B}^* z = c, \\ \mathcal{A}(x) = b, \mathcal{B}(x) \succeq d \\ y \in \mathbf{Y},\ z \succeq 0, \\ \langle z,\, \mathcal{B}(x) - d\rangle = 0. \end{cases} \tag{4.1}$$

For any KKT point $(\bar{x}, \bar{y}, \bar{z}) \in \mathbf{X} \times \mathbf{Y} \times \mathbf{Z}$, $\mathcal{M}(\bar{y}, \bar{z})$ denotes the set of all the Lagrange multipliers at $(\bar{y}, \bar{z})$.

Let $(\bar{y}, \bar{z})$ be an optimal solution to (LD). It is well known that $\mathcal{M}(\bar{y}, \bar{z})$ is nonempty and bounded if and only if problem (LD) satisfies the following Robinson's constraint qualification.

Assumption 4.1. *Let $(\bar{y}, \bar{z})$ be a feasible point to (LD). Robinson's constraint qualification (CQ) [98] is said to hold at $(\bar{y}, \bar{z})$ if*

$$\begin{pmatrix} \mathcal{A}^* & \mathcal{B}^* \\ 0 & \mathcal{I} \end{pmatrix} \begin{pmatrix} \mathbf{Y} \\ \mathbf{Z} \end{pmatrix} + \begin{pmatrix} \{0\} \\ \mathcal{T}_{\mathbf{K}}(\bar{z}) \end{pmatrix} = \begin{pmatrix} \mathbf{X} \\ \mathbf{Z} \end{pmatrix}. \tag{4.2}$$

According to the discussion of maximal monotone operators and their inverses given in Section 3.2, we have the analogous definitions for those operators for the problems (LP) and (LD). The Lagrangian function $l_0 : \mathbf{X} \times \mathbf{Y} \times \mathbf{Z} \to \Re$ for (LP) in extended form is defined as:

$$l(x, y, z) = \begin{cases} \langle c,\, x\rangle + \langle y,\, \mathcal{A}(x) - b\rangle - \langle z,\, \mathcal{B}(x) - d\rangle & \text{if } x \in \mathbf{X}, y \in \mathbf{Y} \text{ and } z \in \mathbf{K}, \\ -\infty & \text{if } x \in \mathbf{X}, y \in \mathbf{Y} \text{ and } z \notin \mathbf{K}. \end{cases} \tag{4.3}$$

The essential objective function of (LP) takes the form as

$$f(x) = \inf_{x\in\mathbf{X}} l(x,y,z) = \begin{cases} \langle c,\, x\rangle & \text{if } x \in \mathcal{F}_{LP}, \\ +\infty & \text{otherwise.} \end{cases} \tag{4.4}$$

while the essential objective function of (LD) takes the form as

$$g(y,z) = \inf_{x\in\mathbf{X}} l(x,y,z) = \begin{cases} \langle b,\, y\rangle + \langle z,\, d\rangle & \text{if } (y,z) \in \mathcal{F}_{LD}, \\ -\infty & \text{otherwise.} \end{cases} \tag{4.5}$$

Assume $\mathcal{F}_{LP}$ and $\mathcal{F}_{LD}$ are nonempty, the maximal monotone operators $T_f = -\partial f$ and $T_g = -\partial g$ and $T_l = \partial l$. And for each $v \in \mathbf{X}$ and $(u_1, u_2) \in \mathbf{Y} \times \mathbf{Z}$, consider the following parameterized problem of (LP),

$$(LP(v,u_1,u_2)) \qquad \begin{array}{ll} \min\limits_{x\in\mathbf{X}} & \langle c,\, x\rangle + \langle v,\, x\rangle \\ \text{s.t.} & \mathcal{A}(x) - u_1 = b, \\ & \mathcal{B}(x) - u_2 \succeq d. \end{array}$$

And it dual problem is

$$(LD(v,u_1,u_2)) \qquad \begin{array}{ll} \max & \langle\, b + u_1, y\,\rangle + \langle\, d + u_2, z\,\rangle \\ \text{s.t.} & \mathcal{A}^* y + \mathcal{B}^* z - v = c, \\ & y \in \mathbf{Y},\ z \succeq 0. \end{array}$$

Since $F_{LP} \neq \emptyset$ and $F_{LD} \neq \emptyset$, then

$$\begin{aligned} T_f^{-1}(v) &= \arg\min_{x\in\mathbf{X}} \{f(x) + \langle v,\, x\rangle\} \\ &= \text{set of all optimal solutions to } (LP(v,0,0)), \end{aligned}$$

and

$$\begin{aligned} T_g^{-1}(u_1,u_2) &= \arg\max_{(y,z)\in\mathbf{Y}\times\mathbf{Z}} \{g(y,z) + \langle\, u_1, y\,\rangle + \langle\, u_2, z\,\rangle\} \\ &= \text{set of all optimal solutions to } (LD(0,u_1,u_2)). \end{aligned}$$

Assume that (LD) satisfies Robinson's CQ (4.2), we have that

$$\begin{aligned} T_l^{-1}(v,u_1,u_2) &= \arg \max_{(x,y,z)\in\mathbf{X}\times\mathbf{Y}\times\mathbf{Z}} \{l_0(x,y,z) - \langle v,\, x\rangle + \langle\, u_1, y\,\rangle + \langle\, u_2, z\,\rangle\} \\ &= \text{set of all } (x,y,z) \text{ satisfying the KKT conditions (4.1)} \\ &\quad\ \text{for } (LP(v,u_1,u_2)). \end{aligned} \tag{4.6}$$

In order to analyze the convergence of the semismooth Newton-CG augmented Lagrangian method for the problem (LD), we need the following result which characterizes the Lipschitz continuity of the corresponding solution mapping at the origin. The result we establish here is stronger than that established in Proposition 15 of [24].

Proposition 4.2. *Suppose that (LD) satisfies Robinson's (CQ) (4.2). Let $(\bar{y},\bar{z}) \in \mathbf{Y}\times\mathbf{Z}$ be an optimal solution to (LD). For any $\bar{x}\in\mathcal{M}(\bar{y},\bar{z})$, $\bar{z}$ and $\mathcal{B}(\bar{x})-d$ have the spectral decomposition as in (3.16). Then the following conditions are equivalent*

(i) $T_g^{-1}(\cdot,\cdot)$ is Lipschitz continuous at $(0,0)\in\mathbf{Y}\times\mathbf{Z}$.

(ii) The second order sufficient condition

$$\sup_{x\in\mathcal{M}(\bar{y},\bar{z})} \Upsilon_{\bar{z}}(\mathcal{B}(x)-d, h_z) > 0 \quad \forall\, (h_y,h_z)\in\mathcal{C}(\bar{y},\bar{z})\setminus\{0\} \tag{4.7}$$

holds at $(\bar{y},\bar{z})$, where $\mathcal{C}(\bar{y},\bar{z})$ denotes the critical cone of problem (LD),

$$\mathcal{C}(\bar{y},\bar{z}) = \{(h_y,h_z)\in\mathbf{Y}\times\mathbf{Z} \mid \mathcal{A}^*h_y + \mathcal{B}^*h_z = 0, (h_z)_{\beta\beta}\succeq 0, (h_z)_{\beta\gamma}=0, (h_z)_{\gamma\gamma}=0\} \tag{4.8}$$

(iii) $(\bar{y},\bar{z})$ satisfies the extended strict primal-dual constraint qualification

$$\begin{pmatrix} \mathcal{A} \\ \mathcal{B} \end{pmatrix}\mathbf{X} + \begin{pmatrix} \{0\} \\ \mathrm{conv}\left(\bigcup_{x\in\mathcal{M}(\bar{y},\bar{z})} \left(\mathcal{T}_{\mathbf{K}}(\mathcal{B}(\bar{x})-d)\cap \bar{z}^{\perp}\right)\right) \end{pmatrix} = \begin{pmatrix} \mathbf{Y} \\ \mathbf{Z} \end{pmatrix} \tag{4.9}$$

where for any set $\mathcal{W}\subset\mathcal{S}^n$, $\mathrm{conv}(\mathcal{W})$ denotes the convex hull of $\mathcal{W}$.

Proof. "(i) $\Leftrightarrow$ (ii)". From [16, Theorem 3.137], we know that (ii) holds if and only if for all $(y,z)\in\mathcal{N}$ such that $(y,z)\in\mathcal{F}_{LD}$, the quadratic growth condition

$$\langle b,\, y\rangle + \langle d,\, z\rangle \geq \langle b,\, \bar{y}\rangle + \langle d,\, \bar{z}\rangle + c\|(y,z)-(\bar{y},\bar{z})\|^2, \tag{4.10}$$

holds at $(\bar{y}, \bar{z})$ for some positive constant c and an open neighborhood $\mathcal{N}$ of $(\bar{y}, \bar{z})$ in $\mathbf{Y} \times \mathbf{Z}$. On the other hand, from [102, Proposition 3], we know that $T_g^{-1}(\cdot)$ is Lipschiz continuous at the origin if and only if the quadratic growth condition (4.10) holds at $(\bar{y}, \bar{z})$. Hence, (i) $\Leftrightarrow$ (ii).

Next we shall prove that (ii) $\Leftrightarrow$ (iii). For notational convenience, let

$$\Gamma := \mathrm{conv}\left(\bigcup_{x \in \mathcal{M}(\bar{y},\bar{z})} \left(\mathcal{T}_{\mathbf{K}}(\mathcal{B}(x) - d) \cap \bar{z}^{\perp}\right)\right). \tag{4.11}$$

"(ii) $\Rightarrow$ (iii)". Denote

$$\mathcal{D} := \begin{pmatrix} \mathcal{A} \\ \mathcal{B} \end{pmatrix} \mathbf{X} + \begin{pmatrix} \{0\} \\ \Gamma \end{pmatrix}.$$

For the purpose of contradiction, we assume that (iii) does not hold, i.e.,

$$\mathcal{D} \neq \begin{pmatrix} \mathbf{Y} \\ \mathbf{Z} \end{pmatrix}.$$

Let $\mathrm{cl}(\mathcal{D})$ and $\mathrm{ri}(\mathcal{D})$ denote the closure of $\mathcal{D}$ and the relative interior of $\mathcal{D}$, respectively. By [99, Theorem 6.3], since $\mathrm{ri}(\mathcal{D}) = \mathrm{ri}(\mathrm{cl}(\mathcal{D}))$, the relative interior of $\mathrm{cl}(\mathcal{D})$, we know that

$$\mathrm{cl}(\mathcal{D}) \neq \begin{pmatrix} \mathbf{Y} \\ \mathbf{Z} \end{pmatrix}.$$

Thus, there exists $B := (B_y, B_z) \in \mathbf{Y} \times \mathbf{Z}$ such that $B \notin \mathrm{cl}(\mathcal{D})$. Let $\overline{B}$ be the metric projection of B onto $\mathrm{cl}(\mathcal{D})$, i.e., $\overline{B} = \Pi_{\mathrm{cl}(\mathcal{D})}(B)$. Let $H = \overline{B} - B \neq 0$. Since $\mathrm{cl}(\mathcal{D})$ is a nonempty closed convex cone, from Zarantonello [138], we know that

$$\langle H, S \rangle = \langle \overline{B} - B, S \rangle \geq 0 \quad \forall\, S \in \mathrm{cl}(\mathcal{D}).$$

In particular, let $H = (h_y, h_z)$, we have that

$$\langle h_y,\, \mathcal{A}(x) \rangle + \langle h_z,\, \mathcal{B}(x) + Q \rangle \geq 0 \qquad \forall\, x \in \mathbf{X} \text{ and } Q \in \Gamma,$$

which implies (by taking $Q = \mathbf{0}$)

$$\langle \mathcal{A}^* h_y + \mathcal{B}^* h_z,\, x \rangle = \langle B_y,\, \mathcal{A}(x) \rangle + \langle h_z,\, \mathcal{B}(x) \rangle \geq 0 \quad \forall\, x \in \mathbf{X}.$$

Thus

$$\mathcal{A}^* h_y + \mathcal{B}^* h_z = 0 \quad \text{and} \quad \langle h_z, Q\rangle \geq 0 \quad \text{for any } Q \in \Gamma. \tag{4.12}$$

Since $0 \neq H \in \mathcal{C}(\bar{y}, \bar{z})$ and (ii) is assumed to hold, there exists $x \in \mathcal{M}(\bar{y}, \bar{z})$ such that

$$\Upsilon_{\bar{z}}(\mathcal{B}(x) - d, h_z) > 0. \tag{4.13}$$

By using the fact that $(x, \bar{y}, \bar{z})$ satisfies (4.1), we can assume that $\bar{z}$ and $(\mathcal{B}(x) - d)$ have the spectral decompositions as in (3.16). Then,

$$\mathcal{T}_{\mathbf{K}}(\mathcal{B}(x) - d) \cap \bar{z}^{\perp} = \{H \in \mathbf{X} \mid H_{\beta\beta} \succeq 0, H_{\alpha\alpha} = 0, H_{\alpha\beta} = 0.\} \tag{4.14}$$

where the index sets α and β given by (3.17). For any $Q \in \mathcal{T}_{\mathbf{K}}(\mathcal{B}(x)-d)\cap\bar{z}^{\perp}$, $\langle h_z, Q\rangle \geq 0$ implies that

$$(h_z)_{\alpha\gamma} = 0, (h_z)_{\beta\gamma} = 0, (h_z)_{\alpha\gamma} = 0, \text{ and } (h_z)_{\beta\beta} \succeq 0. \tag{4.15}$$

By using (4.8), (4.12), and (4.15), we obtain that $H \in \mathcal{C}(\bar{y}, \bar{z})$ and

$$(h_z)_{\alpha\gamma} = 0. \tag{4.16}$$

Therefore, from (2.23) and (3.15), we obtain that

$$\Upsilon_{\bar{z}}(\mathcal{B}(x) - d, h_z) = \sum_{j=1}^{s_1} \sum_{l=s_2+1}^{r} \frac{-\lambda_l}{\lambda_j} \|(h_z)_{jl}\|^2 = 0$$

which contradicts (4.13). This contradiction shows (ii) $\Rightarrow$ (iii).

"(iii) $\Rightarrow$ (ii)". Assume that (ii) does not hold at $(\bar{y}, \bar{z})$. Then there exists $H = (h_y, h_z) \in \mathcal{C}(\bar{y}, \bar{z}) \setminus \{0\}$ such that

$$\sup_{x\in\mathcal{M}(\bar{y},\bar{z})} \Upsilon_{\bar{z}}(\mathcal{B}(x) - d, h_z) = 0. \tag{4.17}$$

Let x be an arbitrary element in $\mathcal{M}(\bar{y}, \bar{z})$. Since $(x, \bar{y}, \bar{z})$ satisfies (4.1), we can assume that $\bar{z}$ and $(\mathcal{B}(x) - d)$ have the spectral decompositions as in (3.16) with index sets in (3.17). From (3.16), (3.27), and (4.17), we have

$$0 \leq \sum_{j=1}^{s_1} \sum_{l=s_2+1}^{r} \frac{-\lambda_l}{\lambda_j} \|(h_z)_{jl}\|^2 = \Upsilon_{\bar{z}}(\mathcal{B}(x) - d, h_z) \leq \sup_{x\in\mathcal{M}(\bar{y},\bar{z})} \Upsilon_{\bar{z}}(\mathcal{B}(x) - d, h_z) = 0,$$

which implies

$$(h_z)_{\alpha\gamma} = 0. \tag{4.18}$$

Then, by using (4.8), (4.14), and (4.18), we have that for any $Q^x \in \mathcal{T}_{\mathbf{K}}(\mathcal{B}(x) - d) \cap \bar{z}^{\perp}$,

$$\langle Q^x, h_z \rangle = \langle Q^x_{\beta\beta}, (h_z)_{\beta\beta} \rangle \geq 0. \tag{4.19}$$

Since (iii) is assumed to hold, there exist $x \in \mathbf{X}$ and $Q \in \Gamma$ such that

$$\begin{cases} -h_y = \mathcal{A}(x) \\ -h_z = \mathcal{B}(x) + Q. \end{cases} \tag{4.20}$$

By Carathéodory's Theorem, there exist an integer $k \leq \frac{n(n+1)}{2} + 1$ and scalars $\alpha_i \geq 0$, $i = 1, 2, \ldots, k$, with $\sum_{i=1}^{k} \alpha_i = 1$, and

$$Q_i \in \bigcup_{x \in \mathcal{M}(\bar{y}, \bar{z})} \left(\mathcal{T}_{\mathbf{K}}(\mathcal{B}(x) - d) \cap \bar{z}^{\perp} \right), \quad i = 1, 2, \ldots, k$$

such that Q can be represented as

$$Q = \sum_{i=1}^{k} \alpha_i Q_i.$$

For each Q_i, there exists a $x^i \in \mathcal{M}(\bar{y}, \bar{z})$ such that $Q_i \in \mathcal{T}_{\mathcal{K}}(\mathcal{B}(x^i) - d) \cap \bar{z}^{\perp}$. Then by using the fact that $H = (h_y, h_z) \in \mathcal{C}(\bar{y}, \bar{z})$ and (4.19), we obtain that

$$\begin{aligned} \langle H, H \rangle &= \langle h_y, -A(x) \rangle + \langle h_z, -\mathcal{B}(x) - Q \rangle \\ &= -\langle \mathcal{A}^* h_y + \mathcal{B}^* h_z, x \rangle - \langle h_z, Q \rangle \\ &= 0 - \sum_{i=1}^{k} \alpha_i \langle h_z, Q_i \rangle \leq 0. \end{aligned}$$

Thus $H = 0$ which contradicts the fact that $H \neq 0$. This contradiction shows that (ii) holds. □

Proposition 4.2 characterizes the Lipschitz continuity of T_g^{-1} at the origin by either the second sufficient condition (4.7) or the extended strict primal-dual constraint qualification (4.9). In particular, if $\mathcal{M}(\bar{y}, \bar{z})$ is a singleton, we have the following simple equivalent conditions.

Corollary 4.3. *Suppose that (LD) satisfies Robinson's (CQ) (4.2). Let $(\bar{y}, \bar{z})$ be an optimal solution to (LD). If $\mathcal{M}(\bar{y}, \bar{z}) = \{\bar{x}\}$, then the following are equivalent:*

(i) $T_g^{-1}(\cdot)$ *is Lipschitz continuous at the origin.*

(ii) The second order sufficient condition

$$\Upsilon_{\bar{z}}(\mathcal{B}(\bar{x}) - d, h_z) > 0 \quad \forall\, (h_y, h_z) \in \mathcal{C}(\bar{y}, \bar{z}) \setminus \{0\} \tag{4.21}$$

holds at $(\bar{y}, \bar{z})$.

(iii) $(\bar{y}, \bar{z})$ satisfies the strict primal-dual constraint qualification

$$\begin{pmatrix} \mathcal{A} \\ \mathcal{B} \end{pmatrix} \mathbf{X} + \begin{pmatrix} \{0\} \\ \mathcal{T}_{\mathbf{K}}(\mathcal{B}(\bar{x}) - d) \cap \bar{z}^{\perp} \end{pmatrix} = \begin{pmatrix} \mathbf{Y} \\ \mathbf{Z} \end{pmatrix} \tag{4.22}$$

Remark 4.4. *Note that for semidefinite programming in [24, Proposition 15], Chan and Sun proved that if the multiplier set is a singleton, then the* strong *second order sufficient condition at $\bar{x}$ for (LP) is equivalent to the constraint nondegenerate condition for (LD). Hence, we can extend to linear programming over symmetric cones, if $\mathcal{M}(\bar{y}, \bar{z})$ is a singleton, then the* strong *second order sufficient condition (with the set $\mathcal{C}(\bar{y}, \bar{z})$ in (4.21) being replaced by the superset $\{(h_y, h_z) \in \mathbf{Y} \times \mathbf{Z} \mid \mathcal{A}^* h_y + \mathcal{B}^* h_z = 0, (h_z)_{\beta\beta} = 0, (h_z)_{\beta\gamma} = 0, (h_z)_{\gamma\gamma} = 0\}$) is equivalent to the constraint nondegenerate condition, in the sense of Robinson [96, 97], at $\bar{x}$ for (LP), i.e,*

$$\begin{pmatrix} \mathcal{A} \\ \mathcal{B} \end{pmatrix} \mathbf{X} + \begin{pmatrix} \{0\} \\ \mathrm{lin}[\mathcal{T}_{\mathbf{K}}(\mathcal{B}(\bar{x}) - d)] \end{pmatrix} = \begin{pmatrix} \mathbf{Y} \\ \mathbf{Z} \end{pmatrix}. \tag{4.23}$$

where $\mathrm{lin}[\mathcal{T}_{\mathbf{K}}(\mathcal{B}(\bar{x}) - d)]$ *denotes the lineality space of $\mathcal{T}_{\mathbf{K}}(\mathcal{B}(\bar{x}) - d)$ defined in (2.19), i.e.,*

$$\mathrm{lin}(\mathcal{T}_{\mathbf{K}}(\mathcal{B}(\bar{x}) - d) = \{H \in \mathbf{Z} \mid H_{\alpha\alpha} = 0, H_{\alpha\beta} = 0, H_{\beta\beta} = 0\}, \tag{4.24}$$

where the index sets α and β are defined in (2.16).

Corollary 4.3 further establishes the equivalence between the second order sufficient condition (4.21) and the strict constraint qualification (4.22) under the condition that $\mathcal{M}(\bar{y}, \bar{z})$ is a singleton.

One may observe that the strict primal-dual constraint qualification condition (4.22) is weaker than the constraint nondegenerate condition (4.23). However, if strict complementarity holds, i.e., $\bar{z} + (\mathcal{B}(\bar{x}) - d) \succ 0$ and hence β is the empty set, then (4.22) and (4.23) coincide. The constraint nondegenerate condition (4.23) is equivalent to the primal nondegeneracy stated in [3, Theorem 6]. Note that under such a condition, the optimal solution $(\bar{y}, \bar{z})$ to (LD) is unique.

Remark 4.5. *In a similar way, we can establish parallel results for T_f^{-1} as for T_g^{-1} in Proposition 4.2 and Corollary 4.3. For brevity, we omit the details.*

4.2 Convergence analysis

To apply the augmented Lagrangian method (3.3) to problem (LP), for some fixed $(y, z) \in \mathbf{Y} \times \mathbf{Z}$, we need to consider the following form of inner problems

$$\min \{\varphi_\sigma(y) := L_\sigma(x, y, z) \mid y \in \mathbf{Y}\}. \tag{4.25}$$

where $L_\sigma(x, y, z)$ is the *augmented Lagrangian* function for problem (LP), defined as

$$\begin{aligned} L_\sigma(x, y, z) &= \langle c, x\rangle - \langle y, \mathcal{A}(x) - b\rangle + \frac{\sigma}{2}\|\mathcal{A}(x) - b\|^2 \\ &\quad + \frac{1}{2\sigma}\Big[\|\Pi_{\mathbf{K}}[z - \sigma(\mathcal{B}(x) - d)]\|^2 - \|z\|^2\Big], \end{aligned} \tag{4.26}$$

where $(x, y, z) \in \mathbf{X} \times \mathbf{Y} \times \mathbf{Z}$.

For finding the optimal solution to (4.25), by the strongly semismoothness of $\Pi_{\mathbf{K}}$, we can apply SNCG algorithm to solve the following nonlinear equation

$$\nabla\varphi_\sigma(x) = c - \mathcal{A}^*(y - \sigma(\mathcal{A}(x) - b)) - \mathcal{B}^*\Pi_{\mathbf{K}}(z - \sigma(\mathcal{B}(x) - d)) = 0.$$

Then in the SNCG algorithm, we choose an element $\widehat{V}_\sigma^0(x)$ as

$$\widehat{V}_\sigma^0(x) = \sigma(\mathcal{A}^*\mathcal{A} + \mathcal{B}^* V_\sigma^0(x)\mathcal{B}) \in \hat{\partial}^2\varphi_\sigma(x), \tag{4.27}$$

where $V_\sigma^0(x) \in \partial\Pi_{\mathbf{K}}(z - \sigma(\mathcal{B}(x) - d))$ defined in (2.15) and $\hat{\partial}^2\varphi_\sigma(x)$ is the alternative form for the generalized Hessian of $\varphi_\sigma(x)$ and has the form

$$\hat{\partial}^2\varphi_\sigma(x) := \sigma\Big(\mathcal{A}^*\mathcal{A} + \mathcal{B}^*\,\partial\Pi_{\mathbf{K}}(z - \sigma(\mathcal{B}(x) - d))\mathcal{B}\Big).$$

Next we shall characterize the property that $\widehat{V}_\sigma$ is positive definite at $\hat{x}$. Since Robinson's CQ (4.2) is assumed to hold, by the definition of g in (4.5), we can deduce from [101, Theorem 17' and 18'] that

$$\min_{x\in\mathbf{X}} \varphi_\sigma(x) = \max_{\xi\in\mathbf{Y},\zeta\succeq 0} \Big\{ g(\xi,\zeta) - \frac{1}{2\sigma}\|(\xi,\zeta)-(y,z)\|^2 \Big\}.$$

Hence, the dual of (4.25) is

$$\begin{array}{ll} \max & \langle b,\,\xi\rangle + \langle d,\,\zeta\rangle - \frac{1}{2\sigma}\|(\xi,\zeta)-(y,z)\|^2 \\ \text{s.t.} & \mathcal{A}^*\xi + \mathcal{B}^*\zeta = c, \\ & \xi\in\mathbf{Y}, \zeta\succeq 0. \end{array} \tag{4.28}$$

The KKT conditions of (4.28) are as follows

$$\left\{ \begin{array}{l} b - \frac{1}{\sigma}(\xi - y) - \mathcal{A}(x) = 0, \\ \frac{1}{\sigma}(\zeta - z) + \mathcal{B}(x) - d \succeq 0, \\ \langle \zeta,\, \frac{1}{\sigma}(\zeta - z) + \mathcal{B}(x) - d\rangle = 0, \\ \mathcal{A}^*\xi + \mathcal{B}^*\zeta = c, \\ x\in\mathbf{X},\ \xi\in\mathbf{Y},\ \zeta\succeq 0. \end{array} \right. \tag{4.29}$$

For a triple $(\hat{x},\hat{\xi},\hat{\zeta})\in\mathbf{X}\times\mathbf{Y}\times\mathbf{Z}$ satisfying the KKT conditions (4.29), $\hat{\zeta}$ and $(\hat{\zeta} - z + \sigma(\mathcal{B}(\hat{x}) - d))$ have the spectral decomposition as (3.16). Then $\bar{x}\in\mathcal{M}(\hat{\xi},\hat{\zeta})$ satisfies the following strict constraint qualification (CQ)

$$\begin{pmatrix} \mathcal{A}^* & \mathcal{B}^* \\ 0 & \mathcal{I} \end{pmatrix} \begin{pmatrix} \mathbf{Y} \\ \mathbf{Z} \end{pmatrix} + \begin{pmatrix} \{0\} \\ [\mathcal{T}_{\mathbf{K}}(\hat{\zeta}) \cap (\mathcal{B}(\hat{x}) - d)]^\perp \end{pmatrix} = \begin{pmatrix} \mathbf{X} \\ \mathbf{Z} \end{pmatrix}.$$

or, equivalently,

$$\mathcal{A}^*\mathbf{Y} + \mathcal{B}^*\left[\mathcal{T}_{\mathbf{K}}(\hat{\zeta}) \cap (\mathcal{B}(\hat{x}) - d)^\perp\right] = \mathbf{X}, \tag{4.30}$$

where $\mathcal{I}$ is the identity mapping from $\mathbf{Z}$ to $\mathbf{Z}$ and $\mathcal{T}_{\mathbf{K}}(\hat{\zeta}) \cap (\mathcal{B}(\hat{x}) - d)^\perp$. i.e.,

$$(\mathcal{T}_{\mathbf{K}}(\hat{\zeta})) \cap (\mathcal{B}(\hat{x}) - d)^\perp = \{h\in\mathbf{Z} \mid h_{\beta\beta}\succeq 0, h_{\beta\gamma} = h_{\gamma\gamma} = 0\}, \tag{4.31}$$

where index sets β and γ are defined in (2.16). Then $\mathcal{M}(\hat{\xi},\hat{\zeta})$ is singleton [16, Proposition 4.50].

Furthermore, the constraint nondegenerate condition holds at $(\hat{\xi},\hat{\zeta})$ to the problem (4.28) if

$$\begin{pmatrix} \mathcal{A}^* & \mathcal{B}^* \\ 0 & \mathcal{I} \end{pmatrix}\begin{pmatrix} \mathbf{Y} \\ \mathbf{Z} \end{pmatrix} + \begin{pmatrix} \{0\} \\ \mathrm{lin}[\mathcal{T}_{\mathbf{K}}(\hat{\zeta})] \end{pmatrix} = \begin{pmatrix} \mathbf{X} \\ \mathbf{Z} \end{pmatrix}.$$

or, equivalently,

$$\mathcal{A}^*\mathbf{Y} + \mathcal{B}^*\left[\mathrm{lin}(\mathcal{T}_{\mathbf{K}}(\hat{\zeta}))\right] = \mathbf{X}, \tag{4.32}$$

where $\mathrm{lin}[\mathcal{T}_{\mathbf{K}}(\hat{\zeta})]$ denotes the lineality space of $\mathcal{T}_{\mathbf{K}}(\hat{\zeta})$ defined in (2.19), i.e.,

$$\mathrm{lin}(\mathcal{T}_{\mathbf{K}}(\hat{\zeta})) = \{h \in \mathbf{Z} \mid h_{\beta\beta} = 0, h_{\beta\gamma} = 0, h_{\gamma\gamma} = 0\}, \tag{4.33}$$

where index sets β and γ are defined in (2.16).

As a special case of convex quadratic programming (P), the following propositions for inner problems (4.25) and (4.28) are analogous to Proposition 3.10 and Corollary 3.12.

Proposition 4.6. *Let $(\hat{x},\hat{\xi},\hat{\zeta}) \in \mathbf{X}\times\mathbf{Y}\times\mathbf{Z}$ be a triple that satisfies the KKT conditions (4.29) and $\hat{\zeta}$ and $(\hat{\zeta} - z + \sigma(\mathcal{B}(\hat{x}) - d))$ have the spectral decomposition as (3.16). Then the following conditions are equivalent:*

(i) The constraint nondegenerate condition (4.32) holds at $(\hat{\xi},\hat{\zeta})$ to the problem (4.28).

(ii) Every $\widehat{V}_\sigma(\hat{x}) \in \hat{\partial}^2\varphi_\sigma(\hat{x})$ is self-adjoint and positive definite.

(iii) Choose $V_\sigma^0(x) \in \partial\Pi_{\mathbf{K}}(z - \sigma(\mathcal{B}(x) - d))$ defined in (4.27), $V_\sigma^0(x)$ is self-adjoint and positive definite.

Moreover, since $\widehat{V}_\sigma^I(\hat{x}) \in \hat{\partial}^2\varphi(\hat{x})$ defined by

$$\widehat{V}_\sigma^I(x) = \sigma(\mathcal{A}^*\mathcal{A} + \mathcal{B}^*V_\sigma^I(x)\mathcal{B}) \in \hat{\partial}^2\varphi_\sigma(x), \tag{4.34}$$

where $V_\sigma^I(x) \in \partial\Pi_{\mathbf{K}}(z - \sigma(\mathcal{B}(x) - d))$ defined in (2.15). Parallel to Corollary 3.12, we can give the following corollary for the positive definiteness of $\widehat{V}_\sigma^I(\hat{x})$.

Corollary 4.7. *Let* $(\hat{x}, \hat{\xi}, \hat{\zeta}) \in \mathbf{X} \times \mathbf{Y} \times \mathbf{Z}$ *be a triple that satisfies the KKT conditions (4.29). If* $\hat{x} \in \mathcal{M}(\hat{\xi}, \hat{\zeta})$ *satisfies the strict constraint qualification (CQ) (4.30), then* $\widehat{V}_\sigma^I(\hat{x})$ *is self-adjoint and positive definite.*

For linear symmetric cone programming, we shall discuss the rate of convergence of SNCG algorithm to solve the problem (4.28).

Theorem 4.8. *Assume that the problem (4.28) satisfies Robinson's CQ (4.2). Let* $\hat{x}$ *be an accumulation point of the infinite sequence* $\{x^j\}$ *generated by SNCG algorithm for solving the inner problem (4.25). Suppose that at each step* $j \geq 0$*, when the practical CG Algorithm 1 terminates, the tolerance* η_j *is achieved (e.g., when* $n_j = m + 1$*), i.e.,*

$$\|\nabla \varphi(x^j) + (\widehat{V}_\sigma(x^j) + \varepsilon_j I)\, d^j\| \leq \eta_j. \tag{4.35}$$

Assume that the constraint nondegenerate condition (4.32) holds at $\hat{\zeta} := \Pi_{\mathbf{K}}(z - \sigma(\mathcal{B}(\hat{x}) - d))$*. Then the whole sequence* $\{x^j\}$ *converges to* $\hat{x}$ *and*

$$\|x^{j+1} - \hat{x}\| = O(\|x^j - \hat{x}\|^{1+\tau}). \tag{4.36}$$

The global convergence of the NAL algorithm for the linear problems (LP) and (LD) is similar to that in Theorem 3.17 for convex quadratic symmetric cone programming. By the explicit form of the problem (LD), we next state the rate of convergence of the NAL algorithm for linear cases.

Theorem 4.9. *Let the NAL algorithm be executed with stopping criteria* (A) *and* (B)*. Assume that (LP) and* (LD) *satisfy Robinson's CQ (3.13) and (4.2) respectively. If the extended strict primal-dual constraint qualification (4.9) holds at* $(\bar{y}, \bar{z})$*, where* $(\bar{y}, \bar{z})$ *is an optimal solution to* (LD)*, then the generated sequence* $\{(y^k, z^k)\} \subset \mathbf{Y} \times \mathbf{K}$ *is bounded and* $\{(y^k, z^k)\}$ *converges to the unique solution* $(\bar{y}, \bar{z})$ *with* $\min(LP) = \max(LD)$*, and*

$$\|(y^{k+1}, z^{k+1}) - (\bar{y}, \bar{z})\| \leq \theta_k \|(y^k, z^k) - (\bar{y}, \bar{z})\| \quad \textit{for all } k \textit{ sufficiently large,}$$

where

$$\theta_k = \left[a_g(a_g^2 + \sigma_k^2)^{-1/2} + \delta_k\right](1 - \delta_k)^{-1} \to \theta_\infty = a_g(a_g^2 + \sigma_\infty^2)^{-1/2} < 1,\ \sigma_k \to \sigma_\infty\,,$$

and a_g is a Lipschitz constant of T_g^{-1} at the origin (cf. Proposition 4.2). The conclusions of Theorem 3.17 about $\{(y^k, z^k)\}$ are valid.

Moreover, if the stopping criterion (B') is also used and the constraint nondegenerate conditions (4.23) and (4.32) hold at $\bar{x}$ and $(\bar{y}, \bar{z})$, respectively, then in addition to the above conclusions the sequence $\{x^k\} \to \bar{x}$, where $\bar{x}$ is the unique optimal solution to (LP), and one has

$$\|x^{k+1} - \bar{x}\| \leq \theta'_k \|(y^{k+1}, z^{k+1}) - (y^k, z^k)\| \quad \textit{for all } k \textit{ sufficiently large,}$$

where $\theta'_k = a_l(1 + \delta'_k)/\sigma_k \to \delta_\infty = a_l/\sigma_\infty$ and a_l is a Lipschitz constant of T_l^{-1} at the origin.

Proof. Conclusions of the first part of Theorem 4.9 follow from the results in [103, Theorem 2] and [102, Theorem 5] combining with Proposition 4.2. By using the fact that T_l^{-1} is Lipschitz continuous near the origin under the assumption that the constraint nondegenerate conditions (4.23) and (4.32) hold, respectively, at $\bar{x}$ and $(\bar{y}, \bar{z})$ [118] and [129], we can directly obtain conclusions of the second part of this theorem from [103, Theorem 2] and [102, Theorem 5]. □

Chapter 5

Numerical results for convex QSDPs

We implemented the semismooth Newton-CG augmented Lagrangian (the NAL algorithm) in MATLAB to solve a variety of large scale convex quadratic programming problems on a PC (Intel Xeon 3.2 GHz with 4G of RAM). We measure the infeasibilities and optimality for the primal and dual problems as follows:

$$R_P = \max\left(\frac{\|b-\mathcal{A}(x)\|}{\max(1,\|b\|)}, \frac{\|\mathcal{B}(x)-S-d\|}{\max(1,\|d\|)}\right), \tag{5.1}$$

$$R_D = \frac{\|\mathcal{Q}(x)+c-\mathcal{A}^*y-\mathcal{B}^*z\|}{\max(1,\|c\|)}, \tag{5.2}$$

$$\text{gap} = \frac{\langle x,\mathcal{Q}(x)\rangle+\langle c,x\rangle-\langle b,y\rangle-\langle d,z\rangle}{1+|\text{pobj}|+|\text{dobj}|} \tag{5.3}$$

where $S=(\Pi_{\mathcal{S}^n_+}(W)-W)/\sigma$ with $W=z-\sigma(\mathcal{B}(x)-d)$,

$$\text{pobj} = \frac{1}{2}\langle x,\mathcal{Q}(x)\rangle+\langle c,x\rangle \tag{5.4}$$

$$\text{dobj} = -\frac{1}{2}\langle x,\mathcal{Q}(x)\rangle+b^Ty+\langle d,z\rangle. \tag{5.5}$$

We do not check the infeasibilities of the conditions $z\succeq 0$, $\mathcal{B}(x)\succeq d$, $\langle z,(\mathcal{B}(x)-d)\rangle=0$, since they are satisfied up to machine precision throughout the NAL algorithm.

In our numerical experiments, we stop the NAL algorithm when

$$\max\{R_D,R_P\} \leq \text{tol}, \tag{5.6}$$

where the tolerance is 10^{-6} for most problems except the EDM problems is 10^{-4}. In solving the subproblem (3.37), we cap the maximum number of Newton iterations to be 50, while in computing the inexact Newton direction from (3.52), we stop the practical CG solver when the maximum number of CG steps exceeds 500, or when the convergence is too slow in that the reduction in the residual norm is exceedingly small.

5.1 Random convex QSCP problems

5.1.1 Random convex QSDP problems

We consider the collection of random sparse convex quadratic problems over symmetric cones which is given as follows,

$$\begin{array}{ll} \min & \frac{1}{2}\langle\, x^s, x^s\,\rangle + \langle c^s, x^s\rangle + \frac{1}{2}\langle\, x^l, Q^l(x^l)\,\rangle + \langle c^l, x^l\rangle \\ \text{s.t.} & \mathcal{A}^s(x^s) + A^l x^l = b, \\ & x^s + \mathcal{B}^l(x^l) \succeq d^s, \end{array} \tag{5.7}$$

where $x^s \in \mathcal{S}^n$, $x^l \in \Re^l$, $c^s \in \mathcal{S}^n$, $c^l \in \Re^l$, Q^l is a positive semidefinite matrix in $\mathcal{S}^l$, $\mathcal{A}^s$ is a linear operator from $\mathcal{S}^n$ to $\Re^m$, A^l is a matrix in $\Re^{m\times l}$, $\mathcal{B}^l$ is a linear operator from $\Re^l$ to $\mathcal{S}^n$, and $d^s \in \mathcal{S}^n$.

In Table 5.1, we list the results obtained by the NAL algorithm for the convex quadratic problem (5.7) with density$= 0.15$. The first six columns give the size of problem (5.7), *mat* and *vec* denote the dimensions of the matrix and vector variables, m is the number of equality constraint, c_s and c_l are the sizes of SDP and linear cone constraints respectively. The middle five columns give the number of outer iteration for solving (P), the total number of inner iterations for solving (3.37), the average number of PCG steps taken to solve (3.52), the objective values pobj and dobj defined in (5.4) and (5.5), respectively. The relative infeasibilities, gap and times are listed in the last four columns.

The results of random QSDPs in Table 5.1 show that the NAL algorithm can solve the QSDPs very efficiently when the dimensions of matrix variables and SDP cone constraints are more than 500.

Table 5.1: Results for the NAL algorithm on computing QSDP problems.

problem	mat;vec	m | $c_s; c_q; c_l$	it| itsub| pcg	pobj	dobj	R_P| R_D| gap	time
Rn2s1l1v12	100; 200	200 | 100; ; 100	14 | 18.9 | 1.4	-8.83290439 4	-8.83260553 4	6.2-7 | 1.2-7 | -1.7-5	2:59
Rn3s1l5v12	100; 200	300 | 100; ; 500	20 | 19.5 | 1.5	-5.34735658 4	-5.34629213 4	8.4-7 | 3.5-7 | -10.0-5	8:39
Rn6s1l5v12	100; 200	600 | 100; ; 500	20 | 22.9 | 1.4	-7.95663787 4	-7.95548555 4	9.1-7 | 1.2-7 | -7.2-5	11:55
Rn2s1l1v15	100; 500	200 | 100; ; 100	13 | 21.1 | 1.5	-2.42008374 5	-2.42004086 5	7.3-7 | 2.0-7 | -8.9-6	5:18
Rn3s1l5v15	100; 500	300 | 100; ; 500	18 | 20.2 | 1.4	-1.30786630 5	-1.30781300 5	5.1-7 | 5.6-7 | -2.0-5	11:14
Rn6s1l5v15	100; 500	600 | 100; ; 500	19 | 24.3 | 1.4	-2.50565465 5	-2.50556185 5	6.8-7 | 3.5-7 | -1.9-5	14:38
Rn1s1l2v18	100; 800	100 | 100; ; 200	11 | 25.2 | 2.3	-7.05008394 5	-7.05007354 5	3.9-7 | 1.2-7 | -7.4-7	7:55
Rn1s1l7v18	100; 800	100 | 100; ; 700	17 | 22.5 | 1.8	-6.56403601 5	-6.56399043 5	8.8-7 | 2.3-7 | -3.5-6	13:43
Rn6s1l2v18	100; 800	600 | 100; ; 200	16 | 26.8 | 1.4	-5.65895744 5	-5.65891378 5	8.8-7 | 2.4-7 | -3.9-6	14:03
Rn6s1l7v18	100; 800	600 | 100; ; 700	18 | 28.7 | 1.8	-4.74074777 5	-4.74068679 5	8.9-7 | 5.2-7 | -6.4-6	25:20
Rn6s1l1v19	100; 900	600 | 100; ; 100	11 | 29.3 | 1.7	-8.61679384 5	-8.61674448 5	5.4-7 | 2.2-7 | -2.9-6	11:24
Rn6s1l3v19	100; 900	600 | 100; ; 300	17 | 23.2 | 1.5	-9.47479388 5	-9.47471060 5	7.1-7 | 2.0-7 | -4.4-6	14:47
Rn6s1l5v19	100; 900	600 | 100; ; 500	18 | 24.2 | 1.4	-7.59621178 5	-7.59614101 5	6.1-7 | 4.5-7 | -4.7-6	18:06
Rn1s2l2v22	200; 200	100 | 200; ; 200	24 | 29.0 | 1.1	-7.13153150 5	-7.13141535 5	6.3-7 | 1.7-7 | -8.1-6	11:27
Rn1s2l7v22	200; 200	100 | 200; ; 700	26 | 23.3 | 1.3	-7.34900174 5	-7.34879089 5	7.0-7 | 7.2-7 | -1.4-5	32:36
Rn2s4l1v42	400; 200	200 | 400; ; 100	23 | 27.1 | 1.0	-5.29390040 6	-5.29382143 6	4.1-7 | 7.5-8 | -7.5-6	38:02
Rn2s4l3v42	400; 200	200 | 400; ; 300	24 | 25.5 | 1.1	-5.62075367 6	-5.62060959 6	5.0-7 | 3.0-7 | -1.3-5	1:06:55

5.1.2 Random convex QSOCP problems

We consider the following convex quadratic second-order cone programming (QSOCP)

$$\begin{array}{ll} \min & \frac{1}{2}\langle x,\, \mathcal{Q}x\rangle + \langle q,\, x\rangle \\ \text{s.t.} & \|\mathcal{A}(x) + b\| \leq \langle c,\, x\rangle + d, \end{array} \tag{5.8}$$

where $\mathcal{Q} : \mathbf{X} \to \mathbf{X}$ is a self-adjoint linear mapping, $\mathcal{A} : \mathbf{X} \to \Re^{n-1}$ is a linear mapping, $q \in \mathbf{X}$, $b \in \Re^{n-1}$ and $d \in \Re$. Thus the inequality constraint in $(SOCP)$ can be written as an affine mapping:

$$\|\mathcal{A}(x) + b\| \leq \langle c,\, x\rangle + d \quad \Leftrightarrow \quad \begin{bmatrix} c^T \\ \mathcal{A} \end{bmatrix} x + \begin{bmatrix} d \\ b \end{bmatrix} \in \mathbf{K}^n,$$

where $\mathbf{K}^n$ denotes the second-order cone of dimension n defined in (1.6). Thus the problem (5.8) can be written as

$$(QSOCP) \qquad \min \ \tfrac{1}{2}\langle x,\, \mathcal{Q}x\rangle + \langle q,\, x\rangle \quad \text{s.t.} \ \widehat{\mathcal{A}}(x) + \hat{b} \in \mathbf{K}^n,$$

where $\widehat{\mathcal{A}} := (c^T; \mathcal{A})$ and $\hat{b} := (d; b)$.

We apply the NAL algorithm to solve the cases of problem (QSOCP). For the first collection of QSOCPs, the operator $\mathcal{Q}$ is the identity mapping if the variable is a matrix and $\mathcal{Q}$ is a random symmetric and positive semidefinite matrix if the variable is a vector with density= 0.25. In table 5.2, c_q is the size of the second-order cone constraint. In the first part of Table 5.2, since there is only one second-order cone constraint in (QSOCP), the NAL algorithm can solve the size of (QSOCP) up to thousands in few minutes. When problem (QSCOP) only has vector variables, it can be solved by the NAL algorithm for very large scale problems shown in the second part of Table 5.2.

Table 5.2: Results for the NAL algorithm on computing QSOCP problems.

problem	mat;vec	m | $c_s; c_q; c_l$	it| itsub| pcg	pobj	dobj	R_P| R_D| gap	time
QSOCP1vs1vl1	100; 100	; | ; 100;	3 | 18.5 | 1.0	-5.12424502 2	-5.12407674 2	4.5-7 | 9.7-7 | -1.6-5	1:26
QSOCP1vs1vl2	100; 200	; | ; 100;	3 | 27.8 | 1.0	3.28292970 3	3.28292794 3	3.2-7 | 8.7-7 | 2.7-7	1:11
QSOCP1vs1vl3	100; 300	; | ; 100;	4 | 40.9 | 1.0	-7.57713359 3	-7.57711360 3	6.3-8 | 5.9-7 | -1.3-6	1:31
QSOCP1vs2vl1	200; 100	; | ; 100;	3 | 30.8 | 1.0	8.03980236 3	8.03986823 3	5.2-8 | 9.7-7 | -4.1-6	3:33
QSOCP1vs2vl2	200; 200	; | ; 100;	4 | 55.8 | 1.0	-7.35128409 4	-7.35127964 4	7.7-8 | 9.6-7 | -3.0-7	7:00
QSOCP1vs2vl3	200; 300	; | ; 100;	5 | 78.5 | 1.0	-1.06081961 5	-1.06082030 5	5.2-8 | 8.1-7 | 3.3-7	6:55
QSOCP3vs1vl1	100; 100	; | ; 300;	3 | 19.9 | 1.0	6.85059278 4	6.85060780 4	3.1-7 | 9.1-7 | -1.1-6	2:07
QSOCP3vs1vl2	100; 200	; | ; 300;	3 | 21.3 | 1.0	1.51573934 4	1.51574899 4	4.3-8 | 9.5-7 | -3.2-6	2:26
QSOCP3vs1vl3	100; 300	; | ; 300;	3 | 28.5 | 1.0	2.60034208 3	2.60035664 3	2.2-7 | 8.8-7 | -2.8-6	2:12
QSOCP3vs2vl1	200; 100	; | ; 300;	3 | 34.9 | 1.0	2.32625214 5	2.32625706 5	5.5-8 | 9.5-7 | -1.1-6	12:18
QSOCP3vs2vl2	200; 200	; | ; 300;	3 | 51.5 | 1.0	1.11186116 5	1.11186335 5	2.1-7 | 7.8-7 | -9.9-7	13:24
QSOCP3vs2vl3	200; 300	; | ; 300;	3 | 54.9 | 1.0	2.35319950 5	2.35320543 5	3.1-7 | 8.9-7 | -1.3-6	15:55
QSOCP5vs1vl1	100; 100	; | ; 500;	3 | 18.5 | 1.0	9.49032754 4	9.49034524 4	2.8-7 | 8.8-7 | -9.3-7	3:16
QSOCP5vs1vl2	100; 200	; | ; 500;	3 | 21.3 | 1.0	1.22792556 5	1.22792725 5	1.7-7 | 8.1-7 | -6.9-7	3:42
QSOCP5vs1vl3	100; 300	; | ; 500;	3 | 27.3 | 1.0	5.94365141 4	5.94367085 4	3.2-7 | 8.7-7 | -1.6-6	3:48
QSOCP5vs2vl1	200; 100	; | ; 500;	3 | 27.8 | 1.0	3.23950750 5	3.23951420 5	9.5-8 | 9.4-7 | -1.0-6	18:00
QSOCP5vs2vl2	200; 200	; | ; 500;	3 | 41.6 | 1.0	4.44822160 5	4.44823116 5	2.2-7 | 8.2-7 | -1.1-6	20:31
QSOCP5vs2vl3	200; 300	; | ; 500;	3 | 48.3 | 1.0	3.64821917 5	3.64822759 5	1.5-7 | 9.1-7 | -1.2-6	21:29
QSOCP4vl1	; 100	; | ; 400;	5 | 7.0 | 1.0	-9.60442211 2	-9.60441343 2	8.0-7 | 8.9-7 | -4.5-7	04
QSOCP7vl5	; 500	; | ; 700;	4 | 8.5 | 1.0	-2.44479022 4	-2.44478973 4	6.0-7 | 8.3-7 | -10.0-8	07
QSOCP12vl10	; 1000	; | ; 1200;	4 | 9.0 | 1.0	-1.00125195 5	-1.00125201 5	6.5-7 | 7.0-7 | 3.3-8	17

Table 5.2: Results for the NAL algorithm on computing QSOCP problems.

problem	mat;vec	$m \mid c_s; c_q; c_l$	it| itsub| pcg	pobj	dobj	R_P| R_D| gap	time
QSOCP12vl15	; 1500	; | ; 1200;	4 | 11.7 | 1.0	-2.31199284 5	-2.31199285 5	1.6-7 | 3.9-7 | 2.7-9	46
QSOCP12vl30	; 3000	; | ; 1200;	3 | 17.7 | 1.0	-9.83541646 5	-9.83541626 5	6.5-7 | 9.7-7 | -1.0-8	6:35
QSOCP20vl10	; 1000	; | ; 2000;	5 | 7.5 | 1.0	-1.17194660 5	-1.17194655 5	2.5-7 | 3.6-7 | -2.3-8	22
QSOCP20vl15	; 1500	; | ; 2000;	4 | 8.8 | 1.0	-2.49417551 5	-2.49417553 5	6.5-7 | 6.4-7 | 2.9-9	38
QSOCP20vl30	; 3000	; | ; 2000;	3 | 12.7 | 1.0	-1.03120005 6	-1.03120010 6	4.9-7 | 9.1-7 | 2.4-8	3:19
QSOCP60vl10	; 1000	; | ; 6000;	6 | 7.0 | 1.0	-1.03135676 5	-1.03135633 5	9.3-7 | 9.9-7 | -2.1-7	47
QSOCP60vl15	; 1500	; | ; 6000;	6 | 7.0 | 1.0	-2.43957158 5	-2.43957141 5	2.8-7 | 3.8-7 | -3.5-8	1:18
QSOCP60vl30	; 3000	; | ; 6000;	5 | 7.5 | 1.0	-1.00314575 6	-1.00314576 6	2.1-7 | 3.1-7 | 1.8-9	3:13
QSOCP80vl30	; 3000	; | ; 8000;	5 | 7.0 | 1.0	-1.00934330 6	-1.00934328 6	5.1-7 | 4.6-7 | -1.2-8	3:36
QSOCP100vl30	; 3000	; | ; 10000;	5 | 7.0 | 1.0	-1.00790624 6	-1.00790614 6	8.4-7 | 8.8-7 | -5.0-8	4:13
QSOCP120vl30	; 3000	; | ; 12000;	6 | 7.0 | 1.0	-1.01767241 6	-1.01767240 6	3.0-7 | 4.0-7 | -3.7-9	5:18

5.2 Nearest correlation matrix problems

A correlation matrix G is a symmetric positive semidefinite matrix with unit diagonal and off-diagonal elements between -1 and 1. Such matrices have many applications, particularly in marketing and financial economics. Unfortunately, due either to paucity of data/information or dynamic nature of the problem at hand, it is not possible to obtain a complete correlation matrix. Some elements of G are unknown. To obtain a valid nearest correlation matrix (NCM) from an incomplete correlation matrix, Higham [51] considered the following problem

$$\begin{array}{lll} (NCM) & \min & \frac{1}{2}\|X - G\|^2 \\ & \text{s.t.} & \operatorname{diag}(X) = e, \\ & & X \in \mathcal{S}^n_+. \end{array}$$

where $e \in \Re^m$ is the vector of all ones. The norm, in the (NCM) problem, can take these forms as follows,

(i) Frobenius norm : $\|A\|_F = \sqrt{\operatorname{trace}(A^T A)}$.

(ii) W-weighted norm: $\|A\|_W = \|W^{1/2} A W^{1/2}\|_F$, where $W \in \mathcal{S}^n_+$.

(iii) H-weighted norm: $\|A\|_H = \|H \circ A\|_F$, where $H = (H_{ij}) \in \mathcal{S}^n$ with $H_{ij} \geq 0$.

To evaluate the performance of our NAL algorithm for solving the (NCM) problem, we set the data matrix as follows,

$$G := (1-\alpha)B + \alpha E$$

where $\alpha \in (0,1)$ (eg. $\alpha = 0.1$), E is a random symmetric matrix with entries in $[-1,1]$, generated by `E = 2*rand(n)-1; E = triu(E) + triu(E,1)'`, and B is generated in the following examples:

- **Randcorr** generates a random correlation matrix with with specified eigenvalues by the MATLAB segment

```
xx = 10.^{(4*[-1:1/(n-1):0])}; B = gallery('randcorr',n*xx/sum(xx).
```

- **Randcorr2** is similar to Example Randcorr except fixing some eigenvalues are zeros,

```
n2 = n/2; xx = [10^(-4/(n-1)).^[0:n2-1], zeros(1,n2)];
xx = n*xx/sum(xx);  B = gallery('randcorr',xx).
```

- **AR1** and **CompSym** can generate two type correlation matrices based on the AR(1) model and compound symmetry model given by [52].

- **NCM387-riskmetric** is the 387×387 1-day correlation matrix (as of June 15, 2006) from the lagged datasets of RiskMetrics (www.riskmetrics.com/stddownload edu.html).

For solving the (NCM) problem, we consider the following classes of test problems:

NCMI. The standard version: The linear operator $\mathcal{Q}$ is chosen to be the identity operator in $\mathcal{S}^n$, i.e., $\mathcal{Q}(X) = X$.

NCMH. The H-weighted version: The linear operator $\mathcal{Q}$ is chosen to be $\mathcal{Q}(X) = H \circ X$, where H can be randomly generated as follows. First generate a random symmetric matrix $H0$ whose elements are picked from the uniform distribution in $[0.1, 10]$. Then for a given $p \in (0, 1)$, we randomly set approximately n^2p elements of $H0$ to 100 and another n^2p elements to 0.01 to simulate the situation where some of the elements in G are fixed and some others are unrestricted. The resulting matrix is chosen to be the weight matrix H. In our experiments, we set $p = 0.01$ or 0.2.

NCMW. The W-weighted version: The linear operator $\mathcal{Q}$ is chosen to be $\mathcal{Q}(X) = WXW$, where $W \in \mathcal{S}^n$ is symmetric and positive definite. We set the data of the matrix W from the choice of the weighted matrix W in [43].

Table 5.3: Results for the NAL algorithm on computing NCMI problems.

problem	mat;vec	m | $c_s; c_q; c_l$	it| itsub| pcg	pobj	dobj	R_P| R_D| gap	time
NCM100-AR1	100;	100 | 100; ;	5 | 7.0 | 1.0	2.23088254 0	2.23089226 0	5.5-7 | 3.8-8 | -1.8-6	01
NCM100-CompSym	100;	100 | 100; ;	4 | 7.0 | 1.0	2.69812614-2	2.69804229-2	7.3-7 | 2.6-9 | 8.0-7	01
NCM100-Randcorr	100;	100 | 100; ;	6 | 7.0 | 1.0	3.49175258 0	3.49175996 0	3.1-7 | 1.9-8 | -9.2-7	01
NCM100-Randcorr	100;	100 | 100; ;	6 | 7.0 | 1.0	3.72839570 0	3.72840342 0	3.0-7 | 1.8-8 | -9.1-7	01
NCM500-AR1	500;	500 | 500; ;	7 | 7.0 | 1.0	1.36256510 2	1.36256635 2	2.8-7 | 9.3-9 | -4.6-7	29
NCM500-CompSym	500;	500 | 500; ;	5 | 7.0 | 1.0	4.88027295 1	4.88028757 1	6.4-7 | 2.0-8 | -1.5-6	18
NCM500-Randcorr	500;	500 | 500; ;	7 | 7.0 | 1.0	1.53391453 2	1.53391596 2	3.0-7 | 1.1-8 | -4.7-7	24
NCM500-Randcorr	500;	500 | 500; ;	7 | 7.0 | 1.0	1.56555429 2	1.56555593 2	3.4-7 | 1.2-8 | -5.2-7	29
NCM1000-AR1	1000;	1000 | 1000; ;	7 | 7.0 | 1.0	6.89909276 2	6.89910324 2	6.9-7 | 1.5-8 | -7.6-7	2:34
NCM1000-CompSym	1000;	1000 | 1000; ;	6 | 7.0 | 1.0	3.80780160 2	3.80780776 2	6.0-7 | 1.3-8 | -8.1-7	2:07
NCM1000-Randcor	1000;	1000 | 1000; ;	7 | 7.0 | 1.0	7.31114578 2	7.31115604 2	6.5-7 | 1.5-8 | -7.0-7	2:33
NCM1000-Randcor	1000;	1000 | 1000; ;	7 | 7.0 | 1.0	7.36882232 2	7.36883271 2	6.6-7 | 1.6-8 | -7.0-7	2:33
NCM387-riskmetr	387;	387 | 387; ;	14 | 7.0 | 1.0	1.47285239 2	1.47285313 2	5.6-7 | 4.9-8 | -2.5-7	25

Since $\mathcal{Q}$ is a strictly positive operator, the performance of results of NAL algorithm on the NCM problems is so fast that the algorithm almost have the quadratic convergence in Table 5.3. Even when the dimensions of variables and constraints is up to 1000, the NCM problems can be solved only in few minutes.

Table 5.4: Results for the NAL algorithm on computing H-norm NCM problems with $p = 0.01$.

problem	mat;vec	m | $c_s; c_q; c_l$	it| itsub| pcg	pobj	dobj	R_P| R_D| gap	time
NCM100-AR1	100;	100 | 100; ;	13 | 36.1 | 1.0	2.64722049 1	2.64742294 1	6.7-7 | 2.5-7 | -3.8-5	11
NCM100-CompSym	100;	100 | 100; ;	14 | 76.8 | 1.1	1.91516041-2	2.09145830-2	8.9-7 | 3.0-7 | -1.7-3	26
NCM100-Randcorr	100;	100 | 100; ;	19 | 27.4 | 1.0	4.62362677 1	4.62391162 1	5.2-7 | 1.6-7 | -3.0-5	12
NCM100-Randcorr	100;	100 | 100; ;	13 | 36.1 | 1.2	4.72084720 1	4.72107928 1	9.5-7 | 5.5-7 | -2.4-5	14
NCM500-AR1	500;	500 | 500; ;	13 | 11.7 | 1.6	2.61602187 3	2.61602860 3	8.3-7 | 5.1-8 | -1.3-6	1:38
NCM500-CompSym	500;	500 | 500; ;	14 | 9.5 | 1.2	5.74879548 2	5.74876282 2	8.6-7 | 1.6-7 | 2.8-6	1:11
NCM500-Randcorr	500;	500 | 500; ;	12 | 8.0 | 1.1	3.01473766 3	3.01473942 3	5.1-7 | 2.1-8 | -2.9-7	1:02
NCM500-Randcorr	500;	500 | 500; ;	15 | 9.3 | 1.2	3.03014635 3	3.03015478 3	5.3-7 | 8.1-8 | -1.4-6	1:24
NCM1000-AR1	1000;	1000 | 1000; ;	18 | 9.0 | 1.1	1.49035680 4	1.49035889 4	7.4-7 | 1.0-7 | -7.0-7	6:50
NCM1000-CompSym	1000;	1000 | 1000; ;	16 | 8.6 | 1.1	6.55333905 3	6.55337763 3	9.4-7 | 2.5-7 | -2.9-6	6:09
NCM1000-Randcor	1000;	1000 | 1000; ;	19 | 8.7 | 1.3	1.58093259 4	1.58093449 4	6.2-7 | 9.1-8 | -6.0-7	7:38
NCM1000-Randcor	1000;	1000 | 1000; ;	18 | 10.1 | 1.2	1.61944132 4	1.61944336 4	4.4-7 | 1.0-7 | -6.3-7	8:03
NCM387-riskmetr	387;	387 | 387; ;	15 | 11.8 | 1.4	4.03067619 3	4.03067559 3	5.2-7 | 1.9-7 | 7.5-8	1:07

Table 5.5: Results for the NAL algorithm on computing H-norm NCM problems with $p = 0.2$.

problem	mat;vec	m | $c_s; c_q; c_l$	it| itsub| pcg	pobj	dobj	R_P| R_D| gap	time
NCM100-AR1	100;	100 | 100; ;	19 | 35.6 | 1.2	1.63274825 1	1.63571056 1	9.4-7 | 4.1-7 | -8.8-4	19
NCM100-CompSym	100;	100 | 100; ;	19 | 55.1 | 1.1	4.11793221-4	2.12560713-3	7.1-7 | 4.8-7 | -1.7-3	19
NCM100-Randcorr	100;	100 | 100; ;	19 | 30.0 | 1.0	3.09043223 1	3.09057409 1	4.2-7 | 7.0-8 | -2.3-5	15
NCM100-Randcorr	100;	100 | 100; ;	17 | 28.5 | 1.1	4.94611854 1	4.94638398 1	8.5-7 | 8.3-8 | -2.7-5	14
NCM500-AR1	500;	500 | 500; ;	13 | 13.0 | 1.4	2.93985274 3	2.93986063 3	7.9-7 | 1.3-7 | -1.3-6	2:08
NCM500-CompSym	500;	500 | 500; ;	11 | 17.0 | 1.4	3.72680567 2	3.72678346 2	9.6-7 | 1.0-7 | 3.0-6	2:47
NCM500-Randcorr	500;	500 | 500; ;	16 | 10.9 | 1.3	3.33870309 3	3.33872491 3	4.7-7 | 8.8-8 | -3.3-6	1:53
NCM500-Randcorr	500;	500 | 500; ;	16 | 9.7 | 1.3	3.44300435 3	3.44303791 3	4.9-7 | 1.4-7 | -4.9-6	1:44
NCM1000-AR1	1000;	1000 | 1000; ;	15 | 10.6 | 1.2	2.00960303 4	2.00960829 4	6.2-7 | 6.8-8 | -1.3-6	9:04
NCM1000-CompSym	1000;	1000 | 1000; ;	16 | 11.4 | 1.3	6.68421785 3	6.68426315 3	7.5-7 | 9.1-8 | -3.4-6	10:28
NCM1000-Randcor	1000;	1000 | 1000; ;	17 | 9.0 | 1.1	2.10044492 4	2.10045529 4	4.3-7 | 1.5-7 | -2.5-6	8:07
NCM1000-Randcor	1000;	1000 | 1000; ;	17 | 9.5 | 1.3	2.18234308 4	2.18234829 4	6.0-7 | 6.9-8 | -1.2-6	9:01
NCM387-riskmetr	387;	387 | 387; ;	21 | 14.5 | 1.2	1.07255923 4	1.07254880 4	7.3-7 | 1.7-7 | 4.9-6	1:41

Tables 5.4 and 5.5, list the results obtained by the QSDP-GAL algorithm for the H-norm NCM problems with $p = 0.01$ and $p = 0.2$ respectively. The results in Tables 5.4 and 5.5 show that the relative gaps can be reduced very small as the desired accuracy of

infeasibilities for the H-norm NCM problems.

Table 5.6: Results for the NAL algorithm on computing W-norm NCM problems.

problem	mat;vec	m | c_s; c_q; c_l	it| itsub| pcg	pobj	dobj	R_P| R_D| gap	time
NCM100-AR1	100;	100 | 100; ;	8 | 11.6 | 1.2	4.01406632-2	4.01515902-2	9.9-7 | 1.0-7 | -1.0-5	04
NCM100-CompSym	100;	100 | 100; ;	17 | 10.7 | 1.0	9.43927680-3	9.44175256-3	4.3-7 | 8.3-8 | -2.4-6	04
NCM100-Randcorr	100;	100 | 100; ;	9 | 10.9 | 1.1	6.25287195-2	6.25273672-2	6.2-7 | 4.6-8 | 1.2-6	04
NCM100-Randcorr	100;	100 | 100; ;	13 | 11.0 | 1.2	7.64308050-2	7.64365845-2	8.8-7 | 9.6-8 | -5.0-6	05
NCM500-AR1	500;	500 | 500; ;	9 | 12.7 | 1.8	1.94451939 0	1.94461715 0	8.4-7 | 9.5-8 | -2.0-5	3:17
NCM500-CompSym	500;	500 | 500; ;	14 | 18.5 | 1.8	9.86698773-1	9.86833418-1	7.5-7 | 1.4-7 | -4.5-5	5:01
NCM500-Randcorr	500;	500 | 500; ;	6 | 14.0 | 1.9	2.03838061 0	2.03862609 0	8.6-7 | 4.7-7 | -4.8-5	5:09
NCM500-Randcorr	500;	500 | 500; ;	6 | 13.5 | 1.9	2.12769503 0	2.12780030 0	2.8-7 | 1.6-7 | -2.0-5	5:18
NCM1000-AR1	1000;	1000 | 1000; ;	23 | 26.5 | 1.4	1.16460867 1	1.16469106 1	7.6-7 | 1.8-7 | -3.4-5	1:29
NCM1000-CompSym	1000;	1000 | 1000; ;	19 | 17.1 | 1.4	5.78548918 0	5.78601682 0	6.6-7 | 2.1-7 | -4.2-5	33:36
NCM1000-Randcor	1000;	1000 | 1000; ;	7 | 12.2 | 2.3	9.11968156 0	9.12194201 0	7.2-7 | 3.4-7 | -1.2-4	33:20
NCM1000-Randcor	1000;	1000 | 1000; ;	9 | 12.0 | 1.9	9.08071778 0	9.08724498 0	4.7-7 | 2.1-7 | -3.4-4	35:52
NCM387-riskmetr	387;	387 | 387; ;	89 | 12.0 | 3.4	7.68457793 0	3.59744521 1	7.7-7 | 1.9-5 | -6.3-1	34:03

In Table 5.6, the relative gap of "NCM387-riskmetric" is not accurate enough although the infeasibilies of primal and dual problems are reached. It is because the real data of market is much worse than the random generated matrices and the operator $\mathcal{Q}$ for the W-norm cases is only positive semidefinite. To overcome this, we can reduce the tolerance and run more iterations.

5.3 Euclidean distance matrix problems

In recent years, the Euclidean distance matrix (EDM) completion problems have received a lot of attention for their many important applications, such as molecular conformation problems in chemistry [85] and multidimensional scaling and multivariate analysis problems in statistics [65]. This section will apply the NAL algorithm to solve (EDM) completion problems for finding a weighted, closest Euclidean distance matrix.

Let $B = (B_{ij})$ be a dissimilarity matrix, defined in (1.2), with nonnegative elements and zero diagonal in $\mathcal{S}^n$. Define the $n \times n$ orthogonal matrix

$$Q := \left(\frac{1}{\sqrt{n}} e \mid V \right), \quad Q^T Q = I. \tag{5.9}$$

where e is the vector of all ones. Thus $V^T e = 0$ and $V^T V = I$, for $V \in \Re^{n\times(n-1)}$. We introduce the linear operators on $\mathcal{S}^n$

$$\theta(B) = \operatorname{diag}(B)\, e^T + e \operatorname{diag}(B)^T - 2(B). \tag{5.10}$$

Then B is a EDM if and only if $B = \theta(X)$ with $Be = 0$ and $B \in \mathcal{S}^n_+$. Let $X = V^T B V$, then since $Be = 0$ we have $B = VXV^T$. Therefore, $VXV^T \in \mathcal{S}^n_+$ if and only if $X \in \mathcal{S}^n_+$.

We consider the following approximate Euclidean distance matrix completion problem from [1]

$$\begin{array}{llll} (EDM) & \min & \frac{1}{2}\|H \circ (B - \theta(X_V))\|_F^2 \\ & \text{s.t.} & (\theta(X_V))_{ij} = B_{ij} \quad \forall\, (i,j) \in \mathcal{E}, \\ & & X \in \mathcal{S}^{n-1}_+, \end{array}$$

where $\circ$ denotes *Hadamard product*, $X_V := VXV^T$ and V is defined in (5.9), B is a dissimilarity matrix, $H \in \mathcal{S}^n$ is a weight matrix with nonnegative elements which typically has the same sparsity pattern as B, $\mathcal{E}$ is a given set of indices. Note that the operators $\mathcal{Q}$ for the (EDM) are positive semidefinite, but not positive definite.

To implement the NAL algorithm, we consider the following classes of QSDPs arising from the (EDM) problem:

EDM1. For the random EDM problems, we can generate n random points, $x^1, x^2, \ldots, x^n$, in the unit cube centered at the origin in $\Re^3$. For a certain cut-off distance R, we then set the dissimilarity matrix B defined in (1.2) as follows,

$$B_{ij} = \begin{cases} \|x^i - x^j\| & \text{if } \|x^i - x^j\| \le R, \\ 0 & \text{otherwise.} \end{cases} \tag{5.11}$$

The non-negative weight matrix H is chosen to be the 0-1 matrix having the same sparsity pattern as B. The set of indices where the distances are fixed is given by $\mathcal{E} = \{(1,j) \mid B_{1j} = 0, j = 1,\ldots,n\}$. We generated four test problems with $n = 50, 100, 200, 400$, and their corresponding dissimilarity matrices B have $16.6\%, 8.6\%, 4.5\%$, and 2.4% nonzero elements, respectively. Note that in the actual QSDP test problems, we added the term $-0.01\langle I, X\rangle$ to the objective function in (EDM) so as to induce a low-rank primal optimal solution.

EDM2. Same as EDM1 but the points are chosen to be the coordinates of the atoms in the following protein molecules, 1PTQ, 1HOE, 1LFB, 1PHT, 1POA, 1AX8, taken from the Protein Data Bank [9]. These six test problems have dimension $n = 401$, 557, 640, 813 and the densities of nonzeros in B are 17.4%, 13.1%, 11.0%, 8.3%, respectively.

Table 5.7: Results for the NAL algorithm on computing EDM problems.

problem	mat;vec	$m \mid c_s; c_q; c_l$	it| itsub| pcg	pobj	dobj	R_P| R_D| gap	time
EDM50	50;	10 | 50; ;	7 | 66.4 | 2.0	6.34541629-2	6.37973333-2	5.3-5 | 8.5-6 | -3.0-4	25
EDM100	100;	12 | 100; ;	7 | 65.6 | 2.2	6.64391571-2	6.88303537-2	6.0-5 | 3.8-5 | -2.1-3	37
EDM200	200;	12 | 200; ;	7 | 70.5 | 2.2	6.63887673-2	6.92644444-2	8.5-5 | 5.4-5 | -2.5-3	1:37
EDM400	400;	15 | 400; ;	12 | 58.4 | 1.9	4.77518288-2	5.00359015-2	7.6-5 | 7.3-5 | -2.1-3	14:48
pdb1PTQ	402;	70 | 402; ;	17 | 61.0 | 1.3	1.09211217 0	1.12228861 0	8.6-5 | 4.6-5 | -9.4-3	18:28
pdb1HOE	558;	42 | 558; ;	16 | 58.6 | 1.4	5.52267196-1	5.67573125-1	9.1-5 | 5.1-5 | -7.2-3	30:24
pdb1LFB	641;	29 | 641; ;	13 | 54.6 | 1.4	1.54148929-1	1.59580279-1	8.4-5 | 6.1-5 | -4.1-3	32:36
pdb1PHT	814;	51 | 814; ;	16 | 57.4 | 1.4	5.26736360-1	5.41945839-1	9.9-5 | 4.7-5 | -7.4-3	1:20:00

Although the optimal solutions to the EDM problems may not satisfy the primal constraint nondegeneracy (3.25) or the condition (3.43), the problems listed in Table 5.7 can be solved by the NAL algorithm with a moderate tolerance 10^{-4}. For the high accuracy of EDM problems, we have to choose a good preconditioner of the generalized Hessian matrix defined in (3.39) to improve the performance of the SNCG algorithm for inner problems.

Chapter 6

Numerical results for linear SDPs

As a special case of QSDPs, the NAL algorithm can also be applied to solve the linear programming over symmetric cones. Because of the explicit form of the problem (LD), the NAL algorithm can solve the dual problems more efficient than the primal problems. Under the same conditions (5.1)−(5.6) of convex quadratic programming in Chapter 5, we will mainly compare the performance of the NAL algorithm with the related algorithms in this section.

6.1 SDP relaxations of frequency assignment problems

Due to the fast implementation of wireless telephone networks and satellite communication projects, the literature on frequency assignment problems (FAP)has grown quickly over the past years. Since the frequency assignment problem is a NP-complete problem, we consider the semidefinite relaxation of frequency assignment problems [36]. Given a network represented by a graph G, a certain type of frequency assignment problem on G can be relaxed into the following SDP:

$$\begin{aligned} \max \quad & \langle C, X \rangle \\ \text{s.t.} \quad & \operatorname{diag}(X) = e, \ \mathcal{A}(X) = b, \\ & \mathcal{B}(X) \leq h, \quad X \in \mathcal{S}^n_+, \end{aligned} \tag{6.1}$$

where the data consist of the matrix $C \in \mathcal{S}^n$, $e \in \Re^n$ is the vector of all ones, $b \in \Re^m$, $h \in \Re^r$, and the linear maps $\mathcal{A} : \mathcal{S}^n \to \Re^m$ and $\mathcal{B} : \mathcal{S}^n \to \Re^r$. The dual of (6.1) has the form as

$$\begin{aligned} \min \quad & e^T y + b^T z + h^T u \\ \text{s.t.} \quad & \mathrm{Diag}(y) + \mathcal{A}^* z + \mathcal{B}^* u - C \in \mathcal{S}^n_+ \\ & y \in \Re^n, z \in \Re^m, u \in \Re^r_+, \end{aligned} \tag{6.2}$$

where $\mathcal{A}^* : \Re^m \to \mathcal{S}^n$ and $\mathcal{B}^* : \Re^r \to \mathcal{S}^n$ are the adjoints of the operators $\mathcal{A}$ and $\mathcal{B}$. We apply the NAL algorithm to solve the problem (6.2) since it only has the vector variable.

Tables 6.1 and 6.2 list the results obtained by the NAL algorithm and the boundary-point method for the SDP relaxation of frequency assignment problems tested in [21], respectively. The details of Table 6.1 and the following tables in this section are almost the same as those for Table 5.1 except that in the second column m is the dimensional of the variables.

For this collection of SDPs, the NAL algorithm outperformed the boundary-point method. While the NAL algorithm can achieve rather high accuracy in $\max\{R_P, R_D, \mathrm{gap}\}$ for all the SDPs, the boundary-point method fails to achieve satisfactory accuracy after the maximum iterations achieved in that the primal and dual objective values obtained have yet to converge close to the optimal values, see [139]. However, although FAP problems are degenerate for (LP) and (LD) problems, the NAL algorithm can achieve the required accuracy $\max\{R_P, R_D\} \leq 10^{-6}$ within moderate CPU time for all the SDPs.

For the FAP problems, previous methods such as the spectral bundle (SB) method [48], the BMZ method [21], and inexact interior-point method [121] largely fail to solve these SDPs to satisfactory accuracy within moderate computer time. For example, the SB and BMZ methods took more than 50 and 3.3 hours, respectively, to solve `fap09` on an SGI Origin2000 computer using a single 300MHz R1200 processor. The inexact interior-point method [121] took more than 2.5 hours to solve the same problem on a 700MHz HP c3700 workstation. Comparatively, our NAL algorithm took only 41 seconds to solve `fap09` to the same accuracy or better.

Table 6.1: Results for NAL Algorithm on the frequency assignment problems.

problem	m | $c_s; c_l$	it| itsub| pcg	pobj	dobj	R_P| R_D| gap	time
fap01	1378 | 52; 1160	20| 109| 33.2	3.28834503-2	3.28832952-2	8.4-7| 1.0-7| 1.5-7	06
fap02	1866 | 61; 1601	20| 81| 21.4	6.90524269-4	7.02036467-4	8.4-7| 3.5-7| -1.1-5	04
fap03	2145 | 65; 1837	20| 102| 38.6	4.93726225-2	4.93703591-2	1.2-7| 2.5-7| 2.1-6	07
fap04	3321 | 81; 3046	21| 173| 43.5	1.74829592-1	1.74844758-1	2.0-7| 6.4-7| -1.1-5	17
fap05	3570 | 84; 3263	21| 244| 56.6	3.08361964-1	3.08294715-1	7.6-6 | 6.2-7| 4.2-5	32
fap06	4371 | 93; 3997	21| 187| 55.3	4.59325368-1	4.59344513-1	7.6-7| 6.8-7| -10.0-6	27
fap07	4851 | 98; 4139	22| 179| 61.4	2.11762487 0	2.11763204 0	9.9-7| 4.9-7| -1.4-6	30
fap08	7260 | 120; 6668	21| 113| 45.0	2.43627884 0	2.43629328 0	2.8-7| 9.9-7| -2.5-6	21
fap09	15225 | 174; 14025	22| 120| 38.4	1.07978114 1	1.07978423 1	8.9-7| 9.6-7| -1.4-6	41
fap10	14479 | 183; 13754	23| 140| 57.4	9.67044948-3	9.74974306-3	1.5-7| 9.3-7| -7.8-5	1:18
fap11	24292 | 252; 23275	25| 148| 69.0	2.97000004-2	2.98373492-2	7.7-7| 6.0-7| -1.3-4	3:21
fap12	26462 | 369; 24410	25| 169| 81.3	2.73251961-1	2.73410714-1	6.0-7| 7.8-7| -1.0-4	9:07
fap25	322924 | 2118; 311044	24| 211| 84.8	1.28761356 1	1.28789892 1	3.2-6 | 5.0-7| -1.1-4	10:53:22
fap36	1154467 | 4110; 1112293	17| 197| 87.4	6.98561787 1	6.98596286 1	7.7-7| 6.7-7| -2.5-5	65:25:07

Table 6.2: Results obtained by the boundary-point method in [73] on the frequency assignment problems. The parameter σ_0 is set to 1 (better than 0.1).

problem	m | $c_s; c_l$	it	pobj	dobj	R_P| R_D| gap	time
fap01	1378 | 52; 1160	2000	3.49239684-2	3.87066984-2	5.4-6 | 1.7-4| -3.5-3	15
fap02	1866 | 61; 1601	2000	4.06570342-4	1.07844848-3	1.6-5 | 7.5-5 | -6.7-4	16
fap03	2145 | 65; 1837	2000	5.02426246-2	5.47858318-2	1.5-5 | 1.5-4| -4.1-3	17
fap04	3321 | 81; 3046	2000	1.77516830-1	1.84285835-1	4.5-6 | 1.7-4| -5.0-3	24
fap05	3570 | 84; 3263	2000	3.11422846-1	3.18992969-1	1.1-5 | 1.6-4| -4.6-3	25
fap06	4371 | 93; 3997	2000	4.60368585-1	4.64270062-1	7.5-6 | 9.8-5 | -2.0-3	27
fap07	4851 | 98; 4139	2000	2.11768050 0	2.11802220 0	2.5-6 | 1.5-5 | -6.5-5	25
fap08	7260 | 120; 6668	2000	2.43638729 0	2.43773801 0	2.6-6 | 3.5-5 | -2.3-4	34
fap09	15225 | 174; 14025	2000	1.07978251 1	1.07982902 1	9.2-7| 9.8-6 | -2.1-5	59
fap10	14479 | 183; 13754	2000	1.70252739-2	2.38972400-2	1.1-5 | 1.1-4| -6.6-3	1:25
fap11	24292 | 252; 23275	2000	4.22711513-2	5.94650102-2	8.8-6 | 1.4-4| -1.6-2	2:31
fap12	26462 | 369; 24410	2000	2.93446247-1	3.26163363-1	6.0-6 | 1.5-4| -2.0-2	4:37
fap25	322924 | 2118; 311044	2000	1.31895665 1	1.35910952 1	4.8-6 | 2.0-4| -1.4-2	8:04:00
fap36	1154467 | 4110; 1112293	2000	7.03339309 1	7.09606078 1	3.9-6 | 1.4-4| -4.4-3	46:59:28

6.2 SDP relaxations of maximum stable set problems

Let G be a simple, undirected graph with the node set V and edge set $\mathcal{E}$. The stability number $\alpha(G)$ is the cardinality of a maximal stable set of G, and

$$\alpha(G) := \{e^T y \,:\, y_i y_j = 0, (i,j) \in \mathcal{E}, y \in \{0,1\}^n\}.$$

The Lovász theta number $\theta(G)$ defined and studied by Lovász in [70] is an upper bound on the stability number $\alpha(G)$ and can be computed as the optimal value of the following SDP problem,

$$\begin{aligned} \theta(G) := \quad \max \quad & \langle ee^T, Y\rangle \\ \text{s.t.} \quad & \langle E_{ij}, Y\rangle = 0 \quad \forall\,(i,j) \in \mathcal{E}, \\ & \langle I, Y\rangle = 1, \quad Y \succeq 0, \end{aligned} \tag{6.3}$$

where $E_{ij} = e_i e_j^{\mathrm{T}} + e_j e_i^{\mathrm{T}}$ and e_i denotes column i of the identity matrix I. It is known that $\alpha(G) \leq \theta(G)_+ \leq \theta(G)$, where

$$\begin{aligned} \theta_+(G) := \quad \max \quad & \langle ee^T, Y\rangle \\ \text{s.t.} \quad & \langle E_{ij}, Y\rangle = 0 \quad \forall\,(i,j) \in \mathcal{E}, \\ & \langle I, Y\rangle = 1, \\ & Y \succeq 0,\; Y \geq 0. \end{aligned} \tag{6.4}$$

Note that the $\theta_+(G)$ problem is reformulated as a standard SDP problem by replacing the constraint $Y \geq 0$ by constraints $Y - X = 0$ and $X \geq 0$. Thus such a reformulation introduces an additional $n(n+1)/2$ linear equality constraints to the SDP problem.

Table 6.3 lists the results obtained by the NAL algorithm for the SDP problems arising from computing $\theta(G)$ for the maximum stable set problems. The first collection of graph instances in Table 6.3 are coming from the randomly generated instances considered in [121], whereas the second collection is from the Seventh DIMACS Implementation Challenge on the Maximum Clique problem [56]. The last collection are graphs arising from coding theory, available from N. Sloane's web page [111].

Observe that the NAL algorithm is not able to achieve the required accuracy level for some of the SDPs from Sloane's collection. It is not surprising that this may happen because many of these SDPs are degenerate at the optimal solution. For example, the problems `1dc.128` and `2dc.128` are degenerate at the optimal solutions $\bar{x}$ even though they are nondegenerate at the optimal solutions $\overline{Y}$.

Compared with the boundary-point method in [73], the iterative solver based primal-dual interior-point method in [121], as well as the iterative solver based modified barrier method in [60], the performance of NAL algorithm is at least as efficient as the boundary-point method on the theta problems for random graphs and much faster than the methods in [121] and [60] on a subset of the large SDPs arising from the first collection of random graphs. Note that the NAL algorithm is more efficient than the boundary-point method on the collection of graphs from DIMACS. For example, the NAL algorithm takes less than 100 seconds to solve the problem `G43` to an accuracy of less than 10^{-6}, while the boundary-point method (with $\sigma_0 = 0.1$) takes more than 3,900 seconds to achieve an accuracy of 1.5×10^{-5}. Such a result for `G43` is not surprising because the rank of the optimal X (equals to 58) is much smaller than n, and as already mentioned in [88], the boundary-point method typically would perform poorly under such a situation.

Table 6.3: Results for the NAL algorithm on computing $\theta(G)$ in (6.3) for the maximum stable set problems.

problem	$m \mid c_s; c_l$	it| itsub| pcg	pobj	dobj	R_P| R_D| gap	time
theta4	1949 | 200;	22| 25| 12.7	5.03212191 1	5.03212148 1	4.9-8| 5.2-7| 4.2-8	05
theta42	5986 | 200;	20| 24| 11.6	2.39317091 1	2.39317059 1	2.2-7| 8.5-7| 6.6-8	06
theta6	4375 | 300;	22| 29| 11.0	6.34770834 1	6.34770793 1	4.5-8| 4.8-7| 3.2-8	12
theta62	13390 | 300;	20| 25| 11.2	2.96412472 1	2.96412461 1	5.8-7| 9.2-7| 1.7-8	14
theta8	7905 | 400;	22| 28| 10.6	7.39535679 1	7.39535555 1	6.5-8| 6.9-7| 8.3-8	23
theta82	23872 | 400;	21| 26| 10.3	3.43668917 1	3.43668881 1	1.4-7| 8.8-7| 5.2-8	27
theta83	39862 | 400;	20| 27| 10.8	2.03018910 1	2.03018886 1	1.2-7| 4.8-7| 5.6-8	35
theta10	12470 | 500;	21| 25| 10.6	8.38059689 1	8.38059566 1	6.9-8| 6.6-7| 7.3-8	36
theta102	37467 | 500;	23| 28| 10.2	3.83905451 1	3.83905438 1	6.9-8| 4.8-7| 1.6-8	50
theta103	62516 | 500;	18| 27| 10.7	2.25285688 1	2.25285667 1	4.4-8| 5.8-7| 4.6-8	1:00
theta104	87245 | 500;	17| 28| 11.2	1.33361400 1	1.33361379 1	6.1-8| 6.5-7| 7.6-8	58
theta12	17979 | 600;	21| 26| 10.3	9.28016795 1	9.28016679 1	9.6-8| 8.1-7| 6.2-8	57
theta123	90020 | 600;	18| 26| 10.9	2.46686513 1	2.46686492 1	3.3-8| 5.2-7| 4.1-8	1:34
theta162	127600 | 800;	17| 26| 10.2	3.70097353 1	3.70097324 1	3.6-8| 5.4-7| 3.8-8	2:53
MANN-a27	703 | 378;	9| 13| 6.2	1.32762891 2	1.32762869 2	9.4-11| 7.0-7| 8.3-8	07
johnson8-4	561 | 70;	3| 4| 3.0	1.39999996 1	1.39999983 1	4.5-9| 1.6-7| 4.4-8	00
johnson16-	1681 | 120;	3| 4| 4.0	7.99998670 0	7.99999480 0	8.1-8| 7.5-7| -4.8-7	01
san200-0.7	5971 | 200;	13| 22| 8.9	3.00000066 1	2.99999980 1	2.3-7| 3.1-7| 1.4-7	04

Table 6.3: Results for the NAL algorithm on computing $\theta(G)$ in (6.3) for the maximum stable set problems.

problem	m | $c_s; c_l$	it| itsub| pcg	pobj	dobj	R_P| R_D| gap	time
c-fat200-1	18367 | 200;	8| 36| 20.3	1.19999983 1	1.19999962 1	1.5-7| 8.3-7| 8.5-8	09
hamming-6-	1313 | 64;	3| 4| 4.2	5.33333334 0	5.33333330 0	4.4-11| 5.8-9| 2.7-9	00
hamming-8-	11777 | 256;	5| 5| 4.0	1.59999983 1	1.59999855 1	7.2-9| 8.0-7| 3.9-7	02
hamming-9-	2305 | 512;	6| 6| 5.2	2.24000000 2	2.24000049 2	1.2-10| 2.4-7| -1.1-7	10
hamming-10	23041 | 1024;	7| 9| 5.6	1.02399780 2	1.02400070 2	7.1-8| 7.1-7| -1.4-6	1:33
hamming-7-	1793 | 128;	4| 5| 4.2	4.26666667 1	4.26666645 1	4.1-12| 6.6-8| 2.6-8	01
hamming-8-	16129 | 256;	4| 4| 4.8	2.56000007 1	2.55999960 1	2.8-9| 2.1-7| 9.0-8	02
hamming-9-	53761 | 512;	4| 6| 6.5	8.53333333 1	8.53333311 1	1.4-11| 3.9-8| 1.3-8	10
brock200-1	5067 | 200;	20| 24| 12.6	2.74566402 1	2.74566367 1	1.2-7| 6.7-7| 6.3-8	06
brock200-4	6812 | 200;	18| 23| 13.0	2.12934757 1	2.12934727 1	1.1-7| 5.8-7| 6.8-8	06
brock400-1	20078 | 400;	21| 25| 10.6	3.97018902 1	3.97018916 1	5.4-7| 9.9-7| -1.7-8	26
keller4	5101 | 171;	17| 21| 15.9	1.40122390 1	1.40122386 1	1.3-7| 4.4-7| 1.3-8	05
p-hat300-1	33918 | 300;	20| 84| 38.7	1.00679674 1	1.00679561 1	5.5-7| 9.4-7| 5.3-7	1:45
G43	9991 | 1000;	18| 27| 11.6	2.80624585 2	2.80624562 2	3.0-8| 4.6-7| 4.2-8	1:33
G44	9991 | 1000;	18| 28| 11.1	2.80583335 2	2.80583149 2	3.6-7| 9.2-7| 3.3-7	2:59
G45	9991 | 1000;	17| 26| 11.5	2.80185131 2	2.80185100 2	3.6-8| 5.8-7| 5.6-8	2:51
G46	9991 | 1000;	18| 26| 11.4	2.79837027 2	2.79836899 2	3.2-7| 9.1-7| 2.3-7	2:53
G47	9991 | 1000;	17| 27| 11.4	2.81893976 2	2.81893904 2	7.0-8| 9.3-7| 1.3-7	2:54
1dc.64	544 | 64;	22| 87| 61.1	1.00000038 1	9.99998513 0	6.9-7| 9.2-7| 8.9-7	06
1et.64	265 | 64;	13| 16| 10.0	1.87999993 1	1.88000161 1	1.2-7| 7.2-7| -4.3-7	01
1tc.64	193 | 64;	14| 25| 14.1	2.00000028 1	1.99999792 1	5.5-7| 9.2-7| 5.7-7	01
1dc.128	1472 | 128;	26| 160| 78.3	1.68422941 1	1.68420185 1	6.4-6 | 6.5-7| 7.9-6	31
1et.128	673 | 128;	14| 25| 11.5	2.92308767 1	2.92308940 1	7.6-7| 4.5-7| -2.9-7	02
1tc.128	513 | 128;	12| 33| 10.7	3.79999935 1	3.79999915 1	1.6-7| 8.5-7| 2.6-8	02
1zc.128	1121 | 128;	10| 16| 8.2	2.06666622 1	2.06666556 1	1.1-7| 5.9-7| 1.6-7	02
1dc.256	3840 | 256;	22| 131| 46.5	3.00000152 1	2.99999982 1	5.1-7| 1.1-8| 2.8-7	1:05
1et.256	1665 | 256;	22| 105| 30.5	5.51142859 1	5.51142381 1	3.2-7| 5.3-7| 4.3-7	52
1tc.256	1313 | 256;	29| 211| 82.2	6.34007911 1	6.33999101 1	7.4-6 | 4.8-7| 6.9-6	2:30
1zc.256	2817 | 256;	13| 17| 8.5	3.79999847 1	3.79999878 1	9.5-8| 4.9-7| -4.1-8	05
1dc.512	9728 | 512;	30| 181| 75.7	5.30311533 1	5.30307418 1	2.0-6 | 4.2-7| 3.8-6	12:07
1et.512	4033 | 512;	16| 90| 40.1	1.04424062 2	1.04424003 2	9.9-7| 7.9-7| 2.8-7	3:48
1tc.512	3265 | 512;	28| 316| 83.4	1.13401460 2	1.13400320 2	3.3-6 | 6.9-7| 5.0-6	28:53
2dc.512	54896 | 512;	27| 258| 61.3	1.17732077 1	1.17690636 1	2.4-5 | 5.0-7| 1.7-4	32:16
1zc.512	6913 | 512;	12| 21| 10.6	6.87499484 1	6.87499880 1	9.0-8| 3.7-7| -2.9-7	44
1dc.1024	24064 | 1024;	26| 130| 64.0	9.59854968 1	9.59849281 1	1.4-6 | 4.9-7| 2.9-6	41:26
1et.1024	9601 | 1024;	19| 117| 76.8	1.84226899 2	1.84226245 2	2.5-6 | 3.5-7| 1.8-6	1:01:14
1tc.1024	7937 | 1024;	30| 250| 79.1	2.06305257 2	2.06304344 2	1.7-6 | 6.3-7| 2.2-6	1:48:04
1zc.1024	16641 | 1024;	15| 22| 12.2	1.28666659 2	1.28666651 2	2.8-8| 3.0-7| 3.3-8	4:15
2dc.1024	169163 | 1024;	28| 219| 68.0	1.86426368 1	1.86388392 1	7.8-6 | 6.8-7| 9.9-5	2:57:56
1dc.2048	58368 | 2048;	27| 154| 82.5	1.74729647 2	1.74729135 2	7.7-7| 4.0-7| 1.5-6	6:11:11
1et.2048	22529 | 2048;	22| 138| 81.6	3.42029313 2	3.42028707 2	6.9-7| 6.3-7| 8.8-7	7:13:55
1tc.2048	18945 | 2048;	26| 227| 78.5	3.74650769 2	3.74644820 2	3.3-6 | 3.7-7| 7.9-6	9:52:09
1zc.2048	39425 | 2048;	13| 24| 14.0	2.37400485 2	2.37399909 2	1.5-7| 7.3-7| 1.2-6	45:16

Table 6.3: Results for the NAL algorithm on computing $\theta(G)$ in (6.3) for the maximum stable set problems.

problem	m | $c_s; c_l$	it| itsub| pcg	pobj	dobj	R_P| R_D| gap	time
2dc.2048	504452 | 2048;	27| 184| 67.1	3.06764717 1	3.06737001 1	3.7-6 | 4.5-7| 4.4-5	15:13:19

Table 6.4: Results for the NAL algorithm on computing $\theta_+(G)$ in (6.4) for the maximum stable set problems.

problem	$m - c_l$ | $c_s; c_l$	it| itsub| pcg	pobj	dobj	R_P| R_D| gap	time
theta4	1949 | 200; 20100	20| 67| 31.3	4.98690157 1	4.98690142 1	4.6-8| 7.9-7| 1.4-8	33
theta42	5986 | 200; 20100	18| 41| 26.0	2.37382088 1	2.37382051 1	5.7-7| 9.8-7| 7.6-8	22
theta6	4375 | 300; 45150	15| 61| 27.7	6.29618432 1	6.29618399 1	2.9-8| 7.6-7| 2.6-8	1:03
theta62	13390 | 300; 45150	16| 38| 22.4	2.93779448 1	2.93779378 1	4.0-7| 6.6-7| 1.2-7	44
theta8	7905 | 400; 80200	13| 52| 29.8	7.34078436 1	7.34078372 1	2.8-7| 7.3-7| 4.3-8	1:54
theta82	23872 | 400; 80200	13| 45| 28.6	3.40643550 1	3.40643458 1	4.0-7| 9.9-7| 1.3-7	2:09
theta83	39862 | 400; 80200	13| 40| 23.0	2.01671070 1	2.01671031 1	1.8-7| 4.5-7| 9.4-8	1:50
theta10	12470 | 500; 125250	12| 54| 32.0	8.31489963 1	8.31489897 1	1.3-7| 8.0-7| 4.0-8	3:35
theta102	37467 | 500; 125250	15| 44| 27.6	3.80662551 1	3.80662486 1	4.5-7| 9.1-7| 8.4-8	3:31
theta103	62516 | 500; 125250	12| 38| 26.5	2.23774200 1	2.23774190 1	1.0-7| 9.3-7| 2.3-8	3:28
theta104	87245 | 500; 125250	14| 35| 22.0	1.32826023 1	1.32826068 1	8.1-7| 8.4-7| -1.6-7	2:35
theta12	17979 | 600; 180300	12| 53| 33.9	9.20908140 1	9.20908772 1	6.5-7| 6.6-7| -3.4-7	5:38
theta123	90020 | 600; 180300	15| 43| 29.2	2.44951438 1	2.44951497 1	7.7-7| 8.5-7| -1.2-7	6:44
theta162	127600 | 800; 320400	14| 42| 26.2	3.67113362 1	3.67113729 1	8.1-7| 4.5-7| -4.9-7	11:24
MANN-a27	703 | 378; 71631	7| 26| 21.5	1.32762850 2	1.32762894 2	2.1-7| 6.8-7| -1.6-7	35
johnson8-4	561 | 70; 2485	5| 6| 7.0	1.39999984 1	1.40000110 1	2.2-8| 5.8-7| -4.4-7	01
johnson16-	1681 | 120; 7260	6| 7| 7.0	7.99999871 0	8.00000350 0	5.3-8| 4.3-7| -2.8-7	01
san200-0.7	5971 | 200; 20100	16| 33| 14.5	3.00000135 1	2.99999957 1	5.9-7| 4.0-7| 2.9-7	11
c-fat200-1	18367 | 200; 20100	7| 48| 42.1	1.20000008 1	1.19999955 1	1.3-7| 9.5-7| 2.1-7	36
hamming-6-	1313 | 64; 2080	6| 7| 7.0	4.00000050 0	3.99999954 0	5.7-9| 6.2-8| 1.1-7	01
hamming-8-	11777 | 256; 32896	8| 10| 7.2	1.59999978 1	1.59999873 1	8.5-9| 3.7-7| 3.2-7	05
hamming-9-	2305 | 512; 131328	3| 8| 8.4	2.24000002 2	2.24000016 2	4.6-8| 5.9-7| -3.1-8	18
hamming-10	23041 | 1024; 524800	8| 17| 10.6	8.53334723 1	8.53334002 1	6.0-8| 7.9-7| 4.2-7	4:35
hamming-7-	1793 | 128; 8256	12| 26| 8.2	3.59999930 1	3.60000023 1	3.8-8| 1.3-7| -1.3-7	03
hamming-8-	16129 | 256; 32896	6| 7| 7.0	2.56000002 1	2.56000002 1	2.0-9| 5.1-9| -2.7-10	05
hamming-9-	53761 | 512; 131328	11| 18| 10.6	5.86666682 1	5.86666986 1	1.1-7| 4.4-7| -2.6-7	42
brock200-1	5067 | 200; 20100	17| 48| 30.7	2.71967178 1	2.71967126 1	3.8-7| 7.0-7| 9.3-8	27
brock200-4	6812 | 200; 20100	18| 40| 23.4	2.11210736 1	2.11210667 1	5.4-8| 9.9-7| 1.6-7	21
brock400-1	20078 | 400; 80200	14| 42| 26.4	3.93309197 1	3.93309200 1	9.5-7| 6.5-7| -3.5-9	1:45
keller4	5101 | 171; 14706	18| 73| 43.3	1.34658980 1	1.34659082 1	6.1-7| 9.7-7| -3.7-7	43
p-hat300-1	33918 | 300; 45150	21| 123| 73.5	1.00202172 1	1.00202006 1	8.7-7| 7.2-7| 7.9-7	6:50
G43	9991 | 1000; 500500	9| 126| 52.2	2.79735847 2	2.79735963 2	9.1-7| 8.1-7| -2.1-7	52:00
G44	9991 | 1000; 500500	8| 122| 51.4	2.79746110 2	2.79746078 2	3.3-7| 6.2-7| 5.7-8	49:32
G45	9991 | 1000; 500500	9| 124| 52.0	2.79317531 2	2.79317544 2	9.3-7| 8.6-7| -2.4-8	50:25

Table 6.4: Results for the NAL algorithm on computing $\theta_+(G)$ in (6.4) for the maximum stable set problems.

problem	$m - c_l \mid c_s; c_l$	it| itsub| pcg	pobj	dobj	R_P| R_D| gap	time
G46	9991 | 1000; 500500	8| 112| 52.2	2.79032493 2	2.79032511 2	3.5-7| 9.6-7| -3.3-8	44:38
G47	9991 | 1000; 500500	9| 102| 53.1	2.80891719 2	2.80891722 2	4.7-7| 6.0-7| -5.1-9	40:27
1dc.64	544 | 64; 2080	12| 107| 39.6	9.99999884 0	9.99998239 0	1.2-7| 9.9-7| 7.8-7	09
1et.64	265 | 64; 2080	12| 24| 17.0	1.88000008 1	1.87999801 1	3.2-8| 6.6-7| 5.4-7	02
1tc.64	193 | 64; 2080	12| 54| 37.9	1.99999995 1	1.99999784 1	7.9-8| 9.3-7| 5.2-7	05
1dc.128	1472 | 128; 8256	28| 277| 117.4	1.66790646 1	1.66783087 1	5.4-5| 2.6-8| 2.2-5	3:16
1et.128	673 | 128; 8256	12| 41| 26.9	2.92309168 1	2.92308878 1	8.3-7| 6.6-7| 4.9-7	08
1tc.128	513 | 128; 8256	14| 51| 28.0	3.80000025 1	3.79999965 1	2.3-7| 4.4-7| 7.9-8	09
1zc.128	1121 | 128; 8256	14| 23| 12.9	2.06667715 1	2.06666385 1	8.5-7| 9.3-7| 3.1-6	04
1dc.256	3840 | 256; 32896	21| 131| 39.3	2.99999987 1	3.00000004 1	4.3-8| 1.7-8| -2.8-8	2:24
1et.256	1665 | 256; 32896	21| 195| 108.4	5.44706489 1	5.44652433 1	2.3-5 | 4.0-7| 4.9-5	8:37
1tc.256	1313 | 256; 32896	23| 228| 137.5	6.32416075 1	6.32404374 1	1.5-5 | 7.5-7| 9.2-6	11:17
1zc.256	2817 | 256; 32896	17| 40| 13.6	3.73333432 1	3.73333029 1	1.7-7| 8.2-7| 5.3-7	21
1dc.512	9728 | 512; 131328	24| 204| 72.9	5.26955154 1	5.26951392 1	2.7-6 | 5.4-7| 3.5-6	36:48
1et.512	4033 | 512; 131328	17| 181| 147.4	1.03625531 2	1.03555196 2	1.3-4| 5.8-7| 3.4-4	51:10
1tc.512	3265 | 512; 131328	28| 396| 143.9	1.12613099 2	1.12538820 2	9.3-5 | 7.9-7| 3.3-4	2:14:55
2dc.512	54896 | 512; 131328	33| 513| 106.2	1.13946331 1	1.13857125 1	2.1-4| 7.7-7| 3.8-4	2:25:15
1zc.512	6913 | 512; 131328	11| 57| 37.3	6.80000034 1	6.79999769 1	4.3-7| 7.6-7| 1.9-7	6:09
1dc.1024	24064 | 1024; 524800	24| 260| 81.4	9.55539508 1	9.55512205 1	1.4-5 | 6.9-7| 1.4-5	5:03:49
1et.1024	9601 | 1024; 524800	20| 198| 155.0	1.82075477 2	1.82071562 2	4.8-6 | 7.0-7| 1.1-5	6:45:50
1tc.1024	7937 | 1024; 524800	27| 414| 124.6	2.04591268 2	2.04236122 2	1.5-4| 7.3-7| 8.7-4	10:37:57
1zc.1024	16641 | 1024; 524800	11| 67| 38.1	1.27999936 2	1.27999977 2	6.4-7| 5.7-7| -1.6-7	40:13
2dc.1024	169163 | 1024; 524800	28| 455| 101.8	1.77416130 1	1.77149535 1	1.6-4| 6.2-7| 7.3-4	11:57:25
1dc.2048	58368 | 2048; 2098176	20| 320| 73.0	1.74292685 2	1.74258827 2	1.9-5| 7.1-7| 9.7-5	35:52:44
1et.2048	22529 | 2048; 2098176	22| 341| 171.5	3.38193695 2	3.38166811 2	6.3-6| 5.7-7| 4.0-5	80:48:17
1tc.2048	18945 | 2048; 2098176	24| 381| 150.2	3.71592017 2	3.70575527 2	3.5-4| 7.9-7| 1.4-3	73:56:01
1zc.2048	39425 | 2048; 2098176	11| 38| 29.3	2.37400054 2	2.37399944 2	2.5-7| 7.9-7| 2.3-7	2:13:04
2dc.2048	504452 | 2048; 2098176	27| 459| 53.4	2.89755241 1	2.88181157 1	1.3-4| 7.2-7| 2.7-3	45:21:42

6.3 SDP relaxations of quadratic assignment problems

The quadratic assignment problem (QAP) is one of fundamental combinatorial optimization problems in the branch of optimization or operations research in mathematics, from the category of the facilities location problems. In this section, we apply our NAL algorithm to compute the lower bound for quadratic assignment problems (QAPs) through SDP relaxations.

Let Π be the set of $n \times n$ permutation matrices. Given matrices $A, B \in \Re^{n\times n}$, the quadratic assignment problem is:

$$v^*_{\rm QAP} := \min\{\langle X, AXB\rangle \,:\, X \in \Pi\}. \tag{6.5}$$

For a matrix $X = [x_1, \ldots, x_n] \in \Re^{n\times n}$, we will identify it with the n^2-vector $x = [x_1; \ldots; x_n]$. For a matrix $Y \in R^{n^2\times n^2}$, we let Y^{ij} be the $n \times n$ block corresponding to $x_i x_j^T$ in the matrix xx^T. It is shown in [87] that $v^*_{\rm QAP}$ is bounded below by the following number:

$$\begin{aligned} v \;:=\; \min \quad & \langle B \otimes A,\, Y\rangle \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^n Y^{ii} = I,\ \langle I,\, Y^{ij}\rangle = \delta_{ij}\ \forall\, 1 \le i \le j \le n, \\ & \langle E,\, Y^{ij}\rangle = 1,\ \forall\, 1 \le i \le j \le n, \\ & Y \succeq 0,\ Y \ge 0, \end{aligned} \tag{6.6}$$

where E is the matrix of ones, and $\delta_{ij} = 1$ if $i = j$, and 0 otherwise. There are $3n(n+1)/2$ equality constraints in (6.6). But two of them are actually redundant, and we remove them when solving the standard SDP generated from (6.6). Note that [87] actually used the constraint $\langle E,\, Y\rangle = n^2$ in place of the last set of the equality constraints in (6.6). But we prefer to use the formulation here because the associated SDP has slightly better numerical behavior. Note also that the SDP problems (6.6) typically do not satisfy the constraint nondegenerate conditions (4.23) and (4.32) at the optimal solutions.

In our experiment, we apply the NAL algorithm to the dual of (6.6) and hence any dual feasible solution would give a lower bound for (6.6). But in practice, our algorithm only delivers an approximately feasible dual solution $\tilde{y}$. We therefore apply the procedure given in [54, Theorem 2] to $\tilde{y}$ to construct a valid lower bound for (6.6), which we denote by $\underline{v}$.

Table 6.5 lists the results of the NAL algorithm on the quadratic assignment instances (6.6). The details of the table are the same as for Table 6.1 except that the objective values are replaced by the best known upper bound on (6.5) under the column "best upper bound" and the lower bound $\underline{v}$. The entries under the column under "%gap" are calculated as follows:

$$\%gap \ = \ \frac{\text{best upper bound} - \underline{v}}{\text{best upper bound}} \times 100\%.$$

We compare our results with those obtained in [22] which used a dedicated augmented Lagrangian algorithm to solve the SDP arising from applying the lift-and-project procedure of Lovász and Schrijver to (6.5). As the augmented Lagrangian algorithm in [22] is designed specifically for the SDPs arising from the lift-and-project procedure, the details of that algorithm is very different from our NAL algorithm. Note that the algorithm in [22] was implemented in C (with LAPACK library) and the results reported were obtained from a 2.4 GHz Pentium 4 PC with 1 GB of RAM (which is about 50% slower than our PC). By comparing the results in Table 6.5 against those in [22, Tables 6 and 7], we can safely conclude that the NAL algorithm applied to (6.6) is superior in terms of CPU time and the accuracy of the approximate optimal solution computed. Take for example the SDPs corresponding to the QAPs `nug30` and `tai35b`, the NAL algorithm obtains the lower bounds with `%gap` of 2.939 and 5.318 in $15,729$ and $37,990$ seconds respectively, whereas the algorithm in [22] computes the bounds with `%gap` of 3.10 and 15.42 in $127,011$ and $430,914$ seconds respectively.

The paper [22] also solved the lift-and-project SDP relaxations for the maximum stable set problems (denoted as N_+ and is known to be at least as strong as θ_+) using a dedicated augmented Lagrangian algorithm. By comparing the results in Table 6.4 against those in [22, Table4], we can again conclude that the NAL algorithm applied to (6.4) is superior in terms of CPU time and the accuracy of the approximate optimal solution computed. Take for example the SDPs corresponding to the graphs `p-hat300-1` and `c-fat200-1`, the NAL algorithm obtains the upper bounds of $\theta_+ = 10.0202$ and $\theta_+ = 12.0000$ in 410 and 36 seconds respectively, whereas the the algorithm in [22] computes the bounds of $N_+ = 18.6697$ and $N_+ = 14.9735$ in $322,287$ and $126,103$ seconds respectively.

Table 6.5: Results for the NAL algorithm on the quadratic assignment problems. The entries under the column "%gap" are calculated with respect to the best solution listed, which is known to be optimal unless the symbol (†) is prefixed.

problem	$m - c_l \mid c_s; c_l$	it| itsub| pcg	best upper bound	lower bound $\underline{v}$	R_P| R_D— %gap	time
bur26a	1051 | 676; 228826	27| 389| 105.9	5.42667000 6	5.42577700 6	2.9-3| 2.8-7| 0.016	4:28:43
bur26b	1051 | 676; 228826	25| 358| 92.3	3.81785200 6	3.81663900 6	2.3-3| 6.1-7| 0.032	3:23:39
bur26c	1051 | 676; 228826	26| 421| 107.5	5.42679500 6	5.42593600 6	3.9-3| 4.7-7| 0.016	4:56:09
bur26d	1051 | 676; 228826	27| 424| 102.3	3.82122500 6	3.81982900 6	3.8-3| 5.0-7| 0.037	4:21:32
bur26e	1051 | 676; 228826	27| 573| 100.0	5.38687900 6	5.38683200 6	7.5-3| 1.7-7| 0.001	5:34:39
bur26f	1051 | 676; 228826	25| 534| 100.9	3.78204400 6	3.78184600 6	3.1-3| 6.2-7| 0.005	5:32:51
bur26g	1051 | 676; 228826	24| 422| 91.0	1.01171720 7	1.01167630 7	3.8-3| 6.6-7| 0.004	3:33:58
bur26h	1051 | 676; 228826	24| 450| 96.8	7.09865800 6	7.09856700 6	2.0-3| 2.3-7| 0.001	3:53:22
chr12a	232 | 144; 10440	24| 314| 82.5	9.55200000 3	9.55200000 3	4.6-7| 4.2-12| 0.000	3:02
chr12b	232 | 144; 10440	23| 374| 106.6	9.74200000 3	9.74200000 3	4.3-7| 5.9-12| 0.000	4:12
chr12c	232 | 144; 10440	25| 511| 103.7	1.11560000 4	1.11560000 4	1.7-3| 5.6-7| 0.000	3:33
chr15a	358 | 225; 25425	27| 505| 110.9	9.89600000 3	9.88800000 3	3.3-3| 3.1-7| 0.081	19:51
chr15b	358 | 225; 25425	23| 385| 94.0	7.99000000 3	7.99000000 3	1.9-4| 3.1-8| 0.000	11:42
chr15c	358 | 225; 25425	21| 382| 82.4	9.50400000 3	9.50400000 3	2.2-4| 2.4-8| 0.000	10:39
chr18a	511 | 324; 52650	32| 660| 111.7	1.10980000 4	1.10960000 4	8.1-3| 1.7-7| 0.018	57:06
chr18b	511 | 324; 52650	25| 308| 136.1	1.53400000 3	1.53400000 3	9.9-5 | 6.9-7| 0.000	35:25
chr20a	628 | 400; 80200	32| 563| 117.8	2.19200000 3	2.19200000 3	4.3-3| 2.9-8| 0.000	1:28:45
chr20b	628 | 400; 80200	25| 375| 98.2	2.29800000 3	2.29800000 3	1.1-3| 1.5-7| 0.000	54:09
chr20c	628 | 400; 80200	30| 477| 101.0	1.41420000 4	1.41400000 4	5.5-3| 5.4-7| 0.014	57:26
chr22a	757 | 484; 117370	26| 467| 116.7	6.15600000 3	6.15600000 3	2.3-3| 9.3-8| 0.000	1:50:37
chr22b	757 | 484; 117370	26| 465| 106.4	6.19400000 3	6.19400000 3	1.8-3| 6.9-8| 0.000	1:47:16
chr25a	973 | 625; 195625	26| 462| 84.7	3.79600000 3	3.79600000 3	1.9-3| 1.4-7| 0.000	3:20:35
els19	568 | 361; 65341	28| 554| 99.5	1.72125480 7	1.72112340 7	1.0-4| 6.5-7| 0.008	51:52
esc16a	406 | 256; 32896	24| 251| 106.3	6.80000000 1	6.40000000 1	9.3-5 | 5.3-7| 5.882	10:48
esc16b	406 | 256; 32896	26| 321| 80.7	2.92000000 2	2.89000000 2	5.0-4| 4.9-7| 1.027	10:10
esc16c	406 | 256; 32896	27| 331| 77.5	1.60000000 2	1.53000000 2	6.6-4| 5.6-7| 4.375	10:42
esc16d	406 | 256; 32896	20| 62| 70.8	1.60000000 1	1.30000000 1	6.1-7| 8.0-7| 18.750	1:45
esc16e	406 | 256; 32896	19| 61| 70.1	2.80000000 1	2.70000000 1	9.7-8| 9.4-7| 3.571	1:42
esc16g	406 | 256; 32896	23| 106| 109.8	2.60000000 1	2.50000000 1	2.9-7| 4.7-7| 3.846	4:26
esc16h	406 | 256; 32896	29| 319| 90.0	9.96000000 2	9.76000000 2	1.4-4| 5.8-7| 2.008	10:52
esc16i	406 | 256; 32896	20| 106| 117.4	1.40000000 1	1.20000000 1	8.6-7| 6.9-7| 14.286	4:51
esc16j	406 | 256; 32896	15| 67| 104.8	8.00000000 0	8.00000000 0	1.6-7| 4.1-7| 0.000	2:41
esc32a	1582 | 1024; 524800	26| 232| 101.9	† 1.30000000 2	1.04000000 2	2.5-5 | 7.8-7| 20.000	4:48:55
esc32b	1582 | 1024; 524800	22| 201| 99.4	† 1.68000000 2	1.32000000 2	1.7-4| 7.8-7| 21.429	3:52:36
esc32c	1582 | 1024; 524800	30| 479| 140.2	† 6.42000000 2	6.16000000 2	6.5-4| 2.1-7| 4.050	11:12:30
esc32d	1582 | 1024; 524800	25| 254| 132.0	† 2.00000000 2	1.91000000 2	5.3-7| 5.6-7| 4.500	5:43:54
esc32e	1582 | 1024; 524800	15| 46| 58.2	2.00000000 0	2.00000000 0	2.2-7| 1.1-7| 0.000	31:11
esc32f	1582 | 1024; 524800	15| 46| 58.2	2.00000000 0	2.00000000 0	2.2-7| 1.1-7| 0.000	31:13
esc32g	1582 | 1024; 524800	15| 38| 50.7	6.00000000 0	6.00000000 0	1.7-7| 3.2-7| 0.000	23:25
esc32h	1582 | 1024; 524800	30| 403| 113.3	† 4.38000000 2	4.23000000 2	9.9-4| 3.0-7| 3.425	8:05:32
had12	232 | 144; 10440	23| 457| 93.8	1.65200000 3	1.65200000 3	2.2-4| 1.4-7| 0.000	5:17

Table 6.5: Results for the NAL algorithm on the quadratic assignment problems. The entries under the column "%gap" are calculated with respect to the best solution listed, which is known to be optimal unless the symbol (†) is prefixed.

problem	$m - c_l$ | $c_s; c_l$	it| itsub| pcg	best upper bound	lower bound $\underline{v}$	R_P| R_D— %gap	time
had14	313 | 196; 19306	28| 525| 99.5	2.72400000 3	2.72400000 3	1.5-3| 7.6-7| 0.000	13:03
had16	406 | 256; 32896	27| 525| 98.7	3.72000000 3	3.72000000 3	1.4-3| 1.2-7| 0.000	22:37
had18	511 | 324; 52650	29| 458| 104.3	5.35800000 3	5.35800000 3	1.5-3| 4.0-7| 0.000	44:30
had20	628 | 400; 80200	32| 568| 96.7	6.92200000 3	6.92200000 3	3.8-3| 2.6-7| 0.000	1:24:06
kra30a	1393 | 900; 405450	27| 313| 68.0	8.89000000 4	8.64280000 4	4.5-4| 6.5-7| 2.781	4:08:17
kra30b	1393 | 900; 405450	28| 289| 68.9	9.14200000 4	8.74500000 4	3.1-4| 7.4-7| 4.343	3:50:35
kra32	1582 | 1024; 524800	31| 307| 78.6	8.89000000 4	8.52980000 4	4.6-4| 6.0-7| 4.052	6:43:41
lipa20a	628 | 400; 80200	18| 243| 70.1	3.68300000 3	3.68300000 3	5.5-7| 2.9-9| 0.000	24:29
lipa20b	628 | 400; 80200	14| 116| 56.2	2.70760000 4	2.70760000 4	1.7-5 | 6.5-7| 0.000	10:10
lipa30a	1393 | 900; 405450	20| 252| 78.2	1.31780000 4	1.31780000 4	2.5-7| 1.1-10| 0.000	3:41:44
lipa30b	1393 | 900; 405450	18| 83| 80.8	1.51426000 5	1.51426000 5	6.9-7| 3.3-8| 0.000	1:23:34
lipa40a	2458 | 1600; 1280800	22| 324| 81.7	3.15380000 4	3.15380000 4	4.1-7| 4.6-11| 0.000	21:02:51
lipa40b	2458 | 1600; 1280800	19| 121| 76.6	4.76581000 5	4.76581000 5	3.9-6 | 1.3-8| 0.000	7:24:25
nug12	232 | 144; 10440	22| 266| 69.6	5.78000000 2	5.68000000 2	1.2-4| 3.6-7| 1.730	2:27
nug14	313 | 196; 19306	24| 337| 62.3	1.01400000 3	1.00800000 3	3.1-4| 8.0-7| 0.592	5:50
nug15	358 | 225; 25425	27| 318| 62.6	1.15000000 3	1.13800000 3	3.0-4| 7.5-7| 1.043	7:32
nug16a	406 | 256; 32896	25| 346| 80.4	1.61000000 3	1.59700000 3	3.3-4| 6.6-7| 0.807	14:15
nug16b	406 | 256; 32896	28| 315| 64.5	1.24000000 3	1.21600000 3	2.8-4| 4.2-7| 1.935	10:20
nug17	457 | 289; 41905	26| 302| 60.6	1.73200000 3	1.70400000 3	2.0-4| 7.7-7| 1.617	12:38
nug18	511 | 324; 52650	26| 287| 59.5	1.93000000 3	1.89100000 3	2.2-4| 3.5-7| 2.021	15:39
nug20	628 | 400; 80200	26| 318| 65.1	2.57000000 3	2.50400000 3	1.5-4| 5.2-7| 2.568	31:49
nug21	691 | 441; 97461	27| 331| 62.5	2.43800000 3	2.37800000 3	1.9-4| 6.6-7| 2.461	40:22
nug22	757 | 484; 117370	28| 369| 86.0	3.59600000 3	3.52200000 3	3.1-4| 5.9-7| 2.058	1:21:58
nug24	898 | 576; 166176	29| 348| 63.7	3.48800000 3	3.39600000 3	1.8-4| 3.6-7| 2.638	1:33:59
nug25	973 | 625; 195625	27| 335| 60.2	3.74400000 3	3.62100000 3	1.8-4| 3.0-7| 3.285	1:41:49
nug27	1132 | 729; 266085	29| 380| 80.1	5.23400000 3	5.12400000 3	1.3-4| 4.5-7| 2.102	3:31:50
nug28	1216 | 784; 307720	26| 329| 80.5	5.16600000 3	5.02000000 3	2.4-4| 6.3-7| 2.826	3:36:38
nug30	1393 | 900; 405450	27| 360| 61.4	6.12400000 3	5.94400000 3	1.3-4| 3.3-7| 2.939	4:22:09
rou12	232 | 144; 10440	25| 336| 106.3	2.35528000 5	2.35434000 5	4.6-4| 1.6-7| 0.040	4:50
rou15	358 | 225; 25425	26| 238| 64.0	3.54210000 5	3.49544000 5	2.5-4| 4.0-7| 1.317	5:48
rou20	628 | 400; 80200	26| 250| 69.9	7.25522000 5	6.94397000 5	1.5-4| 7.5-7| 4.290	27:26
scr12	232 | 144; 10440	19| 255| 99.9	3.14100000 4	3.14080000 4	4.3-4| 7.5-7| 0.006	3:16
scr15	358 | 225; 25425	19| 331| 91.7	5.11400000 4	5.11400000 4	1.3-7| 2.8-7| 0.000	9:42
scr20	628 | 400; 80200	28| 353| 65.2	1.10030000 5	1.06472000 5	2.6-4| 4.9-7| 3.234	34:32
ste36a	1996 | 1296; 840456	26| 318| 93.8	9.52600000 3	9.23600000 3	1.7-4| 4.1-7| 3.044	15:09:10
ste36b	1996 | 1296; 840456	29| 348| 101.0	1.58520000 4	1.56030000 4	1.8-3| 4.3-7| 1.571	19:05:19
ste36c	1996 | 1296; 840456	28| 360| 105.3	8.23911000 6	8.11864500 6	6.3-4| 4.0-7| 1.462	19:56:15
tai12a	232 | 144; 10440	15| 180| 59.8	2.24416000 5	2.24416000 5	1.8-6 | 7.6-8| 0.000	1:28
tai12b	232 | 144; 10440	29| 596| 112.2	3.94649250 7	3.94649080 7	3.7-4| 9.3-9| 0.000	7:40
tai15a	358 | 225; 25425	23| 196| 65.1	3.88214000 5	3.76608000 5	1.3-4| 5.0-7| 2.990	4:58
tai15b	358 | 225; 25425	29| 409| 102.2	5.17652680 7	5.17609220 7	1.5-3| 7.0-7| 0.008	16:04
tai17a	457 | 289; 41905	23| 168| 69.7	4.91812000 5	4.75893000 5	1.4-4| 5.0-7| 3.237	8:21

Table 6.5: Results for the NAL algorithm on the quadratic assignment problems. The entries under the column "%gap" are calculated with respect to the best solution listed, which is known to be optimal unless the symbol (†) is prefixed.

problem	$m - c_l \mid c_s; c_l$	it| itsub| pcg	best upper bound	lower bound $\underline{v}$	R_P| R_D— %gap	time
tai20a	628 | 400; 80200	27| 220| 73.3	7.03482000 5	6.70827000 5	1.9-4| 4.2-7| 4.642	25:32
tai20b	628 | 400; 80200	31| 485| 91.6	1.22455319 8	1.22452095 8	2.9-3| 1.4-7| 0.003	54:05
tai25a	973 | 625; 195625	27| 194| 77.3	1.16725600 6	1.01301000 6	8.0-7| 7.9-7| 13.214	1:17:54
tai25b	973 | 625; 195625	29| 408| 70.4	3.44355646 8	3.33685462 8	2.6-3| 6.2-7| 3.099	2:33:26
tai30a	1393 | 900; 405450	27| 207| 82.4	† 1.81814600 6	1.70578200 6	8.1-5 | 2.0-7| 6.180	3:35:03
tai30b	1393 | 900; 405450	30| 421| 71.6	6.37117113 8	5.95926267 8	1.4-3| 4.9-7| 6.465	6:26:30
tai35a	1888 | 1225; 750925	28| 221| 81.0	2.42200200 6	2.21523000 6	1.5-4| 5.0-7| 8.537	8:09:44
tai35b	1888 | 1225; 750925	28| 401| 58.3	2.83315445 8	2.68328155 8	8.7-4| 6.4-7| 5.290	10:33:10
tai40a	2458 | 1600; 1280800	27| 203| 85.1	3.13937000 6	2.84184600 6	7.5-5 | 5.3-7| 9.477	15:25:52
tai40b	2458 | 1600; 1280800	30| 362| 74.1	6.37250948 8	6.06880822 8	1.7-3| 4.9-7| 4.766	23:32:56
tho30	1393 | 900; 405450	27| 315| 61.1	1.49936000 5	1.43267000 5	2.4-4| 7.3-7| 4.448	3:41:26
tho40	2458 | 1600; 1280800	27| 349| 60.9	† 2.40516000 5	2.26161000 5	2.0-4| 6.5-7| 5.968	17:13:24

6.4 SDP relaxations of binary integer quadratic problems

The binary integer quadratic (BIQ) problem is formulated as follows

$$v^*_{\rm BIQ} := \min\{x^T Q x \,:\, x \in \{0,1\}^n\}, \tag{6.7}$$

where Q is a symmetric matrix (non positive semidefinite) of order n. Problem (6.7) belongs to a class of NP-complete combinatorial optimization problems that have many interesting applications, such as Financial analysis problems [75], CAD problems [61], and models of message traffic management [42]. Since BIQ problems are usually NP-hard which are difficult to obtain exact solutions, we consider the following SDP relaxation of (6.7),

$$\begin{array}{ll} \min & \langle Q,\, Y\rangle \\ \text{s.t.} & \operatorname{diag}(Y) - y = 0, \quad \alpha = 1, \\ & \begin{bmatrix} Y & y \\ y^T & \alpha \end{bmatrix} \succeq 0, \quad Y \ge 0,\ y \ge 0. \end{array} \tag{6.8}$$

Table 6.6 lists the results obtained by the NAL algorithm on the SDPs (6.8) arising from the BIQ instances described in [132]. It is interesting to note that the lower bound obtained from (6.8) is within 10% of the optimal value v^*_{BIQ} for all the instances tested, and for the instances `gka1b`–`gka9b`, the lower bounds are actually equal to v^*_{BIQ}.

Table 6.6: Results for the NAL algorithm on the BIQ problems. The entries under the column "%gap" are calculated with respect to the best solution listed, which is known to be optimal unless the symbol (†) is prefixed.

problem	$m - c_l \mid c_s; c_l$	it| itsub| pcg	best upper bound	lower bound $\underline{v}$	R_P| R_D| %gap	time
be100.1	101 | 101; 5151	27| 488| 70.5	-1.94120000 4	-2.00210000 4	8.6-7| 5.7-7| 3.137	1:45
be100.2	101 | 101; 5151	25| 378| 78.5	-1.72900000 4	-1.79880000 4	8.3-7| 7.6-7| 4.037	1:32
be100.3	101 | 101; 5151	27| 432| 96.3	-1.75650000 4	-1.82310000 4	3.7-7| 7.0-7| 3.792	2:08
be100.4	101 | 101; 5151	27| 505| 101.2	-1.91250000 4	-1.98410000 4	2.4-6| 7.7-7| 3.744	2:37
be100.5	101 | 101; 5151	25| 355| 78.5	-1.58680000 4	-1.68880000 4	8.6-7| 8.8-7| 6.428	1:28
be100.6	101 | 101; 5151	26| 440| 94.4	-1.73680000 4	-1.81480000 4	4.7-6 | 6.3-7| 4.491	2:06
be100.7	101 | 101; 5151	27| 219| 92.3	-1.86290000 4	-1.97000000 4	1.3-7| 4.9-7| 5.749	1:01
be100.8	101 | 101; 5151	25| 265| 47.1	-1.86490000 4	-1.99460000 4	5.1-7| 5.9-7| 6.955	40
be100.9	101 | 101; 5151	28| 526| 72.6	-1.32940000 4	-1.42630000 4	6.4-7| 5.3-7| 7.289	2:01
be100.10	101 | 101; 5151	27| 493| 52.0	-1.53520000 4	-1.64080000 4	6.7-7| 5.8-7| 6.879	1:25
be120.3.1	121 | 121; 7381	26| 384| 112.4	-1.30670000 4	-1.38030000 4	5.9-6 | 4.9-7| 5.633	2:57
be120.3.2	121 | 121; 7381	27| 410| 117.9	-1.30460000 4	-1.36260000 4	4.6-6 | 4.1-7| 4.446	3:16
be120.3.3	121 | 121; 7381	26| 210| 89.2	-1.24180000 4	-1.29870000 4	2.9-7| 4.4-7| 4.582	1:19
be120.3.4	121 | 121; 7381	27| 391| 64.8	-1.38670000 4	-1.45110000 4	6.6-7| 5.5-7| 4.644	1:49
be120.3.5	121 | 121; 7381	27| 489| 99.0	-1.14030000 4	-1.19910000 4	7.8-6 | 2.9-7| 5.157	3:21
be120.3.6	121 | 121; 7381	26| 386| 111.2	-1.29150000 4	-1.34320000 4	7.9-7| 4.3-7| 4.003	2:57
be120.3.7	121 | 121; 7381	27| 412| 111.9	-1.40680000 4	-1.45640000 4	1.0-4| 5.1-7| 3.526	3:16
be120.3.8	121 | 121; 7381	27| 426| 108.5	-1.47010000 4	-1.53030000 4	8.1-5 | 4.0-7| 4.095	3:10
be120.3.9	121 | 121; 7381	27| 418| 89.2	-1.04580000 4	-1.12410000 4	7.5-5 | 6.3-7| 7.487	2:39
be120.3.10	121 | 121; 7381	30| 611| 84.0	-1.22010000 4	-1.29300000 4	1.1-6 | 2.9-7| 5.975	3:36
be120.8.1	121 | 121; 7381	26| 384| 71.5	-1.86910000 4	-2.01940000 4	4.3-7| 6.6-7| 8.041	1:53
be120.8.2	121 | 121; 7381	26| 402| 113.9	-1.88270000 4	-2.00740000 4	4.9-5 | 4.4-7| 6.623	3:11
be120.8.3	121 | 121; 7381	27| 267| 96.2	-1.93020000 4	-2.05050000 4	5.1-7| 5.1-7| 6.233	1:48
be120.8.4	121 | 121; 7381	26| 399| 96.6	-2.07650000 4	-2.17790000 4	3.4-6 | 4.2-7| 4.883	2:42
be120.8.5	121 | 121; 7381	27| 452| 120.1	-2.04170000 4	-2.13160000 4	8.3-7| 5.3-7| 4.403	3:48
be120.8.6	121 | 121; 7381	29| 459| 90.6	-1.84820000 4	-1.96770000 4	1.3-6 | 6.3-7| 6.466	2:53
be120.8.7	121 | 121; 7381	28| 457| 52.5	-2.21940000 4	-2.37320000 4	2.0-7| 4.9-7| 6.930	1:46
be120.8.8	121 | 121; 7381	27| 151| 66.1	-1.95340000 4	-2.12040000 4	8.0-7| 9.7-7| 8.549	43
be120.8.9	121 | 121; 7381	27| 301| 60.4	-1.81950000 4	-1.92840000 4	2.3-7| 4.1-7| 5.985	1:17
be120.8.10	121 | 121; 7381	27| 307| 102.7	-1.90490000 4	-2.00240000 4	4.1-7| 4.1-7| 5.118	2:14
be150.3.1	151 | 151; 11476	27| 538| 84.7	-1.88890000 4	-1.98490000 4	1.3-5 | 5.3-7| 5.082	4:57
be150.3.2	151 | 151; 11476	28| 499| 89.3	-1.78160000 4	-1.88640000 4	1.1-5 | 6.0-7| 5.882	4:51
be150.3.3	151 | 151; 11476	26| 514| 101.8	-1.73140000 4	-1.80430000 4	1.8-6 | 7.6-7| 4.210	5:37
be150.3.4	151 | 151; 11476	27| 233| 98.2	-1.98840000 4	-2.06520000 4	4.9-7| 6.0-7| 3.862	2:28

Table 6.6: Results for the NAL algorithm on the BIQ problems. The entries under the column "%gap" are calculated with respect to the best solution listed, which is known to be optimal unless the symbol (†) is prefixed.

problem	$m - c_l \mid c_s; c_l$	it| itsub| pcg	best upper bound	lower bound $\underline{v}$	R_P| R_D| %gap	time
be150.3.5	151 | 151; 11476	28| 507| 90.4	-1.68170000 4	-1.77680000 4	1.6-5 | 4.1-7| 5.655	4:53
be150.3.6	151 | 151; 11476	27| 517| 95.5	-1.67800000 4	-1.80500000 4	6.7-6 | 5.0-7| 7.569	5:18
be150.3.7	151 | 151; 11476	27| 470| 73.5	-1.80010000 4	-1.91010000 4	6.8-7| 9.1-7| 6.111	3:42
be150.3.8	151 | 151; 11476	27| 377| 84.7	-1.83030000 4	-1.96980000 4	1.3-5 | 6.3-7| 7.622	3:25
be150.3.9	151 | 151; 11476	26| 292| 58.0	-1.28380000 4	-1.41030000 4	3.8-7| 8.8-7| 9.854	1:52
be150.3.10	151 | 151; 11476	27| 438| 121.3	-1.79630000 4	-1.92300000 4	1.6-5 | 3.7-7| 7.053	5:39
be150.8.1	151 | 151; 11476	28| 661| 78.0	-2.70890000 4	-2.91430000 4	9.4-7| 6.6-7| 7.582	5:36
be150.8.2	151 | 151; 11476	27| 272| 87.4	-2.67790000 4	-2.88210000 4	3.5-7| 7.6-7| 7.625	2:34
be150.8.3	151 | 151; 11476	27| 435| 77.9	-2.94380000 4	-3.10600000 4	3.5-7| 8.3-7| 5.510	3:37
be150.8.4	151 | 151; 11476	26| 310| 89.5	-2.69110000 4	-2.87290000 4	8.9-7| 8.6-7| 6.756	3:01
be150.8.5	151 | 151; 11476	27| 500| 113.9	-2.80170000 4	-2.94820000 4	9.4-7| 3.7-7| 5.229	6:06
be150.8.6	151 | 151; 11476	27| 415| 115.6	-2.92210000 4	-3.14370000 4	5.2-6 | 6.8-7| 7.584	4:56
be150.8.7	151 | 151; 11476	27| 446| 127.2	-3.12090000 4	-3.32520000 4	2.8-5 | 2.5-7| 6.546	6:06
be150.8.8	151 | 151; 11476	28| 462| 109.0	-2.97300000 4	-3.16000000 4	5.8-6 | 6.7-7| 6.290	5:23
be150.8.9	151 | 151; 11476	28| 370| 110.7	-2.53880000 4	-2.71100000 4	2.6-7| 5.3-7| 6.783	4:20
be150.8.10	151 | 151; 11476	26| 288| 95.7	-2.83740000 4	-3.00480000 4	5.2-7| 4.7-7| 5.900	2:58
be200.3.1	201 | 201; 20301	29| 615| 89.7	-2.54530000 4	-2.77160000 4	5.6-7| 5.0-7| 8.891	10:29
be200.3.2	201 | 201; 20301	29| 307| 93.2	-2.50270000 4	-2.67600000 4	3.5-7| 5.3-7| 6.925	5:38
be200.3.3	201 | 201; 20301	29| 507| 120.8	-2.80230000 4	-2.94780000 4	5.6-5 | 5.7-7| 5.192	12:09
be200.3.4	201 | 201; 20301	29| 523| 102.1	-2.74340000 4	-2.91060000 4	4.7-6 | 5.4-7| 6.095	10:41
be200.3.5	201 | 201; 20301	28| 466| 116.2	-2.63550000 4	-2.80730000 4	1.4-6 | 5.5-7| 6.519	10:38
be200.3.6	201 | 201; 20301	29| 639| 60.1	-2.61460000 4	-2.79280000 4	9.5-7| 3.7-7| 6.816	7:36
be200.3.7	201 | 201; 20301	29| 534| 93.9	-3.04830000 4	-3.16200000 4	1.1-6 | 5.8-7| 3.730	9:43
be200.3.8	201 | 201; 20301	29| 308| 100.7	-2.73550000 4	-2.92440000 4	6.4-7| 9.0-7| 6.906	5:59
be200.3.9	201 | 201; 20301	28| 482| 87.1	-2.46830000 4	-2.64370000 4	3.2-5 | 3.7-7| 7.106	8:28
be200.3.10	201 | 201; 20301	29| 539| 98.7	-2.38420000 4	-2.57600000 4	5.8-6 | 4.4-7| 8.045	10:25
be200.8.1	201 | 201; 20301	28| 489| 97.5	-4.85340000 4	-5.08690000 4	3.7-5 | 6.2-7| 4.811	9:41
be200.8.2	201 | 201; 20301	29| 192| 74.7	-4.08210000 4	-4.43360000 4	6.1-7| 7.3-7| 8.611	2:46
be200.8.3	201 | 201; 20301	28| 476| 116.1	-4.32070000 4	-4.62540000 4	5.8-7| 9.2-7| 7.052	10:53
be200.8.4	201 | 201; 20301	29| 267| 93.3	-4.37570000 4	-4.66210000 4	8.4-7| 7.2-7| 6.545	4:55
be200.8.5	201 | 201; 20301	28| 521| 93.8	-4.14820000 4	-4.42710000 4	1.7-5 | 7.7-7| 6.723	9:53
be200.8.6	201 | 201; 20301	28| 556| 87.4	-4.94920000 4	-5.12190000 4	2.7-5 | 4.4-7| 3.489	9:48
be200.8.7	201 | 201; 20301	27| 248| 92.6	-4.68280000 4	-4.93530000 4	4.7-7| 6.8-7| 5.392	4:30
be200.8.8	201 | 201; 20301	28| 314| 94.3	-4.45020000 4	-4.76890000 4	7.0-7| 7.7-7| 7.161	5:49
be200.8.9	201 | 201; 20301	29| 543| 115.6	-4.32410000 4	-4.54950000 4	5.8-6 | 3.8-7| 5.213	12:16
be200.8.10	201 | 201; 20301	29| 485| 107.9	-4.28320000 4	-4.57430000 4	6.9-6 | 5.5-7| 6.796	10:15
be250.1	251 | 251; 31626	29| 532| 94.7	-2.40760000 4	-2.51190000 4	4.0-5 | 4.6-7| 4.332	16:41
be250.2	251 | 251; 31626	28| 519| 113.6	-2.25400000 4	-2.36810000 4	3.1-5 | 6.4-7| 5.062	18:51
be250.3	251 | 251; 31626	28| 561| 95.7	-2.29230000 4	-2.40000000 4	2.9-5 | 6.0-7| 4.698	17:17
be250.4	251 | 251; 31626	30| 577| 112.2	-2.46490000 4	-2.57200000 4	4.8-5 | 4.7-7| 4.345	20:42
be250.5	251 | 251; 31626	29| 463| 98.1	-2.10570000 4	-2.23740000 4	9.3-5 | 4.4-7| 6.254	14:30
be250.6	251 | 251; 31626	30| 567| 93.6	-2.27350000 4	-2.40180000 4	2.0-5 | 4.3-7| 5.643	16:39

Table 6.6: Results for the NAL algorithm on the BIQ problems. The entries under the column "%gap" are calculated with respect to the best solution listed, which is known to be optimal unless the symbol (†) is prefixed.

problem	$m - c_l \mid c_s; c_l$	it| itsub| pcg	best upper bound	lower bound $\underline{v}$	R_P| R_D| %gap	time
be250.7	251 | 251; 31626	28| 507| 84.7	-2.40950000 4	-2.51190000 4	5.9-5 | 7.1-7| 4.250	14:00
be250.8	251 | 251; 31626	28| 620| 84.1	-2.38010000 4	-2.50200000 4	1.6-5 | 7.5-7| 5.122	16:50
be250.9	251 | 251; 31626	28| 589| 85.8	-2.00510000 4	-2.13970000 4	1.1-4| 3.6-7| 6.713	17:13
be250.10	251 | 251; 31626	29| 591| 88.9	-2.31590000 4	-2.43550000 4	3.4-5 | 4.8-7| 5.164	16:48
bqp50-1	51 | 51; 1326	25| 463| 119.9	-2.09800000 3	-2.14300000 3	7.1-6 | 6.7-7| 2.145	1:12
bqp50-2	51 | 51; 1326	26| 387| 72.7	-3.70200000 3	-3.74200000 3	2.3-5 | 5.8-7| 1.080	39
bqp50-3	51 | 51; 1326	24| 343| 84.3	-4.62600000 3	-4.63700000 3	8.9-7| 9.9-7| 0.238	40
bqp50-4	51 | 51; 1326	28| 486| 106.6	-3.54400000 3	-3.58300000 3	2.5-4| 5.2-7| 1.100	1:08
bqp50-5	51 | 51; 1326	23| 319| 82.7	-4.01200000 3	-4.07700000 3	3.3-5 | 6.9-7| 1.620	37
bqp50-6	51 | 51; 1326	20| 338| 95.8	-3.69300000 3	-3.71100000 3	1.1-5 | 4.4-7| 0.487	44
bqp50-7	51 | 51; 1326	26| 275| 44.0	-4.52000000 3	-4.64900000 3	2.9-7| 6.2-7| 2.854	18
bqp50-8	51 | 51; 1326	26| 289| 73.3	-4.21600000 3	-4.26900000 3	8.5-7| 6.5-7| 1.257	29
bqp50-9	51 | 51; 1326	21| 225| 57.5	-3.78000000 3	-3.92100000 3	8.3-7| 9.0-7| 3.730	19
bqp50-10	51 | 51; 1326	27| 191| 52.2	-3.50700000 3	-3.62600000 3	4.4-7| 6.5-7| 3.393	14
bqp100-1	101 | 101; 5151	25| 443| 80.5	-7.97000000 3	-8.38000000 3	2.7-7| 8.2-7| 5.144	1:49
bqp100-2	101 | 101; 5151	23| 374| 97.1	-1.10360000 4	-1.14890000 4	5.4-4| 4.8-7| 4.105	1:53
bqp100-3	101 | 101; 5151	26| 451| 122.4	-1.27230000 4	-1.31530000 4	9.9-7| 7.3-7| 3.380	2:40
bqp100-4	101 | 101; 5151	26| 420| 129.4	-1.03680000 4	-1.07310000 4	3.5-5 | 6.5-7| 3.501	2:42
bqp100-5	101 | 101; 5151	28| 515| 84.5	-9.08300000 3	-9.48700000 3	5.0-5 | 3.3-7| 4.448	2:16
bqp100-6	101 | 101; 5151	28| 524| 88.4	-1.02100000 4	-1.08240000 4	6.7-7| 4.6-7| 6.014	2:22
bqp100-7	101 | 101; 5151	28| 572| 81.9	-1.01250000 4	-1.06890000 4	8.5-7| 3.9-7| 5.570	2:19
bqp100-8	101 | 101; 5151	26| 440| 107.4	-1.14350000 4	-1.17700000 4	2.4-5 | 7.8-7| 2.930	2:25
bqp100-9	101 | 101; 5151	27| 482| 101.7	-1.14550000 4	-1.17330000 4	5.0-5 | 6.1-7| 2.427	2:31
bqp100-10	101 | 101; 5151	25| 415| 110.4	-1.25650000 4	-1.29800000 4	3.9-5 | 5.7-7| 3.303	2:18
bqp250-1	251 | 251; 31626	28| 483| 117.7	-4.56070000 4	-4.76630000 4	3.9-7| 6.6-7| 4.508	17:42
bqp250-2	251 | 251; 31626	30| 554| 93.5	-4.48100000 4	-4.72220000 4	4.4-5 | 4.1-7| 5.383	16:23
bqp250-3	251 | 251; 31626	28| 296| 116.4	-4.90370000 4	-5.10770000 4	9.9-7| 7.9-7| 4.160	10:36
bqp250-4	251 | 251; 31626	29| 607| 88.9	-4.12740000 4	-4.33120000 4	1.8-5 | 4.5-7| 4.938	17:37
bqp250-5	251 | 251; 31626	28| 570| 103.7	-4.79610000 4	-5.00040000 4	4.4-5 | 6.9-7| 4.260	19:03
bqp250-6	251 | 251; 31626	28| 477| 113.1	-4.10140000 4	-4.36690000 4	1.9-5 | 7.7-7| 6.473	17:11
bqp250-7	251 | 251; 31626	30| 429| 126.3	-4.67570000 4	-4.89220000 4	8.2-7| 5.9-7| 4.630	16:36
bqp250-8	251 | 251; 31626	28| 748| 73.5	-3.57260000 4	-3.87800000 4	6.3-7| 8.8-7| 8.548	17:34
bqp250-9	251 | 251; 31626	29| 453| 117.0	-4.89160000 4	-5.14970000 4	3.7-7| 3.9-7| 5.276	16:12
bqp250-10	251 | 251; 31626	28| 691| 76.7	-4.04420000 4	-4.30140000 4	8.1-7| 5.1-7| 6.360	16:29
bqp500-1	501 | 501; 125751	30| 357| 117.8	-1.16586000 5	-1.25965000 5	2.9-7| 5.5-7| 8.045	1:00:59
bqp500-2	501 | 501; 125751	30| 637| 94.7	-1.28223000 5	-1.36012000 5	7.9-5 | 7.2-7| 6.075	1:31:17
bqp500-3	501 | 501; 125751	30| 363| 118.9	-1.30812000 5	-1.38454000 5	4.4-7| 4.0-7| 5.842	1:01:47
bqp500-4	501 | 501; 125751	30| 663| 79.9	-1.30097000 5	-1.39329000 5	3.7-6 | 4.3-7| 7.096	1:16:35
bqp500-5	501 | 501; 125751	30| 539| 119.6	-1.25487000 5	-1.34092000 5	4.5-5 | 2.5-7| 6.857	1:36:43
bqp500-6	501 | 501; 125751	30| 485| 124.4	-1.21772000 5	-1.30765000 5	4.1-7| 5.1-7| 7.385	1:28:49
bqp500-7	501 | 501; 125751	31| 648| 87.7	-1.22201000 5	-1.31492000 5	8.1-5 | 5.7-7| 7.603	1:25:26
bqp500-8	501 | 501; 125751	31| 412| 126.3	-1.23559000 5	-1.33490000 5	8.6-7| 4.5-7| 8.037	1:14:37

Table 6.6: Results for the NAL algorithm on the BIQ problems. The entries under the column "%gap" are calculated with respect to the best solution listed, which is known to be optimal unless the symbol (†) is prefixed.

problem	$m - c_l \mid c_s; c_l$	it| itsub| pcg	best upper bound	lower bound $\underline{v}$	R_P| R_D| %gap	time
bqp500-9	501 | 501; 125751	30| 612| 92.7	-1.20798000 5	-1.30289000 5	9.5-5 | 7.3-7| 7.857	1:24:40
bqp500-10	501 | 501; 125751	30| 454| 130.5	-1.30619000 5	-1.38535000 5	7.0-7| 6.4-7| 6.060	1:24:23
gka1a	51 | 51; 1326	20| 309| 57.9	-3.41400000 3	-3.53700000 3	7.7-7| 6.0-7| 3.603	26
gka2a	61 | 61; 1891	24| 281| 57.3	-6.06300000 3	-6.17100000 3	1.4-7| 4.9-7| 1.781	27
gka3a	71 | 71; 2556	25| 398| 68.3	-6.03700000 3	-6.38600000 3	6.6-7| 9.5-7| 5.781	51
gka4a	81 | 81; 3321	25| 567| 106.2	-8.59800000 3	-8.88100000 3	4.2-6 | 6.3-7| 3.291	2:09
gka5a	51 | 51; 1326	24| 284| 55.9	-5.73700000 3	-5.89700000 3	7.7-7| 7.8-7| 2.789	23
gka6a	31 | 31; 496	25| 175| 46.8	-3.98000000 3	-4.10300000 3	4.4-7| 7.2-7| 3.090	10
gka7a	31 | 31; 496	26| 145| 47.2	-4.54100000 3	-4.63800000 3	3.9-7| 5.5-7| 2.136	08
gka8a	101 | 101; 5151	27| 543| 94.1	-1.11090000 4	-1.11970000 4	3.8-5 | 6.6-7| 0.792	2:39
gka1b	21 | 21; 231	7| 42| 23.8	-1.33000000 2	-1.33000000 2	9.8-7| 5.4-7| 0.000	02
gka2b	31 | 31; 496	15| 241| 101.1	-1.21000000 2	-1.21000000 2	8.8-5 | 7.7-7| 0.000	25
gka3b	41 | 41; 861	12| 85| 25.6	-1.18000000 2	-1.18000000 2	2.9-7| 2.4-8| 0.000	04
gka4b	51 | 51; 1326	14| 88| 25.9	-1.29000000 2	-1.29000000 2	2.8-7| 1.2-9| 0.000	04
gka5b	61 | 61; 1891	12| 86| 26.0	-1.50000000 2	-1.50000000 2	7.6-8| 1.7-8| 0.000	05
gka6b	71 | 71; 2556	13| 123| 34.6	-1.46000000 2	-1.46000000 2	3.3-7| 8.1-10| 0.000	10
gka7b	81 | 81; 3321	19| 193| 33.8	-1.60000000 2	-1.60000000 2	8.9-7| 5.3-7| 0.000	16
gka8b	91 | 91; 4186	15| 198| 47.0	-1.45000000 2	-1.45000000 2	5.9-7| 2.3-9| 0.000	28
gka9b	101 | 101; 5151	18| 252| 50.9	-1.37000000 2	-1.37000000 2	3.7-7| 1.2-10| 0.000	44
gka10b	126 | 126; 8001	17| 298| 94.5	-1.54000000 2	-1.55000000 2	1.6-4| 3.4-7| 0.649	2:14
gka1c	41 | 41; 861	24| 371| 103.7	-5.05800000 3	-5.11300000 3	1.5-5 | 3.8-7| 1.087	45
gka2c	51 | 51; 1326	27| 358| 72.0	-6.21300000 3	-6.32000000 3	2.5-7| 5.6-7| 1.722	35
gka3c	61 | 61; 1891	25| 305| 60.0	-6.66500000 3	-6.81300000 3	3.1-7| 9.6-7| 2.221	31
gka4c	71 | 71; 2556	27| 476| 114.7	-7.39800000 3	-7.56500000 3	9.7-7| 4.5-7| 2.257	1:38
gka5c	81 | 81; 3321	28| 304| 94.6	-7.36200000 3	-7.57600000 3	1.2-6 | 3.9-7| 2.907	1:03
gka6c	91 | 91; 4186	27| 427| 108.4	-5.82400000 3	-5.96100000 3	3.0-5 | 6.2-7| 2.352	1:58
gka7c	101 | 101; 5151	26| 396| 82.2	-7.22500000 3	-7.31600000 3	1.9-4| 6.0-7| 1.260	1:43
gka1d	101 | 101; 5151	27| 439| 96.5	-6.33300000 3	-6.52800000 3	1.1-5 | 2.5-7| 3.079	2:09
gka2d	101 | 101; 5151	27| 523| 84.1	-6.57900000 3	-6.99000000 3	1.7-6 | 6.9-7| 6.247	2:15
gka3d	101 | 101; 5151	26| 467| 96.9	-9.26100000 3	-9.73400000 3	1.4-5 | 4.8-7| 5.107	2:21
gka4d	101 | 101; 5151	28| 375| 104.9	-1.07270000 4	-1.12780000 4	1.4-6 | 4.7-7| 5.137	1:56
gka5d	101 | 101; 5151	26| 422| 91.5	-1.16260000 4	-1.23980000 4	2.3-6 | 6.9-7| 6.640	1:57
gka6d	101 | 101; 5151	27| 338| 102.4	-1.42070000 4	-1.49290000 4	1.9-6 | 5.2-7| 5.082	1:42
gka7d	101 | 101; 5151	27| 177| 75.3	-1.44760000 4	-1.53750000 4	6.2-7| 5.8-7| 6.210	40
gka8d	101 | 101; 5151	26| 271| 118.4	-1.63520000 4	-1.70050000 4	2.0-7| 7.1-7| 3.993	1:35
gka9d	101 | 101; 5151	26| 351| 63.9	-1.56560000 4	-1.65330000 4	7.2-7| 6.1-7| 5.602	1:10
gka10d	101 | 101; 5151	26| 213| 78.5	-1.91020000 4	-2.01080000 4	2.0-7| 7.2-7| 5.266	52
gka1e	201 | 201; 20301	29| 530| 97.3	-1.64640000 4	-1.70690000 4	5.2-5 | 7.9-7| 3.675	10:36
gka2e	201 | 201; 20301	29| 367| 103.4	-2.33950000 4	-2.49170000 4	4.7-7| 4.3-7| 6.506	7:23
gka3e	201 | 201; 20301	30| 559| 91.5	-2.52430000 4	-2.68980000 4	1.6-5 | 2.9-7| 6.556	10:22
gka4e	201 | 201; 20301	29| 512| 113.0	-3.55940000 4	-3.72250000 4	1.2-5 | 4.2-7| 4.582	11:25
gka5e	201 | 201; 20301	28| 510| 95.2	-3.51540000 4	-3.80020000 4	3.9-5 | 5.1-7| 8.101	9:46

Table 6.6: Results for the NAL algorithm on the BIQ problems. The entries under the column "%gap" are calculated with respect to the best solution listed, which is known to be optimal unless the symbol (†) is prefixed.

problem	$m - c_l \mid c_s; c_l$	it| itsub| pcg	best upper bound	lower bound $\underline{v}$	R_P| R_D| %gap	time
gka1f	501 | 501; 125751	30| 563| 102.8	†-6.11940000 4	-6.55590000 4	9.9-5 | 5.2-7| 7.133	1:28:54
gka2f	501 | 501; 125751	30| 624| 93.6	†-1.00161000 5	-1.07932000 5	6.6-5 | 5.7-7| 7.759	1:28:11
gka3f	501 | 501; 125751	30| 523| 120.4	†-1.38035000 5	-1.50152000 5	2.8-5 | 6.7-7| 8.778	1:31:34
gka4f	501 | 501; 125751	32| 571| 128.8	†-1.72771000 5	-1.87089000 5	8.7-6 | 4.0-7| 8.287	1:44:43
gka5f	501 | 501; 125751	31| 665| 90.5	†-1.90507000 5	-2.06916000 5	6.6-6 | 7.1-7| 8.613	1:25:48

Chapter 7

Conclusions

Along with recent developments on perturbation analysis of the problems under consideration, we introduced a semismooth Newton-CG augmented Lagrangian algorithm for solving large scale convex quadratic programming over symmetric cones including linear cone, positive semidefinite cone and second-order cone. Based on classic results of proximal point methods [102, 103], we analyze the global and local convergence of our NAL algorithm. Numerical experiments conducted on a variety of large scale convex quadratic symmetric cone programming problems demonstrated that our algorithm is very robust and efficient. However, there are still a number of interesting topics for our future research.

Under the strong second-order sufficient condition (3.26) and the primal constraint nondegeneracy (3.25), we analyze the rate of convergence of the NAL algorithm. However, it is still unclear to us on how to characterize the specific condition of the solution mapping for the dual or the primal-dual of the convex QSCP.

When applying the SNCG algorithm to solve inner problems, we choose the diagonal part of the generalized Hessian matrix as our preconditioner. Of course, this is too simple for some very ill-conditioned problems. Hence, by exploiting the specific structure of the generalized Hessian matrices, one may construct more efficient preconditioners to improve the performance of the semismooth Newton-CG algorithm at least for several subclasses of the problems discussed.

Moreover, there are many interesting applications for nonlinear symmetric cone programming. As an important application of convex QSCPs, the subproblems via the sequential quadratic programming approach for nonlinear symmetric cone optimization problems can be solved by our NAL algorithm. This may open up a way for studying more general nonlinear 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] F. Alizadeh and D. Goldfarb, *Second-order cone programming*, Mathematical Programming, 95 (2003), pp. 3-51.

[3] F. Alizadeh, J.P. A. Haeberly, and O.L. Overton, *Complementarity and nondegeneracy in semidefinite programming*, Mathemtical Programming, 77 (1997), pp. 111-128.

[4] M.F. Anjos, N.J. Higham, P.L. Takouda and H. Wolkowicz, *A semidefinite programming approach for the nearest correlation matrix problem*, Preliminary Research Report, Department of Combanitorics and Optimization, Waterloo, Ontario, 2003.

[5] K.M. Anstreicher, *Recent advances in the solution of quadratic assignment problems*,. Mathematical Programming, 97 (2003), pp. 27-42.

[6] J. Aspnes, D. Goldenberg, and Y. Yang, *The computational complexity of sensor network localization*, in proceedings of the first international workshop on algorithmic aspects of wireless sensor networks, 2004.

[7] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2001.

[8] L. Bergamaschi, J. Gondzio, and G. Zilli, *Preconditioning indefinite systems in interior point methods for optimization*, Computational Optimization and Applications, 28 (2004), pp. 149-171.

[9] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne, *The protein data bank*, Nucl. Acids Res., 28 (2000), pp. 235-242.

[10] D.P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*, Computer Science and Applied Mathematics, Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, 1982.

[11] D.P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*, Athena Scientific, Belmont, Massachusetts, 1996.

[12] P. Biswas and Y. Ye, *A distributed method for solving semidefinite programs arising from ad hoc wireless sensor network localization*, in Multiscale Optimization Methods and Applications, Nonconvex Optim. Appl. 82 (2006), pp. 69-84.

[13] P. Biswas, K.C. Toh and Y. Ye, *A distributed SDP approach for large scale noisy anchor-free graph realization with applications to molecular conformation*, SIAM Journal on Scientific Computing, 30 (2008), pp. 1251-1277.

[14] I.M. Bomze, M. Budinich, P.M. Pardalos, and M. Pelillo, *The maximum clique problem*, in: D.-Z. Du,P.M. Pardalos (eds), Handbook of Combinatorial Optimization, Supplement Volume A. Kluwer Academic Press, Boston, MA, USA, 1999.

[15] J.F. Bonnans and A. Shapiro, *Optimization problems with perturbations: A guided tour*, SIAM Review, 40 (1998), pp. 202-227.

[16] J.F. Bonnans and A. Shapiro, *Perturbation Analysis of Optimization Problems*, Springer, New York, 2000.

[17] B. Borchers and J.G. Young, *Implementation of a primal-dual method for SDP on a shared memory parallel architecture*, Computational Optimization and Applications, 37 (2007), pp. 355-369.

[18] S. Boyd and L. Xiao, *Least-squares covariance matrix adjustment*, SIAM Journal on Matrix Analysis and Applications, 27 (2005), pp. 532-546.

[19] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[20] S. Burer, R. Monterio, and Y. Zhang, *Maximum stable set formulations and heuristics based on continuous optimization*, Mathematical Programming, 94 (2002), pp. 137-166.

[21] S. Burer, R. Monterio, and Y. Zhang, *A computational study of a gradient-based log-barrier algorithm for a class of large scale SDPs*, Mathematical Programming, 95 (2003), pp. 359-379.

[22] S. Burer and D. Vandenbussche, *Solving Lift-and-Project relaxations of binary integer programs*, SIAM Journal on Optimization, 16 (2006), pp. 726-750.

[23] F. Cela, *The quadratic assignment problem: theory and algorithms*, Kluwer, Massachessets, USA, 1998.

[24] Z.X. Chan and D.F. Sun, *Constraint nondegeneracy, strong regularity and nonsigularity in semidefinite programming*, SIAM Journal on optimization, 19 (2008), pp. 370-396.

[25] B. Chazelle, C. Kingsford, and M. Singh, *The side-chain positioning problem: a semidefinite programming formulation with new rounding schemes*, Proceedings of the Paris C. Kanellakis memorial workshop on Principles of computing & knowledge: Paris C. Kanellakis memorial workshop on the occasion of his 50th birthday, pp. 86-94, ACM Press, 2003.

[26] J. Chen, X. Chen, and P. Tseng, *Analysis of nonsmooth vector-valued functions associated with second-order cones*, Mathematical Programming, 101 (2004), pp. 95-117.

[27] X. Chen, H.D. Qi, and P. Tseng, *Analysis of nonsmooth symmetric matrix functions with applications to semidefinite complementarity problems*, SIAM Journal on Optimization, 13 (2003), pp. 960-985.

[28] X. Chen and P. Tseng, *Non-interior continuation methods for solving semidefinite complementarity problems*, Mathematical Programming, 95 (2003), pp. 431-474.

[29] H. Ciria and J. Peraire, *Computation of upper and lower bounds in limit analysis using second-order cone programming and mesh adaptivity*, 9th ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability, 2004.

[30] J.O. Coleman and D.P. Scholnik, *Design of nonlinear-phase FIR filters with second-order cone programming*, Proc. 1999 Midwest Symp. on Circuits and Systems, Las Cruces, NM, August 1999.

[31] G. Crippen and T. Havel, *Distance Geometry and Molecular Conformation*, Wiley, 1988.

[32] F.H. Clarke, *Optimization and Nonsmooth Analysis*, Wiley, New York, 1983.

[33] C.C. Finger *A methodology to stress correlations*, RiskMetrics Monitor, Fourth Quarter (1997), pp. 3-11.

[34] G. Crippen and T. Havel, *Distance geometry and molecular conformation*, Wiley, 1988.

[35] L. Doherty, K.S.J. Pister, and L. El Ghaoui, *Convex position estimation in wireless sensor networks*, in Proc. 20th INFOCOM, IEEE Computer Society, 3 (2001), pp. 1655-1663.

[36] A. Eisenblätter, M. Grötschel, and A. M. C. A. Koster, *Frequency planning and ramification of coloring*, Discuss. Math. Graph Theory, 22 (2002), pp. 51-88.

[37] F. Facchinei, *Minimization of* SC^1 *functions and the Maratos effect*, Operation Research Letters, 17 (1995), pp. 131 - 137.

[38] J. Faraut, A. Korányi, *Analysis on Symmetric Cones*, Clarendon Press, Oxford, 1994.

[39] L. Faybusovich, *Linear systems in Jordan algebras and primal-dual interior-point algorithms*, Journal of Computational and Applied Mathematics , 86 (1997), pp. 149-175.

[40] M. Fukushima, Z.Q. Luo, and P. Tseng, *Smoothing functions for second-order-cone complementarity problems*, SIAM Journal on Optimization, 12 (2002), pp. 436-460.

[41] T. Fushiki, *Estimation of positive semidefinite correlation matrices by using convex quadratic semidefinite programming*, Neural computation, 21 (2009), pp. 2028-2048.

[42] G. Gallo, P.L. Hammer, and B. Simeone, *Quadratic knapsack problems*, Mathematical Programming, 12 (1980), pp. 132-149.

[43] Y. Gao and D.F. Sun, *Calibrating least squares covariance matrix problems with equality and inequality constraints*, Technical report, National University of Singapore, June 2008; Revised in June 2009.

[44] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders, *Preconditioners for indefinite systems arising in optimization*, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 292-311.

[45] G.H. Golub and C.F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore and London, 3rd edition, 1996.

[46] M.S. Gowda, R. Sznajder, and J. Tao, *Some P-properties for linear transformations on Euclidean Jordan algebras*, Linear Algebra and its Applications, 393, pp. 203-232, 2004.

[47] M. Grötschel, L. Lovász and A.. Schrijver, *Relaxations of vertex packing*, Journal of Combinatorial Theory, Series B 40 (1986), pp. 330-343.

[48] C. Helmberg, *Numerical evaluation of SBmethod*, Mathematical Programming, 95 (2003), pp. 381-406.

[49] B.A. Hendrichson, *The molecular problem: Determining conformation from pairwise distances*, Ph.D thesis, Cornell University, 1991.

[50] M.R. Hestenes, *Multiplier and gradient methods*, Journal of Optimization Theory and Applications, 4 (1969), pp. 303-320.

[51] N.J. Higham, *Computing the nearest correlation matrixa problem from finance*, IMA Journal of Numerical Analysis, 22 (2002), pp. 329-343.

[52] J.H. Z. Huang, N.P. Liu, M. Pourahmadi, and L.X. Liu, *Covariance matrix selection and estimation via penalised normal likelihood*, Biometrika, 93 (2006), pp. 85-98.

[53] G. Iyengar, D. Phillips, and C. Stein, *Approximation algorithms for semidefinite packing problems with applications to maxcut and graph coloring*, Lecture Notes in Computer Science, 3509 (2005), pp. 152-166.

[54] C. Jansson, *Termination and verification for ill-posed semidefinite programs*, Technical Report, Informatik III, TU Hamburg-Harburg, Hamburg, June 2005.

[55] F. Jarre and F. Rendl, *An augmented primal-dual method for linear conic programs*, SIAM Journal on Optimization problem, 19 (2008), pp. 808-823.

[56] D. Johnson, G. Pataki, and F. Alizadeh, *The seventh DIMACS implementation challenge: semidefinite and related optimization problems*, Rutgers University, http://dimacs.rutgers.edu/Challenges/, 2000.

[57] L.C. Kong, J. Sun, and N.H. Xiu, *A regularized smoothing Newton method for symmetric cone complementarity problems*, 19 (2008), pp. 1028-1047.

[58] A. Korányi, *Monotone functions on formally real Jordan algebras*, Mathematische Annalen, 269 (1984), pp. 73-76.

[59] M. Kočvara and M. Stingl, *PENNON - a code for convex nonlinear and semidefinite programming*, Optimization Methods and Software, 18 (2003), pp. 317-333.

[60] M. Kočvara and M. Stingl, *On the solution of large-scale SDP problems by the modified barrier method using iterative solvers*, Mathematical Programming, 109 (2007), pp. 413-444.

[61] J. Krarup and P.A. Pruzan, *Computer aided layout design*, Mathematical Programming Study, 9 (1978), pp. 75-94.

[62] H. Lebret, *Synthese de diagrammes de reseaux d'antennes par optimisation convexe*, Ph.D Thesis, Universite de Rennes 1, 1994.

[63] H. Lebret, *Antenna Pattern Synthesis through convex optimization*, in: Franklin T. Luk (Ed.), Proceedings of the SPIE Advanced Signal Processing Algorithms, 1995, pp. 182-192.

[64] H. Lebret and S. Boyd, *Antenna array Pattern Synthesis via convex optimization*, IEEE Transattions on Signal Processing, 45 (1997), pp. 526-532.

[65] S. Lele, *Euclidean distance matrix analysis (EDMA): estimation of mean form and mean form difference*, Mathematical Geology, 25 (1993), pp. 573-602.

[66] T.C. Liang, T.C. Wang, and Y. Ye, *A gradient search method to round the semidefinite programming relaxation solution for ad hoc wireless sensor network localization*, Technical report, Dept of Management Science and Engineering, Stanford University, August 2004.

[67] N. Linial, E. London, and Y. Rabinovich, *The geometry of graphs and some of its algorithmic applications*, Combinatorica, 15 (1995), pp. 215-245.

[68] Y.J. Liu and L.W. Zhang, *Convergence of the augmented Lagrangian method for nonlinear optimization problems over second-order cones*, Journal of Optimization Theory and Applications, 139 (2008), pp. 557-575.

[69] M.S. Lobo, L. Vandenberghe, S. Boyd and H. Lebret, *Applications of second-order cone programming*, Linear Algebra and its Applications, 284 (1998), pp. 193-228.

[70] L. Lovász, *On the Shannon capacity of a graph*, IEEE Transactions on Information Theory, 25 (1979), pp. 1-7.

[71] L. Lovász and A. Schrijver, *Cones of matrices and set-functions, and 0-1 optimization*, SIAM Journal on Optimization, 1 (1991), pp. 166-190.

[72] J. Malick, *A dual approach to semidefinite least-squares problems*, SIAM Journal on Matrix Analysis and Applications, 26 (2004), pp. 272-284.

[73] J. Malick, J. Povh, F. Rendl and A. Wiegele, *Regularization methods for semidefinite programming*, SIAM Journal on Optimization, 20 (2009), pp. 336-356.

[74] C. Mannino and A. Sassano, *An exact algorithm for the maximum stable set problem*, Journal of Computational Optimization and Applications, 3 (1994), pp. 243-258.

[75] R.D. McBride and J.S. Yormark, *An implicit enumeration algorithm for quadratic integer programming*, Management Science, 26 (1980), pp. 282-296.

[76] F.W. Meng, D.F. Sun, G.Y. Zhao, *Semismoothness of solutions to generalized equations and the Moreau-Yosida regularization*, Mathematical Programming Ser. B, 104 (2005), pp. 561-581.

[77] R. Mifflin, *Semismooth and semiconvex functions in constrained optimization*, SIAM Journal on Control and Optimization, 15 (1977), pp. 957-972.

[78] J.J. Moreau, *Déomposition orthogonale d'un espace hilbertien selon deux côes mutuellement polaires*, C.R. Acad. Sci., Paris, 255 (1962), pp. 238-40.

[79] J. Moré and Z. Wu. *ε-optimal solutions to distance geometry problems via global continuation*, Mathematics and Computer Science Division,, 23 (1995), pp. 151-168.

[80] J. Moré and Z. Wu, *Global continuation for distance geometry problems*, SIAM Journal on Optimization., 7 (1997), pp. 814-836.

[81] J.W. Nie and Y.X. Yuan, *A predictor-corrector algorithm for QSDP combining Dikin-type and Newton centering steps*, Annals of Operations Research, 103 (2001), pp. 115-133.

[82] Y.E. Nesterov and A. Nemirovski, *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM, Philadelphia, 1994.

[83] J.S. Pang and L. Qi, *A globally convergent Newton method of convex SC^1 minimization problems*, Journal of Optimization Theory and Application, 85 (1995), pp. 633-648.

[84] P.M. Pardalos, F. Rendl, and H. Wolkowicz, *The quadratic assignment problem: a survey and recent developments*, In Quadratic Assignment and Related Problems, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 16 (1994), pp. 1-42, AMS.

[85] P.M. Pardalos, D. Shalloway, and G. Xue, *Global minimization of nonconvex energy functions: molecular conformation and protein folding*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 23, American Mathematical Society, Providence, RI, 1996.

[86] P. Pardalos and J. Xue, *The maximum clique problem*, Journal of Global Optimization, 4 (1994), pp. 286-301.

[87] J. Povh and F. Rendl, *Copositive and semidefinite relaxations of the quadratic assignment problem*, Discrete Optimization, 6 (2009),pp. 231-241.

[88] J. Povh, F. Rendl, and A. Wiegele, *A boundary point method to solve semidefinite programs*, Computing, 78 (2006), pp. 277-286.

[89] M.J.D. Powell, *A method for nonlinear constraints in minimization problems*, In Fletcher, R. (ed.) Optimization, pp. 283C298, Academic, New York, 1972.

[90] H.D. 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.

[91] H.D. Qi and D.F. Sun, *Correlation stress testing for value-at-risk: an unconstrained convex optimization approach*, to appear in Computational Optimization and Applications.

[92] H.D. Qi and D.F. Sun, *An augmented Lagrangian dual approach for the H-weighted nearest correlation matrix problem*, to appear in IMA Journal of Numerical Analysis.

[93] L.Q. Qi and J. Sun, *A nonsmooth version of Newtons method*, Mathematical Programming, 58 (1993), pp. 353-367.

[94] R. Reams, G. Chatham, W. K. Glunt, D. McDonald, and T. L. Hayden, *Determining protein structure using the distance geometry program APA*, Computers & Chemistry, 23 (1999), pp. 153-163.

[95] F. Rendl and R. Sotirov, *Bounds for the quadratic assignment problem using the bundle method*, Mathematical Programming, 109 (2006), pp. 505-524.

[96] S.M. Robinson, *Local structure of feasible sets in nonlinear programming, Part II: nondegeneracy*, Mathematical Programming Study, 22 (1984), pp. 217-230.

[97] S.M. Robinson, *Local structure of feasible sets in nonlinear programming, Part III: stability and sensitivity*, Mathematical Programming Study, 30 (1987), pp. 45-66.

[98] S.M. Robinson, *Constriant nondegeneracy in variational analysis*, Mathematics of Operations Research, 28 (2003), pp. 201-232.

[99] R.T. Rockafellar, *Convex Analysis*, Princenton University Press, New Jersey, 1970.

[100] R.T. Rockafellar, *A dual approach to solving nonlinear programming problems by unconstrained optimization*, Mathematical Programming, 5 (1973), pp. 354-373.

[101] R.T. Rockafellar, *Conjugate Duality and Optimization*, Regional Conference Series in Applied Mathematics, SIAM Publication, 16, 1974.

[102] R.T. Rockafellar, *Augmented Lagrangains and applications of the proximal point algorithm in convex programming*, Mathematics of Operation Research, 1 (1976), pp. 97-116.

[103] R.T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM Journal on Control and Optimization, 14 (1976), pp. 877-898.

[104] S. Sahni and T. Gonzalez, *P-Complete Aproximation Problems*, Journal of the Association for Computing Machinery, 23 (1976), pp. 555-565.

[105] J.B. Saxe, *Embeddability of weighted graphs in k-space is strongly NP-hard*, In Proceedings of the 17th Allerton Conference on Communication, Control and Computing, pp. 480-489, 1979.

[106] S.H. Schmieta, F. Alizadeh, *Extension of primalCdual interior point algorithms to symmetric cones*, Mathematical Programming, 96 (2003), pp. 409-438.

[107] E.C. Sewell, *A branch and bound algorithm for the stability number of a sparse graph*, INFORMS Journal on Computing, 10 (1998), pp. 438-447.

[108] A. Shapiro, *First and second order analysis of nonlinear semidefinite programs*, Mathematical Programming, Series B, 77 (1997), pp. 301-320.

[109] A. Shapiro, *On differentiability of symmetric matrix valued functions*, Optimization Online, July 2002.

[110] A. Shapiro, *Sensitivity analysis of generalized equations*, Journal of Mathematical Sciences, 15 (2003), pp. 2554-2565.

[111] N. Sloane, *Challenge problems: independent sets in graphs*, http://research.att.com/ njas/doc/graphs.html, 2005.

[112] J.F. Sturm, *Superlinear convergence of an algorithm for monotone linear complementarity problems, when no strictly complementary solution exists*, Mathematics of Operations Research, 24 (1999), pp. 72-94.

[113] J.F. Sturm, *Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones*, Optimization Methods and Software, 11-12 (1999), pp. 625-653.

[114] D.F. Sun, *The strong second order sufficient condition and constraint nondegeneracy in nonlinear semidefinite programming and their implications*, Mathematics of Operations Research, 31 (2006), pp. 761-776.

[115] D.F. Sun and J. Sun, *Semismooth matrix valued functions*, Mathematics of Operations Research, 27 (2002), pp. 150-169.

[116] D.F. Sun and J. Sun, *Löwner's operator and spectral functions in Euclidean Jordan algebras*, Mathematics of Operations Research, 33 (2008), pp. 421-445.

[117] D.F. Sun, J. Sun and L.W. Zhang, *The rate of convergence of the augmented Lagrangian method for nonlinear semidefinite programming*, Mathematical Programming, 114 (2008), pp. 349-391.

[118] D.F. Sun, Y. Wang and L.W. Zhang, *Optimization over Symmetric Cones: Variational and Sensitivity Analysis*, Manuscript, National University of Singapore, December 2007.

[119] N.V. Tretyakov, *A method of penalty estimates for convex programming problems*, Èkonomika i Matematicheskie Metody, 9 (1973), pp. 525-540.

[120] M. Trick, V. Chvatal, W. Cook, D. Johnson, C. McGeoch, and R. Tarjan, *The Second DIMACS Implementation Challenge: NP Hard Problems: Maximum Clique, Graph Coloring, and Satisfiability*, Rutgers University, http://dimacs.rutgers.edu/Challenges/, 1992.

[121] 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.

[122] K.C. Toh, *An inexact primal-dual path-following algorithm for convex quadratic SDP*, Mathematical Programming, 112 (2007), pp. 221-254.

[123] K.C. Toh, R.H. Tutuncu, and M.J. Todd, *Inexact primal-dual path-following algorithms for a special class of convex quadratic SDP and related problems*, Pacific Journal of Optimization, 3 (2007), pp. 135-164.

[124] L.N. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

[125] P. Tseng, *Second-order cone programming relaxation of sensor network localization*, SIAM Journal on Optimization, 18 (2007), pp. 156-185.

[126] K.M. Tsui, S.C. Chan, and K.S. Yeung, *Design of FIR digital filters with prescribed flatness and peak error constraints using second order cone programming*, IEEE Transactions on Circuits and Systems II, 52 (2005), pp. 601-605.

[127] S. Turkay, E. Epperlein, and N. Christofides, *Correlation stress testing for value at risk*, The Journal of Risk, 5 (2003), pp. 75-89.

[128] R.H. Tütüncü, K.C. Toh, and M.J. Todd, *Solving semidefinite-quadratic-linear programs using SDPT3*, Mathematical Programming, 95 (2003), pp. 189-217.

[129] Y. Wang, *Perturbation Analysis of Optimization Problems over Symmetric Cones*, Ph.D thesis, Dalian University of Technology, 2008.

[130] K.Q. Weinberger, F. Sha and L.K. Saul, *Learning a kernel matrix for nonlinear dimensionality reduction*, ACM International Conference Proceeding Series, 69 (2004), pp. 106-115.

[131] B.H. Wellenhoff, H. Lichtenegger, and J. Collins, *Global positions system: theory and practice*, Fourth Edition, Springer Verlag, 1997.

[132] A. Wiegele, *Biq Mac library – a collection of Max-Cut and quadratic 0-1 programming instances of medium size*, Technical report, 2007.

[133] H. Wolkowicz, R. Saigal and L. Vandenberghe, *Handbook of Semidefinite Programming: Theory, Algorithms and Applications*, Kluwer Academic Publishers, 2000.

[134] S.J. Wright, *Primal-Dual Interior Methods*, SIAM, Philadelphia, 1997.

[135] K. Wüthrich, *NMR of proteins and nucleic acids*, John Wiley & Sons, New York, 1986.

[136] Y.Y. Ye, and J.W. Zhang, *An improved algorithm for approximating the radii of point sets*, In RANDOM-APPROX, pp. 178-187, 2003.

[137] J.M. Yoon, Y. Gad, and Z. Wu, *Mathematical modeling of protein structure using distance geometry*, Technical report, Department of Computational & Applied Mathematics, Rice University, 2000.

[138] E.H. Zarantonello, *Projections on convex sets in Hilbert space and spectral theory I and II*, In Zarantonello, E.H. (ed.) Contributions to Nonlinear Functional Analysis, pp. 237-424, Academic, New York, 1971.

[139] X.Y. Zhao, D.F. Sun and K.C. Toh, *A Newton-CG augmented Lagrangian method for semidefinite programming*, Technical report, National University of Singapore, March 2008; Revised in February 2009 and July 2009.

A SEMISMOOTH NEWTON-CG AUGMENTED LAGRANGIAN METHOD FOR LARGE SCALE LINEAR AND CONVEX QUADRATIC SDPS

ZHAO XINYUAN

NATIONAL UNIVERSITY OF SINGAPORE

2009