STUDIES

ON

ALGORITHMS FOR SOLVING GENERALIZED

SECOND-ORDER CONE PROGRAMMING PROBLEMS

TAKAYUKI OKUNO

STUDIES
ON
ALGORITHMS FOR SOLVING GENERALIZED SECOND-ORDER CONE PROGRAMMING PROBLEMS

by

TAKAYUKI OKUNO

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

KYOTO UNIVERSITY
KYOTO 606–8501, JAPAN
JANUARY 2013

Preface

The second-order cone (SOC), also called the Lorentz cone, is a special class of convex cones. So far there have been developed a lot of studies analyzing the structure of the SOC from geometric and algebraic standpoints. The second-order cone programming problem (SOCP) is an optimization problem that contains SOCs in its constraints. It is well-known that SOCP represents a wide class of optimization problems. In fact, standard nonlinear programming problems can be regarded as a subclass of SOCP naturally, and a class of robust optimization problems and convex quadratic programming problems can be reformulated as linear SOCPs (LSOCPs). In the last decade, study on the SOCP has been promoted significantly. For example, a lot of efficient algorithms have been proposed for solving SOCPs, such as primal-dual interior-point method, smoothing Newton method, SQP-type method, and so on. Especially, for solving the LSOCP, several software packages implementing the primal-dual interior point method have been produced, and their superior performance has been reported by a large number of researchers. Also, SOCP finds a wide variety of applications in the real world, such as the antenna array weight design problem, the finite impulse response (FIR) design problem, the portfolio optimization with loss risk constraints, the magnetic shield design problem for maglev trains, and so on. Thanks to such useful algorithms and applications, SOCP attracts much attention from a lot of people in various fields, and is expected to play an important role in the future. However, we are often faced with a number of practical problems that can be formulated as optimization problems with SOC constraints whose structures are quite different from that of the standard SOCP. To deal with such problems in the real world, we need to consider new enhanced SOCP models and develop efficient methods for solving them.

In this thesis, we focus on two classes of problems which are generalizations of the SOCP. The first one is a *semi-infinite second-order cone programming problem* (SISOCP), which is featured as an optimization problem with an infinite number of SOC constraints. The second one is a *mathematical programming problem with SOC complementarity constraints* (MPSOCC). Though the MPSOCC is superficially regraded as an ordinary SOCP, we cannot handle it in the conventional framework established for solving the SOCP because of the complementarity constraints. Many important problems in the real world can be represented as SISOCP or MPSOCC straightforwardly. For example, FIR filter-design and Chebyshev-like approximation problems involving vector-valued functions can be formulated as SISOCPs, and a bilevel-programming problem that possesses a parametric SOCP as a lower-level problem can be expressed as MPSOCC.

The main purpose of the thesis is to propose numerical methods for SISOCP and MPSOCC. To solve the SISOCP, we propose two different algorithms. The first one is based on an exchange-type method, where we generate iteration points that are optima of relaxed SISOCPs with finitely many SOC constraints. The second one is based on a local reduction method, where the SISOCP is reduced to an SOCP locally by means of finitely many implicit functions. Either method has to compute an optimal solution of SOCP in each iteration, which can be obtained by existing algorithms quite efficiently. To solve the MPSOCC, we propose a smoothing SQP method, where we apply the SQP method combined with a smoothing technique to the reformulated MPSOCC without SOC complementarity constraints. By using approximated solutions of smoothed problems, the proposed method attains practical efficiency.

The author hopes that the results of this thesis make some contributions to the further development of research on optimization problems with SOC constraints.

Takayuki Okuno
January 2013

Acknowledgment

First of all, I am deeply grateful to Professor Masao Fukushima of Kyoto University. Throughout my master's and doctoral course, he carefully read my manuscripts and gave me a lot of advices. What I have learned from him is not only basic methodologies for research but also fundamental attitude of researchers. With such a spirit in my mind, I will try to improve myself so that I can become an excellent researcher like him in the future. Secondly, I would like to express my appreciation to Assistant Professor Shunsuke Hayashi of Kyoto University. He always encouraged and supervised me since I came to Professor Fukushima's laboratory. Many of results attained in this thesis are due to his help. Without his presence, I would not have entered the doctoral course. Thirdly, I would like to thank Associate Professor Nobuo Yamashita of Kyoto University. He always answered my trivial question sincerely. Also, his deep insights into the optimization field were incentive for me. His advice often became an hint to promote my research from different viewpoints. In addition, I would like to tender my acknowledgments to Mrs Fumie Yagura who is a secretary of the laboratory. She kindly took care of me and handled my faculty documents.

I am grateful to the OBs and members in the laboratory. It was very pleasant for me to discuss and study the optimization together with them. Especially, I am very thankful to Kenji Ueda, Munetaka Kanayama, Noritoshi Kurokawa, Ryoich Nishimura, Katsunori Kubota, Ryota Otsubo, Daisuke Yamamoto, Kenji Hamatani, and Takahiro Nishi. They helped me when I faced some difficulties. Further, my heartfelt appreciation goes to Assistant Professor Hiroshige Dan of Kansai University. He did not only give encouragement to me but also prepared a letter of recommendation for my job interview while he was busy with his own work.

I also warmly thank my colleagues in Karatedo club of Kyoto University who brought me both physical and mental strengths for these 10 years.

Finally, I would like to express my highest gratitude to my family supporting and encouraging me generously during my student life.

List of Tables

List of Figures

Contents

Chapter 1

Introduction

The second-order cone programming problem (SOCP) is an optimization problem that contains a finite number of so-called *second-order cone* (SOC) constraints. The SOCP has attracted a lot of researchers in various fields, since many practical problems can be represented as an SOCP, and the obtained SOCP can be solved quite efficiently by using existing interior-point solvers [65, 68]. However, we are often faced with practical problems that can be formulated as optimization problems with SOC constraints whose structures are quite different from that of the conventional SOCP model. To deal with such problems in the real world, we need to consider some new enhanced SOCP models and examine their properties precisely. In this thesis, we focus on two classes of optimization problems which are extensions of the SOCP. The first one is a *semi-infinite second-order cone programming problem* (SISOCP) which contains an infinite number of SOC constraints. The second one is a *mathematical program with SOC complementarity constraints* (MPSOCC). The main purpose of the thesis is to propose and develop algorithms for solving them.

In this chapter, we provide with overviews of SOCP, SISOCP, and MPSOCC, and then outline the contents of the thesis.

1.1 Overview of problems

1.1.1 Second-order cone programming problem

The *second-order cone programming problem* (SOCP) is an optimization problem of the following form:

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{Minimize}} \quad & f(x) \\ \text{subject to} \quad & g^0(x) = 0, \\ & g^s(x) \in \mathcal{K}^{m_s} \ (s = 1, 2, \ldots, S), \end{aligned} \tag{1.1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g^s : \mathbb{R}^n \to \mathbb{R}^{m_s}$ are continuously differentiable functions, and $\mathcal{K}^{m_s} \subseteq \mathbb{R}^m$ for each $s = 1, 2, \ldots, S$ denotes the m_s-dimensional second-order cone defined by

$$\mathcal{K}^{m_s} := \begin{cases} \left\{(z_1, \tilde{z}^\top)^\top \in \mathbb{R} \times \mathbb{R}^{m_s-1} \mid z_1 \geq \|\tilde{z}\|\right\} & (m_s \geq 2) \\ \mathbb{R}_+ := \{z \in \mathbb{R} \mid z \geq 0\} & (m_s = 1). \end{cases} \tag{1.1.2}$$

When $m_1 = m_2 = \cdots = m_S = 1$, SOCP (1.1.1) naturally reduces to the standard nonlinear programming problem. Hence, the SOCP include the nonlinear programming problems as a subclass. Studies on the SOCP have been advanced significantly in the last decade in both theoretical and practical aspects. Especially, development of research on the linear SOCP (LSOCP), which consists of linear objective function and affine constraint functions, is notable. The LSOCP has wide applications in economics and engineering such as robust portfolio selection [22, 20], filter design [70, 72, 8, 40], magnetic shield design [58], and so on [37, 1]. The primal-dual interior-point method [37, 1, 69, 42] is well known as an effective algorithm for solving the LSOCP, and some software packages implementing them [65, 68] have been produced. Compared with the LSOCP, the general case (1.1.1) is more complicated and has been studied not so much as the LSOCP. So far, for solving them, there have been proposed the augmented Lagrangian method [36], primal-dual interior point method [76], SQP-type method [33], and so on.

1.1.2 Semi-infinite second-order cone programming problem

The classical semi-infinite programming problem (SIP) is characterized as an optimization problem with a finite number of variables and an infinite number of inequality constraints. The SIP has been studied extensively. It also has wide applications in economics and engineering, e.g., the air pollution control, the robot trajectory planning, the stress of materials, etc.[29, 38]. So far, many efficient algorithms have been proposed for solving SIPs, such as the discretization-type methods [21, 28, 54, 64], local reduction-type methods [24, 30, 46, 50, 51, 66], exchange-type methods [21, 27, 34, 73, 47, 79], Newton-type methods [44, 62, 63, 35, 53, 61, 7, 52], and so on [16, 73, 31, 60].

In the thesis, we consider the following SIP with infinitely many SOC constraints:

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{Minimize}} & f(x) \\ \text{subject to} & g^0(x) = 0, \\ & g^s(x,t) \in \mathcal{K}^{m_s} \quad \text{for all } t \in T^s \ (s = 1, 2, \ldots, S), \end{array} \tag{1.1.3}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g^0 : \mathbb{R}^n \to \mathbb{R}^{m_0}$ and $g^s : \mathbb{R}^n \times \mathbb{R}^{\ell_s} \to \mathbb{R}^{m_s}$ $(s = 1, 2, \ldots, S)$ are continuous functions, and $T^s \subseteq \mathbb{R}^{\ell_s}$ $(s = 1, 2, \ldots, S)$ are nonempty *compact* index sets, and $\mathcal{K}^{m_s}$ is the m_s-dimensional SOC defined by (1.1.2) for each $s = 1, 2, \ldots, S$. We call this problem the *semi-infinite second-order cone programming problem* abbreviated as SISOCP. SISOCP (1.1.3) includes the classical SIP and SOCP as subclasses. In fact, if each index set T_s $(s = 1, 2, \ldots, S)$ consists of finitely many elements, then SISOCP (1.1.3) is nothing but SOCP (1.1.1). On the other hand, when $m_1 = m_2 = \cdots = m_S = 1$, i.e., $\mathcal{K}^{m_s} = \mathbb{R}_+$ for $s = 1, 2, \ldots, S$, SISOCP (1.1.3) reduces to the standard SIP.

There are some important applications of SISOCP (1.1.3). For example, SISOCP (1.1.3) can be used to formulate a Chebyshev-like approximation problem involving vector-valued functions. Specifically, let $X \subseteq \mathbb{R}^\ell$ be a nonempty set, $Y \subseteq \mathbb{R}^n$ be a given compact set, and $\Phi : Y \to \mathbb{R}^m$ and $F : \mathbb{R}^\ell \times Y \to \mathbb{R}^m$ be given functions. Then, we seek a parameter $u \in X$ such that

$\Phi(y) \approx F(u, y)$ for all $y \in Y$. One relevant approach is to solve the following problem:

$$\underset{u \in X}{\text{Minimize}} \quad \max_{y \in Y} \|\Phi(y) - F(u, y)\|. \tag{1.1.4}$$

By introducing the auxiliary variable $r \in \mathbb{R}$, we can transform the above problem to

$$\begin{aligned} &\underset{(u,r) \in X \times \mathbb{R}}{\text{Minimize}} && r \\ &\text{subject to} && \begin{pmatrix} r \\ \Phi(y) - F(u, y) \end{pmatrix} \in \mathcal{K}^{m+1} \ \text{ for all } y \in Y, \end{aligned} \tag{1.1.5}$$

which is of the form (1.1.3).

Another important application for SISOCP (1.1.3) is a *finite impulse response* (FIR) filter-design [55, 72]. Generally, the FIR filter-design is to determine a vector $h := (h_0, h_1, \cdots, h_{n-1})^\top \in \mathbb{R}^n$ such that the *frequency response* function $H : \mathbb{R}^n \times \mathbb{R} \to \mathbb{C}$ defined by $H(h, \omega) := \sum_{k=0}^{n-1} h_k e^{-k\omega\sqrt{-1}}$ satisfies some given conditions for all $\omega \in [\omega_1, \omega_2] \subseteq [0, 2\pi]$. The following problem is called the *Log-Chebyshev approximation FIR filter problem* [72]:

$$\underset{h \in \mathbb{R}^n}{\text{Minimize}} \quad \sup_{\omega \in [0,\pi]} \big| \log |H(h, \omega)| - \log D(\omega) \big|, \tag{1.1.6}$$

where $D : [0, \pi] \to \mathbb{R}$ is a given *desired frequency magnitude* such that $D(\omega) > 0$ for all $\omega \in [0, \pi]$. By letting $R(h, \omega) := |H(h, \omega)|^2$ and using an auxiliary variable $r \in \mathbb{R}$, problem (1.1.6) is expressed as

$$\begin{aligned} &\underset{(r,h) \in \mathbb{R} \times \mathbb{R}^n}{\text{Minimize}} && r \\ &\text{subject to} && 1/r \le R(h, \omega)/D(\omega)^2 \le r, \\ & && R(h, \omega) \ge 0 \ \text{ for all } \omega \in [0, \pi], \end{aligned}$$

which can further be rewritten as

$$\begin{aligned} &\underset{(r,h) \in \mathbb{R} \times \mathbb{R}^n}{\text{Minimize}} && r \\ &\text{subject to} && rD(\omega)^2 - R(h, \omega) \ge 0, \ R(h, \omega) \ge 0, \\ & && \begin{pmatrix} R(h, \omega) + r \\ R(h, \omega) - r \\ 2D(\omega) \end{pmatrix} \in \mathcal{K}^3 \ \text{ for all } \omega \in [0, \pi], \end{aligned} \tag{1.1.7}$$

which is of the form SISOCP (1.1.3) with variables $(r, h) \in \mathbb{R} \times \mathbb{R}^n$. In [39, 70], other kinds of filter design are considered and those design problems are formulated as SISOCPs with infinitely many SOC constraints. However, such problems are actually solved via a uniform discretization.

Although such important applications exist, there have been very few studies on the SISOCP. One of possible reasons is that any closed convex cone including SOC can be represented as an intersection of finitely or infinitely many halfspaces. Then, we may reformulate (1.1.3) as a classical SIP with infinitely many inequality constraints and solve it by using existing SIP algorithms. However, such a reformulation approach brings about serious difficulties since the dimension of the index set may become much larger than that of the original SISOCP (1.1.3). To see this, we consider SISOCP (1.1.3) with a single index set $T \subseteq \mathbb{R}^\ell$ and a semi-infinite SOC constraint

$g(x,t) \in \mathcal{K}^m$ $(t \in T)$. Since $\mathcal{K}^m = \{z \in \mathbb{R}^m \mid z^\top v \geq 0, \forall v \in V\}$, where $V := \{(1,\bar{v})^\top \in \mathbb{R}^m \mid \|\bar{v}\| = 1\}$, SISOCP (1.1.3) can be reformulated as the SIP: min $f(x)$ s.t. $v^\top g(x,t) \geq 0$ for all $(v,t) \in V \times T$. The dimension of $V \times T$ is then equal to $m + \ell - 1$.

In the thesis, we will propose two algorithms for solving SISOCP (1.1.3) while maintaining the SOC structures. The first one is a regularized explicit exchange method, which is developed for the SISOCP with a convex objective function and infinitely many affine SOC constraint functions. It solves a sequence of finitely relaxed SISOCPs to generate iteration points. The second one is a local reduction based SQP-type method, where SISOCP (1.1.3) is locally represented as an SOCP by means of implicit functions and the obtained SOCP is solved by an SQP-type method to generate a search direction.

1.1.3 Mathematical program with SOC complementarity constraints

A mathematical program with equilibrium constraints (MPEC) [41] is an optimization problem whose constraints contain parametric variational inequalities. The MPEC has been studied extensively, since it finds wide applications such as design problems in engineering, equilibrium problems in economics, and game-theoretic multi-level optimization problems. Particularly, variational inequality constraints in MPECs are often written as linear or nonlinear complementarity constraints. Such an MPEC is also called a mathematical program with complementarity constraints (MPCC), for which there have been proposed many algorithms. For example, Fukushima and Tseng [19] proposed an active set algorithm, and proved that any accumulation point of the sequence generated by the algorithm is a B-stationary point under the uniform linear independence constraint qualification (LICQ) on the ε-feasible set. Luo, Pang, and Ralph [41] proposed a piece-wise sequential quadratic programming (SQP) algorithm, and showed that the generated sequence converges to a B-stationary point locally superlinearly or quadratically under the LICQ and the second order sufficient conditions. Fukushima, Luo, and Pang [17] proposed an SQP-type algorithm, and showed that the sequence generated by the algorithm globally converges to a B-stationary point under the nondegeneracy condition at the limit point.

In the thesis, we consider the following mathematical program with linear SOC complementarity constraints (MPSOCC):

$$\begin{array}{ll} \underset{x,y,z}{\text{Minimize}} & f(x,y) \\ \text{subject to} & Ax \leq b, \\ & z = Nx + My + q, \\ & \mathcal{K} \ni y \perp z \in \mathcal{K}, \end{array} \tag{1.1.8}$$

where $f\colon \mathbb{R}^{n+m} \to \mathbb{R}$ is a continuously differentiable function, $A \in \mathbb{R}^{p\times n}$, $b \in \mathbb{R}^p$, $N \in \mathbb{R}^{m\times n}$, $M \in \mathbb{R}^{m\times m}$ and $q \in \mathbb{R}^m$ are given matrices and vectors, $\perp$ denotes the perpendicularity, and $\mathcal{K}$ is the Cartesian product of second-order cones, that is, $\mathcal{K} := \mathcal{K}^{m_1} \times \mathcal{K}^{m_2} \times \cdots \times \mathcal{K}^{m_S} \subseteq \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \cdots \times \mathbb{R}^{m_S} = \mathbb{R}^\ell$. A typical application of MPSOCC (1.1.8) is a bilevel-programming problem [2, 14] that possesses a parametric SOCP as a lower-level problem. By replacing the lower-level SOCP with its Karush-Kuhn-Tucker (KKT) conditions, it is formulated as MPSOCC (1.1.8).

When $m_s = 1$ for all s, i.e., $\mathcal{K} = \mathbb{R}^\ell_+$, MPSOCC (1.1.8) reduces to the MPCC with linear nonnegative complementarity constraints. In addition, since $\mathcal{K} \ni y \perp z \in \mathcal{K}$ and $z = Nx+My+q$ can be rewritten as a certain parametric variational inequality constraint, MPSOCC (1.1.8) can be classified as a special case of MPEC. One may think that MPSOCC (1.1.8) can be regarded as an SOCP by viewing $\mathcal{K} \ni y \perp z \in \mathcal{K}$ as $y \in \mathcal{K}$ and $z \in \mathcal{K}$ and $y^\top z = 0$, and thus existing algorithms for SOCP are applicable for the MPSOCC. However, due to the fact that Mangasarian-Fromovitz constraint qualification or LICQ are violated at all feasible points of the MPSOCC [41], we cannot apply the convergence theories for SOCP to MPSOCC (1.1.8) straightforwardly. Thus, we need to establish specialized theories and algorithms for solving MPSOCC (1.1.8).

A second-order cone complementarity problem (SOCCP) is one of the most important problems relevant to MPSOCC (1.1.8). The SOCCP is characterized as a problem to find a triple $(x, y, z) \in \mathbb{R}^\nu \times \mathbb{R}^n \times \mathbb{R}^n$ satisfying SOC complementarity condition $\mathcal{K} \ni y \perp z \in \mathcal{K}, F(x, y, z) = 0$, where $F : \mathbb{R}^\nu \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^\nu$ is a continuously differentiable vector valued function. The existence and uniqueness properties of solutions to the SOCCP are analyzed in terms of Jordan algebra [23, 67]. For solving SOCCP, a number of algorithms have been proposed such as the smoothing methods [18, 26, 12, 43], matrix splitting methods [25, 74] and so on [10, 9, 49]. Most of them are designed based on the reformulation approach, in which the SOC complementarity condition is reformulated as an equivalent system of equations by means of merit functions involving the Fischer-Burmeister function or the natural residual [18]. As applications of the SOCCP, the robust Nash equilibrium problem in an *N*-person non-cooperative game [45] and a certain equilibrium problem with frictional contact [32, 78] can be formulated as SOCCPs. Particularly, since the KKT conditions for SOCP (1.1.1) are represented as SOCCPs, SOCPs can be solved through the SOCCP.

Compared with the MPCC and SOCCP, there are only a few studies on the MPSOCC. Recently, Yan and Fukushima [77] proposed a smoothing method for solving such a problem. To show convergence of the algorithm, they assume that smoothed subproblems are solved exactly. However, such an approach takes a large amount of costs for computing solutions of those subproblems. In Chapter 5 of the thesis, we propose a more practical algorithm combining the smoothing method with the SQP method, which solves at each iteration a convex quadratic program that approximates a smoothed subproblem, instead of solving the latter problem exactly.

1.2 Outline of the thesis

The thesis is organized as follows. In Chapter 2, as preliminaries, we give some notations, fundamental properties, and mathematical techniques that are necessary in the later arguments. Chapters 3 and 4 are devoted to the study of SISOCP (1.1.3). More precisely, in Chapter 3, we consider SISOCP (1.1.3) with a convex objective function and infinitely many affine SOC constrains. For solving such SISOCPs, we combine a regularization scheme and an exchange method. With the help of the regularization technique, we can verify that, under mild assumptions, generated iteration points globally converge to the optimal solution that attains the least 2-norm value among the optimal solutions of the SISOCP. In Chapter 4, we construct an SQP-type method for solving the SISOCP of the form (1.1.3). This method locally represents infinitely many SOC constraints with finitely many SOC constraints by using implicit functions. We make global and local convergence analyses. In Chapter 5, we propose an algorithm for solving MPSOCC (1.1.8). This method first replaces the SOC complementarity constraints with equality constraints using the smoothing natural residual function, and apply the SQP method to the smoothed problems with decreasing the smoothing parameters. We show that iteration points produced by the proposed method converge to a B-stationary point of MPSOCC (1.1.8) under the strict complementarity condition. In Chapter 6, we end the thesis with some concluding remarks and future works.

Chapter 2

Preliminaries

In this chapter, we give some fundamental notations and properties that will be useful in the subsequent chapters.

2.1 Notations

Throughout the thesis, we use the following notations. We denote the m-dimensional nonnegative and positive cones by

$$\mathbb{R}^m_+ := \{z \in \mathbb{R}^m \mid z_i \geq 0, i = 1, 2, \ldots, m\},$$

and

$$\mathbb{R}^m_{++} := \{z \in \mathbb{R}^m \mid z_i > 0, i = 1, 2, \ldots, m\},$$

respectively. The m-dimensional second-order cone is denoted by

$$\mathcal{K}^m := \begin{cases} \left\{ (z_1, z_2, \ldots, z_m)^\top \in \mathbb{R}^m \mid z_1 \geq \sqrt{\sum_{i=2}^m (z_i)^2} \right\} & (m \geq 2) \\ \mathbb{R}^1_+ & (m = 1). \end{cases}$$

We also denote the set of m-dimensional symmetric positive-semidefinite matrices and the set of symmetric positive-definite matrices by S^m_+ and S^m_{++}, respectively.

For a vector $z := (z_1, z_2, \ldots, z_n)^\top \in \mathbb{R}^n$, the 1-, 2-, and ∞-norms of z are defined by

$$\|z\|_1 := |z_1| + |z_2| + \cdots + |z_n|,$$
$$\|z\|_2 := \sqrt{(z_1)^2 + (z_2)^2 + \cdots + (z_n)^2},$$

and

$$\|z\|_\infty := \max\left(|z_1|, |z_2|, \ldots, |z_n|\right),$$

respectively. Particularly, we often use $\|\cdot\|$ to denote the 2-norm, instead of $\|\cdot\|_2$. For a matrix $M \in \mathbb{R}^{n\times n}$, $\|M\|$ denotes the operator norm defined by

$$\|M\| = \max_{\|x\|_2=1} \|Mx\|_2.$$

For $z_1 + \sqrt{-1}z_2 \in \mathbb{C}$ with $z_1, z_2 \in \mathbb{R}$, its absolute value is defined by

$$|z_1 + \sqrt{-1}z_2| := \sqrt{(z_1)^2 + (z_2)^2}.$$

For any vector $z \in \mathbb{R}^m$, we let

$$(z)_+ := \max(z, 0),$$

where the maximum is taken componentwise. For any scalar function $\psi : \mathbb{R}^n \to \mathbb{R}$ and $z \in \mathbb{R}^n$, we define the function $\psi_+ : \mathbb{R}^n \to \mathbb{R}$ by

$$\psi_+(z) := (\psi(z))_+ .$$

Also, for any vectors $z^i \in \mathbb{R}^{n_i}$ $(i = 1, 2, \ldots, p)$, we often write

$$(z^1, z^2, \ldots, z^p) := ((z^1)^\top, (z^2)^\top, \ldots, (z^p)^\top)^\top \in \mathbb{R}^{n_1+n_2+\cdots+n_p}.$$

Moreover, for any vector $z \in \mathbb{R}^n$ and any vector function $F : \mathbb{R}^n \to \mathbb{R}^m$, we let

$$z := (z_1, \tilde{z}) \in \mathbb{R} \times \mathbb{R}^{n-1}, \quad F(z) := (F_1(z), \tilde{F}(z)) \in \mathbb{R} \times \mathbb{R}^{m-1}.$$

For a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, we define the gradient of f at x by

$$\nabla_x f(x) := \begin{pmatrix} \dfrac{\partial f(x)}{\partial x_1} \\ \vdots \\ \dfrac{\partial f(x)}{\partial x_n} \end{pmatrix}.$$

We often write $\nabla f(x)$ to denote $\nabla_x f(x)$. In addition, if f is twice differentiable, the Hessian of f at x is defined by

$$\nabla^2_{xx} f(x) := \begin{pmatrix} \dfrac{\partial^2 f(x)}{\partial x_1^2} & \cdots & \dfrac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f(x)}{\partial x_n \partial x_1} & \cdots & \dfrac{\partial^2 f(x)}{\partial x_n^2} \end{pmatrix}.$$

We often write $\nabla^2 f(x)$ to denote $\nabla^2_{xx} f(x)$. Moreover, for a differentiable vector function $F : \mathbb{R}^n \to \mathbb{R}^m$, we define the Jacobian of F at x by

$$\nabla_x F(x) := (\nabla F_1(x) \cdots \nabla F_m(x)) = \begin{pmatrix} \dfrac{\partial F_1(x)}{\partial x_1} & \cdots & \dfrac{\partial F_m(x)}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial F_1(x)}{\partial x_n} & \cdots & \dfrac{\partial F_m(x)}{\partial x_n} \end{pmatrix},$$

where $F(x) = (F_1(x), F_2(x), \ldots, F_m(x))^\top$. We often write $\nabla F(x)$ to denote $\nabla_x F(x)$. If F is twice differentiable, for any vector $y := (y_1, y_2, \ldots, y_m)^\top \in \mathbb{R}^m$, we define

$$\nabla^2 F(x) y := \sum_{i=1}^{m} y_i \nabla^2 F_i(x).$$

Furthermore, for a twice differentiable function $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, we define

$$\nabla^2_{xy} g(x,y) := \nabla_y(\nabla_x g(x,y)),$$

where $\nabla_x g(x,y)$ is the gradient of $g(\cdot, y)$ at x and $\nabla_y(\nabla_x g(x,y))$ means the Jacobian of $\nabla_x g(x,\cdot)$ at y.

Finally, in what follows, we list other notations that appear in the thesis.

- $\perp$: the perpendicularity
- I_n: the n-dimensional identity matrix
- $\mathrm{argmin}_{z \in D} h(z)$: the set of minimizers of h over D for a nonempty set $D \subseteq \mathbb{R}^n$ and a function $h : \mathbb{R}^n \to \mathbb{R}$
- $\mathrm{int}\, C$: the interior of a set $C \subseteq \mathbb{R}^n$
- $\mathrm{bd}\, C$: the boundary of a set $C \subseteq \mathbb{R}^n$
- $P_C(z)$: the Euclidean projection of z onto a closed convex set $C \subseteq \mathbb{R}^n$, i.e., $P_C(z) := \mathrm{argmin}_{w \in C} \|z - w\|$
- $\mathcal{N}_C(z)$: the normal cone [56] of a set $C \subseteq \mathbb{R}^n$ at $z \in C$
- $M \succeq (\succ) N$: $M - N \in S^n_+$ (S^n_{++}) for matrices $M, N \in \mathbb{R}^{n \times n}$
- $\mathrm{diag}(z)$: the $n \times n$ diagonal matrix with diagonal elements z_i $(i = 1, 2, \ldots, n)$ for $z \in \mathbb{R}^n$
- $B(z, \delta)$: the closed ball with center z and radius δ, i.e., $\{w \in \mathbb{R}^n \mid \|w - z\| \leq \delta\}$

2.2 Basic properties of convex functions

To begin with, we define the closedness, properness and convexity of functions.

Definition 2.2.1. *For a given function $f : \mathbb{R}^n \to (-\infty, \infty]$, we denote the effective domain of f by*

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

Then, we say that

1. *the function f is proper if* $\mathrm{dom}\, f \neq \emptyset$,
2. *the function f is closed if f is lower-semi-continuous, and*
3. *the function f is convex if*

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y), \quad \forall \theta \in [0,1], \ \forall x, y \in \mathbb{R}^n.$$

Secondly, the *strict/strong* convexity of proper functions is defined as follows:

Definition 2.2.2. *For a given proper function $f : \mathbb{R}^n \to (-\infty, \infty]$, we say that*

1. *the function f is strictly convex if*

$$f(\theta x + (1-\theta)y) < \theta f(x) + (1-\theta)f(y), \quad \forall \theta \in (0,1),\ \forall x, y \in \operatorname{dom} f \text{ such that } x \neq y,$$

2. *the function f is strongly convex if there exists a constant $\sigma > 0$ such that*

$$f(\theta x + (1-\theta)y) + \frac{1}{2}\sigma\theta(1-\theta)\|x-y\|^2 \leq \theta f(x) + (1-\theta)f(y), \quad \forall \theta \in [0,1],\ \forall x, y \in \operatorname{dom} f.$$

Obviously, any strongly or strictly convex functions are convex. However, the converse is not true in general. For example, the convex function $f(x) = x$ is neither strictly convex nor strongly convex. The next proposition gives the necessary and sufficient condition for a differentiable function to be convex.

Proposition 2.2.3. *For a given function $f : \mathbb{R}^n \to (-\infty, \infty]$, suppose that $\operatorname{dom} f$ is an open convex set and f is differentiable on $\operatorname{dom} f$. Then, the function f is (strictly) convex if and only if*

$$f(y) - f(x) - \nabla f(x)^\top (y - x) \geq (>) 0$$

holds for any $x, y \in \operatorname{dom} f$ such that $x \neq y$.

For a given function $f : \mathbb{R}^n \to (-\infty, \infty]$ and $\alpha \in \mathbb{R}$, we denote a level set of f by $L_f(\alpha) := \{x \mid f(x) \leq \alpha\}$. We then give a proposition about level sets, which will play a key role in the subsequent chapters.

Proposition 2.2.4. *For a given convex function $f : \mathbb{R}^n \to (-\infty, \infty]$ and $\alpha \in \mathbb{R}$, the following properties hold.*

1. *$L_f(\alpha)$ is convex,*

2. *if f is closed, then $L_f(\alpha)$ is closed,*

3. *if f is strongly convex, then $L_f(\alpha)$ is bounded.*

Furthermore, provided that the function f is closed, proper, and has at least one nonempty compact level set, then any nonempty level set of f is compact.

We next consider the convex optimization problem

$$\min_{x \in D} \ f(x), \tag{2.2.1}$$

where $f : \mathbb{R}^n \to (-\infty, \infty]$ is a convex function and $D \subseteq \mathbb{R}^n$ is a nonempty closed convex set. Then, we have the following important properties concerning the existence and uniqueness of an optimal solution of (2.2.1):

Proposition 2.2.5.

1. *Suppose that the function f is strictly convex on D and problem* (2.2.1) *has a nonempty optimal solution set. Then, the optimum of* (2.2.1) *is unique.*

2. *Suppose that the function f is strongly convex on D. Then, the optimum of* (2.2.1) *uniquely exists.*

2.3 Basic properties of cones

In this section, we give some definitions and fundamental properties of cones.

Definition 2.3.1. *A set $C \subseteq \mathbb{R}^n$ is called a cone if*

$$\alpha \geq 0, \;\; x \in C \;\; \Rightarrow \;\; \alpha x \in C.$$

Definition 2.3.2. *For a given cone $C \subseteq \mathbb{R}^n$, we define the dual cone C^d by*

$$C^d := \{x \in \mathbb{R}^n \mid x^\top y \geq 0, \;\; \forall y \in C\}.$$

In particular, C is called a self-dual cone if $C = C^d$ holds.

Obviously, a Cartesian product of self-dual cones is also a self-dual cone. It is well-known that $\mathbb{R}^m_+$, $\mathcal{K}^m$ and S^m_+ are typical examples of self-dual cones [15].

Convexity of a cone can be featured by the following equivalency:

Proposition 2.3.3. *Let $C \subseteq \mathbb{R}^n$ be a given cone. Then, C is convex if and only if*

$$x, y \in C \;\; \Rightarrow \;\; x + y \in C.$$

Furthermore, any closed convex cone can also be characterized as the intersection of finitely or infinitely halfspaces.

Proposition 2.3.4. [57, Corollary 11.7.1] *Let $C \subseteq \mathbb{R}^n$ be a given closed convex cone. For any $s \in \mathbb{R}^n$ with $s \neq 0$, define $H(s) := \{y \in \mathbb{R}^n \mid s^\top y \geq 0\}$ and $S := \{s \in \mathbb{R}^n \mid \|s\| = 1, H(s) \supseteq C\}$. Then, we have*

$$C = \bigcap_{s \in S} H(s).$$

The above proposition will be important when deriving the KKT conditions for SISOCP (1.1.3). We next consider the following conic optimization problem:

$$\begin{aligned} &\underset{x \in \mathbb{R}^n}{\text{Minimize}} && f(x) \\ &\text{subject to} && g^j(x) \in C_j \;\; (j = 1, 2, \ldots, J), \end{aligned} \tag{2.3.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g^j : \mathbb{R}^n \to \mathbb{R}^{m_j}$ $(j = 1, 2, \ldots, J)$ are continuously differentiable and $C_j \subseteq \mathbb{R}^{m_j}$ $(j = 1, 2, \ldots, J)$ are closed convex cones with nonempty interior. Let $x^* \in \mathbb{R}^n$ be a local optimum of (2.3.1). Then, the KKT conditions hold at x^* under the following Robinson's constraint qualification.

Definition 2.3.5 (Robinson's constraint qualification)**.** *Let $\bar{x} \in \mathbb{R}^n$ be feasible for* (2.3.1)*. Then, we say that Robinson's constraint qualification holds at $\bar{x}$ if there exists a vector $d \in \mathbb{R}^n$ such that*

$$g^j(\bar{x}) + \nabla g^j(\bar{x})^\top d \in \operatorname{int} C_j \;\; (j = 1, 2, \ldots, J).$$

For details about Robinson's constraint qualification, see [6].

Theorem 2.3.6. *Let $x^* \in \mathbb{R}^n$ be a local optimum of* (2.3.1) *such that Robinson's constraint qualification holds. Then, the KKT conditions hold at x^*, i.e., there exist Lagrange multiplier vectors $\lambda^j \in \mathbb{R}^{m_j}$ $(j = 1, 2, \dots, J)$ such that*

$$\nabla f(x^*) - \sum_{j=1}^{J} \nabla g^j(x^*)\lambda^j = 0, \;\; C_j^d \ni \lambda^j \perp g^j(x^*) \in C_j \;\; (j = 1, 2, \dots, J), \tag{2.3.2}$$

where C_j^d denotes the dual cone of C_j for $j = 1, 2, \dots, J$.

Especially, if each C_j is the m_j-dimensional SOC, i.e., $C_j := \mathcal{K}^{m_j}$ in (2.3.1), the KKT conditions (2.3.2) can be rewritten as

$$\nabla f(x^*) - \sum_{j=1}^{J} \nabla g^j(x^*)\lambda^j = 0, \;\; \mathcal{K}^{m_j} \ni \lambda^j \perp g^j(x^*) \in \mathcal{K}^{m_j} \;\; (j = 1, 2, \dots, J),$$

since $(\mathcal{K}^{m_j})^d = \mathcal{K}^{m_j}$ for $j = 1, 2, \dots, J$ from the self-duality of second-order cones.

2.4 Natural residual associated with SOC complementarity condition

Let $\mathcal{K}$ be a Cartesian product of second-order cones, i.e, $\mathcal{K} := \mathcal{K}^{m_1} \times \mathcal{K}^{m_2} \times \cdots \times \mathcal{K}^{m_\ell}$. In designing algorithms for solving problems involving SOC constraints, we often reformulate the SOC complementarity condition $\mathcal{K} \ni y \perp z \in \mathcal{K}$ as a system of equations by means of the natural residual defined precisely below. To be specific, we first introduce the spectral factorization of a vector with respect to the SOC, $\mathcal{K}^m$.

Definition 2.4.1. *For any vector $z := (z_1, z_2) \in \mathbb{R} \times \mathbb{R}^{m-1}$, we define the spectral factorization with respect to $\mathcal{K}^m$ as*

$$z = \lambda_1 c^1 + \lambda_2 c^2,$$

where λ_1 and λ_2 are the spectral values given by

$$\lambda_j = z_1 + (-1)^j \|z_2\|, \quad j = 1, \; 2,$$

and c^1 and c^2 are the spectral vectors given by

$$c^j = \begin{cases} \dfrac{1}{2}\left(1, \, (-1)^j \dfrac{z_2}{\|z_2\|}\right) & \text{if } z_2 \neq 0 \\ \dfrac{1}{2}(1, \, (-1)^j v) & \text{if } z_2 = 0 \end{cases} \qquad j = 1, \; 2,$$

where $v \in \mathbb{R}^{m-1}$ is an arbitrary vector such that $\|v\| = 1$.

By using the spectral factorization, we can write the Euclidean projection onto $\mathcal{K}^m$ explicitly as follows [18]:

$$P_{\mathcal{K}^m}(z) = (\lambda_1)_+ c^1 + (\lambda_2)_+ c^2,$$

where λ_j and c^j $(j = 1, 2)$ are the spectral values and the spectral vectors of z, respectively. Now, let us define the natural residual for the SOC complementarity condition by using the Euclidean projection.

Definition 2.4.2. *Let* $y := (y^1, y^2, \cdots, y^\ell)$ and $z := (z^1, z^2, \cdots, z^\ell) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \cdots \times \mathbb{R}^{m_\ell} = \mathbb{R}^m$ *be arbitrary vectors. Then, the natural residual function* $\Phi : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m$ *with respect to* $\mathcal{K} = \mathcal{K}^{m_1} \times \mathcal{K}^{m_2} \times \cdots \times \mathcal{K}^{m_\ell}$ *is defined as*

$$\begin{aligned}\Phi(y, z) &:= y - P_{\mathcal{K}}(y - z) \\ &= \begin{pmatrix} \varphi^1(y^1, z^1) \\ \vdots \\ \varphi^\ell(y^\ell, z^\ell) \end{pmatrix} \in \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_\ell},\end{aligned} \tag{2.4.1}$$

where

$$\varphi^i(y^i, z^i) = y^i - P_{\mathcal{K}^{m_i}}(y^i - z^i), \ \ i = 1, 2, \ldots, \ell.$$

Using the natural residual Φ, we can reformulate the SOC complementarity condition as an equivalent system of equations.

Proposition 2.4.3. [18] *We have the following equivalency:*

$$\begin{aligned}\varphi^i(y^i, z^i) = 0 &\iff \mathcal{K}^{m_i} \ni y^i \perp z^i \in \mathcal{K}^{m_i} \quad (i = 1, 2, \ldots, \ell) \\ \Phi(y, z) = 0 &\iff \mathcal{K} \ni y \perp z \in \mathcal{K}.\end{aligned}$$

From the above proposition, we have only to solve the equation $\Phi(y, z) = 0$ for finding $(y, z) \in \mathbb{R}^m \times \mathbb{R}^m$ that satisfies the SOC complementarity condition $\mathcal{K} \ni y \perp z \in \mathcal{K}$. In [26], the authors proved that the natural residual Φ has the Jacobian consistency and semismoothness. By exploiting these properties, there have been proposed several efficient methods such as semismooth and smoothing Newton methods [49, 26].

Chapter 3

A regularized explicit exchange method for convex semi-infinite second-order cone programming problems

3.1 Introduction

In this chapter, we consider SISOCP (1.1.3) with a convex objective function and infinitely many affine SOC constraints, i.e.,

$$\begin{aligned} &\underset{x\in\mathbb{R}^n}{\text{Minimize}} && f(x) \\ &\text{subject to} && A_j(t)^\top x - b_j(t) \in \mathcal{K}^{m_j} \quad \text{for all } t \in T_j \ (j = 1, 2, \ldots, J), \end{aligned} \tag{3.1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable *convex* function, $A : T_j \to \mathbb{R}^{n\times m}$ and $b : T_j \to \mathbb{R}^m$ are continuous functions for $j = 1, 2, \ldots, J$, and $T_j \subseteq\ R^{\ell_j}$ $(j = 1, 2, \ldots, J)$ are nonempty compact index sets. Note that the feasible set of SISOCP (3.1.1) can be expressed as

$$\bigcap_{1\le j\le J} \bigcap_{t\in T_j} \{x \in \mathbb{R}^n \mid A_j(t)^\top x - b_j(t) \in \mathcal{K}^{m_j}\}. \tag{3.1.2}$$

Since $\{x \in \mathbb{R}^n \mid A_j(t)^\top x - b_j(t) \in \mathcal{K}^{m_j}\}$ $(t \in T_j,\ j = 1, 2, \ldots, J)$ are closed convex sets from the closed convexity of $\mathcal{K}^{m_j}$, (3.1.2) is also closed convex. Therefore, SISOCP (3.1.1) is a convex program.

A lot of practical problems can be formulated as (3.1.1). For example, the Chebyshev approximation problem (1.1.4) is transformed to an SISOCP of the form (3.1.1) when the function $F(\cdot, y)$ is affine, and moreover, SISOCP (1.1.7) obtained from the FIR filter problem (1.1.6) obviously takes the form (3.1.1). Hence, it is important to develop efficient algorithms for solving SISOCP (3.1.1).

In this chapter, for solving SISOCP (3.1.1), we propose an algorithm based on the exchange method. Many researchers [21, 27, 34, 73, 47, 79] have studied exchange-type algorithms for solving convex semi-infinite programs. The exchange method solves a relaxed semi-infinite program (SIP) with T_j $(j = 1, 2, \ldots, J)$ replaced by a finite subset $T_j^k \subseteq T_j$ for each j, where T_j^k is updated so that $T_j^{k+1} \subseteq T_j^k \cup \{t_j^1, t_j^2, \cdots, t_j^r\}$ with $\{t_j^1, t_j^2, \cdots, t_j^r\} \subseteq T \setminus T_j^k$. As another scheme analogous to the exchange method, the discretization method [21, 28, 54, 64] should be referred to here. It also solves a sequence of relaxed SIPs with T_j $(j = 1, 2, \ldots, J)$ replaced by finite subset $T_j^k \subseteq T_j$ for each j, but T_j^k is updated so that $T_j^{k+1} = T_j^k \cup \{t_j^1, t_j^2, \cdots, t_j^r\}$ with $\{t_j^1, t_j^2, \cdots, t_j^r\} \subseteq T_j \setminus T_j^k$ and the distance[1] from T_j^k to T_j converges to 0 as k goes to infinity. Although this method is comprehensible and easy to implement, the computational cost tends to be high since the cardinality of T_j^k grows exponentially in the dimension of T_j. However, the exchange method can avoid such a serious drawback by removing unnecessary indices that correspond to inactive constrains. We thus do not adopt the discretization method, but the exchange method for solving SISOCP (3.1.1).

We note that the exchange-type algorithm proposed in this chapter needs to solve a sequence of SOCPs. To such a subproblem, we can apply an existing algorithm such as the interior-point method and the smoothing Newton method [1, 18, 26, 37].

In the subsequent sections, we will propose exchange-type methods for solving SISOCP (3.1.1) and analyze convergence properties of generated iteration points. However, to make a convergence analysis in the more general framework, we consider the following problem:

$$
\begin{aligned}
&\underset{x \in \mathbb{R}^n}{\text{Minimize}} && f(x) \\
&\text{subject to} && g(x,t) \in C \quad \text{for all } t \in T,
\end{aligned}
\tag{3.1.3}
$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function and $g : \mathbb{R}^n \times T \to \mathbb{R}^m$ is a continuous function such that $g(\cdot, t)$ is differentiable for each fixed t, $T \subseteq \mathbb{R}^\ell$ is a compact set and $C \subseteq \mathbb{R}^m$ is a closed convex cone with nonempty interior. We call this problem the semi-infinite conic program, SICP for short. SICP (3.1.3) contains SISOCP (3.1.1) as a subclass. Indeed, by letting

$$
\begin{aligned}
C &:= \mathcal{K}^{m_1} \times \mathcal{K}^{m_2} \times \cdots \times \mathcal{K}^{m_J}, \\
g(x,t) &:= \Big(A_1(t^1)^\top x - b_1(t^1), A_2(t^2)^\top x - b_2(t^2), \ldots, A_J(t^2)^\top x - b_J(t^2) \Big)^\top, \\
t &:= (t^1, t^2, \ldots, t^J),\ t^j \in T_j\ (j = 1, 2, \ldots, J),\ \text{and } T := T_1 \times T_2 \times \cdots \times T_J,
\end{aligned}
$$

SICP (3.1.3) reduces to SISOCP (3.1.1). Notice that another possible choice of C is a symmetric positive semi-definite cone S_+^m. Hence, SICP (3.1.4) also contains semi-infinite programs with infinitely many symmetric positive semi-definite cone constraints.

The main purpose of this chapter is two-fold. First, we study the Karush-Kuhn-Tucker (KKT) conditions for SICP (3.1.3). Although the original KKT conditions for SICP could be described by means of integration and Borel measure, we show that they can be represented by a *finite* number of elements in T under the generalized Robinson constraint qualification. Second, we propose two algorithms for solving the convex SICP (3.1.3) of the following form:

$$
\begin{aligned}
&\underset{x \in \mathbb{R}^n}{\text{Minimize}} && f(x) \\
&\text{subject to} && A(t)^\top x - b(t) \in C \quad \text{for all } t \in T,
\end{aligned}
\tag{3.1.4}
$$

[1] For two sets $X \subseteq Y$, the distance from X to Y is defined as $\operatorname{dist}(X, Y) := \sup_{y \in Y} \inf_{x \in X} \|x - y\|$.

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable *convex* function, $A : T \to \mathbb{R}^{n\times m}$ and $b : T \to \mathbb{R}^m$ are continuous functions, and T and C are as in SICP (3.1.3). The proposed algorithms are based on the exchange method, which solves a sequence of subproblems with *finitely* many conic constraints. The first algorithm is an explicit exchange method, for which we show global convergence under the strict convexity of the objective function. The second algorithm is a regularized explicit exchange method. With the help of regularization, global convergence of the algorithm is established without the strict convexity assumption.

This chapter is organized as follows. In Section 3.2, we discuss the KKT conditions for SICP (3.1.3). In Section 3.3, we propose the explicit exchange method for solving SICP (3.1.4). In Section 3.4, we combine the explicit exchange method with the regularization method and show that the hybrid algorithm is globally convergent for SICP (3.1.4). In Section 3.5, we give some numerical results to examine the efficiency of the proposed algorithm. In Section 3.6, we conclude the chapter with some remarks.

3.2 KKT conditions for SICP

In this section, we derive the Karush-Kuhn-Tucker (KKT) conditions for SICP (3.1.4). The result is not only interesting in itself, but also provides us an important key to analyze global convergence of the algorithm proposed in Section 3.3.

When $m = 1$ and $C = \mathbb{R}_+$, SICP (3.1.3) reduces to the classical semi-infinite program and the KKT conditions are given as follows.

Lemma 3.2.1. [29, Theorem 3.3] *Let $x^* \in \mathbb{R}^n$ be a local optimum of SICP (3.1.3) with $C := \mathbb{R}_+$. Let $T_{\rm act}(x)$ be the set of active indices at $x \in \mathbb{R}^n$, i.e., $T_{\rm act}(x) := \{t \in T \mid g(x,t) = 0\}$. Suppose that the Mangasarian-Fromovitz constraint qualification (MFCQ) holds at x^*, i.e., there exists a vector $d \in \mathbb{R}^n$ such that $\nabla_x g(x^*,t)^\top d > 0$ for any $t \in T_{\rm act}(x^*)$. Then, there exist p indices $t_1, t_2, \dots, t_p \in T_{\rm act}(x^*)$ and Lagrange multipliers $\mu_1, \mu_2, \dots, \mu_p \geq 0$ such that $p \leq n$ and*

$$\nabla f(x^*) - \sum_{i=1}^{p} \mu_i \nabla_x g(x^*, t_i) = 0,$$
$$\mathbb{R}_+ \ni \mu_i \perp g(x^*, t_i) \in \mathbb{R}_+ \quad (i = 1, 2, \dots, p).$$

In the above lemma, the MFCQ plays a key role. However, for SICP (3.1.3), it is difficult to apply the MFCQ in a straightforward manner. We therefore introduce the generalized Robinson constraint qualification (GRCQ), which is defined as follows.

Definition 3.2.2 (Generalized Robinson Constraint Qualification (GRCQ))**.** *Let $x \in \mathbb{R}^n$ be a feasible point of SICP (3.1.3). Then, we say that the Robinson constraint qualification (GRCQ) holds at x if there exists a vector $d \in \mathbb{R}^n$ such that*

$$g(x,t) + \nabla_x g(x,t)^\top d \in \operatorname{int} C \ \text{ for all } t \in T. \tag{3.2.1}$$

When $m = 1$ and $C = \mathbb{R}_+$, the GRCQ reduces to the MFCQ. When g is affine, i.e., $g(x,t) := A(t)^\top x - b(t)$, the GRCQ holds at any feasible point if and only if the Slater constraint qualification holds, i.e., there exists $x_0 \in \mathbb{R}^n$ such that $A(t)^\top x_0 - b(t) \in \operatorname{int} C$ for all $t \in T$. Furthermore, if T consists of finitely elements, the GRCQ is reduced to the Robinson constraint qualification defined by Definition 2.3.5. The next proposition states that any closed convex cone is represented as the intersection of finitely or infinitely many halfspaces generated by a certain compact set.

Proposition 3.2.3. *Let $C \subsetneq \mathbb{R}^m$ be a nonempty closed convex cone. Then,* (i) *there exists a nonempty compact set $S \subseteq \{s \in \mathbb{R}^m \mid \|s\| = 1\}$ such that*

$$C = \{y \in \mathbb{R}^m \mid s^\top y \geq 0, \ \ \forall s \in S\}. \tag{3.2.2}$$

Moreover, we have (ii) $S \subseteq C^d$, *and* (iii) $\operatorname{int} C \subseteq \{y \in \mathbb{R}^m \mid s^\top y > 0, \ \forall s \in S\}$.

Proof. We first show (i). For any $s \in \mathbb{R}^m$ with $s \neq 0$, we define the halfspace $H(s) := \{y \in \mathbb{R}^m \mid s^\top y \geq 0\}$. In addition, let $S := \{s \in \mathbb{R}^m \mid \|s\| = 1, \ H(s) \supseteq C\}$. By Proposition 2.3.4, we have $C = \bigcap_{s \in S} H(s)$. Therefore, it suffices to show the compactness of S. Since the boundedness is evident, we only show the closedness. Choose an arbitrary convergent sequence $\{s^k\} \subseteq S$ such that $\lim_{k\to\infty} s^k = s^*$ and let $z \in C$ be an arbitrary vector. Obviously, we have $\|s^k\| = 1$. Moreover, from $C = \bigcap_{s \in S} H(s) \subseteq H(s^k)$, we have $(s^k)^\top z \geq 0$ for all k. Therefore, by letting $k \to \infty$, we obtain $\|s^*\| = 1$ and $(s^*)^\top z \geq 0$, which implies $z \in H(s^*)$. Since $z \in C$ was arbitrary, we have $C \subseteq H(s^*)$, and hence $s^* \in S$.

Second, we show (ii). Choose $s \in S$ arbitrarily. From (3.2.2), we have $s^\top y \geq 0$ for all $y \in C$, which implies $s \in C^d$.

We finally show (iii). Choose $z \in \operatorname{int} C$ arbitrarily. From the compactness of S, there exists $\bar{s} \in S$ such that $\bar{s} \in \operatorname{argmin}_{s \in S} z^\top s$. To show (iii), we only have to prove $z^\top \bar{s} > 0$. For contradiction, suppose that $z^\top \bar{s} \leq 0$, which together with $\bar{s} \in S \subseteq C^d$ implies $z^\top \bar{s} = 0$. Since $z \in \operatorname{int} C$, we have $z - \delta \bar{s} \in C$ for sufficiently small $\delta > 0$. Then, by using $z^\top \bar{s} = 0$, $z - \delta \bar{s} \in C$ and $\bar{s} \in C^d$, we have $0 \leq (z - \delta \bar{s})^\top \bar{s} = -\delta \|\bar{s}\|^2$, which yields $\bar{s} = 0$. This contradicts the fact $\bar{s} \in S$. □

By using this proposition, we reformulate SICP (3.1.3) as a standard semi-infinite program, whereby we can derive the KKT conditions.

Theorem 3.2.4. *Let $x^* \in \mathbb{R}^n$ be a local optimum of SICP* (3.1.3). *Suppose that the GRCQ holds at x^*. Then, there exist p indices $t_1, t_2, \ldots, t_p \in T$ and Lagrange multipliers $y^1, y^2, \ldots, y^p \in \mathbb{R}^m$ such that $p \leq n$ and*

$$\nabla f(x^*) - \sum_{i=1}^{p} \nabla_x g(x^*, t_i) y^i = 0, \tag{3.2.3}$$

$$C^d \ni y^i \perp g(x^*, t_i) \in C \quad (i = 1, 2, \ldots, p). \tag{3.2.4}$$

Proof. By Proposition 3.2.3, there exists a nonempty compact set $S \subseteq \{s \in \mathbb{R}^m \mid \|s\| = 1\}$ such that SICP (3.1.3) is equivalent to the following semi-infinite program:

$$\begin{array}{ll} \text{Minimize} & f(x) \\ \text{subject to} & s^\top g(x,t) \geq 0 \ \text{ for all } (s,t) \in S \times T. \end{array} \tag{3.2.5}$$

Let $(S \times T)_{\rm act}(x^*) := \{(s,t) \in S \times T \mid s^\top g(x^*,t) = 0\}$. If $(S \times T)_{\rm act}(x^*) = \emptyset$, then we have (3.2.3) and (3.2.4) with $y^i = 0$ for all i. Next, we suppose $(S \times T)_{\rm act}(x^*) \neq \emptyset$. We first show that the MFCQ holds for problem (3.2.5), i.e., there exists a vector $d \in \mathbb{R}^n$ such that

$$(\nabla_x g(x^*,t)s)^\top d > 0 \ \text{ for all } (s,t) \in (S \times T)_{\rm act}(x^*). \tag{3.2.6}$$

By assumption, there exists a vector $d \in \mathbb{R}^n$ satisfying GRCQ (3.2.1), i.e., $g(x^*,t)+\nabla_x g(x^*,t)^\top d \in \operatorname{int} C$ for all $t \in T$. By Proposition 3.2.3, we also have $0 \notin S \subseteq C^d$. Hence, from Proposition 3.2.3 (iii), we have $s^\top(g(x^*,t) + \nabla_x g(x^*,t)^\top d) > 0$ for all $(s,t) \in S \times T$, which implies (3.2.6). Therefore, d satisfies (3.2.6). Now, applying Lemma 3.2.1 to problem (3.2.5), we have p indices $(s^1,t_1),(s^2,t_2),\dots,(s^p,t_p) \in (S \times T)_{\rm act}(x^*)$ and the Lagrange multipliers $\mu_1,\mu_2,\dots,\mu_p \geq 0$ such that $p \leq n$ and

$$\nabla f(x^*) - \sum_{i=1}^{p} \mu_i \nabla_x g(x^*,t_i)s_i = 0, \tag{3.2.7}$$

$$\mathbb{R}_+ \ni \mu_i \perp (s^i)^\top g(x^*,t_i) \in \mathbb{R}_+ \quad (i=1,2,\dots,p). \tag{3.2.8}$$

By letting $y^i := \mu_i s^i$ for each i, we have from (3.2.8) that $0 = \mu_i s_i^\top g(x^*,t_i) = (y^i)^\top g(x^*,t_i)$. We also have $y^i \in C^d$ since $s^i \in S \subseteq C^d$ from Proposition 3.2.3 and $\mu_i \geq 0$. In addition, we have $g(x^*,t_i) \in C$ since x^* is feasible to SICP (3.1.3). Thus, (3.2.7) and (3.2.8) yield (3.2.3) and (3.2.4), respectively. This completes the proof. □

Before closing this section, we give two theorems. The first one states that the KKT conditions are also sufficient for global optimality when the problem is the convex SICP (3.1.4).

Theorem 3.2.5. *Let $x^* \in \mathbb{R}^n$ be feasible to the convex SICP (3.1.4). If there exist p indices $t_1,t_2,\dots,t_p \in T$ and p Lagrange multiplier vectors $y^1,y^2,\dots,y^p \in \mathbb{R}^m$ such that*

$$\nabla f(x^*) - \sum_{i=1}^{p} A(t_i)y^i = 0, \tag{3.2.9}$$

$$C^d \ni y^i \perp A(t_i)^\top x^* - b(t_i) \in C \quad (i=1,2,\dots,p), \tag{3.2.10}$$

then x^* is a global optimum of SICP (3.1.4).

Proof. Let $\ell : \mathbb{R}^n \to \mathbb{R}$ be defined by $\ell(x) := f(x) - \sum_{i=1}^p (y^i)^\top (A(t_i)^\top x - b(t_i))$, and let $\bar{x} \in \mathbb{R}^n$ be an arbitrary feasible point of SICP (3.1.4). Since ℓ is convex and $\nabla \ell(x^*) = \nabla f(x^*) - \sum_{i=1}^p A(t_i)y^i = 0$ by (3.2.9), x^* is a global minimum of ℓ, i.e., $\ell(\bar{x}) - \ell(x^*) \geq 0$. Hence, we have $f(\bar{x}) - f(x^*) = \ell(\bar{x}) - \ell(x^*) + \sum_{i=1}^p (y^i)^\top (A(t_i)^\top \bar{x} - b(t_i)) \geq 0$, where the first equality follows from the definition of ℓ and (3.2.10), and the last inequality follows from $\ell(\bar{x}) - \ell(x^*) \geq 0$, $y^i \in C^d$, and $A(t_i)^\top \bar{x} - b(t_i) \in C$ $(i=1,2,\dots,p)$. We thus conclude that x^* is a global optimum of SICP (3.1.4). □

Next, we enhance Theorem 3.2.4 so that it can elaborate upon the case where C has a Cartesian structure, i.e., $C = C^1 \times \cdots \times C^h \subseteq \mathbb{R}^m = \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_h}$. Consider the following problem:

$$\begin{array}{ll} \text{Minimize} & f(x) \\ \text{subject to} & g^j(x,t^j) \in C^j \ \text{ for all } t^j \in T_j, \ \ j = 1,2,\cdots,h, \end{array} \tag{3.2.11}$$

where $g^j : \mathbb{R}^n \times T_j \to \mathbb{R}^{m_j}$ is continuous, $g^j(\cdot, t^j)$ is differentiable for each fixed t^j, $T_j \subseteq \mathbb{R}^{\ell_j}$ is a nonempty compact set and $C^j \subseteq \mathbb{R}^{m_j}$ is a closed convex cone with nonempty interior for each j. Then, the following theorem holds.

Theorem 3.2.6. *Let $x^* \in \mathbb{R}^n$ be a local optimum of SICP* (3.2.11). *Assume that the GRCQ holds at x^*, i.e., there exists a vector $d \in \mathbb{R}^n$ such that*

$$g_j(x^*, t^j) + \nabla_x g_j(x^*, t^j)^\top d \in \operatorname{int} C^j \ \textit{ for all } t^j \in T_j,\ j = 1, 2, \dots, h. \tag{3.2.12}$$

Then, there exist p indices[2] *$j_1, j_2, \dots, j_p \in \{1, 2, \dots, h\}$ and $(t_i^{j_i}, y_i^{j_i}) \in T_{j_i} \times \mathbb{R}^{m_{j_i}}$ for $i = 1, 2, \dots, p$ such that $p \le n$ and*

$$\nabla f(x^*) - \sum_{i=1}^{p} \nabla_x g_{j_i}(x^*, t_i^{j_i}) y_i^{j_i} = 0 \tag{3.2.13}$$

$$(C^{j_i})^d \ni y_i^{j_i} \perp g_{j_i}(x^*, t_i^{j_i}) \in C^{j_i} \quad (i = 1, 2, \dots, p). \tag{3.2.14}$$

Proof. For each $j = 1, 2, \dots, h$, let $\tilde{t}^j \in \mathbb{R}^{\ell_j} \setminus T_j$ be an arbitrary point and $\tilde{T}_j$ be defined as $\tilde{T}_j := \{\tilde{t}^1\} \times \cdots \times \{\tilde{t}^{j-1}\} \times T_j \times \{\tilde{t}^{j+1}\} \times \cdots \times \{\tilde{t}^h\} \subseteq \mathbb{R}^{\ell_1 + \ell_2 + \cdots + \ell_h}$. Then we can easily see that $\tilde{T}_j \cap \tilde{T}_{j'} = \emptyset$ for any $j \ne j'$. Let

$$t := (t^1, t^2, \dots, t^h) \in \mathbb{R}^{\ell_1 + \ell_2 + \cdots + \ell_h}, \quad T := \bigcup_{j=1}^{h} \tilde{T}_j \subseteq \mathbb{R}^{\ell_1 + \ell_2 + \cdots + \ell_h}, \tag{3.2.15}$$

and $g : \mathbb{R}^n \times T \to \mathbb{R}^{m_1 + m_2 + \cdots + m_h}$ be defined by

$$g(x, t) := (\tilde{g}_1(x, t), \dots, \tilde{g}_h(x, t)), \tag{3.2.16}$$

where

$$\tilde{g}^j(x, t) := \begin{cases} g^j(x, t^j) & (t \in \tilde{T}^j) \\ \zeta^j & (t \notin \tilde{T}^j) \end{cases} \tag{3.2.17}$$

and $\zeta^j \in \operatorname{int} C^j$ is an arbitrary vector. Then, the function g is continuous on $\mathbb{R}^n \times T$ and $g(\cdot, t)$ is differentiable for each $t \in T$. In particular, we have

$$\nabla_x \tilde{g}^j(x, t) := \begin{cases} \nabla_x g^j(x, t^j) & (t \in \tilde{T}_j) \\ 0 & (t \notin \tilde{T}_j). \end{cases} \tag{3.2.18}$$

Then, T is nonempty and compact, and SICP (3.2.11) is equivalent to SICP (3.1.3) with $C = C^1 \times \cdots \times C^h$ and g defined by (3.2.16). By letting $d \in \mathbb{R}^n$ satisfy (3.2.12), we have

$$\tilde{g}^j(x^*, t) + \nabla_x \tilde{g}^j(x^*, t)^\top d = \begin{cases} g^j(x^*, t^j) + \nabla_x g^j(x^*, t^j)^\top d \in \operatorname{int} C^j & (t \in \tilde{T}_j) \\ \zeta^j \in \operatorname{int} C^j & (t \notin \tilde{T}_j) \end{cases}$$

for each $j = 1, 2, \dots, h$, where the first case follows from (3.2.12) and the second one follows from (3.2.17), (3.2.18) and $\zeta^j \in \operatorname{int} C^j$. Therefore, we have $g(x^*, t) + \nabla g(x^*, t)^\top d \in \operatorname{int} C$ for all

[2] Repeated choice of the same index is allowed in the set $\{j_1, j_2, \dots, j_p\}$.

$t \in T$, which implies that the GRCQ holds at x^* for SICP (3.1.3). Hence, by Theorem 3.2.4, there exist $p \le n$, $t_1, t_2, \dots, t_p \in T$ and $y_1, y_2, \dots, y_p \in \mathbb{R}^m$ such that

$$\nabla f(x^*) - \sum_{i=1}^{p} \nabla_x g(x^*, t_i) y_i = 0, \tag{3.2.19}$$

$$C^d \ni y_i \perp g(x^*, t_i) \in C \quad (i = 1, 2, \dots, p). \tag{3.2.20}$$

Let $t_i := (t_i^1, t_i^2, \dots, t_i^h) \in \mathbb{R}^{\ell_1+\ell_2+\cdots+\ell_h}$ and $y_i := (y_i^1, y_i^2, \dots, y_i^h) \in \mathbb{R}^{m_1+m_2+\cdots+m_h}$ for $i = 1, 2, \dots, p$. From (3.2.15), for each i, there exists $j_i \in \{1, 2, \dots, h\}$ such that $t_i \in \tilde{T}_{j_i}$, i.e., $t_i^{j_i} \in T_{j_i}$. Then, we have

$$\begin{aligned}\sum_{i=1}^{p} \nabla_x g(x^*, t_i) y_i &= \sum_{i=1}^{p} \Big(\nabla_x \tilde{g}_1(x^*, t_i), \nabla_x \tilde{g}_2(x^*, t_i), \dots, \nabla_x \tilde{g}_h(x^*, t_i) \Big) \begin{pmatrix} y_i^1 \\ \vdots \\ y_i^h \end{pmatrix} \\ &= \sum_{i=1}^{p} \nabla_x g_{j_i}(x^*, t_i^{j_i}) y_i^{j_i},\end{aligned}$$

where the second equality follows from (3.2.17) and (3.2.18), which together with (3.2.19) implies (3.2.13). In the last, we show (3.2.14). From (3.2.20) and $C^d = (C^1)^d \times (C^2)^d \times \cdots \times (C^h)^d$, we have $(C^j)^d \ni y_i^j \perp \tilde{g}^j(x^*, t_i) \in C^j$ for $j = 1, 2, \dots, h$, which together with $\tilde{g}_{j_i}(x^*, t_i) = g_{j_i}(x^*, t_i^{j_i})$ from (3.2.17) implies (3.2.14) for $i = 1, 2, \dots, p$. The proof is complete. □

3.3 Explicit exchange method for SICP

In this section, we propose an explicit exchange method for solving the convex SICP (3.1.4) and show its global convergence under the assumption that f is strictly convex.

3.3.1 Algorithm

The algorithm proposed in this section requires solving conic programs with *finitely* many constraints as subproblems. Let $\mathrm{CP}(T')$ be the relaxed problem of SICP (3.1.4) with T replaced by a finite subset $T' := \{t_1, t_2, \dots, t_p\} \subseteq T$. Then, $\mathrm{CP}(T')$ can be formulated as follows:

$$\mathrm{CP}(T') \qquad \begin{array}{ll} \text{Minimize} & f(x) \\ \text{subject to} & A(t_i)^\top x - b(t_i) \in C \ \ (i = 1, 2, \dots, p). \end{array}$$

Note that an optimum x^* of $\mathrm{CP}(T')$ satisfies the following KKT conditions:

$$\begin{aligned}&\nabla f(x^*) - \sum_{i=1}^{p} A(t_i) y_{t_i} = 0, \\ &C^d \ni y_{t_i} \perp A(t_i)^\top x^* - b(t_i) \in C \ \ (i = 1, 2, \dots, p),\end{aligned}$$

where y_{t_i} is the Lagrange multiplier vector corresponding to the constraint $A(t_i)^\top x^* - b(t_i) \in C$ for each i.

Now, we propose the following algorithm.

Algorithm 3.1 (Explicit exchange method)

Step 0. Let $\{\gamma_k\} \subseteq \mathbb{R}_{++}$ be a positive sequence such that $\lim_{k\to\infty} \gamma_k = 0$. Choose a finite subset $T^0 := \{t_1^0, \dots, t_\ell^0\} \subseteq T$ for some integer[3] $\ell \geq 0$, and a vector $e \in \operatorname{int} C$. Set $k := 0$.

Step 1. Obtain x^{k+1} and T^{k+1} by the following steps.

Step 1-0 Set $r := 0$, $E^0 := T^k$, and solve CP(E^0) to obtain an optimum v^0.

Step 1-1 Find a $t_{\text{new}}^r \in T$ such that

$$A(t_{\text{new}}^r)^\top v^r - b(t_{\text{new}}^r) \notin -\gamma_k e + C. \tag{3.3.1}$$

If such a t_{new}^r does not exist, i.e.,

$$A(t)^\top v^r - b(t) \in -\gamma_k e + C \tag{3.3.2}$$

for any $t \in T$, then set $x^{k+1} := v^r$, $T^{k+1} := E^r$, and go to Step 2. Otherwise, let

$$\overline{E}^{r+1} := E^r \cup \{t_{\text{new}}^r\},$$

and go to Step 1-2.

Step 1-2 Solve CP$(\overline{E}^{r+1})$ to obtain an optimum v^{r+1} and the Lagrange multipliers y_t^{r+1} for $t \in \overline{E}^{r+1}$.

Step 1-3 Let $E^{r+1} := \{t \in \overline{E}^{r+1} \mid y_t^{r+1} \neq 0\}$. Set $r := r+1$ and return to Step 1-1.

Step 2. If γ_k is sufficiently small, then terminate. Otherwise, set $k := k+1$ and return to Step 1.

Here, $\gamma_k > 0$ plays the role of a relaxation parameter for the feasible set of SICP (3.1.4). Let $X(\gamma) := \{x \in \mathbb{R}^n \mid A(t)^\top x - b(t) \in -\gamma e + C,\ \forall t \in T\}$. Then, $X(0)$ coincides with the feasible set of SICP (3.1.4), and $X(\gamma)$ expands as γ increases. Note that, by the termination criterion (3.3.2) for the inner loop, we have $x^{k+1} \in X(\gamma_k)$ for each k. Hence, we can expect that the distance between x^k and the feasible set of SICP (3.1.4) tends to 0 as k goes to infinity. Moreover, as will be shown in the next subsection, the positivity of γ_k guarantees the inner loop of Step 1 to terminate in a finite number of iterations for each k.

When C is a symmetric cone such as an SOC or a semi-definite cone, a natural choice for the vector $e \in \operatorname{int} C$ is the identity element with respect to Euclidean Jordan algebra [15].[4] Moreover, in Step 1-2, we can employ an existing method such as the primal-dual interior point method, the regularized smoothing method, and so on [1, 18, 26, 33, 37].

Let us denote the optimal values of CP(T') and SICP (3.1.4) by $V(T')$ and $V(T)$, respectively. Since E^{r+1} is obtained by removing the constraints with zero Lagrange multipliers from $\overline{E}^{r+1}$, and the feasible region of CP(E^r) is larger than that of CP$(\overline{E}^{r+1})$, we have

$$V(E^0) \leq V(\overline{E}^1) = V(E^1) \leq \cdots \leq V(E^r) \leq V(\overline{E}^{r+1}) = V(E^{r+1}) \leq \cdots \leq V(T) < \infty. \tag{3.3.3}$$

[3] We allow $\ell = 0$, which means $T^0 = \emptyset$.

[4] For example, if C is $\mathbb{R}_+$, $\mathcal{K}^m$ and $\mathcal{S}_+^m$, then the identity element is 1, $(1, 0, \dots, 0)^\top \in \mathbb{R}^m$, and the $m \times m$ identity matrix, respectively.

In the subsequent convergence analysis, we omit the termination condition in Step 2, so that the algorithm may generate an infinite sequence $\{x^k\}$.

Remark 3.3.1. *Note that the optimal solution set of CP(E^r) contains that of CP($\overline{E}^r$) by the construction of E^r in Step 1-3. Therefore, for each $k \geq 1$, we may simply set $v^0 := x^k$ in Step 1-0 without solving CP(E^0) since CP(E^0) is identical to CP(E^r_*) and x^k solves CP($\overline{E}^r_*$), where E^r_* and $\overline{E}^r_*$ are the finite index sets obtained at the end of Step 1 in the previous outer iteration.*

3.3.2 Global convergence under strict convexity assumption

In the previous subsection, we have proposed the explicit exchange method for solving SICP (3.1.4). In this subsection, we show that the algorithm generates a sequence converging to the optimal solution under the following assumption.

Assumption 3.3.2. *i) Function f is strictly convex over the feasible region of SICP* (3.1.4). *ii) In Step 1-2 of Algorithm 3.1, CP($\overline{E}^{r+1}$) is solvable for each r. iii) A generated sequence $\{v^r\}$ in every Step 1 of Algorithm 3.1 is bounded.*

Notice that all statements i)–iii) automatically hold when f is strongly convex. Under Assumption A, we first show that the inner iterations within Step 1 do not repeat infinitely, which ensures that Algorithm 3.1 is well-defined. To prove this, we provide the following proposition stating that the distance between v^{r+1} and v^r does not tend to zero during the inner iterations in Step 1.

Proposition 3.3.3. *Suppose that Assumption* 3.3.2 *holds. Then, there exists a positive number $N > 0$ such that*

$$\|v^{r+1} - v^r\| \geq N\gamma_k$$

for any $r \geq 0$ and $k \geq 0$.

Proof. Denote $z(v,t) := A(t)^\top v - b(t)$ for simplicity. Due to the continuity of the matrix norm $\|A(t)\| := \max_{\|w\|=1} \|A(t)^\top w\|$ and the compactness of T, there exists a sufficiently large $M > 0$ such that $\|A(t)\| \leq M$ for any $t \in T$. Hence, we have

$$\|z(v^{r+1},t) - z(v^r,t)\| = \|A(t)^\top (v^{r+1} - v^r)\| \leq M\|v^{r+1} - v^r\| \tag{3.3.4}$$

for any $t \in T$.

We next show that $\|z(v^{r+1}, t^r_{\text{new}}) - z(v^r, t^r_{\text{new}})\|$ is bounded below by some positive number for any $r \geq 0$. Since $e \in \operatorname{int} C$, there exists a $\delta > 0$ such that $e + B(0,\delta) \subseteq C$. We therefore have

$$\begin{aligned} z(v^{r+1}, t^r_{\text{new}}) + B(0, \delta\gamma_k) &= -\gamma_k e + z(v^{r+1}, t^r_{\text{new}}) + \gamma_k\,(e + B(0,\delta)) \\ &\subseteq -\gamma_k e + C, \end{aligned} \tag{3.3.5}$$

where the inclusion holds since $e + B(0,\delta) \subseteq C$, $\gamma_k > 0$, $z(v^{r+1}, t^r_{\text{new}}) \in C$, and C is a convex cone[5]. From (3.3.1), we have $z(v^r, t^r_{\text{new}}) \notin -\gamma_k e + C$, which together with (3.3.5) implies that

$$\|z(v^{r+1}, t^r_{\text{new}}) - z(v^r, t^r_{\text{new}})\| \geq \delta\gamma_k. \tag{3.3.6}$$

[5]When C is a convex cone, $\alpha x + \beta y \in C$ holds for any $x, y \in C$ and $\alpha, \beta \geq 0$.

Combining (3.3.4) and (3.3.6) with $N := \delta/M$, we obtain

$$\|v^{r+1} - v^r\| \geq \delta\gamma_k/M = N\gamma_k.$$

□

Theorem 3.3.4. *Suppose that Assumption* 3.3.2 *holds. Then, the inner iterations in Step 1 of Algorithm 3.1 terminate finitely for each* k*.*

Proof. Suppose, for contradiction, that the inner iterations in Step 1 do not terminate finitely at some outer iteration k. (In what follows, k is fixed.) Then, by Assumption 3.3.2 iii), there exist accumulation points v^* and v^{**} of $\{v^r\}$ such that $v^{r_j} \to v^*$ and $v^{r_j+1} \to v^{**}$ as $j \to \infty$. Moreover, we must have $v^* \neq v^{**}$ from Proposition 3.3.3. Denote $z_t^r := A(t)^\top v^r - b(t)$ for simplicity. Since v^r solves $\mathrm{CP}(\overline{E}^r)$, it satisfies the following KKT conditions:

$$\nabla f(v^r) - \sum_{t\in\overline{E}^r} A(t)y_t^r = 0, \tag{3.3.7}$$

$$C^d \ni y_t^r \perp z_t^r \in C \ \ (t \in \overline{E}^r), \tag{3.3.8}$$

where y_t^r are the Lagrange multipliers. From (3.3.3), we have $f(v^1) \leq f(v^2) \leq \cdots \leq V(T) < +\infty$, which implies

$$\lim_{r\to\infty} \left(f(v^{r+1}) - f(v^r)\right) = 0. \tag{3.3.9}$$

Let $F_r := f(v^{r+1}) - f(v^r) - \nabla f(v^r)^\top(v^{r+1} - v^r)$. Then, we have

$$\begin{aligned} f(v^{r+1}) - f(v^r) &= F_r + \nabla f(v^r)^\top(v^{r+1} - v^r) \\ &= F_r + \left(\sum_{t\in\overline{E}^r} A(t)y_t^r\right)^\top (v^{r+1} - v^r) \qquad (3.3.10) \\ &= F_r + \sum_{t\in\overline{E}^r} (y_t^r)^\top z_t^{r+1} - \sum_{t\in\overline{E}^r} (y_t^r)^\top z_t^r \qquad (3.3.11) \\ &= F_r + \sum_{t\in\overline{E}^r} (y_t^r)^\top z_t^{r+1}, \qquad (3.3.12) \end{aligned}$$

where (3.3.10) and (3.3.12) follow from (3.3.7) and (3.3.8), respectively, and (3.3.11) follows from $z_t^r = A(t)^\top v^r - b(t)$ and $z_t^{r+1} = A(t)^\top v^{r+1} - b(t)$. Since f is convex, we have $F_r \geq 0$. In addition, since $y_t^r \in C^d$ and $z_t^{r+1} \in C$, we have $\sum_{t\in\overline{E}^r} (y_t^r)^\top z_t^{r+1} \geq 0$. Therefore, from (3.3.9) and (3.3.12), we have

$$0 = \lim_{r\to\infty} F_r = \lim_{j\to\infty} F_{r_j} = f(v^{**}) - f(v^*) - \nabla f(v^*)^\top(v^{**} - v^*). \tag{3.3.13}$$

However, this contradicts the strict convexity of f since $v^* \neq v^{**}$. Hence, the inner iterations in Step 1 must terminate finitely. □

The next theorem shows global convergence of Algorithm 3.1 under the strict convexity assumption.

Theorem 3.3.5. *Suppose that SICP* (3.1.4) *has a solution and Assumption* 3.3.2 *holds. Let* x^* *be the optimum, and* $\{x^k\}$ *be the sequence generated by Algorithm 3.1. Then, we have*

$$\lim_{k\to\infty} x^k = x^*.$$

Proof. We first show that $\{x^k\}$ is bounded. Let $X(\gamma) := \{x \in \mathbb{R}^n \mid A(t)^\top x - b(t) + \gamma e \in C,\ \forall t \in T\}$ and $L := \{x \in \mathbb{R}^n \mid f(x) \le f(x^*)\}$. Since $x^k \in L \cap X(\gamma_k) \subseteq L \cap X(\overline{\gamma})$ with $\overline{\gamma} := \max_{k\ge 0} \gamma_k$, it suffices to show that $L \cap X(\gamma)$ is bounded for any $\gamma > 0$. By Proposition 3.2.3, there exists a compact set $S \subseteq \mathbb{R}^m$ such that $0 \notin S \subseteq C^d$ and

$$\begin{aligned} X(\gamma) &= \{x \in \mathbb{R}^n \mid s^\top \left(A(t)^\top x - b(t) + \gamma e\right) \ge 0\,,\ \forall (s,t) \in S \times T\} \\ &= \{x \in \mathbb{R}^n \mid (e^\top s)^{-1} \left(s^\top b(t) - (A(t)s)^\top x\right) \le \gamma\,,\ \forall (s,t) \in S \times T\} \\ &= \left\{x \in \mathbb{R}^n \mid h(x) := \max_{(s,t)\in S\times T} (e^\top s)^{-1} \left(s^\top b(t) - (A(t)s)^\top x\right) \le \gamma\right\}, \end{aligned}$$

where the second equality is valid since $e \in \operatorname{int} C$ and $0 \ne s \in S \subseteq C^d$ entail $\min_{s\in S} e^\top s > 0$ from Proposition 3.2.3 (iii). Notice that $h(x) < \infty$ from the compactness of $S \times T$ and continuity of $A(\cdot)$ and $b(\cdot)$. Therefore, function h is closed, proper and convex. Now, let $\overline{h} : \mathbb{R}^n \to (-\infty, +\infty]$ be defined as

$$\overline{h}(x) := \begin{cases} h(x) & (x \in L) \\ \infty & (x \notin L). \end{cases}$$

Then $\overline{h}$ is also closed, proper and convex since L is convex. Notice that

$$L \cap X(\gamma) = \{x \in \mathbb{R}^n \mid \overline{h}(x) \le \gamma\},$$

i.e., $L \cap X(\gamma)$ is a level set of $\overline{h}$. From Proposition 2.2.4, if a closed proper convex function has at least one compact level set, then any nonempty level set is also compact. Moreover, we have $L \cap X(0) = \{x^*\}$ since f is strictly convex. Therefore, $L \cap X(\gamma)$ is compact for any $\gamma \ge 0$.

We next show that $\lim_{k\to\infty} x^k = x^*$. Let $\bar{x}$ be an arbitrary accumulation point of $\{x^k\}$. Then, there exists a subsequence $\{x^{k_j}\} \subseteq \{x^k\}$ such that $\lim_{j\to\infty} x^{k_j} = \bar{x}$. For all j, we have $A(t)^\top x^{k_j} - b(t) + \gamma_{k_j} e \in C$ $(\forall t \in T)$ and $f(x^{k_j}) \le f(x^*)$. Hence, by letting j tend to ∞, we have

$$A(t)^\top \bar{x} - b(t) \in C \quad (\forall t \in T), \tag{3.3.14}$$

$$f(\bar{x}) \le f(x^*) \tag{3.3.15}$$

from the continuity of f and the closedness of C. From (3.3.14), we have $f(\bar{x}) \ge f(x^*)$, which together with (3.3.15) implies $f(\bar{x}) = f(x^*)$. Therefore, $\bar{x}$ solves SICP (3.1.4). Since f is strictly convex, we must have $\bar{x} = x^*$. We thus have $\lim_{k\to\infty} x^k = x^*$. □

3.4 Regularized explicit exchange method for SICP

In the previous section, we have proposed the explicit exchange method for SICP (3.1.4) and analyzed its convergence property. However, to ensure global convergence, we had to assume the strict convexity of the objective function (Assumption 3.3.2). In this section, we propose a new method combining the regularization technique with the explicit exchange method, and establish global convergence without assuming the strict convexity.

3.4.1 Algorithm

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function. Then, the function $f_\varepsilon : \mathbb{R}^n \to \mathbb{R}$ defined by $f_\varepsilon(x) := \frac{1}{2}\varepsilon\|x\|^2 + f(x)$ is strongly convex for any $\varepsilon > 0$. So, if we apply Algorithm 3.1 to the following regularized SICP:

$$\mathrm{RSICP}(\varepsilon) \qquad \begin{array}{ll} \text{Minimize} & f_\varepsilon(x) \\ \text{subject to} & A(t)^\top x - b(t) \in C \quad \text{for all } t \in T, \end{array}$$

then Step 1 always terminates in a finite number of (inner) iterations and the sequence generated by Algorithm 3.1 converges to the unique solution x^*_ε of RSICP(ε).

By introducing a positive sequence $\{\varepsilon_k\}$ converging to 0, we can expect that $x^*_{\varepsilon_k}$ converges to the solution of the original SICP (3.1.4) as k goes infinity. However, since it is computationally prohibitive to solve RSICP(ε_k) exactly for every k, we solve RSICP(ε_k) only approximately by using the explicit exchange method. In the inner iteration of the latter method, we repeatedly solve problems of the form:

$$\mathrm{CP}(\varepsilon_k, T') \qquad \begin{array}{ll} \text{Minimize} & f_{\varepsilon_k}(x) \\ \text{subject to} & A(t_i)^\top x - b(t_i) \in C \quad (i = 1, 2, \dots, p), \end{array}$$

where $T' := \{t_1, t_2, \dots, t_p\} \subseteq T$. The detailed steps of the regularized explicit exchange method are described as follows.

Algorithm 3.2 (Regularized Explicit Exchange Method)

Step 0. Choose positive sequences $\{\gamma_k\} \subseteq \mathbb{R}_{++}$ and $\{\varepsilon_k\} \subseteq \mathbb{R}_{++}$ such that $\lim_{k\to\infty} \gamma_k = \lim_{k\to\infty} \varepsilon_k = 0$. Choose a finite subset $T^0 := \{t^0_1, \dots, t^0_l\} \subseteq T$ for some integer $\ell \geq 0$ and a vector $e \in \operatorname{int} C$. Set $k := 0$.

Step 1. Obtain x^{k+1} and T^{k+1} by the following procedure.

Step 1-0 Set $r := 0$ and $E^0 := T^k$. Solve CP(ε_k, E^0) and let v^0 be an optimum.

Step 1-1 Find $t^r_{\text{new}} \in T$ such that

$$A(t^r_{\text{new}})^\top v^r - b(t^r_{\text{new}}) \notin -\gamma_k e + C. \tag{3.4.1}$$

If such a t^r_{new} does not exist, i.e.,

$$A(t)^\top v^r - b(t) \in -\gamma_k e + C \tag{3.4.2}$$

for any $t \in T$, then set $x^{k+1} := v^r$ and $T^{k+1} := E^r$, and go to Step 2. Otherwise, let

$$\overline{E}^{r+1} := E^r \cup \{t^r_{\text{new}}\},$$

and go to Step 1-2.

Step 1-2 Solve CP(ε_k,$\overline{E}^{r+1}$) to obtain an optimum v^{r+1} and the Lagrange multipliers y^{r+1}_t for $t \in \overline{E}^{r+1}$.

Step 1-3 Let $E^{r+1} := \{t \in \overline{E}^{r+1} \mid y_t^{r+1} \neq 0\}$. Set $r := r + 1$ and return to Step 1-1.

Step 2. If γ_k and ε_k are sufficiently small, then terminate. Otherwise, set $k := k+1$ and return to Step 1.

Algorithm 3.2 differs from Algorithm 3.1 only in the choice of $\{\varepsilon_k\}$ in Step 0 and the subproblems $\mathrm{CP}(\varepsilon_k, \overline{E}^{r+1})$ and $\mathrm{CP}(\varepsilon_k, E^0)$ solved in Step 1. But, we give a full description of Algorithm 3.2 for completeness.

In the subsequent convergence analysis, we omit the termination condition in Step 2, so that the algorithm may generate an infinite sequence $\{x^k\}$. Moreover, to ensure convergence, the sequences of $\{\varepsilon_k\}$ and $\{\gamma_k\}$ are required to satisfy the condition $\gamma_k = O(\varepsilon_k)$.

3.4.2 Global convergence without strict convexity assumption

In this section, we show global convergence of Algorithm 3.2 for SICP (3.1.4) without the strict convexity assumption. Indeed, we only need the following assumption for the proof of convergence.

Assumption 3.4.1. *Function f is convex. Moreover, the Slater constraint qualification (SCQ) holds for SICP (3.1.4), i.e., there exists an $x_0 \in \mathbb{R}^n$ such that $A(t)^\top x_0 - b(t) \in \operatorname{int} C$ for all $t \in T$.*

Notice that, for SICP (3.1.4), the SCQ holds if and only if any feasible point satisfies the GRCQ as studied in Section 3.2.2. We first show that Step 1 terminates finitely.

Proposition 3.4.2. *Suppose that Assumption 3.2 holds. Then, the inner iterations in Step 1 terminate finitely.*

Proof. By Theorem 3.3.4, it suffices to show that conditions i)-iii) in Assumption 3.3.2 hold when Step 1 of Algorithm 3.1 is applied to RSICP(ε) for any $\varepsilon > 0$. Since conditions i) and ii) hold from the strong convexity of f_ε, we only show condition iii). Let x_ε^* be an optimum of RSICP(ε) and $L_\varepsilon^* := \{x \in \mathbb{R}^n \mid f_\varepsilon(x) \leq f_\varepsilon(x_\varepsilon^*)\}$. Then, L_ε^* is compact since f_ε is strongly convex. Moreover, we have $v^r \in L_\varepsilon^*$, i.e., $f_\varepsilon(v^r) \leq f_\varepsilon(x_\varepsilon^*)$ for all r since $\overline{E}^r \subseteq T$. Hence, $\{v^r\}$ is bounded. □

Now, we show that, under Assumption 3.4.1, the generated sequence $\{x^k\}$ is bounded, and Algorithm 3.2 is globally convergent in the sense that the distance from x^k to the solution set of SICP (3.1.4) tends to 0. In the proof, the KKT conditions established in Theorem 3.2.4 plays a critical role.

Theorem 3.4.3. *Suppose that Assumption 3.4.1 holds. Let $\{x^k\}$ be the sequence generated by Algorithm 3.2. Then, the following statements hold.*

i) *If $\{\varepsilon_k\}$ and $\{\gamma_k\}$ are chosen to satisfy $\gamma_k = O(\varepsilon_k)$, then $\{x^k\}$ is bounded.*

ii) *Any accumulation point of $\{x^k\}$ solves SICP (3.1.4).*

Proof. i) Let $x^* \in \mathbb{R}^n$ be an optimum of SICP (3.1.4). Since Assumption 3.4.1 holds, Theorem 3.2.4 can be applied to SICP (3.1.4) to ensure that there exist $t_1, t_2, \ldots, t_p \in T$ and $y^1, y^2, \ldots, y^p \in \mathbb{R}^m$ such that $p \le n$ and

$$\nabla f(x^*) - \sum_{i=1}^{p} A(t_i) y^i = 0, \tag{3.4.3}$$

$$C^d \ni y^i \perp A(t_i)^\top x^* - b(t_i) \in C \quad (i = 1, 2, \ldots, p). \tag{3.4.4}$$

Let $\{x^k\}$ be the sequence generated by Algorithm 3.2. Since x^k solves $\mathrm{CP}(\varepsilon_{k-1}, T^k)$ and x^* is feasible to $\mathrm{CP}(\varepsilon_{k-1}, T^k)$, we have

$$\frac{1}{2}\varepsilon_{k-1}\|x^k\|^2 + f(x^k) \le \frac{1}{2}\varepsilon_{k-1}\|x^*\|^2 + f(x^*). \tag{3.4.5}$$

Multiplying both sides of (3.4.5) by $2/\varepsilon_{k-1}$, we have

$$\begin{aligned}
\|x^k\|^2 &\le \|x^*\|^2 - \frac{2}{\varepsilon_{k-1}}(f(x^k) - f(x^*)) \\
&\le \|x^*\|^2 - \frac{2}{\varepsilon_{k-1}}\nabla f(x^*)^\top (x^k - x^*) \\
&= \|x^*\|^2 - \frac{2}{\varepsilon_{k-1}}\left(\sum_{i=1}^{p} A(t_i) y^i\right)^\top (x^k - x^*),
\end{aligned} \tag{3.4.6}$$

where the second inequality holds since f is convex, and the equality follows from (3.4.3). Moreover, the last term of (3.4.6) satisfies the following inequalities:

$$\begin{aligned}
&-\left(\sum_{i=1}^{p} A(t_i) y^i\right)^\top (x^k - x^*) \\
&= -\sum_{i=1}^{p} (y^i)^\top (A(t_i)^\top x^k - b(t_i) + \gamma_{k-1} e) + \sum_{i=1}^{p} (y^i)^\top (\gamma_{k-1} e) + \sum_{i=1}^{p} (y^i)^\top (A(t_i)^\top x^* - b(t_i)) \\
&\le \sum_{i=1}^{p} (y^i)^\top (\gamma_{k-1} e) \\
&\le p\mu \|e\| \gamma_{k-1},
\end{aligned} \tag{3.4.7}$$

where $\mu := \max\{\|y^1\|, \|y^2\|, \ldots, \|y^p\|\}$, and the first inequality since (3.4.2) and (3.4.4) imply $y^i \in C^d$, $A(t_i)^\top x^k - b(t_i) + \gamma_{k-1} e \in C$ and $(y^i)^\top (A(t_i)^\top x^* - b(t_i)) = 0$. Then, by substituting (3.4.7) into (3.4.6), we have

$$\|x^k\|^2 \le \|x^*\|^2 + 2p\mu\|e\|\gamma_{k-1}/\varepsilon_{k-1}. \tag{3.4.8}$$

Since $\gamma_{k-1} = O(\varepsilon_{k-1})$, $\{\gamma_{k-1}/\varepsilon_{k-1}\}$ is bounded, and hence $\{x^k\}$ is also bounded.

ii) Let $\bar{x}$ be an accumulation point of $\{x^k\}$. Then, taking a subsequence if necessary, we have

$$x^k \to \bar{x}, \qquad \varepsilon_{k-1} \to 0, \qquad \gamma_{k-1} \to 0 \qquad (k \to \infty).$$

First, we show that $\bar{x}$ is feasible to SICP (3.1.4). Since x^k is determined as v^r satisfying (3.4.2) with γ_k replaced by γ_{k-1}, $A(t)^\top x^k - b(t) + \gamma_{k-1} e \in C$ holds for any $t \in T$. Noticing that

C is closed, we have $\lim_{k\to\infty} A(t)^\top x^k - b(t) + \gamma_{k-1}e = A(t)^\top \bar{x} - b(t) \in C$ for any $t \in T$. Hence, $\bar{x}$ is feasible to SICP (3.1.4).

We next show that $\bar{x}$ is optimal to SICP (3.1.4). Let x^* be an arbitrary optimum of SICP (3.1.4). Since $\bar{x}$ is feasible to SICP (3.1.4), we have $f(\bar{x}) \geq f(x^*)$. On the other hand, x^* is feasible to CP(ε_{k-1}, E_k) since the feasible region of SICP (3.1.4) is contained in that of CP(ε_{k-1}, E_k). Hence, we have

$$\frac{1}{2}\varepsilon_{k-1}\|x^k\|^2 + f(x^k) \leq \frac{1}{2}\varepsilon_{k-1}\|x^*\|^2 + f(x^*). \tag{3.4.9}$$

Due to the continuity of f, by letting $k \to \infty$ in (3.4.9), we have $f(\bar{x}) \leq f(x^*)$. Therefore, we obtain $f(x^*) = f(\bar{x})$, which implies that $\bar{x}$ solves SICP (3.1.4). □

From the above theorem, we can see that if we choose $\{\varepsilon_k\}$ and $\{\gamma_k\}$ such that $\gamma_k = O(\varepsilon_k)$, then the generated sequence $\{x^k\}$ has an accumulation point and it solves SICP (3.1.4). Moreover, we can show that, if $\{\varepsilon_k\}$ and $\{\gamma_k\}$ satisfy $\gamma_k = o(\varepsilon_k)$, then $\{x^k\}$ is actually convergent and its limit point is the least 2-norm solution.

Theorem 3.4.4. *Suppose that Assumption* 3.4.1 *holds. Let* $\{\varepsilon_k\}$ *and* $\{\gamma_k\}$ *be chosen such that* $\gamma_k = o(\varepsilon_k)$, *and* $\{x^k\}$ *be a sequence generated by Algorithm* 3.2. *Let* $S^* \subseteq \mathbb{R}^n$ *denote the nonempty solution set of SICP* (3.1.4) *and* $x^* \in \mathbb{R}^n$ *be the least* 2*-norm solution, i.e.,* $x^*_{\min} := \mathrm{argmin}_{x\in S^*}\|x\|$. *Then, we have* $\lim_{k\to\infty} x^k = x^*_{\min}$.

Proof. By Theorem 3.4.3, $\{x^k\}$ is bounded and every accumulation point belongs to S^*. Moreover, $x^*_{\min}$ can be identified uniquely since S^* is closed and convex. Therefore, it suffices to show that $\|\bar{x}\| = \|x^*_{\min}\|$ for any accumulation point $\bar{x}$ of $\{x^k\}$. Note that the inequality (3.4.8) in the proof of Theorem 3.4.3. ii) holds for any $x^* \in S$, in particular, for $x^*_{\min}$. Since $\gamma_k = o(\varepsilon_k)$, by letting $k \to \infty$ in (3.4.8), we obtain $\|\bar{x}\| \leq \|x^*_{\min}\|$. On the other hand, we have $\|\bar{x}\| \geq \|x^*_{\min}\|$, since $\bar{x} \in S^*$ and $x^*_{\min} = \mathrm{argmin}_{x\in S^*}\|x\|$. We thus have $\|\bar{x}\| = \|x^*_{\min}\|$. □

3.5 Numerical experiments

In this section, we report some numerical results. The program is coded in Matlab 2008a and run on a machine with an Intel®Core2 Duo E6850 3.00GHz CPU and 4GB RAM. In this experiment, we consider SISOCP (3.1.1) with a linear objective function and infinitely many second-order cone constraints with respect to a single second-order cone. Actual implementation of Algorithm 3.2 is carried out as follows. In Step 0, we set $e := (1, 0, \ldots, 0)^\top \in \mathrm{int}\,\mathcal{K}^m$. In Step 1-1, to find t^r_{new} satisfying (3.4.1), we first choose N grid points $\bar{t}_1, \bar{t}_2, \ldots, \bar{t}_N$ from the index set T and compute $\lambda\left(A(t)^\top v^r - b(t) + \gamma_k e\right)$ for $t = \bar{t}_1, \bar{t}_2, \ldots, \bar{t}_N \in T$, where $\lambda(\cdot)$ denotes the spectral value of $z \in \mathbb{R}^m$ [18, 26] defined by

$$\lambda(z) := z_1 - \sqrt{z_2^2 + z_3^2 + \cdots + z_m^2}.$$

If we find a $\bar{t} \in \{\bar{t}_1, \bar{t}_2, \ldots, \bar{t}_N\}$ such that $\lambda\left(A(\bar{t})^\top v^r - b(\bar{t}) + \gamma_k e\right) < 0$, then we set $t^r_{\mathrm{new}} := \bar{t}$. Otherwise, we solve

$$\begin{aligned} &\text{Minimize} \;\; \lambda(A(t)^\top v^r - b(t) + \gamma_k e) \\ &\text{subject to} \;\; t \in T, \end{aligned} \tag{3.5.1}$$

and check the nonnegativity of its optimal value.[6] To solve (3.5.1), we apply Newton's method combined with the bisection method when T is one-dimensional, and *fmincon* solver in Matlab Optimization Toolbox when T is multi-dimensional. For both methods, we set the initial point $\bar{t}_0 \in T$ as $\bar{t}_0 := \operatorname{argmin}\{\lambda\left(A(t)^\top v^r - b(t) + \gamma_k e\right) \mid t = \bar{t}_1, \bar{t}_2, \dots, \bar{t}_N\}$. Although there is no theoretical guarantee, in practice, we can expect to find a global optimum of (3.5.1) by taking a sufficiently large N. In Step 1-2, we solve CP(ε, T') by the smoothing method [18, 26]. In Step 1-3, we regard y_t^r as 0 if $\|y_t^r\| \le 10^{-12}$. In Step 2, we terminate the algorithm if $\max(\varepsilon_k, \gamma_k) \le 10^{-5}$. In each experiment reported below, we choose the grid points $\bar{t}_1, \bar{t}_2, \dots, \bar{t}_N \in T$ as in Table 3.1.

Experiment	T	N	$\{\bar{t}_1, \bar{t}_2, \dots, \bar{t}_N\}$
1, 2, 3-1	$[-1, 1]$	101	$\{-1 + \frac{1}{50}p\}_{p=0,1,\dots,100}$
3-2	$[0,1] \times [0,1]$	$2601 (= 51^2)$	$\left\{\frac{1}{50}(p_1, p_2)\right\}_{p_1, p_2 = 0,1,2,\dots,50}$

Table 3.1: Choice of grid points in each experiment

Experiment 1

In the first experiment, we solve the following SICP:

$$\begin{aligned} &\text{Minimize} && c^\top x \\ &\text{subject to} && A(t)^\top x - b(t) \in \mathcal{K}^m \text{ for all } t \in [-1, 1], \end{aligned} \tag{3.5.2}$$

where $\mathcal{K}^m := \{(x_1, x_2, \dots, x_m)^\top \in \mathbb{R}^m \mid x_1 \ge \|(x_2, x_3, \dots, x_m)^\top\|\}$, $c \in \mathbb{R}^n$, $A(t) := (A_{ij}(t)) \in \mathbb{R}^{n \times m}$ with $A_{ij}(t) := \alpha_{ij0} + \alpha_{ij1}t + \alpha_{ij2}t^2 + \alpha_{ij3}t^3$ $(i = 1, 2, \dots, n,\ j = 1, 2, \dots, m)$ and $b(t) := (b_j(t)) \in \mathbb{R}^m$ with $b_1(t) := -\sum_{j=2}^m \sum_{\ell=0}^3 |\beta_j^\ell|$ and $b_j(t) := \beta_{j0} + \beta_{j1}t + \beta_{j2}t^2 + \beta_{j3}t^3$ $(j = 2, \dots, m)$. We choose $\alpha_{ijk}, \beta_{j\ell}$ $(i = 1, 2, \dots, n,\ j = 2, \dots, m,\ k = 0, 1, 2, 3,\ \ell = 0, 1, 2, 3)$ and all components of c randomly from $[-1, 1]$. Note that by the choice of $b_1(t)$, feasibility of (3.5.2) is ensured.[7] In this way, we generate two sets of data $A(t)$, $b(t)$ and c for each of the three pairs $(m, n) = (25, 15)$, $(15, 15)$ and $(10, 15)$, thereby obtaining six problems referred to as Problems $1, 2, \dots, 6$.

In this experiment, using parameters $\{\varepsilon_k\}$ and $\{\gamma_k\}$ such that $\varepsilon_k = 0.5^k, \gamma_k = 0.3^k$, and the initial index set $T^0 := \{-1, 0, 1\}$ in Step 0, we observe the convergence behavior of the algorithm. The results are shown in Table 3.2, where

[6] Notice that $\lambda\left(A(t)^\top x - b(t) + \gamma e\right) \ge 0$ if and only if $A(t)^\top x - b(t) \in -\gamma e + \mathcal{K}^m$.

[7] Note that the origin always lies in the interior of the feasible region, since we have $-b(t) \in \operatorname{int} \mathcal{K}^m$ from $-b_1(t) - \|(-b_2(t), \dots, -b_m(t))^\top\| > 0$ for all $t \in [-1, 1]$.

ite_{out} : the number of outer iterations,

$\{\bar{r}_k\}$: the values of r when the inner termination criterion (3.4.2) is satisfied at the k-th outer iteration for $k = 0, 1, \dots, \text{ite}_{\text{out}} - 1$,

$\bar{r}_{\text{sum}}$: the sum of $\bar{r}_k$'s for all $k = 0, 1, 2, \dots, \text{ite}_{\text{out}} - 1$,

t_{socp} : the number of times the sub-SOCPs ($\text{CP}(\varepsilon_k, E^0)$ and $\text{CP}(\varepsilon_k, \overline{E}^{r+1})$) are solved,

T_{fin} : the index set T^k upon termination of the algorithm,

time(sec) : the CPU time in seconds.

In the column of $\bar{r}_k$, p^q means that we have $\bar{r}_k = p$ in q consecutive iterations. For example, $0^{10}, 2, 1^4$ means that $\bar{r}_k = 0$ $(k = 0, 1, \dots, 9), \bar{r}_{10} = 2$ and $\bar{r}_k = 4$ $(k = 11, 12, 13, 14)$. Notice that we always have $t_{\text{socp}} = \text{ite}_{\text{out}} + \bar{r}_{\text{sum}}$, since we solve sub-SOCPs once at Step 1-0 and $\bar{r}_k$ times at Step 1-3, for each k. Although T_{fin} usually represents an approximate active index set at the optimum, the real active index set is $\{-1, 1\}$ for Problems 2 and 3. This is because the inner termination criterion (3.4.2) is always satisfied with $r = 0$ and therefore the inactive index $t = 0$ is never removed at Step 1-3. From the columns of $\bar{r}_k$, we can see that $\bar{r}_k$ is sometimes large when $k \leq 4$, but it is always 0 or 1 for $k = 7, 8, \dots, 17$. This fact suggests that T_{fin} is usually obtained in the early stage of iterations.

Problem	(m, n)	ite_{out}	$\{\bar{r}_k\}$	$\bar{r}_{\text{sum}}$	t_{socp}	T_{fin}	time(sec)
1	(25, 15)	18	$0^6, 4, 0^8, 1, 0^2$	5	23	$\{-1, -0.296, 1\}$	5.57
2	(25, 15)	18	0^{18}	0	18	$\{-1, 0, 1\}$	2.41
3	(15, 15)	18	0^{18}	0	18	$\{-1, 0, 1\}$	1.84
4	(15, 15)	18	$0^3, 11, 0^2, 1, 0^{11}$	14	32	$\{-1, -0.2, -0.18, 1\}$	12.49
5	(10, 15)	18	$0^2, 13, 0, 3, 0, 3, 0, 1, 0^9$	20	38	$\{-1, -0.48, -0.46, 1\}$	3.83
6	(10, 15)	18	$0^2, 7, 4, 6, 2^2, 0^5, 1, 0^3, 1, 0$	23	41	$\{-1, -0.387, 0.25, 1\}$	12.91

Table 3.2: Convergence behavior for Experiment 1

Experiment 2

In the second experiment, we implement the non-regularized exchange method (Algorithm 3.1) as well as the regularized exchange method (Algorithm 3.2), and compare their performance. In Step 1 of Algorithm 3.1, for $k \geq 1$ we set $v^0 := x^k$ instead of solving $\text{CP}(E^0)$, as suggested in Remark 3.3.1 in Section 3.3. For both methods, the initial index set T^0 is set to be $T_a^0 := \{-1, -0.5, 0, 0.5, 1\}, T_b^0 := \{-1, 0, 1\}$, or $T_c^0 = \{-0.5, 0, 0.5\}$. The parameters are chosen as $\gamma_k = 0.5^k$ for Algorithm 3.1, and $\varepsilon_k = \gamma_k = 0.5^k$ for Algorithm 3.2. Both methods are applied to the same problems as those used in Experiment 1.

Table 3.3 shows the obtained results, where $t^a_{\text{socp}}, t^b_{\text{socp}}$ and t^c_{socp} denote the values of t_{socp} for the initial index sets T_a^0, T_b^0 and T_c^0, respectively, and "F" means that we fail to solve a problem. From the table, we can observe that t_{socp} for the non-regularized method is much less than t_{socp} for the regularized method. This is due to the fact that the regularized exchange method has to solve the sub-SOCP (CP(ε_k, E^0)) at least once in every outer iteration, whereas the non-regularized exchange method does not need to solve it when the inner termination criterion (3.3.2) is satisfied for $r = 0$. However, convergence of the non-regularized exchange method is not guaranteed theoretically since the objective function is linear. Indeed, the non-regularized exchange method fails to solve Problems 4, 5 and 6 with $T^0 = T_c^0$ since their objective functions are most probably unbounded on the feasible sets.[8] On the other hand, the regularized exchange method succeeds in solving all problems for any choice of T^0. This is the main advantage of the regularized exchange method.

		regularized			non-regularized		
Problem	(m,n)	t^a_{socp}	t^b_{socp}	t^c_{socp}	t^a_{socp}	t^b_{socp}	t^c_{socp}
1	$(25, 15)$	23	23	34	5	5	32
2	$(25, 15)$	18	18	20	1	1	11
3	$(15, 15)$	18	18	25	1	1	15
4	$(15, 15)$	27	28	44	4	4	F
5	$(10, 15)$	19	24	29	4	5	F
6	$(10, 15)$	28	30	46	8	8	F

Table 3.3: Comparison of regularized and non-regularized exchange methods

Experiment 3

In the third experiment, we apply Algorithm 3.2 to Chebyshev-like approximation problems for vector-valued functions.

Experiment 3-1. We first consider the vector-valued approximation problem with respect to $H : \mathbb{R} \to \mathbb{R}^3$ and $h : \mathbb{R}^8 \times \mathbb{R} \to \mathbb{R}^3$ defined by

$$H(t) := \begin{pmatrix} e^{t^2} \\ 2te^{t^2} \\ (4t^2+2)e^{t^2} \end{pmatrix}, \quad h(u,t) := \begin{pmatrix} \sum_{\nu=1}^{8} u_\nu t^{\nu-1} \\ \sum_{\nu=2}^{8} (\nu-1) u_\nu t^{\nu-2} \\ \sum_{\nu=3}^{8} (\nu-1)(\nu-2) u_\nu t^{\nu-3} \end{pmatrix}.$$

In order to find a $u \in \mathbb{R}^8$ such that $h(u,t) \approx H(t)$ over $t \in [-1, 1]$, we solve the following problem:

$$\underset{u \in \mathbb{R}^8}{\text{Minimize}} \quad \max_{t \in [-1,1]} \|H(t) - h(u,t)\| . \tag{3.5.3}$$

[8] In fact, for each CP(T_c^0) of Problems 4, 5 and 6, we found a feasible point whose objective function value is less than -10^7.

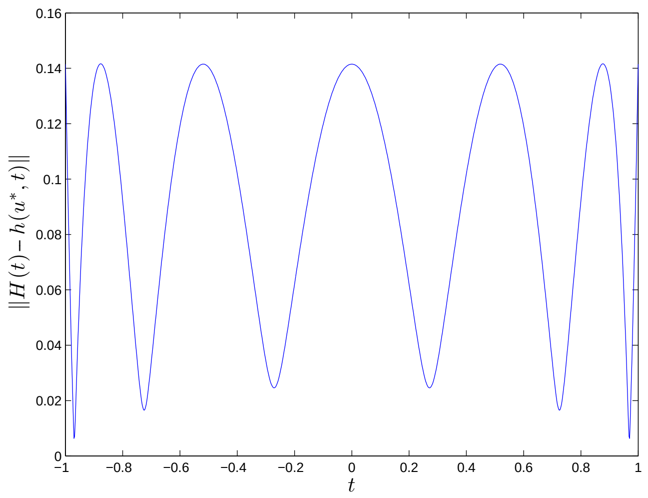

Figure 3.1: The graph of $\|H(t) - h(u^*, t)\|$ ($t \in [-1, 1]$) in Experiment 3-1

Introducing an auxiliary variable $v \in \mathbb{R}$, we can reformulate (3.5.3) as the following SICP with infinitely many four-dimensional second-order cone constraints:

$$\begin{aligned} \underset{(v,u^\top)^\top \in \mathbb{R}\times\mathbb{R}^8}{\text{Minimize}} \quad & v \\ \text{subject to} \quad & \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & t & t^2 & \cdots & t^7 \\ 0 & 0 & 1 & 2t & \cdots & 7t^6 \\ 0 & 0 & 0 & 2 & \cdots & 42t^5 \end{pmatrix} \begin{pmatrix} v \\ u \end{pmatrix} - \begin{pmatrix} 0 \\ e^{t^2} \\ 2te^{t^2} \\ (4t^2+2)e^{t^2} \end{pmatrix} \in \mathcal{K}^4 \\ & \text{for all } t \in [-1, 1]. \end{aligned} \tag{3.5.4}$$

In applying Algorithm 3.2, we set $T_0 := \{-1, 1\}$ and $\varepsilon_k = \gamma_k := 0.5^k$. Then, the algorithm outputs the solution $v^* = 0.1415$, $u^* = (0.9948, 0.0000, 1.0707, 0.0000, 0.3083, 0.0000, 0.3442, 0.0000)^\top$ together with $T_{\text{fin}} = \{-1.00, -0.88, -0.52, 0, 0.52, 0.88, 1.00\}$. Notice that we have $u_2^* = u_4^* = u_6^* = u_8^* = 0$. This is reasonable since $H_1(t)$ and $H_3(t)$ are even functions, whereas $H_3(t)$ is an odd function. Figure 3.1 shows the graph of $\|H(t) - h(u^*, t)\|$ over $t \in [-1, 1]$. From the graph, we can observe that the values of $\|H(t) - h(u^*, t)\|$ is bounded above by $v^* = 0.1415$, and the bound is attained at multiple points in $[-1, 1]$. Actually, those points coincide with $T_{\text{fin}} = \{-1.00, -0.88, -0.52, 0, 0.52, 0.88, 1.00\}$, which correspond to the active constraints at the optimum.

Experiment 3-2. We next consider a vector-valued approximation problem where T is two-dimensional. Let $\tilde{H} : \mathbb{R}^2 \to \mathbb{R}^3$ and $\tilde{h} : \mathbb{R}^8 \times \mathbb{R}^2 \to \mathbb{R}^3$ be defined by

$$\tilde{H}(t_1, t_2) := \begin{pmatrix} \log(t_1 + t_2 + 1) \sin t_1 \\ \sin t_1/(t_1 + t_2 + 1) + \log(t_1 + t_2 + 1) \cos t_1 \\ \sin t_1/(t_1 + t_2 + 1) \end{pmatrix}$$

$v^*, (u^*)^\top$	$0.9730, (-0.1189, 0.2040, 0.2867, -1.0159, 0.9723, 0.1877, -0.3704, 0.1687)$
T_{fin}	$\{(1,0), (0,1), (1,1), (0.64, 0.60), (0.46, 1), (0.60, 0.68), (0.58, 0.74), (0.62, 0.64)\}$
$\{\bar{r}_k\}$	$0^2, 2, 0^2, 3, 1, 4, 0, 1, 5, 3, 12, 6, 1, 2, 0^2$
$\bar{r}_{\text{sum}}$	40
time(sec)	15.40

Table 3.4: Results for Experiment 3-2

and

$$\tilde{h}(u, t_1, t_2) := \begin{pmatrix} \sum_{\nu=1}^{8} u_\nu t_1^{\nu-1} t_2^{8-\nu} \\ \sum_{\nu=2}^{8} u_\nu (\nu-1) t_1^{\nu-2} t_2^{8-\nu} \\ \sum_{\nu=2}^{7} u_\nu (8-\nu) t_1^{\nu-1} t_2^{7-\nu} \end{pmatrix},$$

respectively. In order to find a vector $u := (u_1, u_2, \ldots, u_8)^\top \in \mathbb{R}^8$ such that $\tilde{h}(u, t_1, t_2) \approx \tilde{H}(t_1, t_2)$ for $(t_1, t_2) \in [0,1] \times [0,1]$, we solve the following problem:

$$\underset{u \in \mathbb{R}^8}{\text{Minimize}} \quad \max_{(t_1,t_2) \in [0,1]\times[0,1]} \left\| \tilde{H}(t_1, t_2) - \tilde{h}(u, t_1, t_2) \right\|. \tag{3.5.5}$$

Introducing an auxiliary variable $v \in \mathbb{R}$, we can reformulate (3.5.5) as the following SICP with infinitely many four-dimensional second-order cone constraints and two-dimensional index set:

$$\begin{aligned} &\underset{(v, u^\top)^\top \in \mathbb{R}\times\mathbb{R}^8}{\text{Minimize}} \quad v \\ &\text{subject to} \quad \begin{pmatrix} v \\ \sum_{\nu=1}^{8} u_\nu t_1^{\nu-1} t_2^{8-\nu} \\ \sum_{\nu=2}^{8} u_\nu (\nu-1) t_1^{\nu-2} t_2^{8-\nu} \\ \sum_{\nu=2}^{7} u_\nu (8-\nu) t_1^{\nu-1} t_2^{7-\nu} \end{pmatrix} - \begin{pmatrix} 0 \\ \log(t_1+t_2+1)\sin t_1 \\ \sin t_1/(t_1+t_2+1) + \log(t_1+t_2+1)\cos t_1 \\ \sin t_1/(t_1+t_2+1) \end{pmatrix} \in \mathcal{K}^4 \\ &\qquad\qquad \text{for all } (t_1, t_2) \in [0,1]\times[0,1]. \end{aligned} \tag{3.5.6}$$

In applying Algorithm 3.2, we set $T_0 := \{(0,0), (0,1), (1,0), (1,1)\}$ and $\varepsilon_k = \gamma_k := 0.5^k$. The results are shown in Table 3.4, where $(v^*, (u^*)^\top)^\top$ is the computed optimal solution. From the table, we can observe that Algorithm 3.2 obtained the solution within acceptable time (15.40 seconds). Moreover, the values of $|T_0|$, $|T_{\text{fin}}|$ and $\bar{r}_{\text{sum}}$ indicate that $36 (= |T_0| + \bar{r}_{\text{sum}} - |T_{\text{fin}}|)$ indices are discarded in Step 1-3 in total. Thus, the exchange-scheme in Step 1 worked efficiently to prevent the size of problems $\text{CP}(\varepsilon_k, \overline{E}^{r+1})$ from growing excessively.

3.6 Concluding remarks

For the semi-infinite program with an infinitely many conic constraints (SICP), we have shown that the KKT conditions can be represented with finitely many conic constraints, as long as the generalized Robinson constraint qualification holds. Furthermore, for solving the SICP with a convex objective function and affine conic constrains, we have proposed the explicit exchange method and the regularized explicit exchange method, and established their global convergence.

Finally, we have conducted numerical experiments with the proposed algorithms to examine their effectiveness.

Chapter 4

A local reduction based SQP-type method for semi-infinite second-order cone programming problems

4.1 Introduction

In this chapter, we focus on the following semi-infinite program with an infinite number of second-order cone constraints (SISOCP):

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{Minimize}} \quad & f(x) \\ \text{subject to} \quad & g(x,t) \in \mathcal{K}^m \ \text{ for all } t \in T, \end{aligned} \tag{4.1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \times \mathbb{R}^\ell \to \mathbb{R}^m$ are twice continuously differentiable functions, and T is a nonempty *compact* index set given by

$$T := \{t \in \mathbb{R}^\ell \mid h_i(t) \geq 0,\ i = 1, 2, \dots, p\},$$

where $h_i : \mathbb{R}^\ell \to \mathbb{R}$ are twice continuously differentiable functions for $i = 1, 2, \dots, p$.

We consider the problem (4.1.1) that contains a single SOC with $m \geq 2$ for simplicity of expression, although we can deal with the more general SISOCP that contains multiple SOCs as well as equality constraints, i.e.,

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{Minimize}} \quad & f(x) \\ \text{subject to} \quad & g^0(x) = 0, \\ & g^s(x,t) \in \mathcal{K}^{m_s} \ \text{ for all } t \in T^s \ (s = 1, 2, \dots, S), \end{aligned} \tag{4.1.2}$$

where $g^0 : \mathbb{R}^n \to \mathbb{R}^{m_0}$ and $g^s : \mathbb{R}^n \times \mathbb{R}^{\ell_s} \to \mathbb{R}^{m_s}$ $(s = 1, 2, \dots, S)$ are twice continuously differentiable functions, and $T^s \subseteq \mathbb{R}^{\ell_s}$ $(s = 1, 2, \dots, S)$ are nonempty compact index sets given by $T^s := \{t \in \mathbb{R}^{\ell_s} \mid h_i^s(t) \geq 0,\ i = 1, 2, \dots, p_s\}$ with twice continuously differentiable functions

$h_i^s : \mathbb{R}^{\ell_s} \to \mathbb{R}$ $(i = 1, 2, \dots, p_s)$. It is possible to extend the subsequent analysis for (4.1.1) to the general SISOCP (4.1.2) in a direct manner. In fact, we will show some numerical results for SISOCPs that contain multiple SOCs; see Experiment 3 in Section 4.5.

In the previous chapter, we proposed exchange-type methods to solve semi-infinite programs with infinitely many affine conic constraints. Although an exchange-type algorithm is effective to find an approximate solution, it is not very suitable to obtain an accurate solution. On the other hand, a local reduction-type method is known to have an advantage in computing an accurate solution with fast convergence speed [24, 66, 50, 51]. In this chapter, for solving SISOCP (4.1.1), we propose a local reduction based method combined with a sequential quadratic programming (SQP) method, where, in each iteration, we replace the SISOCP with an SOCP by means of the local reduction method and then generate a search direction by solving a quadratic SOCP that approximates the obtained SOCP.

This chapter is organized as follows. In Section 4.2, we study the local reduction method for SISOCP (4.1.1). We define some concepts and give important propositions to represent the SISOCP locally as an SOCP by using implicit functions. In Section 4.3, we propose an SQP-type method combined with the local reduction method for solving SISOCP (4.1.1). In Section 4.4, we analyze the global and local convergence properties of the proposed algorithm. In Section 4.5, we observe the effectiveness of the algorithm by some numerical experiments. In Section 4.6, we conclude the chapter with some remarks.

4.2 Local reduction of SISOCP to SOCP

In this section, we study the local reduction method for SISOCP (4.1.1). In relation to the constraints in SISOCP (4.1.1), we first consider the problem

$$P(x): \quad \begin{array}{ll} \underset{t\in\mathbb{R}^\ell}{\text{Minimize}} & \lambda(x,t) := g_1(x,t) - \|\tilde{g}(x,t)\| \\ \text{subject to} & t \in T = \{t \in \mathbb{R}^\ell \mid h_i(t) \ge 0,\ i = 1, 2, \dots, p\}, \end{array} \tag{4.2.1}$$

where $g_1(x,t)$ is the first component of $g(x,t) \in \mathbb{R}^m$ and $\tilde{g}(x,t)$ is the vector consisting of the remaining $m-1$ components of $g(x,t)$. We call problem (4.2.1) the lower-level problem of SISOCP (4.1.1) and let

$$\varphi(x) := \max_{t\in T} \left(-\lambda(x,t)\right). \tag{4.2.2}$$

Obviously, the infinitely many SOC constraints $g(x,t) \in \mathcal{K}^m$ $(t \in T)$ are equivalent to the condition $\varphi(x) \le 0$. Hence, SISOCP (4.1.1) can be rewritten equivalently as

$$\underset{x\in\mathbb{R}^n}{\text{Minimize}} \;\; f(x) \;\; \text{subject to} \;\; \varphi(x) \le 0. \tag{4.2.3}$$

Though problem (4.2.3) has only one constraint, treating $\varphi(x) \le 0$ directly is difficult since it is not differentiable everywhere. As a remedy, we take the local reduction method. In this method, at any $\bar{x} \in \mathbb{R}^n$, we find an open neighborhood $U(\bar{x}) \subseteq \mathbb{R}^n$ of $\bar{x}$ and continuously differentiable functions $t_j : U(\bar{x}) \to T$ $(j = 1, 2, \dots, r(\bar{x}))$ such that

$$\varphi(x) = \max_{1\le j\le r(\bar{x})} \left(-\lambda(x, t_j(x))\right)$$

holds for all $x \in U(\bar{x})$, where each $t_j(x)$ represents a local maximum of $-\lambda(x,t)$ on T and $r(\bar{x})$ is a positive integer. This means that $\varphi(x) \geq 0$ may be reduced to the finitely many SOC constraints $g(x, t_j(x)) \in \mathcal{K}^m$ $(j = 1, 2, \dots, r)$ in the set $U(\bar{x})$, i.e., problem (4.2.3) can be transformed locally to

$$\begin{aligned} &\underset{x \in U_\varepsilon(\bar{x})}{\text{Minimize}} && f(x) \\ &\text{subject to} && g(x, t_j(x)) \in \mathcal{K}^m \quad (j = 1, 2, \dots, r(\bar{x})). \end{aligned} \tag{4.2.4}$$

Then, we can expect that existing methods such as an SQP-type method [33] work efficiently for solving the reduced SOCP (4.2.4).

To give more formal treatment of the local reduction method, let $l : \mathbb{R}^n \times \mathbb{R}^\ell \times \mathbb{R}^p \to \mathbb{R}$ denote the Lagrangian of the lower-level problem $P(x)$, i.e.,

$$l(x, t, \alpha) := \lambda(x, t) - h(t)^\top \alpha,$$

where $h(t) := (h_1(t), h_2(t), \dots, h_p(t))^\top$, and $\alpha := (\alpha_1, \alpha_2, \dots, \alpha_p)^\top \in \mathbb{R}^p$ is a Lagrange multiplier vector corresponding to the constraints $h_i(t) \geq 0$ $(i = 1, 2, \dots, p)$. Let $I_a(t)$ denote the active index set at $t \in \mathbb{R}^\ell$, i.e.,

$$I_a(t) := \{i \in \{1, 2, \dots, p\} \mid h_i(t) = 0\}. \tag{4.2.5}$$

Recall that if $t \in T$ is a local optimum of $P(x)$ such that the linear independence constraint qualification holds, i.e., $\{\nabla h_i(t)\}_{i \in I_a(t)}$ are linearly independent, then there exists a unique Lagrange multiplier vector $\alpha := (\alpha_1, \alpha_2, \dots, \alpha_p)^\top \in \mathbb{R}^p$ such that

$$\nabla_t l(x, t, \alpha) = 0, \quad 0 \leq \alpha \perp h(t) \geq 0. \tag{4.2.6}$$

Below, we define the *nondegeneracy* of local optima of $P(x)$.

Definition 4.2.1 (Nondegeneracy). *Let $\bar{x} \in \mathbb{R}^n$ be an arbitrary vector. Let $\bar{t} \in T$ be a local optimum of $P(\bar{x})$ such that the linear independence constraint qualification holds, and $\bar{\alpha} := (\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_p)^\top \in \mathbb{R}^p$ be a Lagrange multiplier vector satisfying* (4.2.6) *with $x := \bar{x}, t := \bar{t}$, and $\alpha := \bar{\alpha}$. Let the function $\lambda(\cdot, \cdot)$ be twice continuously differentiable at $(\bar{x}, \bar{t})$. We say that $\bar{t} \in T$ is nondegenerate if*

(a) *the second-order sufficient condition*

$$v^\top \nabla^2_{tt} l(\bar{x}, \bar{t}, \alpha) v > 0 \text{ for all } v \in C(\bar{t}) \setminus \{0\}$$

with

$$C(t) := \begin{cases} \{v \in \mathbb{R}^\ell \mid v^\top \nabla h_i(t) = 0,\ i \in I_a(t)\} & (I_a(t) \neq \emptyset), \\ \mathbb{R}^\ell & (I_a(t) = \emptyset) \end{cases}$$

holds, and

(b) *the strict complementarity*

$$\alpha_i > 0 \text{ for all } i \in I_a(\bar{t})$$

holds.

Under the nondegeneracy assumption, we have the following proposition.

Proposition 4.2.2. *Let $x \in \mathbb{R}^n$ and $\bar{x} \in \mathbb{R}^n$. Assume that $\bar{t} \in T$ is a nondegenerate local optimum of $P(\bar{x})$ and $\bar{\alpha} := (\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_p)^\top \in \mathbb{R}^p_+$ is a Lagrange multiplier vector corresponding to the constraints $h_i(t) \geq 0$ $(i = 1, 2, \dots, p)$. Furthermore, suppose that $\lambda(\cdot, \cdot)$ is twice continuously differentiable at $(\bar{x}, \bar{t})$. Then, there exist an open neighborhood $U(\bar{x})$ of $\bar{x}$ and twice continuously differentiable functions $t(\cdot) : U(\bar{x}) \to T$ and $\alpha_i(\cdot) : U(\bar{x}) \to \mathbb{R}_+$ $(i = 1, 2, \dots, p)$ such that*

(a) $t(\bar{x}) = \bar{t},\ \alpha_i(\bar{x}) = \bar{\alpha}_i\ (i \in I_a(\bar{t}))$ *and* $\alpha_i(\bar{x}) = 0\ (i \notin I_a(\bar{t}))$,

(b) $t(x)$ *is a nondegenerate local optimum of* $P(x)$ *for each* $x \in U(\bar{x})$ *with a unique Lagrange multiplier vector* $(\alpha_1(x), \alpha_2(x), \dots, \alpha_p(x))^\top \in \mathbb{R}^p_+$,

(c) $\nabla t(\bar{x}) \in \mathbb{R}^{n \times \ell}$ *and* $\nabla \alpha_i(\bar{x}) \in \mathbb{R}^n$ $(i \in I_a(\bar{t}))$ *comprise a unique solution of the linear system*

$$\begin{pmatrix} \nabla^2_{tt} l(\bar{x}, \bar{t}, \bar{\alpha}) & \nabla h_a(\bar{t}) \\ \nabla h_a(\bar{t})^\top & 0 \end{pmatrix} \begin{pmatrix} \nabla t(\bar{x})^\top \\ \nabla \alpha_a(\bar{x})^\top \end{pmatrix} = - \begin{pmatrix} \nabla^2_{tx} \lambda(\bar{x}, \bar{t}) \\ 0 \end{pmatrix}, \tag{4.2.7}$$

where

$$\nabla \alpha_a(\bar{x}) := (\nabla \alpha_i(\bar{x}))_{i \in I_a(\bar{t})} \in \mathbb{R}^{n \times |I_a(\bar{t})|},\ \nabla h_a(\bar{t}) := (\nabla h_i(\bar{t}))_{i \in I_a(\bar{t})} \in \mathbb{R}^{\ell \times |I_a(\bar{t})|};$$

in particular, if $I_a(\bar{t}) = \emptyset$, *then* $\nabla t(\bar{x}) \in \mathbb{R}^{n \times \ell}$ *is a unique solution of the linear system*

$$\nabla^2_{tt} l(\bar{x}, \bar{t}, \bar{\alpha}) \nabla t(\bar{x})^\top = -\nabla^2_{tx} \lambda(\bar{x}, \bar{t}), \tag{4.2.8}$$

(d) *for any* $x \in U(\bar{x})$, *letting* $v(x) := \lambda(x, t(x))$, *we have*

$$\begin{aligned} \nabla v(\bar{x}) &= \nabla_x \lambda(\bar{x}, \bar{t}), \\ \nabla^2 v(\bar{x}) &= \nabla^2_{xx} \lambda(\bar{x}, \bar{t}) - \nabla t(\bar{x}) \nabla^2_{tt} l(\bar{x}, \bar{t}, \bar{\alpha}) \nabla t(\bar{x})^\top. \end{aligned}$$

Proof. Apply the implicit function theorem to the following equations:

$$\nabla_t l(x, t, \alpha) = 0,\ h_i(t) = 0 \ \ (i \in I_a(\bar{t})),$$

which come from the Karush-Kuhn-Tucker (KKT) conditions of $P(x)$. See also [29, 24]. □

Next, let

$$T_{\mathrm{loc}}(x) := \{t \in T \mid t \text{ is a local optimum of } P(x)\}$$

and

$$T_\varepsilon(x) := T_{\mathrm{loc}}(x) \cap \{t \in T \mid \lambda(x, t) \leq \min_{t \in T} \lambda(x, t) + \varepsilon\}$$

for a given constant $\varepsilon > 0$. Now, we show that the infinitely many SOC constraints $g(x, t) \in \mathcal{K}^m$ $(t \in T)$ can locally be represented as finitely many SOC constraints under some assumptions including the following *ε-regularity* condition.

Definition 4.2.3 (ε-regularity)**.** *Let $x \in \mathbb{R}^n$ and $\varepsilon > 0$ be given. We say that x is ε-regular if any $t \in T_\varepsilon(x)$ is nondegenerate and $|T_\varepsilon(x)| < \infty$.*

Proposition 4.2.4. *Let $\bar{x} \in \mathbb{R}^n$ and $\varepsilon > 0$ be given. Suppose that $\bar{x}$ is ε-regular and let $T_\varepsilon(\bar{x}) := \{\bar{t}_1, \bar{t}_2, \ldots, \bar{t}_{r_\varepsilon(\bar{x})}\}$. Furthermore, suppose that the function $\lambda(\cdot,\cdot)$ is twice continuously differentiable at $(\bar{x}, \bar{t}_j)$ for all $j = 1, 2, \ldots, r_\varepsilon(\bar{x})$ and that $\bar{x} \in \mathbb{R}^n$ is regular. Then, there exist an open neighborhood $U_\varepsilon(\bar{x}) \subseteq \mathbb{R}^n$ of $\bar{x}$ and functions $t_1(\cdot), t_2(\cdot), \cdots, t_{r_\varepsilon(\bar{x})}(\cdot) : U_\varepsilon(\bar{x}) \to T$ such that, for each $j = 1, 2, \ldots, r_\varepsilon(\bar{x})$,*

(a) $t_j(\cdot)$ *is twice continuously differentiable,*

(b) $t_j(\bar{x}) = \bar{t}_j$, *and*

(c) $\varphi(x) = \max_{j=1,2,\cdots,r_\varepsilon(\bar{x})}(-\lambda(x, t_j(x)))$ *for all $x \in U_\varepsilon(\bar{x})$, where $\varphi(\cdot)$ is defined by* (4.2.2).

Moreover, SISOCP (4.1.1) *can locally be reduced to the following SOCP:*

$$\begin{aligned} &\underset{x \in U_\varepsilon(\bar{x})}{\text{Minimize}} && f(x) \\ &\text{subject to} && g(x, t_j(x)) \in \mathcal{K}^m, \ (j = 1, 2, \ldots, r_\varepsilon(\bar{x})). \end{aligned}$$

Proof. We omit the proof since it can easily be derived from Proposition 4.2.2 and the implicit function theorem. □

4.3 Local reduction based SQP-type algorithm for the SISOCP

From Theorem 3.2.4, the Karush-Kuhn-Tucker (KKT) conditions for SISOCP (4.1.1) are represented as follows: Let x^* be a local optimum of SISOCP (4.1.1). Then, under suitable constraint qualification, there exist q indices $t_1^*, t_2^*, \ldots, t_q^* \in T_\varepsilon(x^*)$ and Lagrange multipliers $\eta_1^*, \eta_2^*, \ldots, \eta_q^* \in \mathbb{R}^m$ such that $q \leq n$ and

$$\nabla f(x^*) - \sum_{j=1}^{q} \nabla_x g(x^*, t_j^*)\eta_j^* = 0, \tag{4.3.1}$$

$$\mathcal{K}^m \ni \eta_j^* \perp g(x^*, t_j^*) \in \mathcal{K}^m \ (j = 1, 2, \ldots, q). \tag{4.3.2}$$

In this section, we propose an algorithm for finding a vector $x^* \in \mathbb{R}^n$ that satisfies the above KKT conditions. In the algorithm, we combine the local reduction method with the sequential quadratic programming (SQP) method. Let $\varepsilon > 0$ be given and let $x^k \in \mathbb{R}^n$ be a current iterate. Assume that x^k satisfies the *ε-regularity* defined in Definition 4.2.3. Then, from Proposition 4.2.4, there exist some open neighborhood $U_\varepsilon(x^k) \subseteq \mathbb{R}^n$ of x^k and twice continuously differentiable functions $t_j^k : U_\varepsilon(x^k) \to T$ ($j = 1, 2, \ldots, r_\varepsilon(x^k)$) such that SISOCP (4.1.1) can locally be reduced to the following SOCP:

$$\text{SOCP}(x^k, \varepsilon): \quad \begin{aligned} &\underset{x \in U_\varepsilon(x^k)}{\text{Minimize}} && f(x) \\ &\text{subject to} && G_j^k(x) := g(x, t_j^k(x)) \in \mathcal{K}^m \ (j = 1, 2, \ldots, r_\varepsilon(x^k)). \end{aligned}$$

We then generate a search direction $d^k \in \mathbb{R}^n$ by solving the following Quadratic SOCP (QSOCP), which consists of quadratic and linear approximations of the objective function and constraint functions of $\mathrm{SOCP}(x^k,\varepsilon)$, respectively:

$$\mathrm{QSOCP}(x^k,\varepsilon):\quad \begin{array}{ll} \underset{d\in\mathbb{R}^n}{\text{Minimize}} & \nabla f(x^k)^\top d + \frac{1}{2}d^\top B_k d \\ \text{subject to} & G_j^k(x^k) + \nabla G_j^k(x^k)^\top d \in \mathcal{K}^m \ (j=1,2,\ldots,r_\varepsilon(x^k)), \end{array}$$

where $B_k \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix. Note that $G_j^k(x^k)$ and $\nabla G_j^k(x^k)$ are given by

$$G_j^k(x^k) = g(x^k, t_j^k(x^k)), \tag{4.3.3}$$

$$\nabla G_j^k(x^k) = \nabla_x g(x^k, t_j^k(x^k)) + \nabla t_j^k(x^k)\nabla_t g(x, t_j^k(x^k)), \tag{4.3.4}$$

where $t_j^k(x^k) \in T_\varepsilon(x^k)$ and $\nabla t_j^k(x^k)$ can be obtained by solving the lower-level problem $P(x^k)$ and by solving (4.2.7) or (4.2.8). Under some constraint qualification, the optimum d^k of $\mathrm{QSOCP}(x^k,\varepsilon)$ satisfies the following KKT conditions:

$$\nabla f(x^k) + B_k d^k - \sum_{j=1}^{r_\varepsilon(x^k)} \nabla G_j^k(x^k)\eta_j^{k+1} = 0, \tag{4.3.5}$$

$$\mathcal{K}^m \ni \eta_j^{k+1} \perp G_j^k(x^k) + \nabla G_j^k(x^k)^\top d^k \in \mathcal{K}^m \ (j=1,2,\ldots,r_\varepsilon(x^k)), \tag{4.3.6}$$

where $\eta_j^{k+1} \in \mathbb{R}^m$ $(j=1,2,\ldots,r_\varepsilon(x^k))$ are Lagrange multiplier vectors corresponding to the SOC constraints $G_j^k(x^k) + \nabla G_j^k(x^k)^\top d \in \mathcal{K}^m$ $(j=1,2,\ldots,r_\varepsilon(x^k))$. If $d^k=0$, then it follows immediately from (4.3.5) and (4.3.6) that the KKT conditions for solving $\mathrm{SOCP}(x^k,\varepsilon)$ are satisfied at x^k. If, in addition, $\tilde{g}(x^k,t)\neq 0$ holds for all $t\in T_\varepsilon(x^k)$, then it holds that

$$\nabla f(x^k) - \sum_{j=1}^{r_\varepsilon(x^k)} \nabla_x g(x^k, \bar{t}_j^k)\eta_j^{k+1} = 0,$$

$$\mathcal{K}^m \ni \eta_j^{k+1} \perp g(x^k,\bar{t}_j^k) \in \mathcal{K}^m \ (j=1,2,\ldots,r_\varepsilon(x^k)),$$

where $\bar{t}_j^k := t_j^k(x^k)$ $(j=1,2,\ldots,r_\varepsilon(x^k))$, as will be shown by Proposition 4.4.2. These are actually regarded as the KKT conditions (4.3.1) and (4.3.2) of SISOCP (4.1.1). In particular, we can see that x^k is feasible for SISOCP (4.1.1), since $g(x^k,\bar{t}_j^k)\in\mathcal{K}^m$ for all $j=1,2,\ldots,r_\varepsilon(x^k)$ and $\bar{t}_j^k$ $(j=1,2,\ldots,r_\varepsilon(x^k))$ contain all minimizers of the lower-level problem $P(x^k)$.

To generate the next iterate x^{k+1} along the direction d^k, we need to choose a step size. To this end, we perform a line search with the following ℓ_∞-type penalty function:

$$\Phi_\rho(x) := f(x) + \rho\varphi_+(x), \tag{4.3.7}$$

where $\varphi(\cdot)$ is defined by (4.2.2) and $\rho>0$ is a penalty parameter. Notice that the function $\Phi_\rho(\cdot)$ is continuous everywhere. Another plausible choice of a merit function used in the line search is an ℓ_1-type penalty function, i.e., the function (4.3.7) with $\varphi_+(x)$ $\left(=\max_{t\in T}(-\lambda(x,t))_+\right)$ replaced by $\sum_{t\in T_\varepsilon(x)}(-\lambda(x,t))_+$. However, in the semi-infinite case, the ℓ_1-type penalty function has such a serious drawback that it may fail to be continuous at a point where the cardinality of $T_\varepsilon(x)$ changes. Properties of penalty functions for semi-infinite programs are explained in detail together with a specific example in [66]. Now, we formally state the SQP-type algorithm for solving SISOCP (4.1.1).

Algorithm 4.1

Step 0 (Initialization): Choose $x^0 \in \mathbb{R}^n$ and a matrix $B_0 \in S^n_{++}$. Select parameters $\alpha \in (0,1)$, $\beta \in (0,1)$, $\delta > 0$, $\varepsilon > 0$ and $\rho_{-1} > 0$. Set $k := 0$.

Step 1 (Generate a search direction): Solve QSOCP(x^k, ε) to obtain $d^k \in \mathbb{R}^n$ and corresponding Lagrange multipliers $\eta_j^{k+1} \in \mathcal{K}^m$ $(j = 1, 2, \ldots, r_\varepsilon(x^k))$.

Step 2 (Check convergence): If $d^k = 0$, then stop. Otherwise, go to Step 3.

Step 3 (Update penalty parameter): If $\rho_{k-1} \geq \sum_{j=1}^{r_\varepsilon(x^k)} (\eta_j^{k+1})_1$, then set $\rho_k := \rho_{k-1}$. Otherwise, set $\rho_k := \sum_{j=1}^{r_\varepsilon(x^k)} (\eta_j^{k+1})_1 + \delta$.

Step 4 (Armijo line search): Find the smallest nonnegative integer $r_k \geq 0$ satisfying

$$\Phi_{\rho_k}(x^k + \alpha^{r_k} d^k) - \Phi_{\rho_k}(x^k) \leq -\alpha^{r_k} \beta (d^k)^\top B_k d^k.$$

Set $s_k := \alpha^{r_k}$ and $x^{k+1} := x^k + s_k d^k$.

Step 5: Update the matrix B_k to obtain $B_{k+1} \in S^n_{++}$. Set $k := k + 1$ and return to Step 1.

To construct QSOCP(x^k, ε) at each iteration k, we need to obtain the set $T_\varepsilon(x^k)$ by computing all local minimizers of the lower-level problem $P(x^k)$. Moreover, we have to compute $\nabla t_j^k(x^k)$ $(j = 1, 2, \ldots, r_\varepsilon(x^k))$ by solving the linear system (4.2.7) or (4.2.8). In Step 4, we must compute $\max_{t \in T}(-\lambda(x^k + \alpha^r d^k, t))_+$ to evaluate $\Phi_{\rho_k}(x^k + \alpha^r d^k)$ for each r. In Step 5, we may choose B_k as

$$B_k := \nabla^2 f(x^k) - \sum_{j=1}^{r_\varepsilon(x^k)} (\zeta_j^k)_1 W_{kj}, \tag{4.3.8}$$

where

$$W_{kj} := \nabla^2 G_{j1}^k(x^k) - \frac{\nabla^2 \tilde{G}_j^k(x^k) \tilde{G}_j^k(x^k)}{\|\tilde{G}_j^k(x^k)\|} \quad (j = 1, 2, \ldots, r_\varepsilon(x^k))$$

and ζ_j^k $(j = 1, 2, \ldots, r_\varepsilon(x^k))$ are some estimates of Lagrange multiplier vectors corresponding to the constraints $G_j^k(\cdot) \in \mathcal{K}^m$ $(j = 1, 2, \ldots, r_\varepsilon(x^k))$. A specific choice of ζ_j^k $(j = 1, 2, \ldots, r_\varepsilon(x^k))$ will be provided later in the section of numerical experiments. The matrix W_{kj} can be calculated as follows: Let $v_j^k : U_\varepsilon(x^k) \to \mathbb{R}$ be defined by $v_j^k(x) := G_{j1}^k(x) - \|\tilde{G}_j^k(x)\|$ for $j = 1, 2, \ldots, r_\varepsilon(x^k)$. Then, we have

$$\nabla^2 v_j^k(x^k) = W_{kj} - \frac{\nabla \tilde{G}_j^k(x^k) \nabla \tilde{G}_j^k(x^k)^\top}{\|\tilde{G}_j^k(x^k)\|} + \frac{\nabla \tilde{G}_j^k(x^k) \tilde{G}_j^k(x^k) \tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top}{\|\tilde{G}_j^k(x^k)\|^3},$$

which implies that, for $j = 1, 2, \ldots, r_\varepsilon(x^k)$,

$$W_{kj} = \nabla^2 v_j^k(x^k) + \frac{\nabla \tilde{G}_j^k(x^k) \nabla \tilde{G}_j^k(x^k)^\top}{\|\tilde{G}_j^k(x^k)\|} - \frac{\nabla \tilde{G}_j^k(x^k) \tilde{G}_j^k(x^k) \tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top}{\|\tilde{G}_j^k(x^k)\|^3}. \tag{4.3.9}$$

Notice that the right-hand side of the above formula can be evaluated since we have $G_j^k(x^k)$, $\nabla G_j^k(x^k)$ and $\nabla^2 v_j^k(x^k)$ by using (4.3.3), (4.3.4) and Proposition 4.2.2(d) with $\bar{x}$ replaced by x^k, respectively. Thus, we can calculate W_{kj} from (4.3.9). In the subsequent section, we will show quadratic convergence of Algorithm 4.1 in which B_k are chosen as (4.3.8).

Another plausible choice of B_k is to let $B_k = \nabla^2_{xx}\mathcal{L}^k_\varepsilon(x^k, \eta^k)$ for each k, where $\mathcal{L}^k_\varepsilon(\cdot,\cdot)$ denotes the Lagrangian of SOCP(x^k, ε). However, to evaluate $\nabla^2_{xx}\mathcal{L}^k_\varepsilon(x^k, \eta^k)$, we have to compute $\nabla^2 t^k_j(x^k)$ $(j = 1, 2, \dots, r_\varepsilon(x^k))$, and it often brings about some numerical difficulties. On the other hand, computing (4.3.8) does not require any calculation of $\nabla^2 t^k_j(x^k)$.

4.4 Convergence analysis

In this section, we study global and local convergence properties of the proposed algorithm.

4.4.1 Global convergence

To begin with, we make the following assumption:

Assumption 4.4.1. *For each* k,

(a) x^k *is regular,*

(b) $\tilde{g}(x^k, t) \neq 0$ *for all* $t \in T_\varepsilon(x^k)$,

(c) QSOCP(x^k, ε) *is feasible, and the KKT conditions* (4.3.5) *and* (4.3.6) *hold at the unique optimum of* QSOCP(x^k, ε).

By Assumption 4.4.1 (a), SISOCP (4.1.1) can locally be reduced to SOCP(x^k, ε) around x^k for each k. By Assumption 4.4.1 (b), we can ensure the continuous differentiability of $\lambda(x^k, \cdot)$ at each $t \in T_\varepsilon(x^k)$, which is required by the ε-regularity of x^k. Although Assumption 4.4.1 (b) may seem restrictive, $\tilde{g}(x^k, t) = 0$ is unlikely to occur in practice at any local minimizer of $P(x^k)$, since $-\|\tilde{g}(x^k, \cdot)\|$ attains its "sharp" maximum at any $t \in T$ such that $\tilde{g}(x^k, t) = 0$. Under Assumption 4.4.1 (c), QSOCP(x^k, ε) has a unique optimum d^k since $B_k \in S^n_{++}$.

By the following proposition, we can ensure that our algorithm finds a KKT point of SISOCP (4.1.1), when the termination criterion $d^k = 0$ is satisfied.

Proposition 4.4.2. *Suppose that Assumption* 4.4.1 *holds. If* $d^k = 0$, *then the KKT conditions* (4.3.1) *and* (4.3.2) *for SISOCP* (4.1.1) *are satisfied at* x^k *with some Lagrange multiplier vectors* $\eta^{k+1}_1, \eta^{k+1}_2, \dots, \eta^{k+1}_{r_\varepsilon(x^k)} \in \mathbb{R}^m$. *In particular,* x^k *is feasible for SISOCP* (4.1.1).

Proof. From the ε-regularity of x^k, $\bar{t}^k_j := t^k_j(x^k) \in T_\varepsilon(x^k)$ $(j = 1, 2, \dots, r_\varepsilon(x^k))$ are nondegenerate local optima of $P(x^k)$ and then satisfy the KKT conditions of the lower-level problem $P(x^k)$. Thus, we have, for each $j = 1, 2, \dots, r_\varepsilon(x^k)$,

$$\nabla_t g_1(x^k, \bar{t}^k_j) - \frac{\nabla_t \tilde{g}(x^k, \bar{t}^k_j)\tilde{g}(x^k, \bar{t}^k_j)}{\|\tilde{g}(x^k, \bar{t}^k_j)\|} - \sum_{i \in I_a(\bar{t}^k_j)} \alpha^j_i \nabla h_i(\bar{t}^k_j) = 0, \tag{4.4.1}$$

where $I_a(\bar{t}^k_j)$ is defined by (4.2.5) and α^j_i $(i \in I_a(\bar{t}^k_j))$ are Lagrange multipliers. Using the fact that $\nabla t^k_j(x^k)\nabla h_i(\bar{t}^k_j) = 0$ holds for each $i \in I_a(\bar{t}^k_j)$ by Proposition 4.2.2 (c), (4.4.1) yields

$$\nabla t^k_j(x^k)\nabla_t g_1(x^k, \bar{t}^k_j) - \frac{\nabla t^k_j(x^k)\nabla_t \tilde{g}(x^k, \bar{t}^k_j)\tilde{g}(x^k, \bar{t}^k_j)}{\|\tilde{g}(x^k, \bar{t}^k_j)\|} = 0, \quad j = 1, 2, \dots, r_\varepsilon(x^k). \tag{4.4.2}$$

From the KKT conditions (4.3.6) of QSOCP(x^k, ε) with $d^k = 0$ and (4.3.4), we obtain

$$\nabla f(x^k) - \sum_{j=1}^{r_\varepsilon(x^k)} \left(\nabla_x g(x^k, \bar{t}_j^k) + \nabla t_j^k(x^k)\nabla_t g(x^k, \bar{t}_j^k)\right)\eta_j^{k+1} = 0, \tag{4.4.3}$$

$$\mathcal{K}^m \ni \eta_j^{k+1} \perp g(x^k, \bar{t}_j^k) \in \mathcal{K}^m \ (j = 1, 2, \ldots, r_\varepsilon(x^k)). \tag{4.4.4}$$

Notice that (4.4.4) implies $\eta_j^{k+1} = (\eta_j^{k+1})_1(1, -\tilde{g}(x^k, \bar{t}_j^k)^\top/\|\tilde{g}(x^k, \bar{t}_j^k)\|)^\top$ since $\|\tilde{g}(x^k, \bar{t}_j^k)\| \neq 0$ by Assumption 4.4.1 (b), which together with (4.4.2) yields

$$\nabla t_j^k(x^k)\nabla_t g(x^k, \bar{t}_j^k)\eta_j^{k+1} = 0, \quad j = 1, 2, \ldots, r_\varepsilon(x^k).$$

Hence, from (4.4.3), we have $\nabla f(x^k) - \sum_{j=1}^{r_\varepsilon(x^k)} \nabla_x g(x^k, \bar{t}_j^k)\eta_j^{k+1} = 0$. Combining this and (4.4.4), we obtain the desired result.

The feasibility of x^k readily follows, since we have $g(x^k, \bar{t}_j^k) \in \mathcal{K}^m$ $(j = 1, 2, \ldots, r_\varepsilon(x^k))$ from (4.4.4) and $\bar{t}_j^k$ $(j = 1, 2, \ldots, r_\varepsilon(x^k))$ contain global optima of $P(x^k)$. □

We next show that the search direction $d^k \in \mathbb{R}^n$ obtained from QSOCP(x^k, ε) is a descent direction for $\Phi_\rho(\cdot)$ at x^k as long as the penalty parameter ρ is sufficiently large, which ensures that the line search in Step 4 terminates finitely at each iteration. To this end, we begin with proving the following lemma.

Lemma 4.4.3. *Suppose that Assumption* 4.4.1 *holds. Then, we have*

$$\varphi_+(x^k) + \varphi'_+(x^k; d^k) \le 0, \tag{4.4.5}$$

where $\varphi(\cdot)$ *is defined by* (4.2.2).

Proof. By the ε-regularity of x^k and Proposition 4.2.4, there exist an open neighborhood $U_\varepsilon(x^k)$ of x^k and C^2 functions $t_j^k(\cdot) : U_\varepsilon(x^k) \to T$ $(j = 1, 2, \ldots, r_\varepsilon(x^k))$ such that

$$\varphi(x) = \max_{t \in T}(-\lambda(x, t)) = \max_{j=1,2,\cdots,r_\varepsilon(x^k)}(-\lambda(x, t_j^k(x)) = \max_{j=1,2,\cdots,r_\varepsilon(x^k)}\left(\|\tilde{G}_j^k(x)\| - G_{j1}^k(x)\right)$$

for all $x \in U_\varepsilon(x^k)$, where $G_j^k(x) = (G_{j1}^k(x), \tilde{G}_j^k(x)) := (g_1(x, t_j^k(x)), \tilde{g}(x, t_j^k(x)))$ for $j = 1, 2, \ldots, r_\varepsilon(x^k)$. Then, by letting $J(x^k) := \left\{j \in \{1, 2, \ldots, r_\varepsilon(x^k)\} \mid -\lambda(x^k, t_j^k(x^k)) = \varphi(x^k)\right\}$, we have

$$\varphi(x^k) = -\lambda(x^k, t_j^k(x^k)) = \|\tilde{G}_j^k(x^k)\| - G_{j1}^k(x^k) \quad (j \in J(x^k)). \tag{4.4.6}$$

In addition, since $\tilde{G}_j^k(x^k) \neq 0$ from Assumption 4.4.1 (b), it is not difficult to show

$$\varphi'_+(x^k; d^k) = \begin{cases} 0 & \text{if } \varphi(x^k) < 0, \\ \max_{j \in J(x^k)}\left(\dfrac{\tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top d^k}{\|\tilde{G}_j^k(x)\|} - \nabla G_{j1}^k(x^k)^\top d^k\right)_+ & \text{if } \varphi(x^k) = 0, \\ \max_{j \in J(x^k)}\left(\dfrac{\tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top d^k}{\|\tilde{G}_j^k(x)\|} - \nabla G_{j1}^k(x^k)^\top d^k\right) & \text{if } \varphi(x^k) > 0. \end{cases} \tag{4.4.7}$$

Moreover, since d^k is feasible for QSOCP(x^k, ε), we have $G_j^k(x^k) + \nabla G_j^k(x^k)^\top d^k \in \mathcal{K}^m$ $(j = 1, 2, \ldots, r_\varepsilon(x^k))$, which implies that

$$G_{j1}^k(x^k) + \nabla G_{j1}^k(x^k)^\top d^k - \left\| \tilde{G}_j^k(x^k) + \nabla \tilde{G}_j^k(x^k)^\top d^k \right\| \geq 0 \qquad (j = 1, 2, \ldots, r_\varepsilon(x^k)). \tag{4.4.8}$$

Notice that, for any $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^n$,

$$\|u\| + \frac{u^\top v}{\|u\|} \leq \|u + v\| \tag{4.4.9}$$

holds since

$$\|u + v\|^2 - \left(\|u\| + \frac{u^\top v}{\|u\|} \right)^2 = \|v\|^2 - \frac{(u^\top v)^2}{\|u\|^2} \geq \|v\|^2 - \frac{\|u\|^2 \|v\|^2}{\|u\|^2} = 0.$$

Hence, by setting $u := \tilde{G}_j^k(x), v := \nabla \tilde{G}_j^k(x^k)^\top d^k$ in (4.4.9), we have

$$\frac{\tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top d^k}{\|\tilde{G}_j^k(x)\|} + \|\tilde{G}_j^k(x^k)\| \leq \left\| \tilde{G}_j^k(x^k) + \nabla \tilde{G}_j^k(x^k)^\top d^k \right\| \quad (j \in J(x^k)). \tag{4.4.10}$$

To show (4.4.5), we consider three cases (i) $\varphi(x^k) < 0$, (ii) $\varphi(x^k) = 0$, (iii) $\varphi(x^k) > 0$. In case (i), (4.4.7) implies $\varphi_+(x^k) + \varphi'_+(x^k; d^k) = \varphi_+(x^k) = 0$. In case (ii), since $\|\tilde{G}_j^k(x^k)\| - G_{j1}^k(x^k) = -\varphi(x^k) = 0$, it holds that, for $j \in J(x^k)$,

$$\begin{aligned}
&\frac{\tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top d^k}{\|\tilde{G}_j^k(x)\|} - \nabla G_{j1}^k(x^k)^\top d^k \\
&= \|\tilde{G}_j^k(x^k)\| - G_{j1}^k(x^k) + \frac{\tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top d^k}{\|\tilde{G}_j^k(x)\|} - \nabla G_{j1}^k(x^k)^\top d^k \\
&\leq \left\| \tilde{G}_j^k(x^k) + \nabla \tilde{G}_j^k(x^k)^\top d^k \right\| - G_{j1}^k(x^k) - \nabla G_{j1}^k(x^k)^\top d^k \\
&\leq 0,
\end{aligned} \tag{4.4.11}$$

where the first inequality follows from (4.4.10) and the second inequality does from (4.4.8). Then, we obtain from (4.4.7) and (4.4.11)

$$\varphi_+(x^k) + \varphi'_+(x^k; d^k) = \max_{j \in J(x^k)} \left(\frac{\tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top d^k}{\|\tilde{G}_j^k(x)\|} - \nabla G_{j1}^k(x^k)^\top d^k \right)_+ = 0.$$

In case (iii), since $\varphi_+(x^k) = \varphi(x^k)$, we have

$$\begin{aligned}
\varphi_+(x^k) + \varphi'_+(x^k; d^k) &= \varphi(x^k) + \max_{j \in J(x^k)} \left(\frac{\tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top d^k}{\|\tilde{G}_j^k(x)\|} - \nabla G_{j1}^k(x^k)^\top d^k \right) \\
&= \max_{j \in J(x^k)} \left(\frac{\tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top d^k}{\|\tilde{G}_j^k(x)\|} + \varphi(x^k) - \nabla G_{j1}^k(x^k)^\top d^k \right) \\
&= \max_{j \in J(x^k)} \left(\frac{\tilde{G}_j^k(x^k)^\top \nabla \tilde{G}_j^k(x^k)^\top d^k}{\|\tilde{G}_j^k(x)\|} + \|\tilde{G}_j^k(x^k)\| - G_{j1}^k(x^k) - \nabla G_{j1}^k(x^k)^\top d^k \right) \\
&\leq \max_{j \in J(x^k)} \left(\left\| \tilde{G}_j^k(x^k) + \nabla \tilde{G}_j^k(x^k)^\top d^k \right\| - G_{j1}^k(x^k) - \nabla G_{j1}^k(x^k)^\top d^k \right) \\
&\leq 0,
\end{aligned}$$

where the first and third equalities are obtained from (4.4.7) and (4.4.6), respectively, the first inequality follows from (4.4.10), and the last inequality is derived from (4.4.8). Consequently, we have the desired result. □

Proposition 4.4.4. *Suppose that Assumption* 4.4.1 *holds. If* $\rho \geq \sum_{j=1}^{r_\varepsilon(x^k)} (\eta_j^{k+1})_1$, *then we have*

$$\Phi_\rho'(x^k; d^k) \leq -(d^k)^\top B_k d^k. \tag{4.4.12}$$

Furthermore, $\Phi_\rho'(x^k; d^k) < 0$ *holds when* $d^k \neq 0$.

Proof. First note that, for each $j = 1, 2, \ldots, r_\varepsilon(x^k)$, we have

$$\begin{aligned} G_j^k(x^k)^\top \eta_j^{k+1} &= (\eta_j^{k+1})_1 (G_j^k(x^k))_1 + (\tilde{\eta}_j^{k+1})^\top \tilde{G}_j^k(x^k) \\ &\geq (\eta_j^{k+1})_1 (G_j^k(x^k))_1 - \|\tilde{\eta}_j^{k+1}\| \|\tilde{G}_j^k(x^k)\| \\ &\geq (\eta_j^{k+1})_1 (G_j^k(x^k))_1 - (\eta_j^{k+1})_1 \|\tilde{G}_j^k(x^k)\| \\ &= (\eta_j^{k+1})_1 \lambda(x^k, t_j^k(x^k)) \\ &\geq (\eta_j^{k+1})_1 \min_{t \in T} \lambda(x^k, t), \\ &\geq -(\eta_j^{k+1})_1 \varphi_+(x^k), \end{aligned} \tag{4.4.13}$$

where the second and third inequalities hold since $(\eta_j^{k+1})_1 \geq \|\tilde{\eta}_j^k\|$ by $\eta_j^{k+1} \in \mathcal{K}^m$, and the last inequality follows from $\varphi_+(x^k) \geq \varphi(x^k) = -\min_{t \in T} \lambda(x^k, t)$. Then, by noting the KKT conditions (4.3.5) and (4.3.6) of QSOCP(x^k, ε), we obtain

$$\begin{aligned} \nabla f(x^k)^\top d^k &= -(d^k)^\top B_k d^k + \sum_{j=1}^{r_\varepsilon(x^k)} \left(\nabla G_j^k(x^k) \eta_j^{k+1} \right)^\top d^k \\ &= -(d^k)^\top B_k d^k - \sum_{j=1}^{r_\varepsilon(x^k)} G_j^k(x^k)^\top \eta_j^{k+1}, \\ &\leq -(d^k)^\top B_k d^k + \sum_{j=1}^{r_\varepsilon(x^k)} (\eta_j^{k+1})_1 \varphi_+(x^k) \\ &\leq -(d^k)^\top B_k d^k + \rho \varphi_+(x^k), \end{aligned} \tag{4.4.14}$$

where the second equality holds since $(\eta_j^{k+1})^\top (G_j^k(x^k) + \nabla G_j^k(x^k)^\top d^k) = 0$ for each $j = 1, 2, \ldots, r_\varepsilon(x^k)$ by the SOC complementarity conditions in (4.3.6), the first inequality is derived from (4.4.13), and the last inequality is implied by $\rho \geq \sum_{j=1}^{r_\varepsilon(x^k)} (\eta_j^{k+1})_1 \geq 0$ and $\varphi_+(x^k) \geq 0$. By using these facts, we have

$$\begin{aligned} \Phi_\rho'(x^k; d^k) &= \nabla f(x^k)^\top d^k + \rho \varphi_+'(x^k; d^k) \\ &\leq -(d^k)^\top B_k d^k + \rho(\varphi_+(x^k) + \varphi_+'(x^k; d^k)) \\ &\leq -(d^k)^\top B_k d^k, \end{aligned}$$

where the first inequality follows from (4.4.14) and the last inequality does from Lemma 4.4.3. Therefore, (4.4.12) holds.

The latter claim is obvious from (4.4.12), $d^k \neq 0$ and $B_k \in S_{++}^n$. □

Assumption 4.4.5. (a) *There exist* $0 < \gamma_1 \le \gamma_2$ *such that* $\gamma_1\|d\|^2 \le d^\top B_k d \le \gamma_2\|d\|^2$ *for all* $d \in \mathbb{R}^n$ *and* $k = 0, 1, 2, \ldots,$

(b) $\{x^k\}$ *is bounded, and*

(c) $\{d^k\}$ *is bounded.*

For an arbitrary accumulation point $x^* \in \mathbb{R}^n$ *of* $\{x^k\}$*, it holds that*

(d) x^* *is regular, and*

(e) $\tilde{g}(x^*, \bar{t}) \neq 0$ *for any* $\bar{t} \in T_\varepsilon(x^*)$.

Furthermore, let $U_\varepsilon(x^*) \subseteq \mathbb{R}^n$ *and* $t_j(\cdot) : U_\varepsilon(x^*) \to T$ $(j = 1, 2, \ldots, r_\varepsilon(x^*))$ *be an open neighborhood of* x^* *and functions, respectively, such that the conditions* (a)-(c) *in Proposition* 4.2.4 *hold with* $\bar{x}$ *replaced by* x^**. Then,*

(f) *there exists an open neighborhood* $V_\varepsilon(x^*)(\subseteq U_\varepsilon(x^*))$ *of* x^* *such that* $\{t_j(x)\}_{j=1}^{r_\varepsilon(x^*)} = T_\varepsilon(x)$ *holds for any* $x \in V_\varepsilon(x^*)$*, and*

(g) *Slater's constraint qualification holds for* QSOCP(x^*, ε)*, i.e., there exists* $d_0 \in \mathbb{R}^n$ *such that* $G_j(x^*) + \nabla G_j(x^*)^\top d_0 \in \operatorname{int} \mathcal{K}^m$ *for all* $j = 1, 2, \ldots, r_\varepsilon(x^*)$*, where* $G_j(x) := g(x, t_j(x))$ $(j = 1, 2, \ldots, r_\varepsilon(x^*))$.

Assumption 4.4.5 (f) implies that, when $x^k \in V_\varepsilon(x^*)$, we have $G_j^k(x) \equiv G_j(x) := g(x, t_j(x))$ $(j = 1, 2, \ldots, r_\varepsilon(x^*))$, and hence SISOCP (4.1.1) can locally be reduced to the following SOCP around x^k:

$$\min_{x \in V_\varepsilon(x^*)} \ f(x) \quad \text{s.t.} \ G_j(x) \in \mathcal{K}^m, \ j = 1, 2, \ldots, r_\varepsilon(x^*). \tag{4.4.15}$$

In other words, the constraint functions of SOCP(x^k, ε) coincide with those of SOCP(x^*, ε), whenever $x^k \in V_\varepsilon(x^*)$.

Under the above assumptions, we have the following proposition:

Proposition 4.4.6. *Suppose that Assumptions* 4.4.1 *and* 4.4.5 *hold. Let* $\eta_1^{k+1}, \eta_2^{k+1}, \ldots, \eta_{r_\varepsilon(x^k)}^{k+1} \in \mathcal{K}^m$ *be Lagrange multiplier vectors satisfying the KKT conditions* (4.3.5) *and* (4.3.6)*, and denote* $\eta^k := (\eta_1^k, \eta_2^k, \ldots, \eta_{r_\varepsilon(x^k)}^k)$*. Then, it holds that*

(a) $\{\eta^k\}$ *is bounded, and*

(b) *there exist some* $k_0 \ge 0$ *and* $\bar{\rho} > 0$ *such that* $\rho_k = \bar{\rho}$ *for all* $k \ge k_0$.

Proof. We first show (a). For contradiction, suppose that $\{\eta^k\}$ is not bounded. Then, there exists some subsequence $\{\eta^{k+1}\}_{k \in K}$ such that $\lim_{k \in K, k \to \infty} \|\eta^{k+1}\| = \infty$. We may assume, without loss of generality, that $\eta^{k+1} \neq 0$ for all $k \in K$. By Assumption 4.4.5(b), $\{x^k\}_{k \in K}$ is bounded and has at least one accumulation point, say, $x^* \in \mathbb{R}^n$. Again, without loss of generality, we can assume that $\lim_{k \in K, k \to \infty} x^k = x^*$. From Assumption 4.4.5(d), x^* is regular. Then, by Proposition 4.2.4, there exist some open neighborhood $U_\varepsilon(x^*) \subseteq \mathbb{R}^n$ of x^* and C^2 functions

$t_j(\cdot) : U_\varepsilon(x^*) \to T$ $(j = 1, 2, \ldots, r_\varepsilon(x^*))$ such that SISOCP (4.1.1) can locally be reduced to SOCP(x^*, ε) around x^*:

$$\min_{x \in U_\varepsilon(x^*)} \; f(x) \quad \text{s.t. } G_j(x) \in \mathcal{K}^m \; (j = 1, 2, \ldots, r_\varepsilon(x^*))$$

where $G_j(x) := g(x, t_j(x))$ $(j = 1, 2, \ldots, r_\varepsilon(x^*))$. From Assumption 4.4.5 (f), the constraint functions of SOCP(x^k, ε) for $k(\in K) \geq \bar{k}$ are identical to those of SOCP(x^*, ε) for some $\bar{k} \in K$ large enough. Therefore, QSOCP(x^k, ε) can be represented as

$$\min_{d \in \mathbb{R}^n} \frac{1}{2} d^\top B_k d + \nabla f(x^k)^\top d \quad \text{s.t. } G_j(x^k) + \nabla G_j(x^k)^\top d \in \mathcal{K}^m, \; j = 1, 2, \ldots, r_\varepsilon(x^*),$$

and its optimum d^k satisfies the following KKT conditions:

$$\nabla f(x^k) + B_k d^k - \sum_{j=1}^{r_\varepsilon(x^*)} \nabla G_j(x^k) \eta_j^{k+1} = 0,$$
$$\mathcal{K}^m \ni \eta_j^{k+1} \perp G_j(x^k) + \nabla G_j(x^k)^\top d^k \in \mathcal{K}^m \; (j = 1, 2, \ldots, r_\varepsilon(x^*)),$$

from which it follows that

$$\frac{1}{\|\eta^{k+1}\|} \nabla f(x^k) + \frac{B_k d^k}{\|\eta^{k+1}\|} - \sum_{j=1}^{r_\varepsilon(x^*)} \frac{\nabla G_j(x^k) \eta_j^{k+1}}{\|\eta^{k+1}\|} = 0, \tag{4.4.16}$$

$$\mathcal{K}^m \ni \frac{\eta_j^{k+1}}{\|\eta^{k+1}\|} \perp G_j(x^k) + \nabla G_j(x^k)^\top d^k \in \mathcal{K}^m \; (j = 1, 2, \ldots, r_\varepsilon(x^*)) \tag{4.4.17}$$

for all $k(\in K) \geq \bar{k}$.

Note that $\{d^k\}_{k \in K} \subseteq \mathbb{R}^n$ is bounded from Assumption 4.4.5(c) and $\{\eta^{k+1}/\|\eta^{k+1}\|\}_{k(\in K) \geq \bar{k}} \subseteq \mathbb{R}^{m r_\varepsilon(x^*)}$ is also bounded. Let $(d^*, \eta^*) := (d^*, \eta_1^*, \eta_2^*, \ldots, \eta_{r_\varepsilon(x^*)}^*) \in \mathbb{R}^n \times \mathcal{K}^m \times \mathcal{K}^m \times \cdots \times \mathcal{K}^m$ be an arbitrary accumulation point of $\{(d^k, \eta^{k+1}/\|\eta^{k+1}\|)\}_{k(\in K) \geq \bar{k}}$. Without loss of generality, we can assume that

$$\lim_{k \in K, k \to \infty} \left(\frac{\eta^{k+1}}{\|\eta^{k+1}\|}, d^k, x^k \right) = (\eta^*, d^*, x^*). \tag{4.4.18}$$

Then, letting $k \in K, k \to \infty$ in (4.4.16) and (4.4.17) yields

$$\sum_{j=1}^{r_\varepsilon(x^*)} \nabla G_j(x^*) \eta_j^* = 0, \tag{4.4.19}$$

$$\mathcal{K}^m \ni \eta_j^* \perp G_j(x^*) + \nabla G_j(x^*)^\top d^* \in \mathcal{K}^m \; (j = 1, 2, \ldots, r_\varepsilon(x^*)), \tag{4.4.20}$$

since $\{B_k\}$ is bounded from Assumption 4.4.5(a). Furthermore, from Assumption 4.4.5(g), QSOCP(x^*, ε) satisfies Slater's constraint qualification, i.e., there exists some $d_0 \in \mathbb{R}^n$ such that

$$G_j(x^*) + \nabla G_j(x^*)^\top d_0 \in \operatorname{int} \mathcal{K}^m \; (j = 1, 2, \ldots, r_\varepsilon(x^*)). \tag{4.4.21}$$

Now observe that

$$\sum_{j=1}^{r_\varepsilon(x^*)} (\eta_j^*)^\top \left(G_j(x^*) + \nabla G_j(x^*)^\top d_0 \right) = \sum_{j=1}^{r_\varepsilon(x^*)} \left(\nabla G_j(x^*) \eta_j^* \right)^\top (d_0 - d^*) = 0, \tag{4.4.22}$$

where the first equality holds since $(\eta_j^*)^\top \left(G_j(x^*) + \nabla G_j(x^*)^\top d^*\right) = 0$ $(j = 1, 2, \ldots, r_\varepsilon(x^*))$ by (4.4.20), and the second equality follows from (4.4.19). Combining (4.4.22) with (4.4.21) and $\eta_j^* \in \mathcal{K}^m$ $(j = 1, 2, \ldots, r_\varepsilon(x^*))$, we obtain $\eta_j^* = 0$ for $j = 1, 2, \ldots, r_\varepsilon(x^*)$. This is a contradiction since $\|\eta^*\| = 1$ from (4.4.18). Therefore, $\{\eta^k\}$ is bounded.

We next show (b). For contradiction, we suppose that such $\bar{\rho} > 0$ and $k_0 \geq 0$ do not exist. Then, by the update rule in Step 3 of Algorithm 4.1, there exists an infinite subsequence $\{\rho_k\}_{k \in K'}$ of penalty parameters such that, for all $k \in K'$,

$$\rho_{k-1} < \sum_{j=1}^{r_\varepsilon(x^*)} (\eta_j^{k+1})_1 \quad \text{and} \quad \rho_k = \sum_{j=1}^{r_\varepsilon(x^*)} (\eta_j^{k+1})_1 + \delta,$$

from which we have $\rho_k \geq \rho_{k-1} + \delta$ for all $k \in K'$. This implies $\lim_{k\to\infty} \rho_k = \infty$, since $\{\rho_k\}$ is nondecreasing by the update rule. We thus obtain $\lim_{k \in K', k\to\infty} \|\eta^{k+1}\| = \infty$ since $\sum_{j=1}^{r_\varepsilon(x^k)} (\eta_j^k)_1 > \rho_{k-1} \to \infty$ as $k (\in K') \to \infty$. This contradicts the boundedness of $\{\eta^k\}$. □

Now, we establish the global convergence of Algorithm 4.1.

Theorem 4.4.7. *Suppose that Assumptions* 4.4.1 *and* 4.4.5 *hold. Let* $x^* \in \mathbb{R}^n$ *be an arbitrary accumulation point of* $\{x^k\} \subseteq \mathbb{R}^n$. *Then, the KKT conditions* (4.3.1) *and* (4.3.2) *of SISOCP* (4.1.1) *hold at* x^*.

Proof. First, from Proposition 4.4.6, there exists some $\bar{\rho} > 0$ such that $\rho_k = \bar{\rho}$ for all k sufficiently large. For simplicity of expression, we assume that $\rho_k = \bar{\rho}$ for all k.

Choose a subsequence $\{x^k\}_{k \in K}$ such that $\lim_{k \in K, k\to\infty} x^k = x^*$. Since $\{d^k\}_{k \in K}$ is bounded from Assumption 4.4.5 (c), it has at least one accumulation point, say, $d^* \in \mathbb{R}^n$. To prove the desired result, from Proposition 4.4.2, it suffices to show $d^* = 0$. Due to Assumption 4.4.5 (d), (e) and (f), SISOCP (4.1.1) can locally be reduced to SOCP (4.4.15) around x^k for all $k \in K$ sufficiently large. Then, using the facts that $\{B_k\} \subseteq S^n_{++}$ is uniformly bounded by Assumption 4.4.5 (a), d^k is a descent direction of $\Phi_{\bar{\rho}}(\cdot)$ at x^k by Proposition 4.4.4 and $\Phi_{\bar{\rho}}(\cdot)$ is continuous everywhere, we can deduce that $d^* = 0$ in a way similar to the convergence analysis for the SQP-type method for solving the nonlinear SOCP [33]. □

4.4.2 Local convergence

Now, we analyze the convergence rate of Algorithm 4.1. In the remainder of this section, we assume that a sequence $\{(x^k, \eta^k)\}$ generated by Algorithm 4.1 converges to $(x^*, \eta^*) \in \mathbb{R}^n \times \mathbb{R}^{m r_\varepsilon(x^*)}$. Moreover, we let $x^* \in \mathbb{R}^n$ be a ε-regular point such that SISOCP (4.1.1) is locally reduced to the following SOCP(x^*, ε) around x^*:

$$\min_{x \in U_\varepsilon(x^*)} f(x) \text{ s.t. } G_j(x) \in \mathcal{K}^m \ (j = 1, 2, \ldots, r_\varepsilon(x^*)),$$

where, for $j = 1, 2, \ldots, r_\varepsilon(x^*)$, $G_j(x) := g(x, t_j(x))$ with C^2 functions $t_j(\cdot) : U_\varepsilon(x^*) \to T$ and an open neighborhood $U_\varepsilon(x^*) \subseteq \mathbb{R}^n$ of x^* satisfying conditions (a)-(c) in Proposition 4.2.4. We suppose that $\tilde{G}_j(x^*) \neq 0$ $(j = 1, 2, \ldots, r_\varepsilon(x^*))$ and (x^*, η^*) satisfies the KKT conditions for

SOCP(x^*, ε):

$$\nabla f(x^*) - \sum_{j=1}^{r_\varepsilon(x^*)} \nabla G_j(x^*)\eta_j^* = 0,$$
$$\mathcal{K}^m \ni G_j(x^*) \perp \eta_j^* \in \mathcal{K}^m \ (j = 1, 2, \ldots, r_\varepsilon(x^*)),$$

where $\eta^* := (\eta_1^*, \eta_2^*, \ldots, \eta_{r_\varepsilon(x^*)}^*) \in \mathbb{R}^m \times \mathbb{R}^m \times \cdots \times \mathbb{R}^m$. Furthermore, we define the Lagrangian of SOCP(x^*, ε) by

$$\mathcal{L}_\varepsilon(x, \eta) := f(x) - \sum_{j=1}^{r_\varepsilon(x^*)} G_j(x)^\top \eta_j,$$

where $\eta := (\eta_1, \eta_2, \ldots, \eta_{r_\varepsilon(x^*)}) \in \mathbb{R}^m \times \mathbb{R}^m \times \cdots \times \mathbb{R}^m$.

Before discussing the convergence rate of the algorithm, we recall the constraint nondegeneracy and second order sufficient condition for SOCP(x^*, ε). We say that $x^* \in \mathbb{R}^n$ is constraint nondegenerate [5, Definition 16] if

$$\nabla G_j(x^*)^\top \mathbb{R}^n + \operatorname{lin} T_{\mathcal{K}^m}(G_j(x^*)) = \mathbb{R}^m$$

holds for each $j = 1, 2, \ldots, r_\varepsilon(x^*)$, where $T_{\mathcal{K}^m}(z)$ denotes the tangent cone of $\mathcal{K}^m$ at $z \in \mathcal{K}^m$ and $\operatorname{lin} T_{\mathcal{K}^m}(z)$ stands for the largest linear subspace contained by $T_{\mathcal{K}^m}(z)$. The second order sufficient condition (SOSC) for general SOCP is studied in [5, 33, 71]. Under the assumption $\tilde{G}_j(x^*) \neq 0$ $(j = 1, 2, \ldots, r_\varepsilon(x^*))$, the SOSC can be simplified as follows: For all $d \in C_{\mathcal{K}^m}(x^*, \eta^*) \setminus \{0\}$,

$$d^\top \nabla^2_{xx} \mathcal{L}_\varepsilon(x^*, \eta^*) d + d^\top \left(\sum_{j=1}^{r_\varepsilon(x^*)} \mathcal{H}^j_\varepsilon(x^*, \eta^*) \right) d > 0,$$

where

$$\mathcal{H}^j_\varepsilon(x^*, \eta^*) := \begin{cases} -\dfrac{(\eta_j^*)_1}{G_{j1}(x^*)} \nabla G_j(x^*) \begin{pmatrix} 1 & 0 \\ 0 & -I_{m-1} \end{pmatrix} \nabla G_j(x^*)^\top & \text{if } G_j(x^*) \in \operatorname{bd} \mathcal{K}^m \setminus \{0\}, \\ 0 & \text{otherwise}, \end{cases}$$
$$C_{\mathcal{K}^m}(x^*, \eta^*) := \left\{ d \in \mathbb{R}^n \,\middle|\, d^\top \nabla G_j(x^*) \eta_j^* = 0 \text{ for all } j \text{ such that } G_j(x^*) \in \operatorname{bd} \mathcal{K}^m \setminus \{0\} \right\}.$$

Under the above conditions, we can show that the sequence $\{(x^k, \eta^k)\}$ converges to (x^*, η^*) quadratically, by using an argument in [71, Theorem 4.2].

Proposition 4.4.8. *Let $B : \mathbb{R}^n \times \mathbb{R}^{r_\varepsilon(x^*)} \to \mathbb{R}^{n \times n}$ be a function such that $B(x^*, \eta^*) = \nabla_{xx} \mathcal{L}_\varepsilon(x^*, \eta^*)$ and $B(\cdot, \cdot)$ is continuously differentiable at (x^*, η^*). Suppose that Assumption* 4.4.1 *and Assumption* 4.4.5 (d)–(g) *hold. Moreover, let the constraint nondegeneracy condition and SOSC hold at (x^*, η^*). If (x^{k_0}, η^{k_0}) is sufficiently close to (x^*, η^*) for some $k_0 \geq 0$, and if $s_k = 1$, $B_k = B(x^k, \eta^k)$ and $B_k \in S^n_{++}$ for all $k \geq k_0$, then $\{(x^k, \eta^k)\}$ converges to (x^*, η^*) quadratically.*

Proof. From Assumption 4.4.5 (f)–(g), if x^k is sufficiently close to x^*, then we can locally reduce SISOCP (4.1.1) to SOCP(x^*, ε) around x^k. Then, by [71, Theorem 4.2], we obtain the desired result. □

Using the above theorem, we can establish quadratic convergence of Algorithm 4.1 in which B_k are chosen as (4.3.8).

Theorem 4.4.9. *Suppose that the assumptions in Proposition* 4.4.8 *hold. If* (x^{k_0}, η^{k_0}) *is sufficiently close to* (x^*, η^*) *for some* $k_0 \geq 0$*, and if* $s_k = 1$*,* B_k *is chosen as* (4.3.8) *with* $\zeta_j^k := \eta_j^k$ $(j = 1, 2, \ldots, r_\varepsilon(x^*))$ *and* $B_k \in S_{++}^n$ *for all* $k \geq k_0$*, then* $\{(x^k, \eta^k)\}$ *converges to* (x^*, η^*) *quadratically.*

Proof. From Assumption 4.4.5 (f)–(g), if x^k is sufficiently close to x^*, then we can locally reduce SISOCP (4.1.1) to SOCP(x^*, ε) around x^k. Then, by letting

$$B(x, \eta) := \nabla^2 f(x) - \sum_{j=1}^{r_\varepsilon(x^*)} (\eta_j)_1 \left(\nabla^2 G_{j1}(x) - \frac{\nabla^2 \tilde{G}_j(x) \tilde{G}_j(x)}{\|\tilde{G}_j(x)\|} \right),$$

we have $B(x^k, \eta^k) = B_k$ for all k sufficiently large. Hence, from Proposition 4.4.8, we have only to show that $B(\cdot, \cdot)$ is continuously differentiable at (x^*, η^*) and $B(x^*, \eta^*) = \nabla_{xx}^2 \mathcal{L}_\varepsilon(x^*, \eta^*)$. The first claim is obvious since $\tilde{G}_j(x^*) \neq 0$ $(j = 1, 2, \ldots, r_\varepsilon(x^*))$. We prove the second claim. Notice that $\eta_j^* = (\eta_j^*)_1 (1, -\tilde{G}_j(x^*)^\top / \|\tilde{G}_j(x^*)\|)^\top$ for $j = 1, 2, \ldots, r_\varepsilon(x^*)$ since $\tilde{G}_j(x^*) \neq 0$ and $\mathcal{K}^m \ni \eta_j^* \perp G_j(x^*) \in \mathcal{K}^m$ for $j = 1, 2, \ldots, r_\varepsilon(x^*)$. Thus, we have

$$\begin{aligned} \nabla_{xx}^2 \mathcal{L}_\varepsilon(x^*, \eta^*) &= \nabla^2 f(x^*) - \sum_{j=1}^{r_\varepsilon(x^*)} \nabla^2 G_j(x^*) \eta_j^* \\ &= \nabla^2 f(x^*) - \sum_{j=1}^{r_\varepsilon(x^*)} (\eta_j^*)_1 \nabla^2 G_j(x^*) \begin{pmatrix} 1 \\ -\dfrac{\tilde{G}_j(x^*)}{\|\tilde{G}_j(x^*)\|} \end{pmatrix} \\ &= B(x^*, \eta^*). \end{aligned}$$

This completes the proof. □

4.5 Numerical experiments

In this section, we report some numerical results. The program was coded in Matlab 2008a and run on a machine with an Intel®Core2 Duo E6850 3.00GHz CPU and 4GB RAM. Throughout the experiments, we let the index set be given by $T := \{t \in \mathbb{R} \mid h(t) \geq 0\}$, where $h(t) := (t+1, 1-t)^\top$, i.e., $T = [-1, 1]$. The actual implementation of Algorithm 4.1 was carried out as follows. To obtain $T_\varepsilon(x^k)$, we compute local minimizers of the lower-level problem $P(x^k)$. For this purpose, we first compute $\lambda(x^k, t)$ for $t = -1, -0.98, -0.96, \ldots, 0.98, 1$, where $\lambda(\cdot, \cdot)$ is defined as in (4.2.1). We then find local minimizers among $\lambda(x^k, -1), \lambda(x^k, -0.98), \ldots, \lambda(x^k, 1)$ and apply Newton's method with them as starting points. In Step 0, we set the parameters as $\alpha = 0.5$, $\beta = 10^{-5}$, $\delta = 5$, $\varepsilon = 0.1$ and $\rho_{-1} = 10$. The initial point $x^0 \in \mathbb{R}^n$ and the initial matrix $B_0 \in \mathbb{R}^{n \times n}$ are chosen as $x^0 := (10, 10, 10, \ldots, 10)^\top$ and the identity matrix I_n, respectively. In Step 1, we make use of the smoothing method [18, 26] to solve QSOCP(x^k, ε). In Step 2, we stop the algorithm when $\|d^k\| \leq 10^{-7}$ is satisfied. In Step 5, we update the matrix $B_k \in \mathbb{R}^{n \times n}$

by (4.3.8), where the vectors ζ_j^k $(j = 1, 2, \dots, r_\varepsilon(x^k))$ are set as

$$\zeta_j^k := \begin{cases} \eta_i^k & \text{if we find } i \in \{1, 2, \dots, r_\varepsilon(x^k)\} \text{ such that } t_j^k(\cdot) = t_i^{k-1}(\cdot) \\ 0 & \text{otherwise.} \end{cases} \tag{4.5.1}$$

In (4.5.1), we regard $t_j^k(\cdot) = t_i^{k-1}(\cdot)$ when

$$j \in \operatorname{argmin}_{l=1,2,\dots,r_\varepsilon(x^{k-1})} \|t_l^k(x^k) - t_i^{k-1}(x^{k-1}) - \nabla t_i^{k-1}(x^{k-1})^\top (x^k - x^{k-1})\|$$

and

$$\|t_j^k(x^k) - t_i^{k-1}(x^{k-1}) - \nabla t_i^{k-1}(x^{k-1})^\top (x^k - x^{k-1})\| \le 10^{-4}.$$

Moreover, to ensure positive definiteness of B_k, we modify B_k as follows. Let $\alpha_{ki} \in \mathbb{R}$ $(i = 1, 2, \dots, n)$ and $V_k \in \mathbb{R}^{n \times n}$ be scalars and a matrix such that $B_k = V_k \mathrm{diag}(\alpha_{ki})_{i=1}^n V_k^\top$, respectively. Then, for each i, we replace α_{ki} by 10^{-4} if $\alpha_{ki} \le 10^{-5}$, and redefine B_k as $B_k = V_k \mathrm{diag}(\alpha_{ki})_{i=1}^n V_k^\top$.

Experiment 1

In the first experiment, we examine the convergence behavior of Algorithm 4.1 by solving the vector-valued Chebyshev approximation problem (1.1.4). Let $Q : \mathbb{R} \to \mathbb{R}^3$ and $q : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^3$ be defined by

$$Q(t) := \begin{pmatrix} e^{t^2} + \cos t^2 \\ 2te^{t^2} - 2t \sin t^2 \\ (4t^2 + 2)e^{t^2} - 2 \sin t^2 - 4t^2 \cos t^2 \end{pmatrix}$$

and

$$q(u, t) := \begin{pmatrix} \sum_{\nu=1}^n u_\nu t^{\nu-1} \\ \sum_{\nu=2}^n (\nu - 1) u_\nu t^{\nu-2} \\ \sum_{\nu=3}^n (\nu - 1)(\nu - 2) u_\nu t^{\nu-3} \end{pmatrix}.$$

To find a $u \in \mathbb{R}^n$ such that $q(u, t) \approx Q(t)$ over $t \in T$, we solve the following problem:

$$\underset{u \in \mathbb{R}^n}{\text{Minimize}} \ \max_{t \in T} \ \|Q(t) - q(u, t)\|. \tag{4.5.2}$$

As in (1.1.5), by using an auxiliary variable $v \in \mathbb{R}$, we can reformulate (4.5.2) as the following SISOCP with the four-dimensional SOC:

$$\begin{array}{ll} \underset{(v,u) \in \mathbb{R} \times \mathbb{R}^n}{\text{Minimize}} & v \\ \text{subject to} & \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & t & t^2 & \cdots & t^n \\ 0 & 0 & 1 & 2t & \cdots & nt^{n-1} \\ 0 & 0 & 0 & 2 & \cdots & n(n-1)t^{n-2} \end{pmatrix} \begin{pmatrix} v \\ u \end{pmatrix} - \begin{pmatrix} 0 \\ e^{t^2} + \cos t^2 \\ 2te^{t^2} - 2t \sin t^2 \\ (4t^2 + 2)e^{t^2} - 2 \sin t^2 - 4t^2 \cos t^2 \end{pmatrix} \in \mathcal{K}^4 \\ & \text{for all } t \in T. \end{array} \tag{4.5.3}$$

We then apply Algorithm 4.1 to SISOCP (4.5.3) with $n = 6$ and $n = 8$. The obtained results are shown in Tables 4.1 and 4.2, where cpu(s) denotes the running time of Algorithm 4.1 in seconds, and $\mathrm{KKT}(x^k, \eta^k)$ is given by

$$\mathrm{KKT}(x^k, \eta^k) := \begin{pmatrix} \nabla f(x^k) - \sum_{j=1}^{r_\varepsilon(x^k)} \nabla_x g(x^k, \bar{t}_j^k)\eta_i^k \\ \eta_1^k - P_{\mathcal{K}^m}\left(\eta_1^k - g(x^k, \bar{t}_1^k)\right) \\ \vdots \\ \eta_{r_\varepsilon(x^k)}^k - P_{\mathcal{K}^m}\left(\eta_{r_\varepsilon(x^k)}^k - g(x, \bar{t}_{r_\varepsilon(x^k)}^k)\right) \end{pmatrix}$$

and $\eta^k = (\eta_1^k, \dots, \eta_{r_\varepsilon(x^k)}^k) \in \mathbb{R}^m \times \cdots \times \mathbb{R}^m$ is a Lagrange multiplier vector satisfying the KKT conditions (4.3.5) and (4.3.6) for $\mathrm{QSOCP}(x^k, \varepsilon)$. Note that, by Proposition 2.4.3, $\mathrm{KKT}(x^k, \eta^k) = 0$ if and only if (x^k, η^k) satisfies the KKT conditions (4.3.1) and (4.3.2) for SISOCP (4.1.1). From the tables, we can observe that Algorithm 4.1 succeeds in getting an optimal solution for SISOCP (4.5.3). Indeed, x^k and η^k satisfy the KKT conditions for SISOCP (4.5.3) accurately, since $\|\mathrm{KKT}(x^k, \eta^k)\| \le 10^{-10}$ at the last iteration. Also, we can observe that the step size s_k equals 1 in the final stage and $\{x^k\}$ converges to a solution rapidly. In addition, we confirm that $|T_\varepsilon(x^k)|$ becomes constant and the implicit functions $\{t_j^k(\cdot)\}_{j=1}^{r_\varepsilon(x^k)}$ remain unchanged eventually, and hence Assumption 4.4.5 (f) holds.

k	s_k	$\|d^k\|$	$\|\mathrm{KKT}(x^k, \eta^k)\|$	$\|T_\varepsilon(x^k)\|$
1	1.0	1.73e+01	1.47e+02	1
2	0.5	1.47e+01	7.06e-01	2
⋮	⋮	⋮	⋮	⋮
6	1.0	9.18e-04	5.99e-04	5
7	1.0	4.92e-07	2.63e-07	5
8	1.0	7.83e-11	4.20e-11	5
cpu(s): 5.8 seconds				

Table 4.1: Results for Experiment 1 ($n = 6$)

k	s_k	$\|d^k\|$	$\|\mathrm{KKT}(x^k, \eta^k)\|$	$\|T_\varepsilon(x^k)\|$
1	1.0	2.23e+01	5.03e+02	1
2	0.5	1.46e+01	7.06e-01	1
⋮	⋮	⋮	⋮	⋮
10	0.5	7.94e-07	2.51e-06	7
11	1.0	3.97e-07	1.26e-06	7
12	1.0	1.78e-12	5.64e-12	7
cpu(s): 22.4 seconds				

Table 4.2: Results for Experiment 1 ($n = 8$)

Experiment 2

In Experiment 1, we have observed that Algorithm 4.1 obtains accurate solutions with a rapid convergence rate. Thus, if a starting point is chosen near an optimal solution, Algorithm 4.1 is expected to find a solution more efficiently. In this experiment, to produce such a starting point, we use Algorithm 3.2 proposed in Chapter 3, and then use Algorithm 4.1 with an approximate solution computed by Algorithm 3.2. The Algorithm 3.2 was implemented as described in Experiment 3-1 of Chapter 3. The computational results for SISOCP (4.5.3) with $n = 6$ and $n = 8$ are shown in Table 4.3 and Table 4.4, respectively, where

- cpu(s) (Algorithm 3.2): the running time of Algorithm 3.2
- cpu(s) (Algorithm 4.1): the running time of Algorithm 4.1
- cpu(s) (Algorithm 3.2+Algorithm 4.1): the total running time of Algorithm 3.2 and Algorithm 4.1.

From the tables, we observe that the total computational times are much less than those in Experiment 1. In particular, when $n = 8$, Algorithm 4.1 combined with Algorithm 3.2 took only 15.3 seconds in total, while Algorithm 4.1 alone spent 22.4 seconds in Experiment 1.

cpu(s) (Algorithm 3.2)	2.0 seconds
cpu(s) (Algorithm 4.1)	0.49 seconds
cpu(s) (Algorithm 3.2+Algorithm 4.1)	2.49 seconds

Table 4.3: Results for Experiment 2 ($n = 6$)

cpu(s) (Algorithm 3.2)	4.0 seconds
cpu(s) (Algorithm 4.1)	11.3 seconds
cpu(s) (Algorithm 3.2+Algorithm 4.1)	15.3 seconds

Table 4.4: Results for Experiment 2 ($n = 8$)

Experiment 3

In the third experiment, we implemented another SQP-type algorithm, which is also expected to find accurate solutions rapidly, and compared it with Algorithm 4.1 by solving the following SISOCP that contains multiple SOCs:

$$\begin{aligned} &\underset{x\in\mathbb{R}^{10}}{\text{Minimize}} && \tfrac{1}{2}x^\top Mx + c^\top x \\ &\text{subject to} && A^s(t)x - b^s(t) \in \mathcal{K}^{m_s} \text{ for all } t \in T, \\ & && s = 1, 2, \ldots, S, \end{aligned} \tag{4.5.4}$$

where $c \in \mathbb{R}^{10}$, $A^s(t) := (A^s_{ij}(t)) \in \mathbb{R}^{m_s\times 10}$ with $A^s_{ij}(t) := \sum_{\ell=0}^5 \alpha^s_{ij\ell}t^\ell$ $(i = 1, 2, \ldots, m_s, j = 1, 2, \ldots, 10)$ and $b^s(t) := (b^s_i(t)) \in \mathbb{R}^{m_s}$ with $b^s_1(t) := -\sum_{i=2}^{m_s}\sum_{\ell=0}^5 |\beta^s_{i\ell}|$ and $b^s_i(t) := \sum_{\ell=0}^5 \beta^s_{i\ell}t^\ell$

$(i = 2, \dots, m_s)$. The SOCs $\mathcal{K} := \mathcal{K}^{m_1} \times \mathcal{K}^{m_2} \times \cdots \mathcal{K}^{m_S}$ are chosen as in Table 4.5. For each type of SOC $\mathcal{K}$, we generate 50 problems as follows: The problem data $\alpha^s_{1j\ell}, \alpha^s_{ij\ell}, \beta^s_{i\ell}$ $(i = 2, \dots, m_s,\ j = 1, 2, \dots, 10,\ \ell = 0, 1, 2, \dots, 5,\ s = 1, 2, \dots, S)$ are chosen randomly from the interval $[2, -2]$. All components of c are randomly chosen from the interval $[5, -5]$. The matrix M is set to be $M := M_1^\top M_1 + 0.1 I_n$, where $M_1 \in \mathbb{R}^{n \times n}$ is a matrix whose entries are randomly chosen from the interval $[1, -1]$. Notice that, by the choice of $b^s_1(t)$, we can ensure that (4.5.4) is feasible.[1] In Step 3, we use the following penalty function for SISOCP (4.5.4) with multiple SOCs, which is a natural extension of the function defined by (4.3.7):

$$\Phi_\rho(x) := f(x) + \rho \sum_{s=1}^{S} \varphi^s_+(x), \tag{4.5.5}$$

where $\varphi^s(x) := \max_{t \in T} \left(-A^s_1(t)x + b^s_1(t) + \|\tilde{A}^s(t)(x) - \tilde{b}^s(t)\| \right)$ for $s = 1, 2, \dots, S$. Accordingly, we extend the update rule of the penalty parameters $\{\rho_k\}$ in Step 5 as follows:

$$\rho_k := \begin{cases} \rho_{k-1} & \text{if } \rho_{k-1} \geq \displaystyle\max_{s=1,2,\dots,S} \sum_{j=1}^{r^s_\varepsilon(x^k)} (\eta^{k+1}_{sj})_1 \\ \delta + \displaystyle\max_{s=1,2,\dots,S} \sum_{j=1}^{r^s_\varepsilon(x^k)} (\eta^{k+1}_{sj})_1 & \text{otherwise,} \end{cases}$$

where $\delta > 0$ is a given constant and η^{k+1}_{sj} $(j = 1, 2, \dots, r^s_\varepsilon(x^k), s = 1, 2, \dots, S)$ are Lagrange multiplier vectors obtained by solving QSOCP(x^k, ε) for SISOCP (4.5.4).

We next explain another SQP-type algorithm, which we call the QP-based method. For simplicity of expression, we consider the case of SISOCP (4.1.1) with a single SOC. In the QP-based method, we reformulate SOCP(x^k, ε) as the following nonlinear program that does not contain SOC constraints explicitly:

$$\min_{x \in U(x^k)} \ f(x) \ \text{ s.t. } v^k_j(x) \geq 0 \ (j = 1, 2, \dots, r_\varepsilon(x^k)), \tag{4.5.6}$$

where $v^k_j(x) := \lambda(x, t^k_j(x))$ for $j = 1, 2, \dots, r_\varepsilon(x^k)$, and then generate a search direction d^k by solving the following Quadratic Program[2] (QP):

$$\text{QP}(x^k, \varepsilon): \quad \begin{array}{ll} \text{Minimize} & \nabla f(x^k)^\top d + \frac{1}{2} d^\top \tilde{B}_k d \\ \text{subject to} & v^k_j(x^k) + \nabla v^k_j(x^k)^\top d \geq 0 \ (j = 1, 2, \dots, r_\varepsilon(x^k)), \end{array}$$

where $\tilde{B}_k \in S^n_{++}$. We make use of the Hessian of the Lagrangian of (4.5.6). Specifically, we first compute

$$\tilde{D}_k := \nabla^2 f(x^k) - \sum_{j=1}^{r_\varepsilon(x^k)} \xi^k_j \nabla^2 v^k_j(x^k),$$

with

$$\xi^k_j := \begin{cases} \tilde{\xi}^k_i & \text{if we find } i \in \{1, 2, \dots, r_\varepsilon(x^k)\} \text{ such that } t^k_j(\cdot) = t^{k-1}_i(\cdot) \\ 0 & \text{otherwise,} \end{cases}$$

[1] The origin $x = 0$ always lies in the interior of the feasible region, since we have $-b^s(t) \in \operatorname{int} \mathcal{K}^{m_s}$ from $-b^s_1(t) - \|(-b^s_2(t), \dots, -b^s_{m_s}(t))^\top\| > 0$ for all $t \in T$.

[2] We also suppose that Assumption 4.4.1 (b) holds.

where $\tilde{\xi}_i^k \in \mathbb{R}$ $(i = 1, 2, \ldots, r_\varepsilon(x^{k-1}))$ are Lagrange multipliers satisfying the KKT conditions of QP(x^{k-1}, ε). Note that $\nabla v_j^k(x^k)$ and $\nabla^2 v_j^k(x^k)$ for $j = 1, 2, \ldots, r_\varepsilon(x^k)$ can be calculated from Proposition 4.2.2 (d). Similarly to the matrix B_k for Algorithm 4.1, we also ensure the positive definiteness of $\tilde{B}_k$ as follows: Let $\tilde{\alpha}_{ki} \in \mathbb{R}$ $(i = 1, 2, \ldots, n)$ and $\tilde{V}_k \in \mathbb{R}^{n \times n}$ be scalars and a matrix such that $\tilde{B}_k = \tilde{V}_k \mathrm{diag}(\tilde{\alpha}_{ki})_{i=1}^n \tilde{V}_k^\top$, respectively. Then, for each i, we replace $\tilde{\alpha}_{ki}$ by 10^{-4} if $\tilde{\alpha}_{ki} \leq 10^{-5}$, and redefine $\tilde{B}_k$ as $\tilde{B}_k = \tilde{V}_k \mathrm{diag}(\tilde{\alpha}_{ki})_{i=1}^n \tilde{V}_k^\top$. We use the penalty function defined by (4.3.7) (by (4.5.5) for problem (4.5.4)) and determine a step size by the Armijo line search. We update the penalty parameters $\{\rho_k\}$ as follows: If $\rho_{k-1} \geq \sum_{j=1}^{r_\varepsilon(x^k)} \tilde{\xi}_j^{k+1}$, then we set $\rho_k := \rho_{k-1}$. Otherwise, we set $\rho_k := \sum_{j=1}^{r_\varepsilon(x^k)} \tilde{\xi}_j^{k+1} + \delta$, where $\delta > 0$ is a given constant.

We extend the above QP-based method to (4.5.4) and implement it. The choice of parameters in the QP-based method is the same as in Algorithm 4.1. Moreover, we solve QP(x^k, ε) with the solver *quadprog* in MATLAB Optimization Toolbox.

The obtained results are shown in Table 4.5, where each column represents the following:

- $\mathrm{ite}_{\max}$: the maximum number of iterations among 50 problems for each $\mathcal{K}$
- $\mathrm{ite}_{\min}$: the minimum number of iterations among 50 problems for each $\mathcal{K}$
- $\mathrm{ite}_{\mathrm{ave}}$: the average number of iterations over 50 problems for each $\mathcal{K}$
- cpu(s): the average time in seconds over 50 problems for each $\mathcal{K}$

For all the generated problems, both algorithms successfully obtain optimal solutions. From the table, we can observe that Algorithm 4.1 tends to perform better than the QP-based method. In particular, when $\mathcal{K} = 10$, both the number of iterations and the computational time for Algorithm 4.1 are less than half of those for the QP-based method. This fact suggests that Algorithm 4.1 may exploit the structure of SOC more effectively than the QP-based method.

	Algorithm 4.1				QP-based method			
$\mathcal{K}$	$\mathrm{ite}_{\max}$	$\mathrm{ite}_{\min}$	$\mathrm{ite}_{\mathrm{ave}}$	cpu(s)	$\mathrm{ite}_{\max}$	$\mathrm{ite}_{\min}$	$\mathrm{ite}_{\mathrm{ave}}$	cpu(s)
$\mathcal{K}^{10}$	19	3	6.22	1.04	49	6	13.28	1.91
$\mathcal{K}^{30}$	12	3	5.34	2.06	41	6	12.52	4.09
$\mathcal{K}^{50}$	11	3	5.54	2.66	31	7	13.36	4.04
$\mathcal{K}^{20} \times \mathcal{K}^{30}$	17	3	5.56	2.95	24	7	11.48	4.55
$\mathcal{K}^{20} \times \mathcal{K}^{15} \times \mathcal{K}^{15}$	13	3	5.95	5.91	23	7	11.46	6.36

Table 4.5: Comparison of Algorithm 4.1 and the QP-based method (Experiment 3)

4.6 Concluding remarks

For solving the semi-infinite program with an infinite number of SOC constraints, we proposed the local reduction based SQP-type method. We studied the global and local convergence properties of the proposed algorithm. Finally, in the numerical experiments, we actually implemented and examined its effectiveness. For the sake of comparison, we also implemented a regularized explicit exchange method and another SQP-type method, and observed good performance of the proposed algorithm.

Chapter 5

A smoothing SQP method for mathematical programs with SOC complementarity constraints

5.1 Introduction

In the previous chapters, we have studied algorithms for solving semi-infinite programs with infinitely many SOC constraints. In this chapter, we turn to a class of mathematical programs with equilibrium constraints. Especially, we focus on the following mathematical program with SOC complementarity constraints, abbreviated as MPSOCC:

$$\begin{array}{ll} \underset{x,y,z}{\text{Minimize}} & f(x,y) \\ \text{subject to} & Ax \leq b, \\ & z = Nx + My + q, \\ & \mathcal{K} \ni y \perp z \in \mathcal{K}, \end{array} \tag{5.1.1}$$

where $f\colon \mathbb{R}^{n+m} \to \mathbb{R}$ is a continuously differentiable function, $A \in \mathbb{R}^{p\times n}$, $b \in \mathbb{R}^p$, $N \in \mathbb{R}^{m\times n}$, $M \in \mathbb{R}^{m\times m}$ and $q \in \mathbb{R}^m$ are given matrices and vectors, and $\mathcal{K}$ is the Cartesian product of second-order cones, that is, $\mathcal{K} := \mathcal{K}^{m_1} \times \mathcal{K}^{m_2} \times \cdots \times \mathcal{K}^{m_\ell}$. Throughout the chapter, we suppose that $m_i \geq 2$ for each i.

The main purpose of this chapter is to develop an algorithm for solving MPSOCC (5.1.1). As mentioned in Chapter 1, Yan and Fukushima [77] proposed a smoothing method for solving such problems. In their convergence theory, it is supposed that smoothed subproblems are solved exactly. However, it can hardly be expected in practice. To overcome such a difficulty, we propose to combine an SQP-type method with the smoothing method. The proposed method replaces the SOC complementarity condition of MPSOCC (5.1.1) with a certain vector equation by using a smoothed natural residual function, thereby yielding convex quadratic programming subproblems which can be solved efficiently by any state-of-the-art method such as the active-set method and interior point method. While Yan and Fukushima's method solves the smoothed subproblem exactly at each iteration, our method only solves convex quadratic programming subproblems that approximates the smoothed subproblem.

Although our method may be viewed as an extension of the SQP method in [17], the convergence analysis is quite different since it exploits the particular properties of the natural residual associated with the SOC complementarity condition.

This chapter is organized as follows. In Section 5.2, we introduce the Cartesian P_0 and the Cartesian P matrices that play a key role to establish the well-definedness of the proposed algorithm. In Section 5.3, we reformulate MPSOCC (5.1.1) as a nonlinear programming problem by replacing the second-order cone complementarity constraints by equivalent nonsmooth equality constraints. In Section 5.4, we recall a smoothing technique to deal with the nonsmooth constraints. In Section 5.5, we propose an SQP-type algorithm for solving problem (5.1.1) and show that the proposed method is well-defined. In Section 5.6, we prove that the proposed algorithm possesses the global convergence property under the strict complementarity assumptions. In Section 5.7, we give some numerical examples. In Section 5.8, we end this chapter with some concluding remarks.

5.2 Cartesian P_0 and P matrices

In this section, we introduce the Cartesian P_0 and the Cartesian P matrices. The concept of the Cartesian $P_0\,(P)$ matrix is a natural extension of the well-known $P_0\,(P)$ matrix [13]. Although the Cartesian $P_0\,(P)$ matrix can be defined not only for the SOC but also for the semidefinite cone [11] and the symmetric cone [23], we restrict ourselves to the case of the SOCs.

Definition 5.2.1. *Suppose that the Cartesian structure of $\mathcal{K} \subseteq \mathbb{R}^m$ is given as $\mathcal{K} := \mathcal{K}^{m_1} \times \mathcal{K}^{m_2} \times \cdots \times \mathcal{K}^{m_\ell}$. Then, $M \in \mathbb{R}^{m\times m}$ is called*

(a) *a Cartesian P_0 matrix if, for every nonzero $z = (z^1, \dots, z^\ell) \in \mathbb{R}^m = \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_\ell}$, there exists an index $i \in \{1, \dots, \ell\}$ such that $(z^i)^\top (Mz)^i \geq 0$;*

(b) *a Cartesian P matrix if, for every nonzero $z = (z^1, \dots, z^\ell) \in \mathbb{R}^m = \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_\ell}$, there exists an index $i \in \{1, \dots, \ell\}$ such that $(z^i)^\top (Mz)^i > 0$.*

Here, $(Mz)^i \in \mathbb{R}^{m_i}$ denotes the i-th subvector of $Mz \in \mathbb{R}^m$ conforming to the Cartesian structure of $\mathcal{K}$.

Notice that the definition of the Cartesian P_0 (P) property depends on the Cartesian structure of $\mathcal{K}$. In what follows, we assume that the Cartesian structure of $\mathcal{K}$ is always given as $\mathcal{K} = \mathcal{K}^{m_1} \times \mathcal{K}^{m_2} \times \cdots \times \mathcal{K}^{m_\ell}$. The definition of the "classical" $P_0\,(P)$ matrix corresponds to the case where $\mathcal{K} = \mathbb{R}^m_+$. It is easily seen that every Cartesian $P_0\,(P)$ matrix is a P_0 (P) matrix [48].

The following proposition implies that the Cartesian $P_0\,(P)$ property is preserved under a nonsingular block-diagonal transformation.

Proposition 5.2.2. *Let $M \in \mathbb{R}^{m\times m}$ be any matrix, and $H_i \in \mathbb{R}^{m_i\times m_i} (i = 1, 2, \dots, \ell)$ be arbitrary nonsingular matrices. Let the matrix $M' \in \mathbb{R}^{m\times m}$ be defined by*

$$M' := H^\top M H, \quad H := \begin{pmatrix} H_1 & & 0 \\ & \ddots & \\ 0 & & H_\ell \end{pmatrix}.$$

Then, the following statements hold.

(a) *If M is a Cartesian P_0 matrix, then M' is a Cartesian P_0 matrix.*

(b) *If M is a Cartesian P matrix, then M' is a Cartesian P matrix.*

Proof. We first show (b). Let $z = (z^1, \dots, z^\ell) \in \mathbb{R}^m = \mathbb{R}^{m_1} \times \cdots \times \mathbb{R}^{m_\ell}$ be an arbitrary nonzero vector. We show that there exists an $i \in \{1, 2, \dots, \ell\}$ such that $(z^i)^\top (M'z)^i > 0$. Note that

$$\begin{aligned}
(M'z)^i &= \left(\begin{pmatrix} H_1 & & 0 \\ & \ddots & \\ 0 & & H_\ell \end{pmatrix}^\top M \begin{pmatrix} H_1 & & 0 \\ & \ddots & \\ 0 & & H_\ell \end{pmatrix} z \right)^i = \left(\begin{pmatrix} H_1 & & 0 \\ & \ddots & \\ 0 & & H_\ell \end{pmatrix}^\top M \begin{pmatrix} H_1 z^1 \\ \vdots \\ H_\ell z^\ell \end{pmatrix} \right)^i \\
&= \begin{pmatrix} H_1^\top \sum_{k=1}^{\ell} M_{1k} H_k z^k \\ \vdots \\ H_\ell^\top \sum_{k=1}^{\ell} M_{\ell k} H_k z^k \end{pmatrix}^i \\
&= \left(H_i^\top \sum_{k=1}^{\ell} M_{ik} H_k z^k \right).
\end{aligned}$$

where $(\cdot)^i$ and $(\cdot)_{ik}$ denote the i-th subvector and the (i,k)-th block entry, respectively, conforming to the Cartesian structure of $\mathcal{K}$. Hence, we have

$$(z^i)^\top (M'z)^i = (z^i)^\top H_i^\top \sum_{k=1}^{\ell} M_{ik} H_k z^k = \left(H_i z^i\right)^\top \sum_{k=1}^{\ell} M_{ik} H_k z^k = ((Hz)^i)^\top (MHz)^i.$$

Since M is a Cartesian P matrix and $Hz \neq 0$ from the nonsingularity of H, we have $(z^i)^\top (M'z)^i = ((Hz)^i)^\top (MHz)^i > 0$ for some i. Hence, M' is a Cartesian P matrix.

We omit the proof of (a) since it can be shown in a similar manner to (b). □

5.3 Reformulation of MPSOCC and B-stationary points of MPSOCC

In Chapter 2, we introduced the natural residual function Φ and observed that SOC complementarity condition $\mathcal{K} \ni y \perp z \in \mathcal{K}$ can be represented as $\Phi(y, z) = 0$ equivalently. In this section, we rewrite MPSOCC (5.1.1) as the following problem where the SOC complementarity constraint is replaced by the equivalent equality constraint involving the natural residual function Φ:

$$\begin{array}{ll}
\underset{x,y,z}{\text{Minimize}} & f(x, y) \\
\text{subject to} & Ax \leq b, \\
& z = Nx + My + q, \\
& \Phi(y, z) = 0.
\end{array} \tag{5.3.1}$$

We also call this problem MPSOCC. MPSOCC (5.3.1) is a nonsmooth optimization problem since Φ is not differentiable everywhere. However, as is shown below, Φ is continuously differentiable at any (y,z) satisfying the following strict complementarity condition:

Definition 5.3.1 (Strict complementarity). *Suppose that $(y,z) \in \mathbb{R}^m \times \mathbb{R}^m$ satisfies the SOC complementarity condition $\mathcal{K} \ni y \perp z \in \mathcal{K}$. Moreover, decompose y and z as $y = (y^1, y^2, \ldots, y^\ell)$ and $z = (z^1, z^2, \ldots, z^\ell) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \cdots \times \mathbb{R}^{m_\ell} = \mathbb{R}^m$ conforming to the Cartesian structure of $\mathcal{K}$. Then, we say that strict complementarity holds at (y,z) if, for every $i = 1, 2, \ldots, \ell$, one of the following three conditions holds:*

(i) $y^i \in \operatorname{int} \mathcal{K}^{m_i},\ z^i = 0$;

(ii) $y^i = 0,\ z^i \in \operatorname{int} \mathcal{K}^{m_i}$;

(iii) $y^i \in \operatorname{bd} \mathcal{K}^{m_i} \setminus \{0\},\ z^i \in \operatorname{bd} \mathcal{K}^{m_i} \setminus \{0\},\ (y^i)^\top z^i = 0$.

Lemma 5.3.2. *Let $y, z \in \mathbb{R}^m$ be chosen so that $y - z \notin \operatorname{bd}(\mathcal{K} \cup -\mathcal{K})$. Then, the function $\Phi : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m$ defined by (2.4.1) is continuously differentiable at (y,z), and the following equality holds:*

$$\nabla_y \Phi(y,z) + \nabla_z \Phi(y,z) = I_m,$$

where $I_m \in \mathbb{R}^{m \times m}$ denotes the identity matrix.

Proof. It suffices to consider the case where $\mathcal{K} = \mathcal{K}^m$. Let $\lambda_1, \lambda_2 \in \mathbb{R}$ be the spectral values of $y - z$ defined as in Definition 2.4.1. Note that, from $y - z \notin \operatorname{bd}(\mathcal{K}^m \cup -\mathcal{K}^m)$, we have $\lambda_1, \lambda_2 \neq 0$. Then, from [26, Proposition 4.8], the Clarke subdifferential $\partial P_{\mathcal{K}^m}(y - z)$ is explicitly given as

$$\partial P_{\mathcal{K}^m}(y-z) = \begin{cases} I_m & (\lambda_1 > 0,\ \lambda_2 > 0), \\ \dfrac{\lambda_2}{\lambda_2 - \lambda_1} I_m + W & (\lambda_1 < 0,\ \lambda_2 > 0), \\ O & (\lambda_1 < 0,\ \lambda_2 < 0), \end{cases}$$

where

$$W := \frac{1}{2} \begin{pmatrix} -r_1 & r_2^\top \\ r_2 & -r_1 r_2 r_2^\top \end{pmatrix}, \qquad (r_1, r_2) := \frac{(y_1 - z_1, y_2 - z_2)}{\|y_2 - z_2\|}.$$

Thus $P_{\mathcal{K}^m}$ is differentiable at $y - z$. This fact readily implies the continuous differentiability of Φ at (y,z) since $\Phi(y,z) = y - P_{\mathcal{K}^m}(y - z)$. We next show the second half of the proposition. By an easy calculation, we have

$$\nabla_y \Phi(y,z) \quad = \quad I_m - \nabla P_{\mathcal{K}^m}(y - z).$$

Similarly, we have $\nabla_z \Phi(y,z) = \nabla P_{\mathcal{K}^m}(y - z)$. Hence we obtain the desired equality. □

Proposition 5.3.3. *Let $(\overline{y}, \overline{z}) \in \mathbb{R}^m \times \mathbb{R}^m$ satisfy the strict complementarity condition. Then, Φ is continuously differentiable at $(\overline{y}, \overline{z})$.*

Proof. The strict complementarity condition readily yields $\overline{y} - \overline{z} \notin \operatorname{bd}(\mathcal{K} \cup -\mathcal{K})$. Hence, Φ is continuously differentiable at $(\overline{y}, \overline{z})$ by Lemma 5.3.2. □

Now, let $X := \{x \in \mathbb{R}^n \mid Ax \le b\}$, and let $\overline{w} := (\overline{x}, \overline{y}, \overline{z}) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$ be a feasible point of MPSOCC (5.1.1) that satisfies the strict complementarity condition. By Proposition 5.3.3, MPSOCC (5.3.1) can be viewed as a smooth optimization problem in a small neighborhood of $\overline{w}$. Then, the Karush-Kuhn-Tucker (KKT) conditions on $\overline{w}$ are represented as

$$\begin{pmatrix} \nabla_x f(\overline{x}, \overline{y}) \\ \nabla_y f(\overline{x}, \overline{y}) \\ 0 \end{pmatrix} + \begin{pmatrix} N^T \\ M^T \\ -I \end{pmatrix} u + \begin{pmatrix} 0 \\ \nabla_y \Phi(\overline{y}, \overline{z}) \\ \nabla_z \Phi(\overline{y}, \overline{z}) \end{pmatrix} v \in -\mathcal{N}_X(\overline{x}) \times \{0\}^{2m},$$
$$\Phi(\overline{y}, \overline{z}) = 0,\ A\overline{x} \le b,\ \overline{z} = N\overline{x} + M\overline{y} + q, \tag{5.3.2}$$

where $\{0\}^{2m} := \{0\} \times \{0\} \times \cdots \times \{0\} \subseteq \mathbb{R}^{2m}$, and $u \in \mathbb{R}^\ell, v \in \mathbb{R}^m$ and $\eta \in \mathbb{R}^m$ are Lagrange multipliers.

We next consider the stationarity of MPSOCC (5.1.1) or (5.3.1). So far, several kinds of stationary points have been studied in the literature of MPECs, e.g., see [59]. Among them, a Bouligand- or B-stationary point is the most desirable, since it is directly related to the first order optimality condition. Specifically, a B-stationary point for MPSOCC (5.1.1) is defined as follows:

Definition 5.3.4 (B-stationarity)**.** *Let $\mathcal{F} \subseteq \mathbb{R}^{n+2m}$ denote the feasible set of MPSOCC* (5.1.1)*. We say that $\overline{w} := (\overline{x}, \overline{y}, \overline{z}) \in \mathcal{F}$ is a B-stationary point of MPSOCC* (5.1.1) *if $(-\nabla f(\overline{x}, \overline{y}), 0) \in \mathcal{N}_{\mathcal{F}}(\overline{w})$ holds.*

In what follows, we show that a point satisfying KKT conditions (5.3.2) is a B-stationary point. For this purpose, we give three useful lemmas.

Lemma 5.3.5. [56, Proposition 6.41] *Let $C := C_1 \times C_2 \times \cdots \times C_s$ for nonempty closed sets $C_i \subseteq \mathbb{R}^{n_i}$. Choose $\overline{\zeta} := (\overline{\zeta}_1, \overline{\zeta}_2, \ldots, \overline{\zeta}_s) \in C_1 \times C_2 \times \cdots \times C_s$. Then, we have $\mathcal{N}_C(\overline{\zeta}) = \mathcal{N}_{C_1}(\overline{\zeta}_1) \times \mathcal{N}_{C_2}(\overline{\zeta}_2) \times \cdots \times \mathcal{N}_{C_s}(\overline{\zeta}_s)$.*

Lemma 5.3.6. [56, Chapter 6-C] *For a continuously differentiable function $F : \mathbb{R}^p \to \mathbb{R}^q$, let $D := \{\zeta \in \mathbb{R}^p \mid F(\zeta) = 0\}$. Choose $\overline{\zeta} \in D$ arbitrarily. If $\nabla F(\overline{\zeta})$ has full column rank, then we have*

$$\mathcal{N}_D(\overline{\zeta}) = \nabla F(\overline{\zeta})\mathbb{R}^q := \{\zeta \in \mathbb{R}^p \mid \zeta = \nabla F(\overline{\zeta})v,\ v \in \mathbb{R}^q\}.$$

Lemma 5.3.7. [56, Theorem 6.14] *For a continuously differentiable function $F : \mathbb{R}^p \to \mathbb{R}^q$ and a closed set $C \subseteq \mathbb{R}^p$, let $D := \{\zeta \in C \mid F(\zeta) = 0\}$. Choose $\overline{\zeta} \in D$ arbitrarily. Then we have*

$$\mathcal{N}_D(\overline{\zeta}) \supseteq \nabla F(\overline{\zeta})\mathbb{R}^q + \mathcal{N}_C(\overline{\zeta}).$$

Now, we show that the KKT conditions (5.3.2) are sufficient conditions for $\overline{w} = (\overline{x}, \overline{y}, \overline{z})$ to be B-stationary.

Proposition 5.3.8. *Let $\overline{w} := (\overline{x}, \overline{y}, \overline{z})$ be a feasible point of MPSOCC* (5.3.1)*. Suppose that the strict complementarity condition holds at $(\overline{y}, \overline{z})$. If $\overline{w}$ satisfies the KKT conditions* (5.3.2)*, then $\overline{w}$ is a B-stationary point of MPSOCC* (5.1.1)*.*

Proof. Let $Y := \{(y,z) \in \mathbb{R}^m \times \mathbb{R}^m \mid \Phi(y,z) = 0\}$. We first note that the strict complementarity at $(\overline{y}, \overline{z})$ implies the continuous differentiability of Φ at $(\overline{y}, \overline{z})$ from Lemma 5.3.2. Choose $v \in \mathbb{R}^m$ such that $\nabla\Phi(\overline{y}, \overline{z})v = 0$. From Lemma 5.3.2, we then have $\nabla_y\Phi(\overline{y}, \overline{z})v = 0$ and $\nabla_z\Phi(\overline{y}, \overline{z})v = (I - \nabla_y\Phi(\overline{y}, \overline{z}))v = 0$, which readily imply $v = 0$, and thus $\nabla\Phi(\overline{y}, \overline{z})$ has full column rank. Therefore, by Lemma 5.3.6 with $p = q := m$, $D := Y$ and $F := \Phi$, we have

$$\mathcal{N}_Y(\overline{y}, \overline{z}) = \nabla\Phi(\overline{y}, \overline{z})\mathbb{R}^m. \tag{5.3.3}$$

Then, it holds that

$$\mathcal{N}_{X\times Y}(\overline{w}) = \mathcal{N}_X(\overline{x}) \times \mathcal{N}_Y(\overline{y}, \overline{z}) = \mathcal{N}_X(\overline{x}) \times \nabla\Phi(\overline{y}, \overline{z})\mathbb{R}^m, \tag{5.3.4}$$

where the first equality follows from Lemma 5.3.5 and the second equality follows from (5.3.3). Now, let $\mathcal{F} \subseteq \mathbb{R}^{n+2m}$ denote the feasible set of MPSOCC (5.3.1), i.e., $\mathcal{F} = \{(x,y,z) \in X \times Y \mid Nx + My - z + q = 0\}$. Then, from Lemma 5.3.7, we have

$$\mathcal{N}_{\mathcal{F}}(\overline{w}) \supseteq \left(N^\top, M^\top, -I\right)^\top \mathbb{R}^m + \mathcal{N}_{X\times Y}(\overline{w}). \tag{5.3.5}$$

Combining (5.3.4) with (5.3.5), we obtain

$$\mathcal{N}_{\mathcal{F}}(\overline{w}) \supseteq \left(N^\top, M^\top, -I\right)^\top \mathbb{R}^m + \mathcal{N}_X(\overline{x}) \times \nabla\Phi(\overline{y}, \overline{z})\mathbb{R}^m,$$

which together with the KKT conditions (5.3.2) implies $(-\nabla f(\overline{x}, \overline{y}), 0) \in \mathcal{N}_{\mathcal{F}}(\overline{w})$. Thus, $\overline{w}$ is a B-stationary point. □

5.4 Smoothing function of natural residual

The natural residual function Φ given in Definition 2.4.2 is not differentiable everywhere, and therefore, we cannot employ a derivative-based algorithm such as Newton's method to solve MPSOCC (5.3.1). To overcome such a difficulty, we will utilize a smoothing technique.

Definition 5.4.1. *Let $\Psi : \mathbb{R}^m \to \mathbb{R}^m$ be a nondifferentiable function. Then, the function $\Psi_\mu : \mathbb{R}^m \to \mathbb{R}^m$ parametrized by $\mu > 0$ is called a smoothing function of Ψ if it satisfies the following properties: For any $\mu > 0$, Ψ_μ is differentiable on $\mathbb{R}^m$; for any $z \in \mathbb{R}^m$, it holds that $\lim_{\mu\to 0+} \Psi_\mu(z) = \Psi(z)$.*

A smoothing function of the natural residual function can be constructed by means of the Chen-Mangasarian (CM) function $\hat{g} : \mathbb{R} \to \mathbb{R}$ [18].

Definition 5.4.2. *A differentiable convex function $\hat{g} : \mathbb{R} \to \mathbb{R}_+$ is called a CM function if*

$$\lim_{\alpha\to-\infty} \hat{g}(\alpha) = 0, \qquad \lim_{\alpha\to\infty} (\hat{g}(\alpha) - \alpha) = 0, \qquad 0 < \hat{g}'(\alpha) < 1 \qquad (\alpha \in \mathbb{R}). \tag{5.4.1}$$

Notice that, if function $p_\mu : \mathbb{R} \to \mathbb{R}$ is defined by $p_\mu(\alpha) := \mu\hat{g}(\alpha/\mu)$ with a CM function $\hat{g}$ and a positive parameter μ, then it becomes a smoothing function for $p(\alpha) := \max\{0, \alpha\}$. Thanks to this fact, we can next provide a smoothing function P_μ for the projection operator $P_{\mathcal{K}}$.

Definition 5.4.3. *Let $z \in \mathbb{R}^m$ be an arbitrary vector decomposed as $z = (z^1, z^2, \cdots, z^\ell) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \cdots \times \mathbb{R}^{m_\ell} = \mathbb{R}^m$ conforming to the given Cartesian structure of $\mathcal{K}$. For an arbitrary CM function $\hat{g} : \mathbb{R} \to \mathbb{R}$, let $g : \mathbb{R}^m \to \mathbb{R}^m$ be defined as*

$$\begin{aligned} g(z) &:= \begin{pmatrix} g^1(z^1) \\ \vdots \\ g^\ell(z^\ell) \end{pmatrix}, \qquad (5.4.2) \\ g^i(z) &:= \hat{g}(\lambda_{i1})c^{i1} + \hat{g}(\lambda_{i2})c^{i2}, \end{aligned}$$

where $\lambda_{ij} \in \mathbb{R}$ and $c^{ij} \in \mathbb{R}^{m_i}$ $((i,j) \in \{1, 2, \cdots, \ell\} \times \{1, 2\})$ are the spectral values and the spectral vectors of subvectors z^i with respect to $\mathcal{K}^{m_i}$, respectively. Then, the smoothing function P_μ: $\mathbb{R}^m \to \mathbb{R}^m$ of $P_\mathcal{K}$ is given as

$$P_\mu(z) := \mu g(z/\mu).$$

Now, by using the above smoothing function P_μ, we can define the smoothing function for the natural residual Φ.

Definition 5.4.4. *Let $\mu > 0$ be arbitrary. Let $g : \mathbb{R}^m \to \mathbb{R}^m, g^i : \mathbb{R}^{m_i} \to \mathbb{R}^{m_i}$ $(i = 1, 2, \ldots, \ell)$, and $P_\mu : \mathbb{R}^m \to \mathbb{R}^m$ be defined as in Definition 5.4.3. Then, the smoothing function $\Phi_\mu : \mathbb{R}^m \to \mathbb{R}^m$ for the natural residual Φ is given as*

$$\begin{aligned} \Phi_\mu(y, z) &:= y - P_\mu(y - z) \\ &= y - \mu g\left(\frac{y - z}{\mu}\right) \\ &= \begin{pmatrix} y^1 - \mu g^1\left(\dfrac{y^1 - z^1}{\mu}\right) \\ \vdots \\ y^\ell - \mu g^\ell\left(\dfrac{y^\ell - z^\ell}{\mu}\right) \end{pmatrix}. \qquad (5.4.3) \end{aligned}$$

Before closing this subsection, we provide the following propositions which will be used in the subsequent analyses.

Proposition 5.4.5. *Let $P_\mu : \mathbb{R}^m \to \mathbb{R}^m$ be defined as in Definition 5.4.3, and choose $\overline{z} \notin \mathrm{bd}\,(\mathcal{K} \cup -\mathcal{K})$ arbitrarily. Let $\{z^k\} \subseteq \mathbb{R}^m$ and $\{\mu_k\} \subseteq \mathbb{R}_{++}$ be arbitrary sequences such that $z^k \to \overline{z}$ and $\mu_k \to 0$ as $k \to \infty$. Then, we have*

$$\nabla P_\mathcal{K}(\overline{z}) = \lim_{k \to \infty} \nabla P_{\mu_k}(z^k). \qquad (5.4.4)$$

Proof. For simplicity, we consider the case where $\mathcal{K} = \mathcal{K}^m$. Let $\overline{z}$ and z^k be decomposed as $\overline{z} = \overline{\lambda}_1 \overline{c}^1 + \overline{\lambda}_2 \overline{c}^2$ and $z^k = \lambda_1^k c_k^1 + \lambda_2^k c_k^2$, where $\overline{\lambda}_i$, $\lambda_i^k \in \mathbb{R}$ are spectral values, and $\overline{c}^i$, $c_k^i \in \mathbb{R}^m$ $(i = 1, 2)$ are spectral vectors of $\overline{z}$ and z^k, respectively. Since $\overline{z} \notin \mathrm{bd}\,(\mathcal{K}^m \cup -\mathcal{K}^m)$, $P_{\mathcal{K}^m}$ is differentiable at $\overline{z}$ and $\nabla P_{\mathcal{K}^m}(\overline{z})$ is given as

$$\nabla P_{\mathcal{K}^m}(\overline{z}) = \begin{cases} I_m & (\overline{\lambda}_1 > 0,\ \overline{\lambda}_2 > 0), \\ \dfrac{\overline{\lambda}_2}{\overline{\lambda}_2 - \overline{\lambda}_1} I_m + W & (\overline{\lambda}_1 < 0,\ \overline{\lambda}_2 > 0), \\ O & (\overline{\lambda}_1 < 0,\ \overline{\lambda}_2 < 0), \end{cases}$$

where

$$W := \frac{1}{2}\begin{pmatrix} -r_1 & r_2^\top \\ r_2 & -r_1 r_2 r_2^\top \end{pmatrix}, \ (r_1, r_2) := \frac{(\overline{z}_1, \overline{z}_2)}{\|\overline{z}_2\|}, \ \overline{z} := (\overline{z}_1, \overline{z}_2) \in \mathbb{R} \times \mathbb{R}^{m-1}.$$

On the other hand, by [18, Proposition 5.2], $\nabla P_{\mu_k}(z^k)$ is written as

$$\nabla P_{\mu_k}(z^k) = \begin{cases} \hat{g}'(z_1^k/\mu_k) I_m & (z_2^k = 0), \\ \begin{pmatrix} b_{\mu_k} & \dfrac{c_{\mu_k}(z_2^k)^\top}{\|z_2^k\|} \\ \dfrac{c_{\mu_k} z_2^k}{\|z_2^k\|} & a_{\mu_k} I_{m-1} + (b_{\mu_k} - a_{\mu_k}) \dfrac{z_2^k {z_2^k}^\top}{\|z_2^k\|^2} \end{pmatrix} & (z_2^k \neq 0), \end{cases}$$

where

$$a_{\mu_k} = \frac{\hat{g}(\lambda_2^k/\mu_k) - \hat{g}(\lambda_1^k/\mu_k)}{\lambda_2^k/\mu_k - \lambda_1^k/\mu_k}, \ b_{\mu_k} = \frac{1}{2}\left(\hat{g}'\left(\frac{\lambda_2^k}{\mu_k}\right) + \hat{g}'\left(\frac{\lambda_1^k}{\mu_k}\right)\right),$$

$$c_{\mu_k} = \frac{1}{2}\left(\hat{g}'\left(\frac{\lambda_2^k}{\mu_k}\right) - \hat{g}'\left(\frac{\lambda_1^k}{\mu_k}\right)\right), \ z^k := (z_1^k, z_2^k) \in \mathbb{R} \times \mathbb{R}^{m-1},$$

and $\hat{g}$ is defined as in Definition 5.4.3. Note that, from the definition of $\hat{g}$, we have $\hat{g}(\alpha) - \alpha \to 0$, $\hat{g}(-\alpha) \to 0$, $\hat{g}'(\alpha) \to 1$ and $\hat{g}'(-\alpha) \to 0$ as $\alpha \to \infty$. Then, it follows that

$$\lim_{k\to\infty} a_{\mu_k} = \begin{cases} 1 & (0 < \overline{\lambda}_1 \leq \overline{\lambda}_2) \\ \overline{\lambda}_2/(\overline{\lambda}_2 - \overline{\lambda}_1) & (\overline{\lambda}_1 < 0 < \overline{\lambda}_2) \\ 0 & (\overline{\lambda}_1 \leq \overline{\lambda}_2 < 0), \end{cases}$$

$$\lim_{k\to\infty} b_{\mu_k} = \begin{cases} 1 & (0 < \overline{\lambda}_1 \leq \overline{\lambda}_2) \\ 1/2 & (\overline{\lambda}_1 < 0 < \overline{\lambda}_2) \\ 0 & (\overline{\lambda}_1 \leq \overline{\lambda}_2 < 0), \end{cases}$$

and

$$\lim_{k\to\infty} c_{\mu_k} = \begin{cases} 0 & (0 < \overline{\lambda}_1 \leq \overline{\lambda}_2) \\ 1/2 & (\overline{\lambda}_1 < 0 < \overline{\lambda}_2) \\ 0 & (\overline{\lambda}_1 \leq \overline{\lambda}_2 < 0). \end{cases}$$

From this fact, it is not difficult to observe that $\nabla P_{\mathcal{K}^m}(\overline{z}) = \lim_{k\to\infty} \nabla P_{\mu_k}(z^k)$. □

Proposition 5.4.6. [18, Proposition 5.1] *Let $\Phi : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m$ and $\Phi_\mu : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m$ be defined by* (2.4.1) *and* (5.4.3)*, respectively. Let $\rho := \hat{g}(0)$. Then, for any y, $z \in \mathbb{R}^n$ and $\mu > \nu > 0$, we have*

$$\rho(\mu - \nu)e \succeq_{\mathcal{K}} \Phi_\nu(y, z) - \Phi_\mu(y, z) \succ_{\mathcal{K}} 0,$$
$$\rho\mu e \succeq_{\mathcal{K}} \Phi(y, z) - \Phi_\mu(y, z) \succ_{\mathcal{K}} 0,$$

where $e := (e^1, e^2, \ldots, e^\ell) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \cdots \times \mathbb{R}^{m_\ell}$ with $e^i := (1, 0, 0, \ldots, 0)^\top \in \mathbb{R}^{m_i}$ for $i = 1, 2, \ldots, \ell$.

Proposition 5.4.7. [18, Corollary 5.3 and Proposition 6.1] *Let $\Phi : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m$* and Φ_μ : $\mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m$ *be defined by* (2.4.1) *and* (5.4.3)*, respectively. Let $g : \mathbb{R}^m \to \mathbb{R}^m$ be defined by* (5.4.2)*. Then, the following statements hold.*

(a) *Function g is continuously differentiable and $\nabla g(z) = \mathrm{diag}(\nabla g^1(z^1), \dots, \nabla g^\ell(z^\ell))) \in \mathbb{R}^{m\times m}$ is symmetric for any $z \in \mathbb{R}^m$, where the latter matrix denotes the block-diagonal matrix with block-diagonal elements $\nabla g^i(z^i)$, $i = 1, 2, \dots, \ell$.*

(b) *For any $y, z \in \mathbb{R}^m$, we have*

$$\nabla_y \Phi_\mu(y, z) = I_m - \nabla g\left(\frac{y - z}{\mu}\right), \quad \nabla_z \Phi_\mu(y, z) = \nabla g\left(\frac{y - z}{\mu}\right),$$

where $I_m \in \mathbb{R}^{m\times m}$ denotes the identity matrix.

(c) *For any $y, z \in \mathbb{R}^m$, we have*

$$O \prec \nabla_y \Phi_\mu(y, z) \prec I_m, \quad O \prec \nabla_z \Phi_\mu(y, z) \prec I_m, \quad O \prec \nabla g\left(\frac{y - z}{\mu}\right) \prec I_m.$$

Proposition 5.4.8. *Let $\Phi : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m$ and $\Phi_\mu : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m$ be defined by* (2.4.1) *and* (5.4.3)*, respectively. Let $\rho := \hat{g}(0)$. Then, $\|\Phi_\mu(y, z) - \Phi(y, z)\| \le \sqrt{2}\rho\mu$ for any $(\mu, y, z) \in \mathbb{R}_{++} \times \mathbb{R}^m \times \mathbb{R}^m$.*

Proof. For simplicity, we only consider the case where $\mathcal{K} = \mathcal{K}^m$. Let $\Phi(y, z) - \Phi_\mu(y, z) = \lambda_1 c^1 + \lambda_2 c^2$, where $\lambda_i \in \mathbb{R}$ and $c^i \in \mathbb{R}^m$ $(i = 1, 2)$ are the spectral values and spectral vectors of $\Phi(y, z) - \Phi_\mu(y, z)$. Since $e = c^1 + c^2$ and $\rho\mu e \succeq_{\mathcal{K}^m} \Phi(y, z) - \Phi_\mu(y, z) \succ_{\mathcal{K}^m} 0$ from Proposition 5.4.6, we have $\rho\mu(c^1 + c^2) \succeq_{\mathcal{K}^m} \lambda_1 c^1 + \lambda_2 c^2 \succ_{\mathcal{K}^m} 0$, which implies $0 < \lambda_1 \le \lambda_2 \le \rho\mu$. Hence, we obtain

$$\|\Phi(y, z) - \Phi_\mu(y, z)\| = \|\lambda_1 c^1 + \lambda_2 c^2\| \le \lambda_1\|c^1\| + \lambda_2\|c^2\| \le \sqrt{2}\rho\mu,$$

where the first inequality is due to the triangle inequality and $0 < \lambda_1 \le \lambda_2$, and the last inequality follows from $\|c^1\| = \|c^2\| = 1/\sqrt{2}$ and $\lambda_1 \le \lambda_2 \le \rho\mu$. This completes the proof. ☐

Proposition 5.4.9. *Let $\Phi_\mu : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m$ be defined by* (5.4.3)*. Then, for any $\mu > \nu > 0$ and $(y, z) \in \mathbb{R}^m \times \mathbb{R}^m$, it holds that*

$$\|\Phi_\nu(y, z)\|_1 - \|\Phi_\mu(y, z)\|_1 \le m\rho(\mu - \nu),$$

where $\rho = \hat{g}(0)$.

Proof. We first assume $\mathcal{K} = \mathcal{K}^m$. From Proposition 5.4.6, we have

$$\rho(\mu - \nu)e - (\Phi_\nu(y, z) - \Phi_\mu(y, z)) \in \mathcal{K}^m, \tag{5.4.5}$$

$$\Phi_\nu(y, z) - \Phi_\mu(y, z) \in \mathcal{K}^m, \tag{5.4.6}$$

where $e = (1, 0, \dots, 0)^\top \in \mathbb{R}^m$. Moreover, for any $w = (w_1, w_2, \dots, w_m)^\top \in \mathcal{K}^m$, we have

$$w_1 \ge |w_i| \quad (i = 1, \dots, m), \tag{5.4.7}$$

since $w_1 \geq \sqrt{w_2^2 + \cdots + w_m^2}$. Therefore, for each $i = 1, 2, \ldots, m$, we have

$$\begin{aligned} \rho(\mu - \nu) &\geq \big(\Phi_\nu(y, z) - \Phi_\mu(y, z)\big)_1 \\ &\geq \big|\big(\Phi_\nu(y, z) - \Phi_\mu(y, z)\big)_i\big| \\ &\geq |\Phi_\nu(y, z)_i| - |\Phi_\mu(y, z)_i|, \end{aligned} \tag{5.4.8}$$

where the first inequality holds from (5.4.3) and (5.4.5), the second equality holds from (5.4.6) and (5.4.7), and the last equality holds from the triangle inequality. Summing up (5.4.8) for all i, we obtain the desired conclusion. When $\mathcal{K} = \mathcal{K}^{m_1} \times \cdots \times \mathcal{K}^{m_\ell}$, we can prove it in a similar way. □

5.5 Algorithm

In this section, we propose an SQP type algorithm for MPSOCC (5.1.1). The SQP method solves a quadratic programming (QP) problem in each iteration to determine the search direction. This method is known as one of the most efficient methods for solving nonlinear programming problems. In the remainder of the chapter, to apply the SQP method, we mainly consider MPSOCC (5.3.1) equivalent to MPSOCC (5.1.1). We should notice, however, that the SQP method cannot be applied directly to MPSOCC (5.3.1), since $\Phi(y, z)$ is not differentiable everywhere. We thus consider the following problem where the smooth equality constraint $\Phi_\mu(y, z) = 0$ replaces $\Phi(y, z) = 0$ in each iteration

$$\begin{array}{ll} \underset{x,y,z}{\text{Minimize}} & f(x, y) \\ \text{subject to} & Ax \leq b, \\ & z = Nx + My + q, \\ & \Phi_\mu(y, z) = 0. \end{array} \tag{5.5.1}$$

Given a current iterate (x^k, y^k, z^k) satisfying $Ax^k \leq b^k$ and $z^k = Nx^k + My^k + q$, we then generate the search direction (dx^k, dy^k, dz^k) by solving the following QP subproblem, which consists of quadratic and linear approximations of the objective and constraint functions of problem (5.5.1) with $\mu = \mu_k$, respectively, at (x^k, y^k, z^k):

$$\begin{array}{ll} \underset{dx,dy,dz}{\text{Minimize}} & \nabla f(x^k, y^k)^\top \begin{pmatrix} dx \\ dy \end{pmatrix} + \dfrac{1}{2} \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix}^\top B_k \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} \\ \text{subject to} & Adx \leq b - Ax^k, \\ & \begin{pmatrix} N & M & -I_m \\ 0 & \nabla_y \Phi_{\mu_k}(y^k, z^k)^\top & \nabla_z \Phi_{\mu_k}(y^k, z^k)^\top \end{pmatrix} \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = -\begin{pmatrix} 0 \\ \Phi_{\mu_k}(y^k, z^k) \end{pmatrix}, \end{array} \tag{5.5.2}$$

where $B_k \in \mathbb{R}^{(n+2m)\times(n+2m)}$ is a positive definite symmetric matrix. In the numerical experiments in Section 5.7, B_k will be updated by using the modified Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula. Note that the Karush-Kuhn-Tucker (KKT) conditions of QP (5.5.2) can be written as

$$
\begin{aligned}
&\begin{pmatrix} \nabla_x f(x^k, y^k) \\ \nabla_y f(x^k, y^k) \\ 0 \end{pmatrix} + B_k \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} + \begin{pmatrix} N^\top \\ M^\top \\ -I_m \end{pmatrix} u + \begin{pmatrix} 0 \\ \nabla_y \Phi_{\mu_k}(y^k, z^k) \\ \nabla_z \Phi_{\mu_k}(y^k, z^k) \end{pmatrix} v + \begin{pmatrix} A^\top \\ 0 \\ 0 \end{pmatrix} \eta = 0, \\
&\begin{pmatrix} N & M & -I_m \\ 0 & \nabla_y \Phi_{\mu_k}(y^k, z^k)^\top & \nabla_z \Phi_{\mu_k}(y^k, z^k)^\top \end{pmatrix} \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = - \begin{pmatrix} 0 \\ \Phi_{\mu_k}(y^k, z^k) \end{pmatrix}, \\
&0 \le (b - Ax^k - Adx) \perp \eta \ge 0,
\end{aligned}
\tag{5.5.3}
$$

where $(\eta, u, v) \in \mathbb{R}^p \times \mathbb{R}^m \times \mathbb{R}^m$ denotes the Lagrange multipliers.

For simplicity of notation, we denote

$$w := (x, y, z) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m, \ dw := (dx, dy, dz) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m.$$

Also, we define the ℓ_1 penalty function by

$$\theta_{\mu,\alpha}(w) := f(x, y) + \alpha \|\Phi_\mu(y, z)\|_1, \tag{5.5.4}$$

where $\alpha > 0$ is the penalty parameter. Note that this function has the directional derivative $\theta'_{\mu,\alpha}(w; dw)$ for any w and dw.

Algorithm 5.1

Step 0: Choose parameters $\delta \in (0, \infty)$, $\beta \in (0, 1)$, $\rho \in (0, 1)$, $\sigma \in (0, 1)$, $\mu_0 \in (0, \infty)$, $\alpha_{-1} \in (0, \infty)$ and a symmetric positive definite matrix $B_0 \in \mathbb{R}^{(n+2m)\times(n+2m)}$. Choose $w^0 = (x^0, y^0, z^0) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$ such that $Nx^0 + My^0 + q = z^0$ and $Ax^0 \le b$. Set $k := 0$.

Step 1: Solve QP subproblem (5.5.2) to obtain the optimum $dw^k = (dx^k, dy^k, dz^k)$ and the Lagrange multipliers (η^k, u^k, v^k).

Step 2: If $dw^k = 0$, then let $w^{k+1} := w^k$, $\alpha_k := \alpha_{k-1}$ and go to Step 3. Otherwise, update the penalty parameter by

$$\alpha_k := \begin{cases} \alpha_{k-1} & \text{if } \ \alpha_{k-1} \ge \|v^k\|_\infty + \delta, \\ \max\{\|v^k\|_\infty + \delta, \ \alpha_{k-1} + 2\delta\} & \text{otherwise.} \end{cases} \tag{5.5.5}$$

Then, set the step size $\tau_k := \rho^L$, where L is the smallest nonnegative integer satisfying the Armijo condition

$$\theta_{\mu_k,\alpha_k}(w^k + \rho^L dw^k) \le \theta_{\mu_k,\alpha_k}(w^k) + \sigma \rho^L \theta'_{\mu_k,\alpha_k}(w^k; dw^k). \tag{5.5.6}$$

Let $w^{k+1} := w^k + \tau_k dw^k$, and go to Step 3.

Step 3: Terminate if a certain criterion is satisfied. Otherwise, let $\mu_{k+1} := \beta\mu_k$ and update B_k to determine a symmetric positive definite matrix B_{k+1}. Return to Step 1 with k replaced by $k + 1$.

In the remainder of this section, we establish the well-definedness of Algorithm 5.1. We first show the feasibility of QP subproblem (5.5.2). In general, a QP subproblem generated by the SQP method may not be feasible, even if the original nonlinear programming problem is feasible. However, in the present case, we can show that QP subproblem (5.5.2) is always feasible under the Cartesian P_0 property of the matrix M. To this end, the following lemma will be useful.

Lemma 5.5.1. *Let $M \in \mathbb{R}^{m\times m}$ be a Cartesian P_0 matrix. Let $H_i \in \mathbb{R}^{m_i\times m_i}$ $(i = 1, 2, \ldots, \ell)$ be positive definite matrices with $m = \sum_{i=1}^{\ell} m_i$, and $H \in \mathbb{R}^{m\times m}$ be a block diagonal matrix with block diagonal elements H_i $(i = 1, \ldots, \ell)$. Then, $H + M$ is nonsingular.*

Proof. The matrix $H+M$ can easily be shown to be a Cartesian P matrix, which is nonsingular. □

The next proposition shows the feasibility and solvability of QP subproblem (5.5.2). In the proof, the matrix

$$D_k := \begin{pmatrix} M & -I_m \\ \nabla_y \Phi_{\mu_k}(y^k, z^k)^\top & \nabla_z \Phi_{\mu_k}(y^k, z^k)^\top \end{pmatrix} \tag{5.5.7}$$

plays an important role.

Proposition 5.5.2 (Feasibility of QP subproblem)**.** *Let M be a Cartesian P_0 matrix, and $\{w^k\}$ be a sequence generated by Algorithm* 5.1*. Then,* (i) *$Ax^k \le b$ and $z^k = Nx^k + My^k + q$ hold for all k, and* (ii) *QP subproblem* (5.5.2) *is feasible and hence has a unique solution for all k.*

Proof. Since (i) can be shown easily, we only show (ii). Since the objective function of QP (5.5.2) is strongly convex, it suffices to show the feasibility. We first show that the matrix D_k defined by (5.5.7) is nonsingular. Note that, by Proposition 5.4.7(c), $\nabla_z \Phi_{\mu_k}(y^k, z^k)$ is nonsigular. Let $\tilde{D}_k$ be the Schur complement of the matrix $\nabla_z \Phi_{\mu_k}(y^k, z^k)^\top$ with respect to D_k, that is,

$$\begin{aligned} \tilde{D}_k &:= M + \left(\nabla_z \Phi_{\mu_k}(y^k, z^k)^\top\right)^{-1} \nabla_y \Phi_\mu(y^k, z^k)^\top \\ &= M + \left(\nabla g \left(\frac{y^k - z^k}{\mu_k}\right)^\top\right)^{-1} \left(I_m - \nabla g \left(\frac{y^k - z^k}{\mu_k}\right)^\top\right) \\ &= M + \operatorname{diag}\left(\left(\nabla g^i \left(\frac{y^{i,k} - z^{i,k}}{\mu_k}\right)^\top\right)^{-1} - I_{m_i}\right)_{i=1}^{\ell}, \end{aligned}$$

where $y^{i,k}$ and $z^{i,k}$ denote the i-th subvectors of y^k and z^k, respectively, conforming to the Cartesian structure of $\mathcal{K}$, and each equality follows from Proposition 5.4.7. Since M is a Cartesian P_0 matrix and $(\nabla g^i((y^{i,k} - z^{i,k})/\mu_k)^\top)^{-1} - I_{m_i} \in \mathbb{R}^{m_i\times m_i}$ is positive definite from Proposition 5.4.7 (c), Lemma 5.5.1 ensures that $\tilde{D}_k$ is nonsingular, and hence D_k is nonsingular. It then follows from (i) that

$$dx = 0, \qquad \begin{pmatrix} dy \\ dz \end{pmatrix} = -D_k^{-1} \begin{pmatrix} 0 \\ \Phi_{\mu_k}(y^k, z^k) \end{pmatrix}$$

comprise a feasible solution to (5.5.2). This completes the proof. □

The following proposition shows that the search direction dw^k produced in Step 1 of Algorithm 5.1 is a descent direction of the penalty function θ_{μ_k,α_k} defined by (5.5.4). It guarantees the well-definedness of the line search in Step 2 in the sense that there exists a finite L satisfying the Armijo condition (5.5.6).

Proposition 5.5.3 (Descent direction). *Let $\{w^k\}$ and $\{dw^k\}$ be sequences generated by Algorithm 5.1. Then, we have*

(a) $\theta'_{\mu_k,\alpha_k}(w^k; dw^k) = \nabla_x f(x^k, y^k)^\top dx^k + \nabla_y f(x^k, y^k)^\top dy^k - \alpha_k \|\Phi_{\mu_k}(y^k, z^k)\|_1$,

(b) $\theta'_{\mu_k,\alpha_k}(w^k; dw^k) \leq -(dw^k)^\top B_k dw^k$

for each k. Moreover, if $\Phi_{\mu_k}(y^k, z^k) \neq 0$, then the inequality in (b) *holds strictly.*

Proof. We first show (a). Let J_+^k, J_0^k, and $J_-^k \subseteq \{1, 2, \ldots, m\}$ be the index sets defined by

$$
\begin{aligned}
J_+^k &= \{j \mid \Phi_{\mu_k}(y^k, z^k)_j > 0\}, \\
J_0^k &= \{j \mid \Phi_{\mu_k}(y^k, z^k)_j = 0\}, \\
J_-^k &= \{j \mid \Phi_{\mu_k}(y^k, z^k)_j < 0\},
\end{aligned}
$$

where $\Phi_{\mu_k}(y^k, z^k)_j \in \mathbb{R}$ denotes the j-th component of $\Phi_{\mu_k}(y^k, z^k) \in \mathbb{R}^m$. Then, we have

$$
\begin{aligned}
\theta'_{\mu_k,\alpha_k}(w^k;\, dw^k) \;=\;& \nabla f(x^k, y^k)^\top \begin{pmatrix} dx^k \\ dy^k \end{pmatrix} + \alpha_k \sum_{j \in J_+^k} \left[\nabla \Phi_{\mu_k}(y^k, z^k)\right]_j^\top \begin{pmatrix} dy^k \\ dz^k \end{pmatrix} \\
&+ \alpha_k \sum_{j \in J_0^k} \left| \left[\nabla \Phi_{\mu_k}(y^k, z^k)\right]_j^\top \begin{pmatrix} dy^k \\ dz^k \end{pmatrix} \right| - \alpha_k \sum_{j \in J_-^k} \left[\nabla \Phi_{\mu_k}(y^k, z^k)\right]_j^\top \begin{pmatrix} dy^k \\ dz^k \end{pmatrix}, \qquad (5.5.8)
\end{aligned}
$$

where $[\nabla \Phi_{\mu_k}(y^k, z^k)]_j$ denotes the j-th column vector of $\nabla \Phi_{\mu_k}(y^k, z^k)$. Since

$$
\left[\nabla \Phi_{\mu_k}(y^k, z^k)\right]_j^\top \begin{pmatrix} dy^k \\ dz^k \end{pmatrix} = -\Phi_{\mu_k}(y^k, z^k)_j
$$

from the constraint of QP subproblem (5.5.2), we have

$$
\theta'_{\mu_k,\alpha_k}(w^k; dw^k) = \nabla_x f(x^k, y^k)^\top dx^k + \nabla_y f(x^k, y^k)^\top dy^k - \alpha_k \|\Phi_{\mu_k}(y^k, z^k)\|_1. \qquad (5.5.9)
$$

We next show (b). Taking the inner product of $dw^k = (dx^k, dy^k, dz^k)$ and both sides of the first equality in the KKT conditions (5.5.3) with $dw = dw^k$, $\eta = \eta^k$, $u = u^k$, $v = v^k$ for the subproblem (5.5.2), we obtain

$$
\begin{aligned}
&\nabla f(x^k, y^k)^\top \begin{pmatrix} dx^k \\ dy^k \end{pmatrix} + (dw^k)^\top B_k dw^k + (u^k)^\top (N dx^k + M dy^k - dz^k) \\
&+ (v^k)^\top \nabla \Phi_{\mu_k}(y^k, z^k) \begin{pmatrix} dy^k \\ dz^k \end{pmatrix} + (\eta^k)^\top A dx^k = 0. \qquad (5.5.10)
\end{aligned}
$$

Moreover, from the constraints of the subproblem (5.5.2) and the KKT conditions (5.5.3), we have

$$Ndx^k + Mdy^k - dz^k = 0, \tag{5.5.11}$$

$$\nabla\Phi_{\mu_k}(y^k, z^k)^\top \begin{pmatrix} dy^k \\ dz^k \end{pmatrix} = -\Phi_{\mu_k}(y^k, z^k), \tag{5.5.12}$$

and

$$0 = (\eta^k)^\top(b - Ax^k - Adx^k) = -(\eta^k)^\top Adx^k + (\eta^k)^\top(b - Ax^k) \geq -(\eta^k)^\top Adx^k, \tag{5.5.13}$$

where the inequality is due to $\eta^k \geq 0$ and $b - Ax^k \geq 0$ from (5.5.3). Substituting (5.5.11)–(5.5.13) into (5.5.10), we have

$$\nabla f(x^k, y^k)^\top \begin{pmatrix} dx^k \\ dy^k \end{pmatrix} + (dw^k)^\top B_k dw^k - (v^k)^\top \Phi_{\mu_k}(y^k, z^k) \leq 0.$$

Furthermore, from (5.5.9), we obtain

$$\begin{aligned} \theta'_{\mu_k,\alpha_k}(w^k; dw^k) &\leq -(dw^k)^\top B_k dw^k + (v^k)^\top \Phi_{\mu_k}(y^k, z^k) - \alpha_k \|\Phi_{\mu_k}(y^k, z^k)\|_1 \\ &= -(dw^k)^\top B_k dw^k + \sum_{j \in J_+^k} (v_j^k - \alpha_k) \left[\Phi_{\mu_k}(y^k, z^k)\right]_j \\ &\quad + \sum_{j \in J_-^k} (v_j^k + \alpha_k) \left[\Phi_{\mu_k}(y^k, z^k)\right]_j \\ &\leq -(dw^k)^\top B_k dw^k, \end{aligned}$$

where the last inequality follows from $\alpha_k > \|v^k\|_\infty$ and the definitions of J_+^k and J_-^k. Moreover, if $\Phi_{\mu_k}(y^k, z^k) \neq 0$, then the last inequality holds strictly since $J_+^k \cup J_-^k \neq \emptyset$. This completes the proof of (b). □

5.6 Convergence analysis

In this section, we study the convergence property of the proposed algorithm. To begin with, we make the following assumption.

Assumption 5.6.1. *Let sequences $\{w^k\}$ and $\{B_k\}$ be produced by Algorithm* 5.1*.*

(a) *$\{w^k\}$ is bounded.*

(b) *There exist constants $\gamma_1, \gamma_2 > 0$ such that $\gamma_1\|d\|^2 \leq d^\top B_k d \leq \gamma_2\|d\|^2$ for any $d \in \mathbb{R}^{n+2m}$ and k.*

(c) *There exists a constant $c > 0$ such that $\|D_k^{-1}\| \leq c$ for any k, where D_k is the matrix defined by* (5.5.7)*.*

Assumption 5.6.1 (b) means that $\{B_k\}$ is bounded and uniformly positive definite. Assumption 5.6.1 (c) holds if and only if any accumulation point of $\{D_k\}$ is nonsingular. The next proposition provides a sufficient condition under which Assumption 5.6.1 (c) holds.

Proposition 5.6.2. *Suppose that M is a Cartesian P matrix and Assumption* 5.6.1 (a) *holds. Then, Assumption* 5.6.1 (c) *holds.*

Proof. Let $(\overline{E}_y, \overline{E}_z)$ be an arbitrary accumulation point of $\{(\nabla_y \Phi_{\mu_k}(y^k, z^k), \nabla_z \Phi_{\mu_k}(y^k, z^k))\}$. Then, it suffices to show that the matrix

$$D_\infty := \begin{pmatrix} M & -I_m \\ \overline{E}_y & \overline{E}_z \end{pmatrix}$$

is nonsingular. By Proposition 5.4.7, $\overline{E}_y$ and $\overline{E}_z$ satisfy the following three properties:

1. $\overline{E}_y + \overline{E}_z = I_m$;
2. $0 \preceq \overline{E}_y \preceq I_m$ and $0 \preceq \overline{E}_z \preceq I_m$;
3. $\overline{E}_y$ and $\overline{E}_z$ are symmetric and have the block-diagonal structure conforming to the Cartesian structure of $\mathcal{K} = \mathcal{K}^{m_1} \times \cdots \times \mathcal{K}^{m_\ell}$.

From (a) and (b), there exists an orthogonal matrix $H \in \mathbb{R}^{m \times m}$,

$$H\overline{E}_y H^\top = \mathrm{diag}(\alpha_i)_{i=1}^m, \; H\overline{E}_z H^\top = \mathrm{diag}(1-\alpha_i)_{i=1}^m, \quad 0 \le \alpha_i \le 1 \; (i = 1, 2, \ldots, m), \tag{5.6.1}$$

where α_i $(i = 1, 2, \ldots, m)$ are the eigenvalues of $\overline{E}_y$. Moreover, from (c), H has the same block-diagonal structure as $\overline{E}_y$ and $\overline{E}_z$. Hence, $\tilde{M} := HMH^\top$ is a Cartesian P matrix by Proposition 5.2.2.

Now, let $\tilde{D}_\infty \in \mathbb{R}^{2m \times 2m}$ be defined as

$$\tilde{D}_\infty := \begin{pmatrix} H & 0 \\ 0 & H \end{pmatrix} D_\infty \begin{pmatrix} H^\top & 0 \\ 0 & H^\top \end{pmatrix} = \begin{pmatrix} \tilde{M} & -I_m \\ \mathrm{diag}(\alpha_i)_{i=1}^m & \mathrm{diag}(1-\alpha_i)_{i=1}^m \end{pmatrix}, \tag{5.6.2}$$

and let $(\zeta, \eta) \in \mathbb{R}^m \times \mathbb{R}^m$ be an arbitrary vector such that $\tilde{D}_\infty \binom{\zeta}{\eta} = 0$. Then, we have

$$\tilde{M}\zeta = \eta, \tag{5.6.3}$$

$$\alpha_i \zeta_i + (1-\alpha_i)\eta_i = 0 \quad (i = 1, 2, \ldots, m). \tag{5.6.4}$$

If $\alpha_i = 0$, then we have $(\tilde{M}\zeta)_i = \eta_i = 0$ from (5.6.3) and (5.6.4). If $\alpha_i = 1$, then we have $\zeta_i = 0$ from (5.6.4). If $0 < \alpha_i < 1$, then we have $\zeta_i(\tilde{M}\zeta)_i = \zeta_i \eta_i = -\alpha_i(1-\alpha_i)^{-1}\zeta_i^2 \le 0$. Thus, for all i, we have $\zeta_i(\tilde{M}\zeta)_i \le 0$. Since $\tilde{M}$ is a Cartesian P matrix and every Cartesian P matrix is a P matrix, we must have $\zeta = \eta = 0$. Hence, $\tilde{D}_\infty$ is nonsingular. From (5.6.2) and the nonsingularity of H, matrix D_∞ is also nonsingular.

□

The following three lemmas play crucial roles in establishing the convergence theorem for the algorithm.

Lemma 5.6.3. *Let $\{w^k\}$ be a sequence generated by Algorithm* 5.1*, and $dw^k := (dx^k, dy^k, dz^k) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$ be the unique optimum of QP subproblem* (5.5.2) *for each k. Let $\{(\mu_k, \tau_k)\} \subseteq \mathbb{R}_{++} \times \mathbb{R}_{++}$ be a sequence converging to $(0, 0)$ and $\alpha > 0$ be a fixed scalar. Suppose that $\{w^k\}$*

satisfies Assumption 5.6.1(a). *In addition, assume that* $\{dw^k\}$ *has an accumulation point, and* $\overline{w} := (\overline{x}, \overline{y}, \overline{z})$ *and* $\overline{dw} := (\overline{dx}, \overline{dy}, \overline{dz})$ *are arbitrary accumulation points of* $\{w^k\}$ *and* $\{dw^k\}$, *respectively. Then we have*

$$\limsup_{k\to\infty} \left(\frac{\theta_{\mu_k,\alpha}(w^k + \tau_k dw^k) - \theta_{\mu_k,\alpha}(w^k)}{\tau_k} - \theta'_{\mu_k,\alpha}(w^k; dw^k) \right) \le 0$$

provided $\overline{y} - \overline{z} \notin \mathrm{bd}\,(\mathcal{K} \cup -\mathcal{K})$ *and* $\overline{dw} \neq 0$.

Proof. Taking subsequences if necessary, we may suppose $w^k \to \overline{w}$ and $dw^k \to \overline{dw}$. Moreover, we have $\theta'_{\mu_k,\alpha}(w^k; dw^k) = \nabla_x f(x^k, y^k)^\top dx^k + \nabla_y f(x^k, y^k)^\top dy^k - \alpha\|\Phi_{\mu_k}(y^k, z^k)\|_1$ as shown in (5.5.9). Thus, we have only to show

$$\limsup_{k\to\infty} \left(\frac{\theta_{\mu_k,\alpha}(w^k + \tau_k dw^k) - \theta_{\mu_k,\alpha}(w^k)}{\tau_k} \right) \le \nabla f(\overline{x}, \overline{y})^\top \begin{pmatrix} \overline{dx} \\ \overline{dy} \end{pmatrix} - \alpha\|\Phi(\overline{y}, \overline{z})\|_1.$$

From the mean-value theorem and the continuity of ∇f, we have

$$\lim_{k\to\infty} \frac{f(x^k + \tau_k dx^k, y^k + \tau_k dy^k) - f(x^k, y^k)}{\tau_k} = \nabla f(\overline{x}, \overline{y})^\top \begin{pmatrix} \overline{dx} \\ \overline{dy} \end{pmatrix}.$$

Therefore, it suffices to show that

$$\limsup_{k\to\infty} \frac{|\Phi_{\mu_k}(y^k + \tau_k dy^k, z^k + \tau_k dz^k)_j| - |\Phi_{\mu_k}(y^k, z^k)_j|}{\tau_k} \le -|\Phi(\overline{y}, \overline{z})_j| \tag{5.6.5}$$

for each j. By Definition 5.4.4 and Proposition 5.4.7, we have $\nabla_y \Phi_{\mu_k}(y^k, z^k) = I_m - \nabla P_{\mu_k}(y^k - z^k)$ and $\nabla_z \Phi_{\mu_k}(y^k, z^k) = \nabla P_{\mu_k}(y^k - z^k)$, which together with the constraints of QP subproblem (5.5.2) yield

$$\begin{aligned} -\Phi_{\mu_k}(y^k, z^k) &= \nabla_y \Phi_{\mu_k}(y^k, z^k)^\top dy^k + \nabla_z \Phi_{\mu_k}(y^k, z^k)^\top dz^k \\ &= (I - \nabla P_{\mu_k}(y^k - z^k)^\top) dy^k + \nabla P_{\mu_k}(y^k - z^k)^\top dz^k \\ &= dy^k - \nabla P_{\mu_k}(y^k - z^k)^\top (dy^k - dz^k). \end{aligned} \tag{5.6.6}$$

Hence, we have

$$\begin{aligned} & \Phi_{\mu_k}(y^k + \tau_k dy^k, z^k + \tau_k dz^k) - \Phi_{\mu_k}(y^k, z^k) \\ = \;& \tau_k dy^k - \Big(P_{\mu_k}(y^k + \tau_k dy^k - (z^k + \tau_k dz^k)) - P_{\mu_k}(y^k - z^k) \Big) \\ = \;& \tau_k \left(-\Phi_{\mu_k}(y^k, z^k) + \nabla P_{\mu_k}(y^k - z^k)^\top (dy^k - dz^k) \right) \\ & - P_{\mu_k}(y^k - z^k + \tau_k(dy^k - dz^k)) + P_{\mu_k}(y^k - z^k) \\ = \;& -\tau_k \Phi_{\mu_k}(y^k, z^k) + \tau_k \delta^k, \end{aligned} \tag{5.6.7}$$

where the first equality is due to (5.4.3), the second equality follows from (5.6.6), and $\delta^k := (\delta_1^k, \delta_2^k, \ldots, \delta_m^k) \in \mathbb{R}^m$ is given by

$$\delta_j^k := \nabla P_{\mu_k}(y^k - z^k)_j^\top (dy^k - dz^k) - \tau_k^{-1} \Big(P_{\mu_k}(y^k - z^k + \tau_k(dy^k - dz^k))_j - P_{\mu_k}(y^k - z^k)_j \Big). \tag{5.6.8}$$

To show (5.6.5), we consider three cases: (i) $\Phi(\overline{y},\overline{z})_j = 0$, (ii) $\Phi(\overline{y},\overline{z})_j > 0$ and (iii) $\Phi(\overline{y},\overline{z})_j < 0$. In case (i), we first notice that

$$\frac{|\Phi_{\mu_k}(y^k+\tau_k dy^k, z^k+\tau_k dz_k)_j| - |\Phi_{\mu_k}(y^k,z^k)_j|}{\tau_k} \le \frac{|\Phi_{\mu_k}(y^k+\tau_k dy^k, z^k+\tau_k dz_k)_j - \Phi_{\mu_k}(y^k,z^k)_j|}{\tau_k}$$
$$= |\Phi_{\mu_k}(y^k,z^k)_j - \delta_j^k|, \tag{5.6.9}$$

where the equality follows from (5.6.7). By applying the mean-value theorem in (5.6.8), we can find $\zeta_{kj} \in (0,1)$ for each k such that

$$\delta_j^k = \left(\nabla P_{\mu_k}(y^k - z^k) - \nabla P_{\mu_k}\left(y^k - z^k + \zeta_{kj}\tau_k(dy^k - dz^k)\right)\right)_j^\top (dy^k - dz^k). \tag{5.6.10}$$

Since Proposition 5.4.5 and the boundedness of $\{dw^k\}$ imply

$$\lim_{k\to\infty} \nabla P_{\mu_k}(y^k - z^k) - \nabla P_{\mu_k}(y^k - z^k + \zeta_{kj}\tau_k(dy^k - dz^k)) = 0, \tag{5.6.11}$$

we obtain $\lim_{k\to\infty}\delta_j^k = 0$. Moreover, by Proposition 5.4.8, we have $\lim_{k\to\infty}\Phi_{\mu_k}(y^k,z^k)_j = \Phi(\overline{y},\overline{z})_j = 0$. Then, by letting $k\to\infty$ in (5.6.9), we obtain (5.6.5). In case (ii) and case (iii), we have

$$|\Phi_{\mu_k}(y^k+\tau_k dy^k, z^k+\tau_k dz^k)_j| - |\Phi_{\mu_k}(y^k,z^k)_j| = -\Phi_{\mu_k}(y^j,z^k)_j + \delta_j^k$$

and

$$|\Phi_{\mu_k}(y^k+\tau_k dy^k, z^k+\tau_k dz^k)_j| - |\Phi_{\mu_k}(y^k,z^k)_j| = \Phi_{\mu_k}(y^j,z^k)_j - \delta_j^k,$$

respectively. Then, a similar argument to that in case (i) leads to the desired inequality (5.6.5). □

Lemma 5.6.4. *Let $\{w^k\}$ be a sequence generated by Algorithm* 5.1*. Suppose that Assumption* 5.6.1 *holds. Then, we have the following statements.*

(i) *$\{dw^k\}$ and $\{(u^k,v^k)\}$ are bounded.*

(ii) *There exists k_0 such that $\alpha_k = \alpha_{k_0}$ for all $k \ge k_0$.*

(iii) *The sequences $\{\theta_{\mu_k,\alpha_k}(w^k)\}$ and $\{\theta_{\mu_k,\alpha_k}(w^{k+1})\}$ converge to the same limit.*

Proof. We first prove (i). Let

$$\begin{pmatrix} d\tilde{y}^k \\ d\tilde{z}^k \end{pmatrix} := -D_k^{-1}\begin{pmatrix} 0 \\ \Phi_{\mu_k}(y^k,z^k) \end{pmatrix} \tag{5.6.12}$$

and $d\tilde{w}^k := (0,\ d\tilde{y}^k,\ d\tilde{z}^k)$, where D_k is defined by (5.5.7). Then, $d\tilde{w}^k$ is a feasible point of QP subproblem (5.5.2). Note that the objective function of QP subproblem (5.5.2) is rewritten as

$$\frac{1}{2}(dw - B_k^{-1}g^k)^\top B_k(dw - B_k^{-1}g^k) + \text{constant},$$

where $g^k := (\nabla f(x^k,y^k),0)$. Since dw^k is the optimum of QP subproblem (5.5.2), we have

$$(d\tilde{w}^k - B_k^{-1}g^k)^\top B_k(d\tilde{w}^k - B_k^{-1}g^k) \ge (dw^k - B_k^{-1}g^k)^\top B_k(dw^k - B_k^{-1}g^k).$$

This together with Assumption 5.6.1 (b) implies

$$\gamma_2 \|d\tilde{w}^k - B_k^{-1} g^k\|^2 \geq \gamma_1 \|dw^k - B_k^{-1} g^k\|^2. \tag{5.6.13}$$

Now, notice that $\{d\tilde{w}^k\}$ is bounded from (5.6.12), Assumption 5.6.1 (a), (c) and Proposition 5.4.8. In addition, $\{B_k^{-1}\}$ and $\{g^k\}$ are also bounded from Assumption 5.6.1 (a), (b). Thus, we have the boundedness of $\{dw^k\}$ from (5.6.13). On the other hand, from the first equality of the KKT conditions (5.5.3), we have

$$\begin{pmatrix} u^k \\ v^k \end{pmatrix} = -(D_k^\top)^{-1} \left(\begin{pmatrix} \nabla_y f(x^k, y^k) \\ 0 \end{pmatrix} + \tilde{B}_k dw^k \right),$$

where $\tilde{B}_k$ is the $2m \times (n+2m)$ matrix consisting of the last $2m$ rows of B_k. This equation together with Assumption 5.6.1 and the boundedness of $\{dw^k\}$ yields the boundedness of $\{(u^k, v^k)\}$.

We next prove (ii). From the update rule (5.5.5), we can easily see that $\{\alpha_k\}$ is nondecreasing. Moreover, if

$$\|v^k\|_\infty > \alpha_{k-1} - \delta, \tag{5.6.14}$$

then we have $\alpha_k = \max\{\|v^k\|_\infty + \delta, \alpha_{k-1} + 2\delta\} \geq \alpha_{k-1} + 2\delta$, that is, α_k increases at least by 2δ at a time. Let $\hat{K} := \{k \,|\, \|v^k\|_\infty > \alpha_{k-1} - \delta\}$. If $|\hat{K}| = \infty$, then $\alpha_k \to \infty$ as $k \to \infty$ and hence $\{\|v^k\|_\infty\}$ is unbounded from (5.6.14). However this contradicts (i). Thus we have (ii).

We finally show (iii). Since we have (ii), there exist $\overline{\alpha}$ and k_0 such that $\overline{\alpha} = \alpha_k$ for all $k \geq k_0$. In what follows, we suppose $k \geq k_0$. Since $\mu_{k+1} \leq \mu_k$, Proposition 5.4.9 together with (5.5.4) implies

$$\begin{aligned} \theta_{\mu_{k+1},\overline{\alpha}}(w^{k+1}) + \overline{\alpha} m \rho \mu_{k+1} &\leq \theta_{\mu_k,\overline{\alpha}}(w^{k+1}) + \overline{\alpha} m \rho \mu_k && (5.6.15) \\ &\leq \theta_{\mu_k,\overline{\alpha}}(w^k) + \overline{\alpha} m \rho \mu_k, && (5.6.16) \end{aligned}$$

where the last inequality follows from the Armijo condition (5.5.6) and Proposition 5.5.3 (b). From (5.6.16), $\{\theta_{\mu_k,\overline{\alpha}}(w^k) + \overline{\alpha} m \rho \mu_k\}$ is a monotonically nonincreasing sequence. In addition, $\{\theta_{\mu_k,\overline{\alpha}}(w^k) + \overline{\alpha} m \rho \mu_k\}$ is bounded, since $\{\theta_{\mu_k,\overline{\alpha}}(w^k)\}$ is bounded from Assumption 5.6.1 (a). Therefore, $\{\theta_{\mu_k,\overline{\alpha}}(w^k) + \overline{\alpha} m \rho \mu_k\}$ is convergent. This fact together with $\lim_{k\to\infty} \mu_k = 0$ and (5.6.16) yields that $\{\theta_{\mu_k,\overline{\alpha}}(w^{k+1})\}$ and $\{\theta_{\mu_k,\overline{\alpha}}(w^k)\}$ must converge to the same limit. □

Finally, we show that a sequence generated by the algorithm globally converges to B-stationary point of MPSOCC (5.3.1) under the assumption that any accumulation point $\overline{w} = (\overline{x}, \overline{y}, \overline{z})$ satisfies $\overline{y} - \overline{z} \notin \mathrm{bd}\,(\mathcal{K} \cup -\mathcal{K})$, which is equivalent to the strict complementarity condition given in Definition 5.3.1 if $\overline{y}$ and $\overline{z}$ satisfy the SOC complementarity condition.

Theorem 5.6.5. *Let $\{w^k\}$ be a sequence generated by Algorithm 5.1. Suppose that $\{w^k\}$ satisfies Assumption 5.6.1. Let $\overline{w} = (\overline{x}, \overline{y}, \overline{z})$ be an arbitrary accumulation point of $\{w^k\}$. If $(\overline{y}, \overline{z})$ satisfies $\overline{y} - \overline{z} \notin \mathrm{bd}\,(\mathcal{K} \cup -\mathcal{K})$, then $\overline{w}$ is a B-stationary point of MPSOCC (5.1.1).*

Proof. By Lemma 5.6.4(ii), there exists some constant $\alpha > 0$ and k_0 such that $\alpha_k = \alpha$ for all $k \geq k_0$. In this proof, we suppose without loss of generality that $\alpha_k = \alpha$ holds for all k.

We first show that

$$\lim_{k\to\infty} \|dw^k\| = 0. \tag{5.6.17}$$

From Proposition 5.5.3 (b) and Assumption 5.6.1 (b), we have

$$\theta'_{\mu_k,\alpha}(w^k; dw^k) \le -(dw^k)^\top B_k dw^k \le -\gamma_1 \|dw^k\|^2, \tag{5.6.18}$$

which together with Armijo condition (5.5.6) yields

$$\begin{aligned} \theta_{\mu_k,\alpha}(w^{k+1}) &\le \theta_{\mu_k,\alpha}(w^k) + \sigma\tau_k \theta'_{\mu_k,\alpha}(w^k; dw^k) \\ &\le \theta_{\mu_k,\alpha}(w^k) - \gamma_1\sigma\tau_k \|dw^k\|^2. \end{aligned}$$

Hence, from Lemma 5.6.4 (iii), we obtain

$$\lim_{k\to\infty} \tau_k \|dw^k\|^2 = 0.$$

Now, assume for contradiction that (5.6.17) does not hold. Then, there exists an infinite index set $K \subseteq \{0, 1, \dots\}$ such that

$$\lim_{\substack{k\to\infty \\ k\in K}} \|dw^k\| > 0, \tag{5.6.19}$$

and hence

$$\lim_{\substack{k\to\infty \\ k\in K}} \tau_k = 0.$$

Let ℓ_k be the smallest nonnegative integer L satisfying (5.5.6), i.e., $\rho^{\ell_k} = \tau_k$. Then, from the definition of ℓ_k, we have

$$\theta_{\mu_k,\alpha}(w^k + \rho^{\ell_k - 1} dw^k) > \theta_{\mu_k,\alpha}(w^k) + \sigma\rho^{\ell_k-1}\theta'_{\mu_k,\alpha}(w^k; dw^k),$$

which implies

$$\xi_k := \frac{\theta_{\mu_k,\alpha}(w^k + \rho^{\ell_k-1}dw^k) - \theta_{\mu_k,\alpha}(w^k)}{\rho^{\ell_k-1}} - \theta'_{\mu_k,\alpha}(w^k; dw^k) > -(1-\sigma)\theta'_{\mu_k,\alpha}(w^k; dw^k). \tag{5.6.20}$$

By Lemma 5.6.3 together with $\lim_{k\to\infty,\ k\in K} \rho^{\ell_k-1}=0$ and $\overline{y} - \overline{z} \notin \mathrm{bd}\,(\mathcal{K} \cup -\mathcal{K})$, we have $\limsup_{k\to\infty,\ k\in K} \xi_k \le 0$. Moreover, we have from (5.6.18)

$$-(1-\sigma)\theta'_{\mu_k,\alpha}(w^k; dw^k) \ge (1-\sigma)\gamma_1 \|dw^k\|^2. \tag{5.6.21}$$

From (5.6.20) and (5.6.21), we must have $\lim_{k\to\infty,\ k\in K} \|dw^k\| = 0$. However, this contradicts (5.6.19), and hence we have (5.6.17).

Next, we show that $\overline{w}$ satisfies the KKT conditions (5.3.2) of MPSOCC (5.3.1). Let $\{(\eta^k, u^k, v^k)\}$ be the sequence of multipliers corresponding to $\{dw^k\}$. Then, $\{(u^k, v^k)\}$ is bounded from Lemma 5.6.4 (i). Hence, there exist vectors $\overline{u}$, $\overline{v}$ and an index set K' such that $\lim_{k\to\infty, k\in K'}(w^k, u^k, v^k) = (\overline{w}, \overline{u}, \overline{v})$. By (5.5.3) and letting $X := \{x \in \mathbb{R}^n \mid Ax \le b\}$ we have

$$\zeta^k \in -\mathcal{N}_X(x^k) \times \{0\}^{2m}, \tag{5.6.22}$$

$$\begin{pmatrix} N & M & -I_m \\ 0 & \nabla_y \Phi_{\mu_k}(y^k, z^k)^\top & \nabla_z \Phi_{\mu_k}(y^k, z^k)^\top \end{pmatrix} \begin{pmatrix} dx^k \\ dy^k \\ dz^k \end{pmatrix} = -\begin{pmatrix} 0 \\ \Phi_{\mu_k}(y^k, z^k) \end{pmatrix}, \tag{5.6.23}$$

$$0 \le b - Ax^k - Adx^k, \tag{5.6.24}$$

where

$$\zeta^k := \begin{pmatrix} \nabla_x f(x^k, y^k) \\ \nabla_y f(x^k, y^k) \\ 0 \end{pmatrix} + B_k \begin{pmatrix} dx^k \\ dy^k \\ dz^k \end{pmatrix} + \begin{pmatrix} N^\top \\ M^\top \\ -I_m \end{pmatrix} u^k + \begin{pmatrix} 0 \\ \nabla_y \Phi_{\mu_k}(y^k, z^k) \\ \nabla_z \Phi_{\mu_k}(y^k, z^k) \end{pmatrix} v^k, \quad (5.6.25)$$

and (5.6.22) follows from the first equation of (5.5.3) with $\mathcal{N}_X(x^k) = \{A^\top \eta \mid \eta \geq 0,\ \eta^\top(Ax^k - b) = 0\}$. By letting $k \in K'$ tend to ∞ in (5.6.23) and (5.6.24), we have

$$0 \leq b - A\overline{x}, \quad \Phi(\overline{y}, \overline{z}) = 0. \quad (5.6.26)$$

Note that (5.6.26) implies $\overline{x} \in X$. In addition, note that $\{\zeta^k\}_{k\in K'}$ is a convergent sequence satisfying (5.6.22), since $\{B_k\}$ is bounded from Assumption 5.6.1 and $\lim_{k\to\infty} \nabla\Phi_{\mu_k}(y^k, w^k) = \nabla\Phi(\overline{y}, \overline{w})$ from the strict complementarity condition. Then, these facts together with (5.6.17) and the closedness of the point-to-set map $\mathcal{N}_X(\cdot)$ yield

$$\begin{pmatrix} \nabla_x f(\overline{x}, \overline{y}) \\ \nabla_y f(\overline{x}, \overline{y}) \\ 0 \end{pmatrix} + \begin{pmatrix} N^T \\ M^T \\ -I_m \end{pmatrix} \overline{u} + \begin{pmatrix} 0 \\ \nabla_y \Phi(\overline{y}, \overline{z}) \\ \nabla_z \Phi(\overline{y}, \overline{z}) \end{pmatrix} \overline{v} = \lim_{k\to\infty, k\in K'} \zeta^k \in -\mathcal{N}_X(\overline{x}) \times \{0\}^{2m}.$$

This together with (5.6.26) means that $\overline{w}$ satisfies the KKT conditions (5.3.2) of MPSOCC (5.3.1). By Proposition 5.3.8, $\overline{w}$ is a B-stationary point of MPSOCC (5.1.1). □

5.7 Numerical experiments

In this section, we implement Algorithm 5.1 for solving problem (5.1.1) and report some numerical results. The program is coded in Matlab 2008a and run on a machine with an Intel®Core2 Duo E6850 3.00GHz CPU and 4GB RAM. In Step 0 of the algorithm, we set the parameters as

$$\delta := 1, \quad \alpha_{-1} := 10, \quad \sigma := 10^{-3}, \quad \rho := 0.9.$$

The choice of smoothing parameters $\{\mu_k\}$, i.e., μ_0 and β, and a starting point w^0 vary with the experiment. We let B_0 be the identity matrix, and update B_k by the modified BFGS formula:

$$B_{k+1} := B_k - \frac{B_k s^k (B_k s^k)^\top}{(s^k)^\top B_k s^k} + \frac{\zeta^k (\zeta^k)^\top}{(s^k)^\top \zeta^k}$$

with $s^k = w^{k+1} - w^k$ and $\zeta^k = \theta_k \tilde{\zeta}^k + (1-\theta_k) B_k s^k$, where $\tilde{\zeta}^k := \nabla_w L_{\mu_{k+1}}(w^{k+1}, u^k, v^k, \eta^k) - \nabla_w L_{\mu_k}(w^k, u^k, v^k, \eta^k)$, L_μ denotes the Lagrangian function defined by $L_\mu(w, u, v, \eta) := f(x, y) + \Phi_\mu(y, z)v + (Nx + My + q - z)u + (Ax - b)\eta$, and θ_k is determined by

$$\theta_k := \begin{cases} 1 & \text{if } (s^k)^\top \tilde{\zeta}^k \geq 0.2 (s^k)^\top B_k s^k \\ \dfrac{0.8 (s^k)^\top B_k s^k}{(s^k)^\top (B_k s^k - \tilde{\zeta}^k)} & \text{otherwise.} \end{cases}$$

In Step 1, we use the *quadprog* solver in Matlab Optimization Toolbox for solving the QP subproblems. In Step 3, we terminate the algorithm if the following condition is satisfied:

$$\|\Phi(y^k, z^k)\|_\infty + \|dw^k\|_\infty \leq 10^{-7}. \quad (5.7.1)$$

The rationale for using (5.7.1) is as follows. If $\Phi(y^k, z^k) = 0$, i.e., $\mathcal{K} \ni y^k \perp z^k \in \mathcal{K}$ holds, then $w^k = (x^k, y^k, z^k)$ is feasible to MPSOCC (5.1.1), since the remaining constraints $Ax^k \le b$ and $z^k = Nx^k + My^k + q$ always hold from Proposition 5.5.2. Moreover, if $\|dw^k\|_\infty = 0$, then w^k satisfies the KKT conditions (5.3.2) of MPSOCC (5.3.1). Thus, by Theorem 5.6.5, $\|\Phi(y^k, z^k)\|_\infty + \|dw^k\|_\infty = 0$ indicates that w^k is a B-stationary point under the assumption $y^k - z^k \notin \mathrm{bd}\,(\mathcal{K} \cup -\mathcal{K})$, which is equivalent to the strict complementarity condition in Definition 5.3.1 if $\mathcal{K} \ni y^k \perp z^k \in \mathcal{K}$. Hence, (5.7.1) is appropriate for a stopping criterion of the algorithm. As the CM function, we choose $\hat{g}(\alpha) := ((\alpha^2 + 4)^{1/2} + \alpha)/2$.

Experiment 1

In the first experiment, we solve the following test problem of the form (5.1.1):

$$\begin{array}{ll} \underset{x,y,z}{\text{Minimize}} & \|x\|^2 + \|y\|^2 \\ \text{subject to} & Ax \le b, \\ & z = Nx + My + q, \\ & \mathcal{K} \ni y \perp z \in \mathcal{K}, \end{array} \tag{5.7.2}$$

where $(x, y, z) \in \mathbb{R}^{10} \times \mathbb{R}^m \times \mathbb{R}^m$, and each element of $A \in \mathbb{R}^{10\times 10}$, $N \in \mathbb{R}^{m\times 10}$ is randomly chosen from $[-1, 1]$. Moreover, each element of $b \in \mathbb{R}^{10}$ is randomly chosen from $[0, 1]$. In addition, $M \in \mathbb{R}^{m\times m}$ is a positive semi-definite symmetric matrix generated by $M = M_1 M_1^\top + 0.01I$, and $M_1 \in \mathbb{R}^{m\times m}$ is a matrix whose entries are randomly chosen from $[-1, 1]$. The vector $q \in \mathbb{R}^m$ is set to be $q := \xi_z - M\xi_y$ with $\xi_y \in \mathbb{R}^m$ and $\xi_z \in \mathbb{R}^m$, whose components are randomly chosen from $[-1, 1]$. We choose different Cartesian structures for $\mathcal{K}$, and generate 50 problems for each $\mathcal{K}$. In applying Algorithm 5.1, we set an initial point $w^0 = (x^0, y^0, z^0) := (0, \xi_y, \xi_z) \in \mathbb{R}^{10} \times \mathbb{R}^m \times \mathbb{R}^m$, so that $Ax^0 \le b$ and $z^0 = Nx^0 + My^0 + q$ are satisfied. We choose smoothing parameters $\{\mu_k\}$ as $\mu_k := 100 \times 0.8^k$.

The obtained results are shown in Tables 5.1 and 5.2, where $(\mathcal{K}^\nu)^\kappa := \mathcal{K}^\nu \times \mathcal{K}^\nu \times \cdots \times \mathcal{K}^\nu \subseteq \mathbb{R}^{\nu\kappa}$ and each column represents the following:

- $\sharp$ite: the average number of iterations among 50 test problems for each $\mathcal{K}$;
- cpu(s): the average cpu-time in second among 50 test problems for each $\mathcal{K}$;
- non(%): percentage of test problems whose solutions obtained by Algorithm 5.1 satisfy the strict complementarity condition in Definition 5.3.1.

Recall that convergence to a B-stationary point is proved under the strict complementarity condition. Hence, the value of "non" represents the percentage of problems for which the algorithm successfully finds B-stationary points. From Table 5.1, we can observe that $\sharp$ite does not change so much although the values of cpu(s) tends to be larger as m increases. From Table 5.2, we can see that non(%) tends to be less than 100 if $\mathcal{K}$ includes $\mathcal{K}^1$ or $\mathcal{K}^2$. Indeed, when $\mathcal{K} = (\mathcal{K}^1)^{100}$ and $(\mathcal{K}^2)^{50}$, the values of "non(%)" is 74 and 86, respectively, whereas it becomes 100 when the dimension of all SOCs in $\mathcal{K}$ is larger than 10.

m	$\mathcal{K}$	$\sharp$ite	cpu(s)	non(%)
10	$\mathcal{K}^{10}$	57.42	0.400	100
20	$\mathcal{K}^{20}$	56.30	0.471	100
30	$\mathcal{K}^{30}$	55.78	0.568	100
40	$\mathcal{K}^{40}$	55.44	0.726	100
50	$\mathcal{K}^{50}$	55.06	0.987	100
60	$\mathcal{K}^{60}$	54.96	1.388	100
70	$\mathcal{K}^{70}$	54.74	1.797	100
80	$\mathcal{K}^{80}$	54.66	2.130	100
90	$\mathcal{K}^{90}$	54.40	2.437	100
100	$\mathcal{K}^{100}$	54.20	2.930	100

Table 5.1: Results for problems with a single SOC complementarity constraint (Experiment 1)

m	$\mathcal{K}$	$\sharp$ite	cpu(s)	non(%)
100	$\mathcal{K}^{100}$	54.20	2.930	100
100	$(\mathcal{K}^{50})^2$	55.18	3.037	100
100	$\mathcal{K}^{50} \times \mathcal{K}^{20} \times \mathcal{K}^{30}$	56.28	3.016	100
100	$(\mathcal{K}^{10})^{10}$	64.10	3.687	100
100	$\mathcal{K}^{50} \times \mathcal{K}^{20} \times (\mathcal{K}^{10})^2 \times \mathcal{K}^5 \times (\mathcal{K}^1)^5$	78.22	4.558	98
100	$(\mathcal{K}^2)^{50}$	78.68	6.012	86
100	$(\mathcal{K}^1)^{100}$	87.84	6.685	74

Table 5.2: Results for problems with multiple SOC complementarity constraints (Experiment 1)

Experiment 2.

In the second experiment, we apply Algorithm 5.1 to a bilevel programming problem with a robust optimization problem in the *lower-level*. Bilevel programming has wide application such as network design and production planning [2, 14]. On the other hand, robust optimization is known to be a powerful methodology to treat optimization problems with uncertain data [3, 4]. In this experiment, we solve the following problem:

$$\begin{array}{ll} \underset{(x,y)\in\mathbb{R}^4\times\mathbb{R}^4}{\text{Minimize}} & \|x - Cy\|^2 + \sum_{i=1}^4 x_i \\ \text{subject to} & 0 \le x_i \le 5 \ (i = 1, 2, 3, 4), \\ & 1 \le -x_1 + 2x_2 + x_4 \le 3, \\ & 1 \le x_2 + x_3 - x_4 \le 2, \\ & y \text{ solves } P(x), \end{array} \tag{5.7.3}$$

with

$$P(x): \ \underset{y\in\mathbb{R}^4}{\text{Minimize}} \max_{\tilde{x}\in U_r(x)} \tilde{x}^\top y + \frac{1}{2} y^\top M y,$$

where $r \geq 0$ is an uncertainty parameter, $U_r(x) \subseteq \mathbb{R}^4$ is an uncertainty set defined by $U_r(x) := \{\tilde{x} \in \mathbb{R}^4 \mid \|\tilde{x} - x\| \leq r\}$, and

$$M := \begin{pmatrix} 2 & 2 & 0 & -1 \\ 2 & 4 & -2 & 0 \\ 0 & -2 & 2 & 0 \\ -1 & 0 & 0 & 6 \end{pmatrix}, \ C := \begin{pmatrix} -1 & 1 & 0 & 1 \\ 0 & 2 & 2 & 3 \\ 0 & 0 & 3 & 2 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

For solving problem (5.7.3), we introduce an auxiliary variable $\gamma \in \mathbb{R}$ to reformulate the lower-level minimax problem $P(x)$ as the following SOCP:

$$\begin{array}{ll} \underset{(\gamma,y)\in\mathbb{R}\times\mathbb{R}^4}{\text{Minimize}} & \frac{1}{2}y^\top My + x^\top y + r\gamma \\ \text{subject to} & \begin{pmatrix} \gamma \\ y \end{pmatrix} \in \mathcal{K}^5. \end{array}$$

Furthermore, the above SOCP can be rewritten as the following SOC complementarity problem:

$$\mathcal{K}^5 \ni \begin{pmatrix} \gamma \\ y \end{pmatrix} \perp \begin{pmatrix} r \\ My + x \end{pmatrix} \in \mathcal{K}^5.$$

Thus, we can convert problem (5.7.3) to the following problem:

$$\begin{array}{ll} \underset{(x,y,z,\gamma)\in\mathbb{R}^4\times\mathbb{R}^4\times\mathbb{R}^5\times\mathbb{R}}{\text{Minimize}} & \|x - Cy\|^2 + \sum_{i=1}^4 x_i \\ \text{subject to} & 0 \leq x_i \leq 5 \ (i = 1, 2, 3, 4), \\ & 1 \leq -x_1 + 2x_2 + x_4 \leq 3, \\ & 1 \leq x_2 + x_3 - x_4 \leq 2, \\ & z = \begin{pmatrix} r \\ My + x \end{pmatrix}, \ \mathcal{K}^5 \ni \begin{pmatrix} \gamma \\ y \end{pmatrix} \perp z \in \mathcal{K}^5, \end{array} \tag{5.7.4}$$

which is of the form (5.1.1). For the sake of comparison, problem (5.7.4) is solved not only by Algorithm 5.1, but also by the smoothing method [77], which is described as follows:

Smoothing method

Step 0. Choose a positive sequence $\{\tau_\ell\}$ such that $\tau_\ell \to 0$. Set $\ell := 0$.

Step 1. Find a stationary point $w^\ell = (x^\ell, y^\ell, z^\ell)$ of the smoothed problem (5.5.1) with $\mu = \tau_\ell$.

Step 2. If w^ℓ is feasible for MPSOCC (5.1.1), then stop. Otherwise, set $\ell := \ell + 1$ and go to Step 1.

Each algorithm is implemented in the following way: In Step 0 of Algorithm 5.1, we set smoothing parameters $\mu_k := 0.8^k$ $(k \geq 0)$ and a starting point $x^0 := (1, 1, 1, 1), (\gamma_0, y^0, z^0) := (0, 0, 0) \in \mathbb{R} \times \mathbb{R}^4 \times \mathbb{R}^5$. The choice of the other parameters is the same as in Experiment 1. In Step 0 of the smoothing method, we set $\tau_\ell := 0.8^\ell$ for $\ell \geq 0$. In Step 1, for solving problem (5.5.1) with $\mu = \tau_\ell$, we apply Algorithm 5.1 with slight modification, where the smoothing parameter μ_k is fixed to τ_ℓ for all k and the termination criterion is replaced by $\|dw^k\|_\infty \le 10^{-7}$. In Step 2, we stop the smoothing method when $\|\Phi(y^\ell, z^\ell)\|_\infty \le 10^{-7}$ is satisfied.[1]

We then test both the methods to problem (5.7.4) with $r = 0.02, 0.04, 0.06, 0.08$ and 0.10. The obtained results are shown in Tables 5.3 and 5.4, whose columns represent the following:

- (x^*, y^*, γ^*): the value of x, y, γ obtained by Algorithm 5.1;
- $(\lambda_1^*, \lambda_2^*)$: spectral values of $\begin{pmatrix}\gamma^* \\ y^*\end{pmatrix} + z^*$ with respect to $\mathcal{K}^5$ defined as in Definition 2.4.1, where $z^* := (r, M\begin{pmatrix}\gamma^* \\ y^*\end{pmatrix} + x^*)$;
- ite_{out}: the number of outer iterations;
- $\sharp$QP: the number of QP-subproblems (5.5.2) solved in each trial.

From Table 5.3, for all r, we can observe $(\lambda_1^*, \lambda_2^*) > 0$, which means $\begin{pmatrix}\gamma^* \\ y^*\end{pmatrix} + z^* \in \operatorname{int} \mathcal{K}^5$, i.e., the strict complementarity condition holds at the obtained solution. Hence, Algorithm 5.1 finds a B-stationary point of problem (5.7.4) successfully. From Table 5.4, we cannot find a significant difference between the values of ite_{out} for the two methods. However, it is observed that the value of $\sharp$QP in the smoothing method tends to be much larger than that in Algorithm 5.1. Indeed, when $r = 0.02$, the smoothing method has $\sharp\text{QP} = 218$, which is almost five times larger than $\sharp\text{QP} = 44$ in Algorithm 5.1. This fact suggests that the smoothing method needs to solve a number of QP-subproblems in Step 1 for solving each smoothed problem (5.5.1) with fixed μ, while Algorithm 5.1 only solves one QP subproblem (5.5.2) for each smoothed problem (5.5.1). As a result, the computational cost in the smoothing method tends to be larger than Algorithm 5.1.

r	x^*	y^*	γ^*	$(\lambda_1^*, \lambda_2^*)$
0.02	$(0.9236, 0.9618, 0.0382, 0.0000)$	$(-0.6152, 0.1120, 0.0915, 0.1020)$	0.6402	$(0.040, 1.280)$
0.04	$(0.9495, 0.9747, 0.0253, 0.0000)$	$(-0.6170, 0.1106, 0.0950, 0.1018)$	0.6421	$(0.080, 1284)$
0.06	$(0.9754, 0.9877, 0.0123, 0.0000)$	$(-0.6189, 0.1092, 0.0985, 0.1016)$	0.6442	$(0.120, 1.288)$
0.08	$(1.0021, 1.0007, 0.0000, 0.0007)$	$(-0.6218, 0.1084, 0.1021, 0.1014)$	0.6474	$(0.160, 1.295)$
0.10	$(1.0416, 1.0139, 0.0000, 0.0139)$	$(-0.6356, 0.1123, 0.1044, 0.1011)$	0.6616	$(0.200, 1.323)$

Table 5.3: Results for Algorithm 5.1 (Experiment 2)

5.8 Concluding remarks

In this chapter, we have considered the mathematical program with SOC complementarity constraints. We have proposed an algorithm based on the smoothing and the sequential quadratic programming (SQP) methods, in which we replace the SOC complementarity constraints with smooth equality constraints by means of the natural residual and its smoothing function, and apply the SQP method while decreasing the smoothing parameter gradually. We have shown

[1] The remaining constraints $Ax^\ell \le b$ and $z^\ell = Nx^\ell + My^\ell + q$ are automatically satisfied since w^ℓ is feasible to the smoothed problem (5.5.1).

	Algorithm 5.1			smoothing method		
r	cpu(s)	ite_{out}	♯QP	cpu(s)	ite_{out}	♯QP
0.02	0.207	45	44	1.005	43	218
0.04	0.213	43	42	0.941	42	205
0.06	0.213	43	41	0.908	41	199
0.08	0.214	42	40	0.873	40	188
0.10	0.209	41	40	0.875	40	188

Table 5.4: Comparison of Algorithm 5.1 and the smoothing method

that the proposed algorithm possesses the global convergence property under the Cartesian P_0 and the strict complementarity assumptions. We have further confirmed the efficiency of the algorithm through numerical experiments.

Chapter 6

Conclusion

In this thesis, we have considered two different types of optimization problems that are generalizations of the second-order cone programming problem (SOCP). One is a semi-infinite second-order cone programming problem (SISOCP) and the other is a mathematical programming problem with SOC complementarity constraints (MPSOCC).

The main contributions of the thesis can be summarized as follows:

- In Chapter 3, we have considered a special case of SISOCP that consists of a convex objective function and infinitely many affine SOC constrains. For such a convex SISOCP, we have proposed two exchange-type methods that produce iteration points by solving a sequence of relaxed SISOCPs with finitely many SOC constraints. The first one is an explicit exchange method. Assuming strict convexity of the objective function, we have established its global convergence. The second one is a regularized explicit exchange method which is a hybrid of the explicit exchange method and regularization method. With the help of a regularization scheme, we have succeeded in ensuring its global convergence without strict convexity. Especially, the choice of two parameters given in the algorithm plays a crucial role in the convergence analysis. In the numerical experiments, we have examined the behavior of the latter proposed method and observed its efficiency.

- In Chapter 4, for solving SISOCP of the more general form, we have proposed an SQP-type method based on the local reduction technique. In this approach, we represent the SISOCP as SOCP locally by means of finitely many implicit functions, and apply the SQP-type method to the latter SOCP. We have established its global convergence by using the max-type penalty function as a merit function. Furthermore, we have analyzed local convergence behavior of generated iteration points. The convergence rate has been proved to be quadratic. In the numerical experiments, for the sake of comparison, we have also implemented another SQP-type method, and then observed good performance of the proposed method.

- In Chapter 5, we have focused on MPSOCC and proposed a smoothing SQP algorithm for solving it. This method first replaces the SOC complementarity constraints with nondifferentiable equality constraints by using the natural residual. Then, an SQP-type method together with a smoothing technique is applied to the reformulated MPSOCC. We have proved that QP subproblems are always consistent if the Cartesian P_0 property holds. Moreover, we have also shown that the generated iterative sequence globally converges to a B-stationarity point under strict complementarity. Finally, we have conducted some numerical experiments and observed efficiency of the proposed method.

We finally discuss further research concerning SISOCP and MPSOCC.

- We have some room to improve the proposed methods for solving SISOCP. Actually, all algorithms require exact optima of SOCP-subproblems to generate iteration points. Although they can be found efficiently by using existing algorithms, we have to spend quite a lot of computational cost for that purpose. Therefore, to solve SISOCP more practically, it is desired to refine the proposed algorithms so that they allow inexact solutions of SOCP without loss of good convergence properties.

- In the proposed algorithm for solving MPSOCC, we gradually decrease the smoothing parameter to zero to ensure the global convergence. However, as in smoothing-type Newton methods [26, 12], the speed of decreasing the parameter should play a key role in establishing rapid local convergence. For fast convergence, an effective way of decreasing the parameter should be explored.

==

Bibliography

[1] F. Alizadeh and D. Goldfarb, *Second-order cone programming*, Mathematical Programming, 95 (2003), pp. 3–51.

[2] J. Bard et al., *Practical Bilevel Optimization: Algorithms and Applications*, Springer, 1998.

[3] A. Ben-Tal and A. Nemirovski, *Robust convex optimization*, Mathematics of Operations Research, 23 (1998), pp. 769–805.

[4] ——, *Robust solutions of uncertain linear programs*, Operations Research Letters, 25 (1999), pp. 1–13.

[5] J. F. Bonnans and H. Ramírez C, *Perturbation analysis of second-order cone programming problems*, Mathematical Programming, 104 (2005), pp. 205–227.

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

[7] S. Bütikofer and D. Klatte, *A nonsmooth Newton method with path search and its use in solving $C^{1,1}$ programs and semi-infinite problems*, SIAM Journal on Optimization, 20 (2010), pp. 2381–2412.

[8] S. Chan, H. Chen, and C. Pun, *The design of digital all-pass filters using second-order cone programming (SOCP)*, IEEE Transactions on Circuits and Systems II: Express Briefs, 52 (2005), pp. 66–70.

[9] J.-S. Chen and S.-H. Pan, *A descent method for a reformulation of the second-order cone complementarity problem*, Journal of Computational and Applied Mathematics, 213 (2008), pp. 547–558.

[10] J.-S. Chen and P. Tseng, *An unconstrained smooth minimization reformulation of the second-order cone complementarity problem*, Mathematical Programming, 104 (2005), pp. 293–327.

[11] X. Chen and H. Qi, *Cartesian P-property and its applications to the semidefinite linear complementarity problem*, Mathematical programming, 106 (2006), pp. 177–201.

[12] X. Chen, D. Sun, and J. Sun, *Complementarity functions and numerical experiments on some smoothing Newton methods for second-order-cone complementarity problems*, Computational Optimization and Applications, 25 (2003), pp. 39–56.

[13] R. Cottle, J.-S. Pang, and R. Stone, *The linear complementarity problem*, Society for Industrial Applied Mathematics, 2009.

[14] S. Dempe, *Foundations of bilevel programming*, Springer, 2002.

[15] J. Faraut and A. Korányi, *Analysis on Symmetric Cones*, Oxford University Press, New York, 1994.

[16] C. A. Floudas and O. Stein, *The adaptive convexification algorithm: A feasible point method for semi-infinite programming*, SIAM Journal on Optimization, 18 (2007), pp. 1187–1208.

[17] M. Fukushima, Z.-Q. Luo, and J.-S. Pang, *A globally convergent sequential quadratic programming algorithm for mathematical programs with linear complementarity constraints*, Computational Optimization and Applications, 10 (1998), pp. 5–34.

[18] M. Fukushima, Z.-Q. Luo, and P. Tseng, *Smoothing functions for second-order cone complementarity problems*, SIAM Journal on Optimization, 12 (2001), pp. 436–460.

[19] M. Fukushima and P. Tseng, *An implementable active-set algorithm for computing a B-stationary point of a mathematical program with linear complementarity constraints*, SIAM Journal on Optimization, 12 (2002), pp. 724–739.

[20] L. E. Ghaoui, M. Oks, and F. Oustry, *Worst-case value-at-risk and robust portfolio optimization: A conic programming approach*, Operations Research, 51 (2003), pp. 543–556.

[21] M. A. Goberna and M. A. López, *Semi-Infinite Programming: Recent Advances*, Kluwer Academic Publishers, Dordrecht, 2001.

[22] D. Goldfarb and G. Iyengar, *Robust portfolio selection problems*, Mathematics of Operations Research, 28 (2003), pp. 1–38.

[23] M. S. Gowda, R. Sznajder, and J. Tao, *Some P-properties for linear transformations on Euclidean Jordan algebras*, Linear Algebra and Its Applications, 393 (2004), pp. 203–232.

[24] G. Gramlich, R. Hettich, and E. W. Sachs, *Local convergence of SQP methods in semi-infinite programming*, SIAM Journal on Optimization, 5 (1995), pp. 641–658.

[25] S. Hayashi, T. Yamaguchi, N. Yamashita, and M. Fukushima, *A matrix-splitting method for symmetric affine second-order cone complementarity problems*, Journal of Computational and Applied Mathematics, 175 (2005), pp. 335–353.

[26] S. Hayashi, N. Yamashita, and M. Fukushima, *A combined smoothing and regularization method for monotone second-order cone complementarity problems*, SIAM Journal on Optimization, 15 (2005), pp. 593–615.

[27] S. Hayashi and S.-Y. Wu, *An explicit exchange algorithm for linear semi-infinite programming problems with second-order cone constrains*, SIAM Journal on Optimization, 20 (2009), pp. 1527–1546.

[28] R. Hettich, *An implementation of a discretization method for semi-infinite programming*, Mathematical Programming, 34 (1986), pp. 354–361.

[29] R. Hettich and K. O. Kortanek, *Semi-infinite programming: Theory, methods, and applications*, SIAM Review, 35 (1993), pp. 380–429.

[30] R. Hettich and W. V. Honstede, *On quadratically convergent methods for semi-infinite programming*, in Semi-Infinite Programming, R. Hettich, ed., Springer, Berlin, 1979, pp. 97–111.

[31] S. Ito, Y. Liu, and K. Teo, *A dual parametrization method for convex semi-infinite programming*, Annals of Operations Research, 98 (2000), pp. 189–213.

[32] Y. Kanno, J. Martins, and A. Pinto da Costa, *Three-dimensional quasi-static frictional contact by using second-order cone linear complementarity problem*, International Journal for Numerical Methods in Engineering, 65 (2006), pp. 62–83.

[33] H. Kato and M. Fukushima, *An SQP-type algorithm for nonlinear second-order cone programs*, Optimization Letters, 1 (2007), pp. 129–144.

[34] H. C. Lai and S.-Y. Wu, *On linear semi-infinite programming problems*, Numerical Functional Analysis and Optimization, 13 (1992), pp. 287–304.

[35] D. Li, L. Qi, J. Tam, and S.-Y. Wu, *A smoothing Newton method for semi-infinite programming*, Journal of Global Optimization, 30 (2004), pp. 169–194.

[36] Y. Liu and L. Zhang, *Convergence analysis of the augmented Lagrangian method for nonlinear second-order cone optimization problems*, Nonlinear Analysis: Theory, Methods & Applications, 67 (2007), pp. 1359–1373.

[37] 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.

[38] M. López and G. Still, *Semi-infinite programming*, European Journal of Operational Research, 180 (2007), pp. 491–518.

[39] W. S. Lu and T. Hinamoto, *A second-order cone programming approach for minimax design of 2-D FIR filters with low group delay*, in Proc. IEEE Int. Symp. Circuits Systems, 2006, pp. 2521–2524.

[40] W. Lu, T. Saramaki, and R. Bregovic, *Design of practically perfect-reconstruction cosine-modulated filter banks: A second-order cone programming approach*, IEEE Transactions on Circuits and Systems I: Regular Papers, 51 (2004), pp. 552–563.

[41] Z.-Q. Luo, J.-S. Pang, and D. Ralph, *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press, Cambridge, 1996.

[42] R. Monteiro and T. Tsuchiya, *Polynomial convergence of primal-dual algorithms for the second-order cone program based on the MZ-family of directions*, Mathematical Programming, 88 (2000), pp. 61–83.

[43] Y. Narushima, N. Sagara, and H. Ogasawara, *A smoothing Newton method with Fischer-Burmeister function for second-order cone complementarity problems*, Journal of Optimization Theory and Applications, 149 (2011), pp. 79–101.

[44] Q. Ni, C. Ling, L. Qi, and K. Teo, *A truncated projected Newton-type algorithm for large-scale semi-infinite programming*, SIAM Journal on Optimization, 16 (2006), pp. 1137–1154.

[45] R. Nishimura, S. Hayashi, and M. Fukushima, *Robust Nash equilibria in n-person non-cooperative games: Uniqueness and reformulation*, Pacific Journal of Optimization, 5 (2009), pp. 237–259.

[46] T. Okuno and M. Fukushima, *Local reduction based SQP-type method for semi-infinite programs with an infinite number of second-order cone constraints*, submitted to Journal of Global Optimization.

[47] T. Okuno, S. Hayashi, and M. Fukushima, *A regularized explicit exchange method for semi-infinite programs with an infinite number of conic constraints*, SIAM Journal on Optimization, 22 (2012), pp. 1009–1028.

[48] S.-H. Pan and J.-S. Chen, *A regularization method for the second-order cone complementarity problem with the cartesian P_0-property*, Nonlinear Analysis: Theory, Methods & Applications, 70 (2009), pp. 1475–1491.

[49] ——, *A semismooth Newton method for SOCCPs based on a one-parametric class of SOC complementarity functions*, Computational Optimization and Applications, 45 (2010), pp. 59–88.

[50] A. Pereira, M. Costa, and E. Fernandes, *Interior point filter method for semi-infinite programming problems*, Optimization, 60 (2011), pp. 1309–1338.

[51] A. Pereira and E. Fernandes, *A reduction method for semi-infinite programming by means of a global stochastic approach*, Optimization, 58 (2009), pp. 713–726.

[52] L. Qi, C. Ling, X. Tong, and G. Zhou, *A smoothing projected Newton-type algorithm for semi-infinite programming*, Computational Optimization and Applications, 42 (2009), pp. 1–30.

[53] L. Qi, S.-Y. Wu, and G. Zhou, *Semismooth Newton methods for solving semi-infinite programming problems*, Journal of Global Optimization, 27 (2003), pp. 215–232.

[54] R. Reemtsen, *Discretization methods for the solution of semi-infinite programming problems*, Journal of Optimization Theory and Applications, 71 (1991), pp. 85–103.

[55] R. Reemtsen and J. Rückmann, *Semi-Infinite Programming*, Kluwer Academic Publishers, Boston, 1998.

[56] R. T. Rockafellar and R. J. -B. Wets, *Variational Analysis*, Springer, New York, 1998

[57] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, 1970.

[58] T. Sasakawa and T. Tsuchiya, *Optimal magnetic shield design with second-order cone programming*, SIAM Journal on Scientific Computing, 24 (2003), pp. 1930–1950.

[59] H. Scheel and S. Scholtes, *Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity*, Mathematics of Operations Research, 25 (2000), pp. 1–22.

[60] T. Shiu and S.-Y. Wu, *Relaxed cutting plane method with convexification for solving nonlinear semi-infinite programming problems*, Computational Optimization and Applications, 53 (2012), pp. 1–23.

[61] O. Stein and P. Steuermann, *The adaptive convexification algorithm for semi-infinite programming with arbitrary index sets*, Mathematical Programming, 136 (2012), pp. 183–207.

[62] O. Stein and A. Tezel, *The semismooth approach for semi-infinite programming under the reduction ansatz*, Journal of Global Optimization, 41 (2008), pp. 245–266.

[63] ——, *The semismooth approach for semi-infinite programming without strict complementarity*, SIAM Journal on Optimization, 20 (2009), pp. 1052–1072.

[64] G. Still, *Discretization in semi-infinite programming: The rate of convergence*, Mathematical Programming, 91 (2001), pp. 53–69.

[65] J. F. Sturm, *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optimization Methods and Software, 11 (1999), pp. 625–653.

[66] Y. Tanaka, M. Fukushima, and T. Ibaraki, *A globally convergent SQP method for semi-infinite nonlinear optimization*, Journal of Computational and Applied Mathematics, 23 (1988), pp. 141–153.

[67] J. Tao and M. S. Gowda, *Some P-properties for nonlinear transformations on Euclidean Jordan algebras*, Mathematics of Operations Research, 30 (2005), pp. 985–1004.

[68] K. C. Toh, M. J. Todd, and R. H. Tütüncü, *SDPT3—a MATLAB software package for semidefinite programming, version 2.1*, Optimization Methods and Software, 11 (1999), pp. 545–581.

[69] T. Tsuchiya, *A convergence analysis of the scaling-invariant primal-dual path-following algorithms for second-order cone programming*, Optimization Methods and Software, 11 (1999), pp. 141–182.

[70] 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 Trans. Circuits Syst. II, 52 (2005), pp. 601–605.

[71] Y. Wang and L. Zhang, *Properties of equation reformulation of the Karush–Kuhn–Tucker condition for nonlinear second order cone optimization problems*, Mathematical Methods of Operations Research, 70 (2009), pp. 195–218.

[72] S. P. Wu, S. Boyd, and L. Vandenberghe, *FIR filter design via spectral factorization and convex optimization*, in Applied and Computational Control, Signals, and Circuits., 1 (1999), pp. 215–245.

[73] S.-Y. Wu, D. H. Li, L. Qi and G. Zhou, *An iterative method for solving KKT system of the semi-infinte programming*, Optimization Methods and Software, 20 (2005), pp. 629–643.

[74] H. Xu and J. Zeng, *A multisplitting method for symmetrical affine second-order cone complementarity problem*, Computers & Mathematics with Applications, 55 (2008), pp. 459–469.

[75] H. Yamamura, T. Okuno, S. Hayashi, and M. Fukushima, *A smoothing SQP method for mathematical programs with linear second-order cone complementarity constraints*, Pacific Journal of Optimization, to appear.

[76] H. Yamashita and H. Yabe, *A primal-dual interior point method for nonlinear optimization over second-order cones*, Optimization Methods and Software, 24 (2009), pp. 407–426.

[77] T. Yan and M. Fukushima, *Smoothing method for mathematical programs with symmetric cone complementarity constraints*, Optimization, 60 (2011), pp. 113–128.

[78] H. Zhang, J. Li, and S. Pan, *New second-order cone linear complementarity formulation and semi-smooth newton algorithm for finite element analysis of 3D frictional contact problem*, Computer Methods in Applied Mechanics and Engineering, 200 (2011), pp. 77–88.

[79] L. Zhang, S.-Y. Wu, and M. López, *A new exchange method for convex semi-infinite programming*, SIAM Journal on Optimization, 20 (2010), pp. 2959–2977.