AN INEXACT AUGMENTED LAGRANGIAN METHOD FOR SECOND-ORDER CONE PROGRAMMING WITH APPLICATIONS

LING LIANG *, DEFENG SUN †, AND KIM-CHUAN TOH ‡

Abstract. In this paper, we adopt the augmented Lagrangian method (ALM) to solve convex quadratic second-order cone programming problems (SOCPs). Fruitful results on the efficiency of the ALM have been established in the literature. Recently, it has been shown in [12] that, if the quadratic growth condition holds at an optimal solution for the dual problem, then the KKT residual converges to zero R-superlinearly when the ALM is applied to the primal problem. Moreover, Cui et al. [11] provided sufficient conditions for the quadratic growth condition to hold under the metric subregularity and bounded linear regularity conditions for solving composite matrix optimization problems involving spectral functions. Here, we adopt these recent ideas to analyze the convergence properties of the ALM when applied to SOCPs. To the best of our knowledge, no similar work has been done for SOCPs so far. In our paper, we first provide sufficient conditions to ensure the quadratic growth condition for SOCPs. With these elegant theoretical guarantees, we then design an SOCP solver and apply it to solve various classes of SOCPs such as minimal enclosing ball problems, classical trust-region subproblems, square-root Lasso problems, and DIMACS Challenge problems. Numerical results show that the proposed ALM based solver is efficient and robust compared to the existing highly developed solvers such as Mosek and SDPT3.

Key words. second-order cone programming, augmented Lagrangian method, quadratic growth condition, trust-region subproblem, minimal enclosing ball problem, square-root Lasso problem

AMS subject classifications. 90C06, 90C22, 90C25

1. Introduction.

Denote the standard d-dimensional second-order cone (also called ice cream cone *or* Lorentz cone) in $\mathbb{R}^d$ $(d > 1)$ as

$$\mathcal{K}^d := \left\{ x = (x_0, x_t)^\top \in \mathbb{R} \times \mathbb{R}^{d-1} \mid x_0 \geq \|x_t\| \right\}.$$

Let $\mathcal{K}$ be the Cartesian product of r second-order cones, i.e.,

$$\mathcal{K} = \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_r} \subseteq \mathbb{R}^n,$$

where $n = n_1 + \cdots + n_r$. In this paper, we consider the following convex quadratic second-order cone programs (SOCPs)

$$\text{(P)} \qquad \min_{x=(x_1;x_2;x_3)} \; f^0(x) := \frac{1}{2}\langle x_1,\, Hx_1\rangle - \langle b,\, x_2\rangle + \delta_{\mathcal{K}}(x_3)$$

$$\text{s.t.} \qquad -Hx_1 + A^\top x_2 + x_3 = c,\; x_1 \in \operatorname{Ran}(H) \subseteq \mathbb{R}^n,\; x_2 \in \mathbb{R}^m,\; x_3 \in \mathbb{R}^n,$$

where $H \in \mathbb{S}^n_+$ (the cone of $n \times n$ symmetric positive semidefinite matrices) and $A \in \mathbb{R}^{m\times n}$ are given matrices, $\operatorname{Ran}(H)$ denotes the range space of H, $c \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$ are given vectors, and $\delta_{\mathcal{K}}(\cdot)$ is the indicator function for the symmetric cone $\mathcal{K}$. In the above, $(x_1; x_2; x_3)$ denotes the concatenation of the vectors x_1, x_2, x_3. For notational simplicity, we denote $\mathbb{X} := \operatorname{Ran}(H) \times \mathbb{R}^m \times \mathbb{R}^n$ for the rest of this paper.

*Department of Mathematics, National University of Singapore, 10 Lower Kent Ridge Road, Singapore 119076 (liang.ling@u.nus.edu).

†Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Hong Kong (defeng.sun@polyu.edu.hk). The research of this author is supported in part by Hong Kong Research Grant Council under grant PolyU 153014/18P.

‡Department of Mathematics, and Institute of Operations Research and Analytics, National University of Singapore, 10 Lower Kent Ridge Road, Singapore 119076 (mattohkc@nus.edu.sg). This author is supported in part by the Ministry of Education, Singapore, under its Academic Research Fund Tier 3 grant (MOE-2019-T3-1-010).

The dual problem associated with (P) is given by

$$\text{(D)} \quad \max_{y} \; g^0(y) := -\frac{1}{2}\langle y,\, Hy\rangle - \langle c,\, y\rangle - \delta_{\mathcal{K}}(y)$$
$$\text{s.t.} \;\; Ay = b,\; y \in \mathbb{R}^n.$$

We should mention that in this paper, our naming convention of the primal and dual problems are opposite of the convention adopted in the interior-point methods literature.

Let $\mathrm{SOL_P}$ and $\mathrm{SOL_D}$ be the solution sets of (P) and (D), respectively. The Karush-Kuhn-Tucker (KKT) optimality condition for (P) and (D) is given as follows:

$$(1.1) \quad -Hx_1 + A^\top x_2 + x_3 = c, \quad Ay - b = 0, \quad H(x_1 - y) = 0, \quad \mathcal{K} \ni x_3 \perp y \in \mathcal{K}.$$

We assume for the rest of this paper that the KKT condition (1.1) admits at least one solution. Under this assumption, it is well known that $(\bar{x}, \bar{y})$ solves the KKT condition (1.1) if and only if $\bar{x} \in \mathrm{SOL_P}$ and $\bar{y} \in \mathrm{SOL_D}$.

Note that problems (P) and (D) cover the standard primal and dual linear SOCP problems by simply dropping the quadratic term in the objective function, respectively. One may also observe that problem (P) or (D) can be reformulated as a linear SOCP with additional affine and rotated quadratic cone constraints. To explain the procedure, we consider problem (D) as an illustrative example. Recall that a d-dimensional $(d \geq 3)$ rotated quadratic cone is defined by

$$\mathcal{K}_r^d := \left\{x = (x_1, x_2, \cdots, x_d) \in \mathbb{R}^d \mid 2x_1x_2 \geq x_3^2 + \cdots + x_d^2,\; x_1, x_2 \geq 0\right\}.$$

From the positive semidefiniteness of H, there exists $R \in \mathbb{R}^{n\times n}$ such that $H = R^\top R$, and hence we can rewrite problem (D) as $\min_{y,t} \left\{ t + \langle c,\, y\rangle \;\middle|\; Ay = b,\; y \in \mathcal{K},\; \|Ry\|^2 \leq 2t \right\}$. Observe that the constraint $\|Ry\|^2 \leq 2t$ is equivalent to $(t, 1, Rx) \in \mathcal{K}_r^{n+2}$. Therefore, (D) can be reformulated as

$$(1.2) \quad \min_{y,t,s,z} \left\{ t + \langle c,\, y\rangle \;\middle|\; Ay = b,\; Ry - z = 0,\; s = 1,\; y \in \mathcal{K},\; (t, s, z) \in \mathcal{K}_r^{n+2} \right\}.$$

From the constraints in (1.2), we can infer the following potential disadvantages for transforming the quadratic term in the objective into the constraints: (1) One needs to introduce an affine constraint with coefficient matrix of size $(n + 1) \times (2n + 1)$. Thus, when n is large, which is usually the case, this additional affine constraint will increase the difficulty of computing the search direction (e.g., when an interior point method is used, one needs to solve a large linear system to compute the Newton direction). (2) Introducing the extra variables (y, z, s) naturally would increase the computational complexity in solving the problem. (3) The factorization $H = R^\top R$ to begin with can be expensive to compute. The above disadvantages have motivated us to deal with (P) and (D) directly.

Optimization problems with second-order cone constraints have been studied for quite a long time and still receive constant attention to date. There is a large body of literature on the topic. For comprehensive surveys and numerous important applications of SOCPs, we refer the readers to [1, 22, 25] and references therein. Here, we mention some recent literature in the next three paragraphs to capture the main research topics on SOCPs.

Optimization problems with second-order cone constraints are of great interest theoretically due to their non-polyhedral nature. In fact, theoretical results on variational analysis for SOCPs have been well-developed. For example, Bonnans and Shapiro [7] performed rigorous and systematic perturbation analysis for nonlinear SOCPs. Outrata and Sun [33] then computed the limiting (Mordukhovich) coderivative of the metric projection onto a second-order cone, which can be used to provide a sufficient condition for the Aubin property of the solution map of a complementarity problem, as well as to derive certain necessary optimality conditions. Very recently, Hang et al. [17] conducted a second-order variational analysis for SOCPs without imposing any nondegeneracy assumptions.

The importance of SOCPs comes from their modeling power. Indeed, applications of SOCPs have grown dramatically over the years in engineering, control, management science and statistics, see for instance [4, 6, 16, 27, 30, 39, 43, 47]. As illustrative examples, we consider minimal enclosing ball problems [47], classical trust-region subproblems [30] and square-root Lasso problems [6] in this paper.

As driven by the needs in applications, many algorithms have also been developed for solving SOCPs. Among them, the most well-developed ones are interior point methods (IPMs). In particular, primal-dual IPMs have been shown to have superior theoretical and practical efficiency and they are widely used to solve SOCPs to high precision. For references on primal-dual IPMs for solving SOCPs, we recommend [2, 9, 29, 31, 32, 44]. However, IPMs are sometimes not scalable for large-scale problems due to the high expense needed to solve the large linear system of equations in each iteration. Besides IPMs, smoothing Newton methods [10, 15] and semismooth Newton methods [21] have also been applied to solve the KKT system directly. However, limited numerical implementations and experiments were conducted in these works. Therefore, the practical performance of these algorithms remains unclear. Finally, the augmented Lagrangian method (ALM) has also been applied to general nonlinear programming problems with second-order cone constraint in [18, 24]. Both papers focus on analyzing the local fast convergence rate of the ALM under some strong conditions such as the uniform second-order growth condition and the second-order sufficient condition, but with different approaches. Nevertheless, the practical performance of the ALM is not considered in both works. Therefore, the contributions in [18, 24] are mainly on the theoretical development.

Continuing the research theme on algorithmic development just mentioned above, the present paper aims to design a highly efficient and scalable algorithm for solving large-scale SOCPs. Our algorithmic design is motivated by the recent success in developing an ALM framework for solving semidefinite programming (SDP) problems. Specifically, in [46], an inexact ALM combined with a semismooth Newton method has been shown to be highly efficient and scalable for solving large-scale SDP problems. Thus, it is natural for us to apply a similar ALM framework to solve SOCPs directly. Note that this ALM framework together with its convergence analysis is well established based on the theoretical work of Rockafellar [35, 36]. Along this line, various papers (see e.g., [12, 26]) have extended Rockafellar's work by relaxing some restrictive conditions for convergence. For instance, Cui et al. [12] showed recently that under the calmness condition for the dual solution mapping (equivalently, the quadratic growth condition for the dual problem), the ALM applied to a primal convex composite conic programming problem has an asymptotic R-superlinear convergence rate in term of the KKT residual. Moreover, Cui et al. [11] showed that under the metric subregularity and bounded linear regularity conditions, the quadratic growth condition can be guaranteed for matrix optimization problems involving symmetric spectral functions. Therefore, we can borrow these ideas to establish the fast convergence rate of the ALM when applied to SOCPs. To the best of our knowledge, no such work has been done for SOCPs so far.

Our contributions in this paper can thus be summarized as follows.

- Theoretically, we provide sufficient conditions for ensuring the quadratic growth condition for the dual problem (D) under the bounded linear regularity condition and the metric subregularity condition. In particular, we revisit the fact that if a strictly complementary solution exists, then the quadratic growth condition holds for problem (D). Thus, sufficient conditions for the R-superlinear convergence of the KKT residual generated by the ALM can also be obtained.

- Numerically, we develop a highly efficient and robust SOCP solver for large-scale SOCPs. Our numerical results show that the solver is comparable to existing state-of-the-art linear SOCP solvers such as the highly powerful commercial solver Mosek and the efficient open source solver SDPT3, when solving some large-scale linear SOCPs. More specifically, we apply our SOCP solver to solve minimal enclosing ball (MEB) problems, square-root Lasso problems and some linear SOCPs in DIMACS implementation challenge. For the SOCPs arising from the MEB problems, we show that any feasible solution to the primal problem is constraint nondegenerate and hence the semismooth Newton method employed to solve the ALM subproblems is guaranteed to attain at least a superlinear convergence rate.
- For solving the convex quadratic SOCPs (P) and (D), we deal with the quadratic objective functions directly in a concise manner. We do not need to transform the problem into a much larger linear SOCP problem with an additional rotated quadratic cone constraint. The great computational benefit of our approach is demonstrated via the numerical results for solving the classical trust-region subproblems.

The rest of the paper is organized as follows. In Section 2, we introduce some preliminaries and notation which will be used in this paper. Recently developed convergence results of the ALM and related topics on the quadratic growth condition for the dual problem (D) are presented in Section 3 and Section 4. A highly efficient semismooth Newton method for solving the ALM subproblems is presented in Section 5 with some well-known convergence properties. In Section 6, we design an SOCP solver based on the proposed ALM. Moreover, we discuss the efficient implementation of the solver and conduct extensive numerical experiments to illustrate the efficiency and robustness of the proposed algorithm. Finally, we conclude the paper in Section 7.

2. Preliminaries.

In this section, we first list some notation and present some basic material on the projection operator onto the standard second-order cone.

2.1. Notation and definitions.

We use $\mathbb{Y}$, $\mathbb{Z}$ and $\mathbb{W}$ to denote generic finite-dimensional real Euclidean spaces. For a given closed convex cone $\mathcal{C}$, we use $\mathcal{C}^\circ$ and $\mathcal{C}^*$ to denote the polar and dual cones of $\mathcal{C}$, respectively. We use $N_{\mathcal{C}}(x)$ and $\mathcal{T}_{\mathcal{C}}(x)$ to denote the normal and tangent cones of $\mathcal{C}$ at a point $x \in \mathcal{C}$, respectively.

For a given convex function $f : \mathbb{W} \to [-\infty, +\infty]$, its effective domain is denoted as $\mathrm{dom}\,(f)$ and the subdifferential at the point $x \in \mathrm{dom}(\,f)$ is denoted as $\partial f(x)$. We use f^* to denote the convex conjugate function of f, i.e., $f^*(z) = \sup_x \{\langle z,\, x\rangle - f(x) \mid x \in \mathrm{dom}\,(f)\}$. Let $D \subseteq \mathbb{W}$ be a set. We use $\delta_D(\cdot)$ to denote the indicator function over the set D. If the set D is closed and convex, then the metric projection of $x \in \mathbb{W}$ onto D is defined by $\Pi_D(x) := \arg\min\{\|x - s\| \mid s \in D\}$. Moreover, the distance for a point $x \in \mathbb{W}$ to the set D is given by $\mathrm{dist}\,(x, D) := \inf_{x \in D} \|x - d\|$. For more useful properties related to convex functions and convex sets, we refer the readers to the monograph of Rockafellar [37].

For a proper closed convex function $p : \mathbb{W} \to (-\infty, +\infty]$, the proximal mapping of $p(\cdot)$ is defined as $\mathrm{Prop}_p(u) := \arg\min_x \{p(x) + \frac{1}{2}\|x - u\|^2\},\ u \in \mathbb{W}$. Note that $x = \mathrm{Prop}_p(u)$ if and only if $u - x \in \partial p(x)$.

The following definitions on the Lipschitz-like continuity for a set-valued mapping are commonly involved in derivation of the convergence rate for the ALM.

DEFINITION 2.1. 1. *A set-valued mapping* $\Phi : \mathbb{W} \rightrightarrows \mathbb{Y}$ *is Lipschitz continuous at* $u \in \mathbb{W}$ *with modulus* $\kappa \geq 0$ *if* $\Phi(u) = \{v\}$ *and there exists a positive constant* ϵ

such that

$$\|v'-v\| \le \kappa\,\|u'-u\|, \quad \forall\, v' \in \Phi(u'), \quad u' \in \mathbb{B}_\epsilon(u).$$

2. *A set-valued mapping* $\Phi : \mathbb{W} \rightrightarrows \mathbb{Y}$ *is upper Lipschitz continuous at* $u \in \mathbb{W}$ *with modulus* $\kappa \ge 0$ *if there exists a positive constant* ϵ *such that*

$$\operatorname{dist}(v', \Phi^{-1}(u)) \le \kappa\,\|u'-u\|, \quad \forall\, v' \in \Phi(u'), \quad u' \in \mathbb{B}_\epsilon(u).$$

Next, we define some mappings that are closely related to the perturbation theory of optimization problems. We will use these mappings to analyze the convergence property of the proposed ALM.

Let $l : \mathbb{X} \times \mathbb{R}^n \to [-\infty, +\infty]$ be the Lagrangian function in the extended form:

$$l(x,y) := \begin{cases} f^0(x) + \langle y, -Hx_1 + A^\top x_2 + x_3 - c\rangle & x \in \operatorname{dom}(f^0), \\ +\infty & x \notin \operatorname{dom}(f^0). \end{cases}$$

Denote the essential objective functions of (P) and (D), respectively, by

$$f(x) := \sup_y\ l(x,y) = \begin{cases} f^0(x) & -Hx_1 + A^\top x_2 + x_3 = c, \\ +\infty & \text{otherwise}, \end{cases}$$

$$g(x) := \inf_x\ l(x,y) = \begin{cases} g^0(y) & Ay = b, \\ -\infty & \text{otherwise}. \end{cases}$$

Note that the functions $l(\cdot)$, $f(\cdot)$ and $g(\cdot)$ are convex-concave, convex and concave, respectively. Therefore, their subdifferentials are well-defined. In particular, we can define the following set-valued mappings $T_l : \mathbb{X} \times \mathbb{R}^n \rightrightarrows \mathbb{X} \times \mathbb{R}^n$, $T_f : \mathbb{X} \rightrightarrows \mathbb{X}$ and $T_g : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ by

$$T_l(x,y) := \left\{(u,v) \in \mathbb{X} \times \mathbb{R}^n \mid (u,-v) \in \partial l(x,y)\right\}, \quad (x,y) \in \mathbb{X} \times \mathbb{R}^n,$$

$T_f := \partial f$, and $T_g := -\partial g$, respectively.

Consider the following linearly perturbed form of problem (P) with perturbation parameters $(u,v) \in \mathbb{X} \times \mathbb{R}^n$:

$$(\mathrm{P}(u,v)) \quad \min_x\ \left\{ f^0(x) - \langle x, u\rangle \;\middle|\; -Hx_1 + A^\top x_2 + x_3 + v - c = 0 \right\}.$$

Then according to [35], the inverse mapping of three mappings T_l, T_f and T_g are well-defined (since T_l, T_f and T_g are shown to be maximal monotone operators), and can be viewed as the solution mappings of their corresponding perturbed problems. Indeed, one can verify that

$$\begin{cases} T_l(u,v)^{-1} &= \text{ the set of all KKT points to } (\mathrm{P}(u,v)), \\ T_f(u)^{-1} &= \text{ the set of all optimal solution to } (\mathrm{P}(u,0)), \\ T_g(v)^{-1} &= \text{ the set of all optimal solution to } (\mathrm{D}(0,v)), \end{cases}$$

where $(\mathrm{D}(u,v))$ is the ordinary dual of $(\mathrm{P}(u,v))$ for any $(u,v) \in \mathbb{X} \times \mathbb{R}^n$. Therefore, we may call T_l^{-1} the KKT solution mapping, T_f^{-1} the primal solution mapping and T_g^{-1} the dual solution mapping.

2.2. Projection onto the second-order cone. We next recall some important properties on the projection onto the second-order cone. We will pay particular attention to the differential properties for the projection mapping $\Pi_K(\cdot)$, where for notational simplicity we use K to denote a single second-order cone in $\mathbb{R}^d$, i.e.,

$$K := \{x = (x_0, x_t)^\top \in \mathbb{R}^d \mid x_0 \geq \|x_t\|\}.$$

The following lemma provides an exact formula of the projection onto the second-order cone (see e.g., [15]).

LEMMA 2.2. *For any $x = (x_0, x_t)^\top \in \mathbb{R}^d$, the projection onto the second-order cone K is given by*

$$\Pi_K(x) = \begin{cases} x & \|x_t\| \leq x_0, \\ 0 & \|x_t\| \leq -x_0, \\ \frac{1}{2}(x_0 + \|x_t\|)\Big(1, \frac{x_t}{\|x_t\|}\Big)^\top & \text{otherwise.} \end{cases}$$

Since $\Pi_K(\cdot)$ is a globally Lipschitz continuous mapping with modulus 1 on $\mathbb{R}^d$, i.e.,

$$\|\Pi_K(x) - \Pi_K(y)\| \leq \|x - y\|, \quad \forall\, x,\, y \in \mathbb{R}^d,$$

it is well known that by Rademacher's Theorem [13], $\Pi_K(\cdot)$ is Fréchet differentiable almost everywhere on any open set $\mathcal{O} \subseteq \mathbb{R}^d$. Thus, we can define the B-subdifferential of $\Pi_K(\cdot)$ at a point $x \in \mathbb{R}^d$ as

$$\partial_B \Pi_K(x) := \{\lim_{i\to\infty} J\Pi_K(x^i) \mid x^i \to x,\ J\Pi_K(x^i) \text{ exists}\},$$

where $J\Pi_K(x)$ denotes the Jacobian of $\Pi_K(\cdot)$ at $x \in \mathbb{R}^d$ if it exists. Then, for any $x \in \mathbb{R}^d$, the Clarke generalized Jacobian of $\Pi_K(x)$, namely $\partial_C \Pi_K(x)$, is defined as the convex hull of $\partial_B \Pi_K(x)$. The following proposition gives the concrete expression of the elements in $\partial_B \Pi_K(x)$. We refer the readers to [34, 21, 33] for more details.

PROPOSITION 2.3. *Given an arbitrary point $x = (x_0, x_t)^\top \in \mathbb{R}^d$, each element $V \in \partial_B \Pi_K(x)$ has the following representations:*

1. *If $x_0 \neq \pm\|x_t\|$, $\Pi_K(\cdot)$ is continuously differentiable near x with*

$$J\Pi_K(x) = \begin{cases} 0 & x_0 < -\|x_t\|, \\ I_d & x_0 > \|x_t\|, \\ \frac{1}{2}\begin{pmatrix} 1 & \frac{x_t^\top}{\|x_t\|} \\ \frac{x_t}{\|x_t\|} & (1 + \frac{x_0}{\|x_t\|})I_{d-1} - \frac{x_0}{\|x_t\|^3} x_t x_t^\top \end{pmatrix} & -\|x_t\| < x_0 < \|x_t\|. \end{cases}$$

2. *If $x_t \neq 0$ and $x_0 = \|x_t\|$, then*

$$V \in \left\{ I_d, \frac{1}{2}\begin{pmatrix} 1 & \frac{x_t^\top}{\|x_t\|} \\ \frac{x_t}{\|x_t\|} & 2I_{d-1} - \frac{x_t}{\|x_t\|}\frac{x_t^\top}{\|x_t\|} \end{pmatrix} \right\}.$$

3. *If $x_t \neq 0$ and $x_0 = -\|x_t\|$, then*

$$V \in \left\{ \mathbf{0}, \frac{1}{2}\begin{pmatrix} 1 & \frac{x_t^\top}{\|x_t\|} \\ \frac{x_t}{\|x_t\|} & \frac{x_t}{\|x_t\|}\frac{x_t^\top}{\|x_t\|} \end{pmatrix} \right\}.$$

4. *If $x_t = 0$ and $x_0 = 0$, then*

$$V \in \left\{\mathbf{0}, I_d\right\} \bigcup \left\{ \frac{1}{2}\begin{pmatrix} 1 & \omega^\top \\ \omega & (1+\rho)I_{d-1} - \rho\omega\omega^\top \end{pmatrix} \;:\; |\rho| \le 1,\ \|\omega\| = 1 \right\}.$$

Recall that $\mathcal{K} = \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_r} \in \mathbb{R}^n$ is the Cartesian product of r second-order cones. It is clear that for any $x = (x_1; \cdots; x_r) \in \mathbb{R}^n$,

$$V := \mathrm{Diag}(V_1, \cdots, V_r) \in \partial_B \Pi_{\mathcal{K}}(x), \quad V_j \in \partial_B \Pi_{\mathcal{K}^{n_j}}(x_j), \quad 1 \le j \le r.$$

To apply the semismooth Newton method for solving the ALM subproblems presented later in the paper, we also need the concept of semismoothness.

DEFINITION 2.4. *Let $\Phi : \mathbb{W} \to \mathbb{Y}$ be a locally Lipschitz continuous function on the open set $\mathcal{O} \subseteq \mathbb{W}$. Φ is said to be semismooth at a point $x \in \mathcal{O}$ if Φ is directionally differentiable at x and for any $V \in \partial_C \Phi(x + \Delta x)$,*

$$\Phi(x + \Delta x) - \Phi(x) - V\Delta x = o(\|\Delta x\|), \quad \Delta x \to 0.$$

Φ is said to be strongly semismooth at $x \in \mathcal{O}$ if Φ is semismooth at x and for any $V \in \partial_C \Phi(x + \Delta x)$,

$$\Phi(x + \Delta x) - \Phi(x) - V\Delta x = o(\|\Delta x\|^2), \quad \Delta x \to 0.$$

Φ is said to be a (strongly) semismooth function on $\mathcal{O}$ if it is (strongly) semismooth for every point $x \in \mathcal{O}$.

The next lemma shows that $\Pi_{\mathcal{K}}(\cdot)$ is strongly semismooth on $\mathbb{R}^n$. For a proof of this lemma, see [10, 19].

LEMMA 2.5. *The projection mapping $\Pi_{\mathcal{K}}(\cdot)$ is strongly semismooth everywhere.*

3. Convergence results of the ALM. In this section, we analyze the convergence properties of the ALM applied to problem (P). Even though the theory has been highly developed, we present certain important results here to make our paper self-contained.

Let $\sigma > 0$ be a given penalty parameter. The augmented Lagrangian function associated with problem (P) for any $(x, y) \in \mathbb{X} \times \mathbb{R}^n$ is defined as

$$L_\sigma(x, y) := f^0(x) + \frac{1}{2\sigma}\left(\left\| \sigma\big(-Hx_1 + A^\top x_2 + x_3 - c\big) + y \right\|^2 - \|y\|^2 \right).$$

At the $(k+1)$-th iteration, for a given sequence of penalty parameters $0 < \sigma_k \uparrow \sigma_\infty \le \infty$ and an initial point $y^0 \in \mathbb{R}^n$, the inexact ALM performs the following scheme:

$$\text{(3.1)} \qquad \begin{cases} x^{k+1} := (x_1^{k+1}, x_2^{k+1}, x_3^{k+1}) \;\approx\; \arg\min\limits_{x} \{f_k(x) := L_{\sigma_k}(x, y^k)\}, \\ y^{k+1} \;:=\; y^k + \sigma(-Hx_1^{k+1} + A^\top x_2^{k+1} + x_3^{k+1} - c), \quad k \ge 0. \end{cases}$$

The rate of convergence for the ALM can be obtained by considering its connection with the dual proximal point algorithm (PPA). This connection was explored in Rockafellar's classical papers [35, 36]. More specifically, by combining Theorem 4 and Theorem 5 in [35], one obtains the following fundamental convergence result for ALM.

THEOREM 3.1. *Assume that $\mathrm{SOL_D}$ is nonempty, i.e., $T_g^{-1}(0) \neq \emptyset$. Let $\{(x^k, y^k)\}$ be the infinite sequence generated by the ALM in (3.1) under the following criterion for inexact computation*

$$\text{(A)} \quad f_k(x^{k+1}) - \inf\ f_k \le \frac{\epsilon_k^2}{2\sigma_k},$$

where $\{\epsilon_k\}$ *is a summable sequence in* $\mathbb{R}$*. Then the whole sequence* $\{y^k\}$ *converges to some* $y^\infty \in \mathrm{SOL}_D$*.*

If T_g^{-1} *is Lipschitz continuous at the origin with modulus* $\kappa_g > 0$ *and the ALM is also executed under the following criterion*

$$(\mathrm{B}) \quad f_k(x^{k+1}) - \inf\ f_k \le \frac{\delta_k^2}{2\sigma_k} \left\| y^{k+1} - y^k \right\|^2$$

with a summable sequence $\{\delta_k\}$*. Then* $y^k \to y^\infty$ *as* $k \to \infty$*, where in this case,* y^∞ *is the unique solution for problem (D). Furthermore, it holds that*

$$\left\| y^{k+1} - y^\infty \right\| \le \frac{\kappa_g(\kappa_g^2 + \sigma_k^2)^{-1/2} + \delta_k}{1 - \delta_k} \left\| y^k - y^\infty \right\|$$

for all k *sufficiently large.*

Remark 3.2. Note that the Lipschitz continuity assumption on T_g^{-1} is rather restrictive, since it requires the solution set $T_g^{-1}(0)$ to be a singleton. In [26], Luque extended Rockafellar's original results by relaxing the Lipschitz continuity condition to the upper Lipschitz continuity condition. The latter condition is satisfied if the corresponding set-valued mapping is piecewise polyhedral (see Sun's PhD thesis [40] for more discussions on these mappings). However, in this paper, we consider the mapping involving the non-polyhedral second-order cone; thus, more relaxed conditions might be needed.

The classical convergence results for the ALM (or equivalently PPA) are of great value both theoretically and numerically. However, there are two practical issues to be resolved. Firstly, we can only obtain the rate of convergence for the dual sequence $\{y^k\}$ generated by the ALM, but the rate of convergence for the primal sequence $\{x^k\}$ is not known. Even though [12, Proposition 3] has provided a convergence result for $\{x^k\}$ under the upper Lipschitz continuity condition of T_l^{-1}, the Lipschitz-like condition is quite restrictive as explained in [12]. Thus, instead of requiring the convergence of $\{x^k\}$ when designing a solver, in our opinion, a more reasonable requirement is the convergence of the KKT residual of the computed primal-dual sequence $\{(x^k, y^k)\}$. Secondly, the stopping criteria used in the theoretical analysis are not implementable since they require some unknown information (e.g., $\inf\ f_k$). Fortunately, these issues are resolved in [12] by conducting finer analysis of the ALM applied to the dual problem. We shall summarize these results in the rest of this subsection.

To proceed, we first need the following definition of quadratic growth condition and assumption of Robinson constraint qualification.

DEFINITION 3.3. *The quadratic growth condition holds at an optimal solution* $\bar{y} \in \mathrm{SOL_D}$ *if there exist positive constants* κ *and* ϵ *such that*

$$-g^0(y) \ge -g^0(\bar{y}) + \kappa\,\mathrm{dist}^2(y, \mathrm{SOL_D}), \quad \forall y \in \mathbb{B}_\epsilon(\bar{y}) \cap \{y \in \mathbb{R}^n \mid Ay = b\}\,. \tag{3.2}$$

Assumption 3.4. The solution set $\mathrm{SOL_D}$ for the problem (D) is non-empty and the following Robinson constraint qualification (RCQ) of the problem (D) hold at some $\bar{y} \in \mathrm{SOL_D}$:

$$0 \in \mathrm{int}\left\{ \begin{pmatrix} b \\ \bar{y} \end{pmatrix} + \begin{pmatrix} A & 0 \\ 0 & I_n \end{pmatrix} \begin{pmatrix} \mathbb{R}^n \\ \mathbb{R}^n \end{pmatrix} - \begin{pmatrix} \{0\} \\ \mathcal{K} \end{pmatrix} \right\}.$$

By [7, Theorem 3.9], the optimal solution set $\mathrm{SOL_P}$ to the problem (P) is nonempty and bounded under the Assumption 3.4.

For any $k \geq 0$, $y^k \in \mathbb{R}^n$, $x_1 \in \mathbb{R}^n$ and $x_2 \in \mathbb{R}^m$, denote

$$
\begin{cases}
\tilde{y}^k(x_1, x_2) := \Pi_{\mathcal{K}}\left[y^k + \sigma_k\big(-Hx_1 + A^\top x_2 - c\big)\right], \\
\tilde{x}^k(x_1, x_2) := \big(x_1, x_2, \tilde{y}^k(x_1, x_2)\big)^\top \in \mathbb{X}, \\
e^k(x_1, x_2) := \begin{pmatrix} Hx_1 - H\tilde{y}^k(x) \\ -b + A\tilde{y}^k(x) \\ 0 \end{pmatrix}.
\end{cases} \tag{3.3}
$$

Let $\{\hat{\epsilon}_k\}$ and $\{\hat{\delta}_k\}$ be two summable nonnegative sequences. For inexact computations, we adopt the following stopping criteria

$$
\begin{aligned}
&(\mathrm{A}') \quad \left\|e^{k+1}\right\| \leq \frac{\hat{\epsilon}_k^2/\sigma_k}{1 + \|x^{k+1}\| + \|y^{k+1}\|} \min\left\{1, \frac{1}{\|Hy^{k+1}\| + \|y^{k+1} - y^k\|/\sigma_k + 1/\sigma_k}\right\}, \\
&(\mathrm{B}') \quad \left\|e^{k+1}\right\| \leq \frac{(\hat{\delta}_k^2/\sigma_k)\left\|y^{k+1} - y^k\right\|^2}{1 + \|x^{k+1}\| + \|y^{k+1}\|} \min\left\{1, \frac{1}{\|Hy^{k+1}\| + \|y^{k+1} - y^k\|/\sigma_k + 1/\sigma_k}\right\},
\end{aligned}
$$

where

$$
y^{k+1} := \tilde{y}^k(x_1^{k+1}, x_2^{k+1}), \quad x^{k+1} := \tilde{x}^k(x_1^{k+1}, x_2^{k+1}), \quad e^{k+1} := e^k(x_1^{k+1}, x_2^{k+1}).
$$

We can see that the above stopping criteria are truly implementable and hence they are more useful for practical purposes than the classical ones (i.e., criteria (A) and (B)).

Based on the KKT optimality condition (1.1), we define the natural map

$$
R^{\mathrm{nat}}(x, y) := \begin{pmatrix} Hx_1 - Hy \\ -b + Ay \\ x_3 - \Pi_{\mathcal{K}}(x_3 - y) \\ Hx_1 - A^\top x_2 - x_3 + c \end{pmatrix}, \quad \forall\, x = (x_1, x_2, x_3) \in \mathbb{X},\ y \in \mathbb{R}^n. \tag{3.4}
$$

The following theorem is taken from [12, Theorem 2] which provides the R-superlinear convergence of the KKT residual.

THEOREM 3.5. *Suppose that Assumption 3.4 holds. Let $\{(x^k, y^k)\}$ be an infinite sequence generated by the ALM in (3.1) under the criterion (A'). Then the sequence $\{y^k\}$ is bounded and converges to some $y^\infty \in \mathrm{SOL_D}$. Moreover, the sequence $\{x^k\}$ is also bounded with all of its limit points in $\mathrm{SOL_P}$.*

If criterion (B') is also executed in the ALM and the quadratic growth condition holds at y^∞ with modulus $\hat{\kappa}_g > 0$. Then there exist a positive constant α and an integer $\bar{k} \geq 0$ such that for all $k \geq \bar{k}$, $\alpha\hat{\delta}_k < 1$ and

$$
\mathrm{dist}\,(y^{k+1}, \mathrm{SOL_D}) \leq \theta_k \mathrm{dist}\,(y^k, \mathrm{SOL_D}), \quad \left\|R^{\mathrm{nat}}(x^{k+1}, y^{k+1})\right\| \leq \theta_k' \mathrm{dist}\,(y^k, \mathrm{SOL_D}),
$$

where

$$
\theta_k := \frac{1}{1 - \alpha\hat{\delta}_k}\left(\alpha\hat{\delta}_k + \frac{\alpha\hat{\delta}_k}{\sqrt{1 + \sigma_k^2\hat{\kappa}_g^2}}\right),
$$

$$
\theta_k' := \frac{1}{1 - \alpha\hat{\delta}_k}\left(\max\left\{1, \frac{1}{\sigma_k}\right\} + \frac{\hat{\delta}_k^2}{\sigma_k}\left\|y^{k+1} - y^k\right\|\right).
$$

One can observe from the above theorem that when $\sigma_k \uparrow \sigma_\infty \leq \infty$,

$$\theta_k \to \theta_\infty := \frac{1}{\sqrt{1+\sigma_\infty^2 \hat{\kappa}_g^2}}, \quad \theta'_k \to \theta'_\infty := \max\left\{1, \frac{1}{\sigma_\infty}\right\}.$$

Thus θ_∞ can be arbitrarily close to zero if σ_∞ is sufficiently large. This implies that the linear convergence rate for the sequence $\{\mathrm{dist}(y^k, \mathrm{SOL}_D)\}$ can be arbitrarily small. Moreover, since $\theta'_\infty \leq 1$, the KKT residual also converges as rapidly as $\{\mathrm{dist}(y^k, \mathrm{SOL}_D)\}$. These convergence properties may explain partially the highly efficiency of the ALM, as we shall see in our numerical experiments.

4. Quadratic growth condition.

In this section, we analyze the quadratic growth condition for the dual problem (D), which serves as a sufficient condition for the KKT residual generated by the ALM to achieve the R-superlinear convergence rate (see Theorem 3.5). In the recent work of Cui et al. [11], two types of sufficient conditions were proposed to ensure the quadratic growth condition. Here in this section, we will follow one of the available frameworks in [11] to provide a sufficient condition for the quadratic growth condition under the bounded linear regularity and metric subregularity conditions.

Recall that since $H \succeq \mathbf{0}$, there exists $R \in \mathbb{R}^{n\times n}$ such that $H = R^\top R$. Denote $F_D := \{ y \in \mathbb{R}^n \mid Ay = b \}$. Then problem (D) can be reformulated as

$$\max_y \left\{ g^0(y) := -\frac{1}{2}\|Ry\|^2 - \langle c, y\rangle - p(y) \;\middle|\; y \in F_D \right\},$$

where $p(\cdot) = \delta_{\mathcal{K}}(\cdot)$. Moreover, the KKT optimality condition (1.1) can be rewritten as

$$0 \in R^\top R y + c + \partial p(y) - A^\top x_2, \quad Ay - b = 0, \quad \forall\, (x_2, y) \in \mathbb{R}^m \times \mathbb{R}^n. \tag{4.1}$$

Take any $\bar{y} \in \mathrm{SOL_D}$. Denote

$$\bar{\zeta} := R\bar{y}, \quad \overline{\mathcal{V}} := \{ y \in \mathbb{R}^n \mid Ry = \bar{\zeta} \}$$

and define the set-valued mapping $\mathcal{G} : \mathbb{X} \rightrightarrows \mathbb{R}^n$ as

$$\mathcal{G}(x) := (\partial p)^{-1}\left(A^\top x_2 - R^\top \bar{\zeta} - c\right), \quad \forall\, x = (x_1, x_2, x_3) \in \mathbb{X}.$$

Then, we have the following characterization for the optimal solution set $\mathrm{SOL_D}$.

PROPOSITION 4.1. *Assume that* $\bar{y} \in \mathrm{SOL_D}$ *and* $\bar{x} = (\bar{x}_1, \bar{x}_2, \bar{x}_3) \in \mathrm{SOL_P}$. *Then the optimal solution set* $\mathrm{SOL_D}$ *can be characterized as*

$$\mathrm{SOL_D} = \overline{\mathcal{V}} \cap \mathcal{G}(\bar{x}) \cap F_D.$$

Proof. We only have to show that for any $y, y' \in \mathrm{SOL_D}$, it holds that $Ry = Ry'$. This is equivalent to say that the value Ry is invariant over $y \in \mathrm{SOL_D}$. However, such a fact is already well known in the literature, see for instance [28]. ∎

Next, we recall the concept of bounded linear regularity of a collection of closed convex sets. This concept is useful for analyzing error bound properties for constrained optimization problems.

DEFINITION 4.2. *Let* $D_1, \cdots, D_q$ *be some closed convex sets in a finite dimensional Euclidean space* $\mathbb{W}$. *Suppose that* $D := D_1 \cap \cdots \cap D_q$ *is non-empty. The collection* $\{D_1, \cdots, D_q\}$ *is said to be boundedly linearly regular if for every bounded set* $B \subseteq \mathbb{W}$, *there exists a positive constant* κ *such that*

$$\mathrm{dist}\,(x, D) \;\leq\; \kappa \max\{\mathrm{dist}\,(x, D_1), \cdots, \mathrm{dist}\,(x, D_q)\}, \quad \forall\, x \in B.$$

However, checking the condition in Definition 4.2 is not a trivial task. In [5, Corollary 3], the authors established the following simpler sufficient condition.

PROPOSITION 4.3. *Let $D_1, \cdots, D_q$ be some closed convex sets in a finite dimensional Euclidean space $\mathbb{W}$. Suppose that $D_1, \cdots, D_{q_1}$ are polyhedral for some $0 \le q_1 \le q$. Then a sufficient condition for the collection $\{D_1, \cdots, D_q\}$ to be boundedly linearly regular is*

$$\bigcap_{1\le i\le q_1} D_i \ \cap \bigcap_{q_1+1\le i\le q} \mathrm{ri}\,(D_i) \ \neq \ \emptyset.$$

We next introduce the definition of metric subregularity.

DEFINITION 4.4. *A multifunction $\Phi : \mathbb{W} \rightrightarrows \mathbb{Y}$ is said to be metrically subregular at $\bar{x} \in \mathbb{W}$ for $\bar{v} \in \mathbb{Y}$ if $(x, v) \in \mathrm{gph}\,(\Phi)$ and there exist positive constants κ and ϵ such that*

$$\mathrm{dist}\,(x, \Phi^{-1}(\bar{v})) \le \kappa\,\mathrm{dist}\,(\bar{v}, \Phi(x)), \quad \forall\, x \in \mathbb{B}_\epsilon(\bar{x}).$$

For a general multifunction, it could be difficult to check the metric subregularity directly since the graph of the multifunction at the reference point may contain infinitely many points. Fortunately, when the multifunction is the subdifferential of a proper closed convex function, it has a more convenient characterization as shown in the next proposition.

PROPOSITION 4.5. *Let $\mathbb{W}$ be a real Hilbert space endowed with the inner product $\langle\cdot, \cdot\rangle$ and $p : \mathbb{W} \to (-\infty, +\infty]$ be a proper closed convex function. Let $(\bar{v}, \bar{x}) \in \mathbb{W} \times \mathbb{W}$ such that $\bar{v} \in \partial p(\bar{x})$. Then ∂p is metrically subregular at $\bar{x}$ for $\bar{v}$ if and only if there exist positive constants κ and ϵ such that*

$$p(x) \ge p(\bar{x}) + \langle \bar{v},\, x - \bar{x}\rangle + \kappa\,\mathrm{dist}\,(x, (\partial p)^{-1}(\bar{v})), \quad \forall\, x \in \mathbb{B}_\epsilon(\bar{x}).$$

The proof of Proposition 4.5 can be found in [3, Theorem 3.3]. Next proposition states that $\partial p(\cdot) = \partial\delta_{\mathcal{K}}(\cdot) = \mathcal{N}_{\mathcal{K}}(\cdot)$ is indeed metrically subregular.

PROPOSITION 4.6. *Let $\mathcal{K} = \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_r} \subseteq \mathbb{R}^n$ be the Cartesian product of some second-order cones with $\mathcal{K}^{n_i} \subseteq \mathbb{R}^{n_i}$, $i = 1, \cdots, r$, and $n = n_1 + \cdots + n_r$. For any $(x, v) \in \mathrm{gph}(\partial\delta_{\mathcal{K}})$ i.e., $v \in \mathcal{N}_{\mathcal{K}}(x)$, $\partial\delta_{\mathcal{K}}(\cdot)$ is metrically subregular at x for v.*

Proof. Since the metric subregularity of $\mathcal{N}_{\mathcal{K}}$ is implied by the metric subregularity of each $\mathcal{N}_{\mathcal{K}^{n_i}}$ for $i = 1, \ldots, r$, we only need to check that for a standard second-order cone K, $\mathcal{N}_K(\cdot)$ is metrically subregular at any point on its graph. The latter has been shown in [41] as a special case of the results for the p-order conic constraint system. Thus, the proof is completed. ∎

After all the previous preparations, we are now able to provide a sufficient condition for the quadratic growth condition for problem (D) to hold. The next theorem, which is taken from [11], provides a general framework to establish the sufficient condition for the quadratic growth condition. To make the paper self-contained and to explain the idea more clearly, we provide a proof that is restricted to SOCPs.

THEOREM 4.7. *Assume that $\mathrm{SOL_D}$ is nonempty and that there exists $\bar{x} \in \mathrm{SOL_P}$ such that the collection $\{\overline{\mathcal{V}}, \mathcal{G}(\bar{x})\}$ is boundedly linearly regular. Then the quadratic growth condition holds for problem (D) at any point $\bar{y} \in \mathrm{SOL_D}$.*

Proof. Let $\bar{y} \in \mathrm{SOL_D}$, $\bar{x} = (\bar{x}_1, \bar{x}_2, \bar{x}_3) \in \mathrm{SOL_P}$ and $\epsilon > 0$. Then for any $y \in F_D \cap \mathbb{B}_\epsilon(\bar{y})$ we have that there exist $\kappa_2 > 0$ and $\kappa_3 > 0$ such that

$$\begin{aligned} \mathrm{dist}^2(y, \mathrm{SOL_D}) &= \mathrm{dist}^2\left(y, \overline{\mathcal{V}} \cap \mathcal{G}(\bar{x})\right) \\ &\le \kappa_2\left[\mathrm{dist}^2\left(y, \overline{\mathcal{V}}\right) + \mathrm{dist}^2\left(y, \mathcal{G}(\bar{x})\right)\right] \\ &\le \kappa_3\left[\left\|Ry - \bar{\zeta}\right\|^2 + \mathrm{dist}^2\left(y, (\partial p)^{-1}(A^\top x_2 - \bar{\eta})\right)\right], \end{aligned}$$

under the assumption that $\{\overline{\mathcal{V}}, \mathcal{G}(\bar{x})\}$ is boundedly linearly regular. Note that in the last inequality of above, the first term makes use of Hoffman's error bound [20].

By Proposition 4.5, Proposition 4.6 and $(\bar{y}, A^\top \bar{x}_2 - R^\top \bar{\zeta} - c) \in \mathrm{gph}(\partial p)$, we know that there exists $\kappa_p > 0$ such that for any $y \in \mathbb{B}_\epsilon(\bar{y})$, it holds that (by shrinking ϵ if necessary),

$$p(y) - p(\bar{y}) \geq \langle A^\top \bar{x}_2 - R^\top \bar{\zeta} - c,\, y - \bar{y}\rangle + \kappa_p \mathrm{dist}^2\,(y, (\partial p)^{-1}(A^\top \bar{x}_2 - R^\top \bar{\zeta} - c)).$$

By combining all the obtained inequalities, we have that for any $y \in F_D \cap \mathbb{B}_\epsilon(\bar{y})$,

$$\begin{aligned}
-g^0(y) &= \frac{1}{2}\|Ry\|^2 + \langle c,\, y\rangle + p(y) \\
&\geq \frac{1}{2}\|\bar{\zeta}\|^2 + \langle \bar{\zeta},\, Ry - \bar{\zeta}\rangle + \frac{1}{2}\left\|Ry - \bar{\zeta}\right\|^2 + \langle c,\, y\rangle \\
&\quad + p(\bar{y}) + \langle A^\top \bar{x}_2 - \bar{\eta},\, y - \bar{y}\rangle + \kappa_p\, \mathrm{dist}^2\left(y, (\partial p)^{-1}(A^\top \bar{x}_2 - R^\top \bar{\zeta} - c)\right) \\
&= -g^0(\bar{y}) + \frac{1}{2}\left\|Ry - \bar{\zeta}\right\|^2 + \kappa_p\, \mathrm{dist}^2\big(y, (\partial p)^{-1}(A^\top \bar{x}_2 - R^\top \bar{\zeta} - c)\big) \\
&\geq -g^0(\bar{y}) + \kappa_3^{-1} \min\{\kappa_p, \frac{1}{2}\}\, \mathrm{dist}^2\,(x, \mathrm{SOL_D}),
\end{aligned}$$

which is exactly the quadratic growth condition for problem (D). Therefore, the proof is completed. ∎

By the definitions of $\overline{\mathcal{V}}$ and $\mathrm{F_D}$, it is obvious that both sets are polyhedral. However, $\mathcal{G}(\cdot)$ is not always polyhedral. Indeed, let $\bar{x}_3 := -A^\top \bar{x}_2 + R^\top \bar{\zeta} + c = ((\bar{x}_3)_1, \cdots, (\bar{x}_3)_r)^T \in \mathbb{R}^n$, since (see e.g., [8])

$$(\partial \delta_{\mathcal{K}^{n_i}})^{-1}\,(-(\bar{x}_3)_i) = \mathcal{N}_{(\mathcal{K}^{n_i})^\circ}(-(\bar{x}_3)_i) = \begin{cases} \{0\} & (\bar{x}_3)_i \in \mathrm{int}\,\mathcal{K}^{n_i}, \\ \mathcal{K}^{n_i} & (\bar{x}_3)_i = 0, \\ \mathbb{R}_+(-(\bar{x}_3)_{i,0}, (\bar{x}_3)_{i,t}) & (\bar{x}_3)_i \in \mathrm{bd}\,\mathcal{K}^{n_i} \backslash \{0\}, \end{cases}$$

we can see that $\mathcal{G}(\bar{x})$ is polyhedral if and only if $(\bar{x}_3)_i \neq 0, \forall\, 1 \leq i \leq r$. As a consequence, given $\bar{x}_2 \in \mathbb{R}^m$, let J be the index set defined as $J := \{i \mid (\bar{x}_3)_i = 0\}$, if there exists $\bar{y} = (\bar{y}_1, \cdots, \bar{y}_r)^\top \in \mathbb{R}^n$ such that $(\bar{x}_2, \bar{y})$ solves the KKT system (4.1) and $\bar{y}_i \in \mathrm{int}\,\mathcal{K}^{n_i}$, $\forall\, i \in J$. Then by Proposition 4.3, the collection $\{\overline{\mathcal{V}}, \mathcal{G}(\bar{x})\}$ is boundedly linearly regular and hence the quadratic growth condition for the problem (D) holds at any optimal solution. The aforementioned conclusion on $(\bar{x}_2, \bar{y})$ is summarized as follows.

COROLLARY 4.8. *Let $(\bar{x}_2, \bar{y})$ be a solution of the KKT system* (4.1) *and $\bar{x}_3 = -A^\top \bar{x}_2 + R^\top \bar{\zeta} + c$. If $(\bar{x}_3, \bar{y})$ is a strictly complementary solution, i.e., $\bar{y}_i + (\bar{x}_3)_i \in \mathrm{int}\,\mathcal{K}^{n_i}$ holds for all block components $i = 1, \cdots, r$. Then the quadratic growth condition holds at any solution of the problem* (D).

Proof. By [1, Corollary 24], we know that $(\bar{x}_2, \bar{y})$ is strictly complementary if and only if for each block $1 \leq i \leq r$, either both $\bar{y}_i$ and $(\bar{x}_3)_i$ are nonzero and in the $\mathrm{bd}\,\mathcal{K}^{n_i}$, or if one is zero, the other is in the interior of $\mathcal{K}^{n_i}$. Then by the above discussions, the conclusion can be derived in a straight-forward way. ∎

5. Solving the ALM subproblem by an inexact semismooth Newton method.

In this section, we propose an inexact semismooth Newton method for solving subproblems arsing from the ALM in (3.1) applied to the problem (P).

For given y and σ, denote $\tilde{y}(x_1, x_2, y) := Hx_1 - A^\top x_2 - y/\sigma + c$. Recall that the ALM subproblem is given by

$$(5.1) \qquad (x_1^+, x_2^+, x_3^+) = \underset{(x_1,x_2,x_3)\in\mathbb{X}}{\mathrm{argmin}} \left\{ \begin{array}{l} \frac{1}{2}\langle x_1,\, Hx_1\rangle - \langle b,\, x_2\rangle + \delta_{\mathcal{K}}(x_3) \\ +\frac{1}{2\sigma}\left(\|\sigma(x_3 - \tilde{y}(x_1, x_2, y))\|^2 - \|y\|^2\right) \end{array} \right\}.$$

By simple calculations, we have

$$x_3^+ = \Pi_{\mathcal{K}}\left(\tilde{y}(x_1^+, x_2^+, y)\right). \tag{5.2}$$

Therefore, by using the Moreau identity, we obtain that to solve the problem (5.1), it is equivalent to solve

$$\min_{x_1, x_2} \ \psi(x_1, x_2) := \frac{1}{2}\langle x_1,\, Hx_1\rangle - \langle b,\, x_2\rangle + \frac{1}{2\sigma}\left(\|\Pi_{\mathcal{K}}\left[-\sigma\tilde{y}(x_1, x_2, y)\right]\|^2 - \|y\|^2\right). \tag{5.3}$$

Once x_1^+ and x_2^+ have been computed, we can obtain x_3^+ via (5.2). Furthermore, to solve the above unconstrained minimization problem with respect to $(x_1, x_2) \in \mathrm{Ran}(H) \times \mathbb{R}^m$, it is equivalent to solve the following system of nonsmooth equations:

$$\nabla\psi(x_1, x_2) = \begin{pmatrix} Hx_1 - H\Pi_{\mathcal{K}}\left[-\sigma\tilde{y}(x_1, x_2, y)\right] \\ -b + A\Pi_{\mathcal{K}}\left[-\sigma\tilde{y}(x_1, x_2, y)\right] \end{pmatrix} = 0. \tag{5.4}$$

Since $\Pi_{\mathcal{K}}(\cdot)$ is strongly semismooth everywhere (by Proposition 2.5), it is desirable to apply a semismooth Newton method to solve the above system of nonsmooth equations as one could expect a superlinear *or* even quadratic convergence rate. To this end, for any $(x_1, x_2) \in \mathrm{Ran}(H) \times \mathbb{R}^m$, we define

$$\hat{\partial}^2\psi(x_1, x_2) := \begin{pmatrix} H & \\ & 0 \end{pmatrix} + \sigma\begin{pmatrix} H \\ -A \end{pmatrix} \partial_C \Pi_{\mathcal{K}}\left[-\sigma\tilde{y}(x_1, x_2, y)\right] \begin{pmatrix} H & -A^\top \end{pmatrix}.$$

Then $\hat{\partial}^2\psi(x_1, x_2)$ can serve as a replacement of the (hard-to-characterize) generalized Hessian of ψ at (x_1, x_2), namely $\partial^2\psi(x_1, x_2)$, in the sense that for any $d_1 \in \mathrm{Ran}(H)$ and $d_2 \in \mathbb{R}^m$,

$$\hat{\partial}^2\psi(x_1, x_2)\left(\begin{pmatrix} d_1 \\ d_2 \end{pmatrix}\right) = \partial^2\psi(x_1, x_2)\left(\begin{pmatrix} d_1 \\ d_2 \end{pmatrix}\right).$$

Next we present the well-known inexact semismooth Newton method (as Algorithm 5.1) in [46] to solve (5.3) as in Algorithm 5.1.

The convergence of Algorithm 5.1 is given by the next theorem under the following assumption.

Assumption 5.1. The linear mapping $A : \mathbb{R}^n \to \mathbb{R}^m$ is onto and there exists $\hat{y} \in \mathrm{int}\,\mathcal{K}$ such that $A\hat{y} = b$.

THEOREM 5.2. *Suppose that Assumption 5.1 holds. Then Algorithm 5.1 generates a bounded sequence $\{(x_1^j, x_2^j)\}$ such that any of its accumulation point is an optimal solution to problem* (5.3).

Readers may refer to [46, Theorem 3.4] for a proof of Theorem 5.2. To obtain a fast superlinear convergence rate *or* even a quadratic convergence rate of Algorithm 5.1, one needs the positive definiteness of the coefficient matrix in the linear system at the solution point. Establishing conditions that ensure the positive definiteness of the coefficient matrix is important for the convergence analysis. Next theorem provides the convergence rate of the algorithm under the constraint nondegeneracy condition, whose proof can be done by combining the results from [23, Proposition 3.1, Theorem 3.2] and [46, Proposition 3.2, Theorem 3.5].

THEOREM 5.3. *Suppose that Assumption 5.1 holds. Let $(\hat{x}_1, \hat{x}_2)$ be an accumulation point of the infinite sequence $\{(x_1^j, x_2^j)\}$ generated by Algorithm* iSSN *for problem* (5.3). *Let*

Algorithm 5.1 Algorithm iSSN: An inexact semismooth Newton method (iSSN(y, σ)).

Given $\hat{\nu} \in (0,1)$, $\tau \in (0,1]$, $\tau_1, \tau_2 \in (0,1)$, and $\mu \in (0,1/2)$, $\delta \in (0,1)$. Choose $(x_1^0, x_2^0) \in \mathrm{Ran}(H) \times \mathbb{R}^m$. Perform the following iterations for $j = 0,1,2,\cdots$,

Step 1. Set $\epsilon_j := \tau_1 \min\left\{\tau_2, \left\|\nabla\psi(x_1^j, x_2^j)\right\|\right\}$ and $\nu_j := \min\left\{\hat{\nu}, \left\|\nabla\psi(x_1^j, x_2^j)\right\|^{1+\tau}\right\}$.
Find $(d_1^j, d_2^j) \in \mathrm{Ran}(H) \times \mathbb{R}^m$ by solving the following linear system approximately

$$M_j \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} + \epsilon_j \begin{pmatrix} 0 \\ d_2 \end{pmatrix} + \nabla\psi(x_1^j, x_2^j) = 0, \quad M_j \in \hat{\partial}^2\psi(x_1^j, x_2^j)$$

in the sense that

$$\left\| M_j \begin{pmatrix} d_1^j \\ d_2^j \end{pmatrix} + \epsilon_j \begin{pmatrix} 0 \\ d_2^j \end{pmatrix} + \nabla\psi(x_1^j, x_2^j) \right\| \le \nu_j.$$

Step 2. Set $\alpha_j = \delta^{m_j}$ where m_j is the smallest non-negative integer m for which

$$\psi(x_1^j + \delta^m d_1^j, x_2^j + \delta^m d_2^j) \le \psi(x_1^j, x_2^j) + \mu\delta^m \langle \nabla\psi(x_1^j, x_2^j), \begin{pmatrix} d_1^j \\ d_2^j \end{pmatrix} \rangle.$$

Step 3. Set $x_1^{j+1} = x_1^j + \alpha_j d_1^j$ and $x_2^{j+1} = x_2^j + \alpha_j d_2^j$.

$\hat{y} := \Pi_{\mathcal{K}}\left(-\sigma\tilde{y}(\hat{x}_1, \hat{x}_2, y)\right)$. *Assume that the following constraint nondegeneracy condition holds*

$$A \,\mathrm{lin}\left(\mathcal{T}_{\mathcal{K}}(\hat{y})\right) = \mathbb{R}^m,$$

where $\mathrm{lin}\left(\mathcal{T}_{\mathcal{K}}(\hat{y})\right)$ *denotes the lineality space of the tangent cone of* $\mathcal{K}$ *at* $\hat{y}$*. Then, the whole sequence* $\{(x_1^j, x_2^j)\}$ *converges to* $(\hat{x}_1, \hat{x}_2)$ *and*

$$\left\|(x_1^{j+1}, x_2^{j+1}) - (\hat{x}_1, \hat{x}_2)\right\| = O\left(\left\|(x_1^j, x_2^j) - (\hat{x}_1, \hat{x}_2)\right\|^{1+\tau}\right).$$

Remark 5.4. The constraint nondegeneracy condition in the above theorem could be hard to verify since the accumulation point $(\hat{x}_1, \hat{x}_2)$ is usually not known. Fortunately, for some special problems one may check that this condition holds at any feasible solution. For such an example, see Theorem 6.1 in Section 6.3 on solving minimal enclosing ball problems.

Note that under the constraint nondegeneracy condition, one can show that every element in $\hat{\partial}^2\psi(\hat{x}_1, \hat{x}_2)$ is self-adjoint and positive definite on $\mathrm{Ran}(H) \times \mathbb{R}^m$, see [23, Theorem 3.2]. It is also clear that if H is not positive definite on $\mathbb{R}^n$, then $\mathrm{Ran}(H) \neq \mathbb{R}^n$. Thus, if one replaces $\mathrm{Ran}(H)$ by any linear subspace of $\mathbb{R}^n$ strictly containing $\mathrm{Ran}(H)$ in the formulation of problem (P), then the local fast convergence rate for Algorithm iSSN will be lost. As a result, the restriction $x_1 \in \mathrm{Ran}(H)$ in problem (P) in fact plays a crucial role in our algorithmic framework. We will discuss later in Section 6.1 on how to implement the restriction $(d_1, d_2) \in \mathrm{Ran}(H) \times \mathbb{R}^m$ when solving the linear system in Algorithm iSSN.

We end this section by emphasizing that our ALM equipped with a semismooth Newton method for solving the ALM subproblems is an inner-outer loop algorithm. By our convergence analysis, both the inner loop and the outer loop have fast convergence rates under some technical assumptions. Thus, our present algorithm is a "fast+fast" algorithm.

6. Numerical implementation and experiments. In this section, we aim to design an efficient solver for the following SOCP problem

$$\min_{x_1,x_2,x_3} \left\{ \frac{1}{2}\langle x_1,\, Hx_1\rangle - \langle b,\, x_2\rangle \;\middle|\; \begin{array}{c} -Hx_1 + A^\top x_2 + x_3 = c, \\ x_3 = \big((x_3)_0, (x_3)_1, \cdots, (x_3)_r\big) \in \mathcal{K} \end{array} \right\}, \tag{6.1}$$

where $\mathcal{K} := \mathbb{R}^{n_0}_+ \times \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_r}$ with $\mathcal{K}^{n_i}$ being the second-order cone in $\mathbb{R}^{n_i}$ for $1 \le i \le r$, $c = (c_0, c_1, \cdots, c_r)^T \in \mathbb{R}^{n_0} \times \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_r}$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m\times n}$, and $H \in \mathbb{S}^n_+$ are given data with $n = n_0 + n_1 + \cdots + n_r$. Moreover, if we treat A as a linear mapping such that $A : \mathbb{R}^n \to \mathbb{R}^m$, then it has the following form:

$$Ay := \sum_{i=0}^{r} A_i y_i, \quad A_i \in \mathbb{R}^{m\times n_i},\ 0 \le i \le r, \quad \forall\, y = (y_0; y_1; \cdots; y_r) \in \mathbb{R}^n.$$

Note that in (6.1), we consider additionally a nonnegative constraint since it often appears in real world applications. However, all the theoretical development in the previous parts of the paper can easily be extended to include the additional nonnegative constraint since the cone $\mathbb{R}^{n_0}_+$ is polyhedral.

In the remaining part of this section, we first discuss some implementation details for the proposed ALM. Next, we apply our SOCP solver to solve minimal enclosing ball problems, trust-region subproblems, square-root Lasso problems and some linear SOCPs problems in the DIMACS challenge dataset. We also mention here that the purpose of our numerical experiments is to compare the efficiency of our proposed ALM against other well-known linear SOCP solvers. Therefore, we do not compare the performance of our solver with specialized solvers for each application that are presented in the rest of this section.

6.1. On the efficient implementation of the ALM for SOCP. In this subsection, we present some implementation details for our ALM solver. In particular we discuss how to solve the Newton systems efficiently when the input data possesses certain sparsity structures.

Firstly, we consider solving systems arising in linear SOCPs. Let us focus on the case when A is a sparse matrix. For any given (y, x_2) and $\sigma > 0$, it is shown in Section 5 that the crucial task for solving the ALM subproblem is to solve a linear system in the following form:

$$Md := \left(\epsilon I_m + \sum_{i=0}^{r} M_i\right) d = \mathrm{rhs}, \quad d \in \mathbb{R}^m, \tag{6.2}$$

where $M_i := A_i V_i A_i^\top$, for $1 \le i \le r$, ϵ is a small positive number, rhs is a given vector and

$$V_0 \in \partial_B \Pi_{\mathbb{R}^n_+}\left(y_0 + \sigma(A_0^\top x_2 - c_0)\right),$$

$$V_i \in \partial_B \Pi_{\mathcal{K}^{n_i}}\left(y_i + \sigma(A_i^\top x_2 - c_i)\right), \quad 1 \le i \le r.$$

From the description of the elements in $\partial_B \Pi_{\mathcal{K}^{n_i}}(\cdot)$ presented in Section 2, we can see that V_i takes the following form

$$V_i = \frac{1}{2}\begin{pmatrix} 1 & \omega_i^\top \\ \omega_i & (1+\rho_i)I_{n_i-1} - \rho_i\omega_i\omega_i^\top \end{pmatrix}, \quad |\rho_i| \le 1,\ \|\omega_i\| = 1.$$

Then, for any $1 \le i \le r$, M_i can be rewritten as

$$M_i \;=\; A_i V_i A_i^\top \;=\; \frac{1+\rho_i}{2} A_i A_i^\top + \frac{1}{2}\left(A_{i,1},\ A_{i,2}w_i\right) \begin{pmatrix} -\rho_i & 1 \\ 1 & -\rho_i \end{pmatrix} \left(A_{i,1},\ A_{i,2}w_i\right)^\top,$$

where $A_i = (A_{i,1}, A_{i,2})$ with $A_{i,1} \in \mathbb{R}^m$ and $A_{i,2} \in \mathbb{R}^{m\times(n_i-1)}$.

The presence of the outer-product terms in the formulation of the matrix $M_i = A_i V_i A_i^\top$ can cause numerical issue in the following sense. If the vector $A_{i,1}$ *or* $A_{i,2}w_i$ is dense, even when $A_iA_i^\top$ is a sparse matrix, M_i will still be a dense matrix. In this case, directly solving (6.2) based on Cholesky factorization will be time-consuming. To overcome the aforementioned issue, we will apply the following dense-column handling technique to exploit the possibly sparse part of the matrix M_i.

Let us assume that the coefficient matrix M can be written as $M = M_{\rm sp} + UDU^\top$ where $M_{\rm sp}$ is a sparse symmetric positive definite matrix, U has only a few columns and D is an invertible diagonal matrix. Then, we can solve the linear system (6.2) by solving the following slightly larger but sparse linear system

$$\text{(6.3)} \qquad \mathcal{M}\begin{pmatrix} d \\ d_u \end{pmatrix} = \begin{pmatrix} \text{rhs} \\ 0 \end{pmatrix}, \quad \mathcal{M} := \begin{pmatrix} M_{\rm sp} & U \\ U^T & -D^{-1}, \end{pmatrix}, \quad d_u := DU^\top d.$$

To obtain an accurate approximate solution to the linear system (6.3), it is desirable to solve the above linear system via a preconditioned symmetric quasi-minimal residual method (PSQMR) [14] with the preconditioner computed based on the following analytical expression of $\mathcal{M}^{-1}$:

$$\mathcal{M}^{-1} = \begin{pmatrix} M_{\rm sp}^{-1} - M_{\rm sp}^{-1}US^{-1}U^\top M_{\rm sp}^{-1} & M_{\rm sp}^{-1}US^{-1} \\ S^{-1}U^\top M_{\rm sp}^{-1} & -S^{-1} \end{pmatrix},$$

where $S = D^{-1} + U^\top M_{\rm sp}^{-1} U$. It can be readily seen that for a given vector $(h_1; h_2)$, $\mathcal{M}^{-1}(h_1; h_2)$ can be evaluated efficiently as follows:

$$\lambda_1 = M_{\rm sp}^{-1}h_1, \quad \lambda_2 = S^{-1}\left(U^\top \lambda_1 - h_2\right), \quad \mathcal{M}^{-1}(h_1;h_2) = (\lambda_1 - M_{\rm sp}^{-1}U\lambda_2; \lambda_2).$$

However, when the size of the matrix S (which is twice the number of second-order cones) is large or there is no obvious sparsity structure in the linear system (6.2), the aforementioned technique may be time-consuming. In this case, we would apply the PSQMR directly to solve the system (6.2) with diagonal preconditioner.

Next, we consider the case when $H \neq 0$. We then need to solve the following linear system as described in Algorithm iSSN:

$$\text{(6.4)} \qquad M\begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = \begin{pmatrix} H + \sigma HVH & -\sigma HVA^\top \\ -\sigma AVH & \epsilon I_m + \sigma AVA^\top \end{pmatrix}\begin{pmatrix} d_1 \\ d_2 \end{pmatrix} = \begin{pmatrix} HR_1 \\ R_2 \end{pmatrix}$$

such that $(d_1, d_2) \in \text{Ran}(H) \times \mathbb{R}^m$ and

$$\text{(6.5)} \qquad \left\| M\begin{pmatrix} d_1 \\ d_2 \end{pmatrix} - \begin{pmatrix} HR_1 \\ R_2 \end{pmatrix} \right\| \le \nu,$$

where $\epsilon > 0$, $\sigma > 0$, $\nu > 0$ are given parameters, R_1, R_2 are given vectors and $V \in \partial_C \Pi_{\mathcal{K}}[-\sigma\tilde{y}(x_1, x_2, y)]$ at the given point (x_1, x_2, y). Given the fact that one requires the condition $d_1 \in \text{Ran}(H)$ to establish the convergence of Algorithm iSSN, however, in practice this condition may bring numerical issues in computing the Newton direction. Fortunately, we can fully overcome this difficulty via solving the following simplified system

$$\text{(6.6)} \qquad \hat{M}\begin{pmatrix} \hat{d}_1 \\ \hat{d}_2 \end{pmatrix} = \begin{pmatrix} I_n + \sigma VH & -\sigma VA^\top \\ -\sigma AVH & \epsilon I_m + \sigma AVA^\top \end{pmatrix}\begin{pmatrix} \hat{d}_1 \\ \hat{d}_2 \end{pmatrix} = \begin{pmatrix} R_1 \\ R_2 \end{pmatrix}$$

such that $(\hat{d}_1, \hat{d}_2) \in \mathbb{R}^n \times \mathbb{R}^m$, with the residual

$$\left\| \hat{M} \begin{pmatrix} \hat{d}_1 \\ \hat{d}_2 \end{pmatrix} - \begin{pmatrix} R_1 \\ R_2 \end{pmatrix} \right\| \leq \frac{1}{\max\{1, \lambda_{max}(H)\}} \nu,$$

where $\lambda_{max}(H)$ is the maximum eigenvalue of H. Then, simple calculations show that $(d_1, d_2) := (\Pi_{\mathrm{Ran}(H)}(\hat{d}_1), \hat{d}_2)$ solves (6.4) satisfying (6.5). Moreover, one can verify that $H\Pi_{\mathrm{Ran}(H)}(\hat{d}_1) = H\hat{d}_1$ and $\langle \Pi_{\mathrm{Ran}(H)}(\hat{d}_1),\, H\Pi_{\mathrm{Ran}(H)}(\hat{d}_1)\rangle = \langle \hat{d}_1,\, H\hat{d}_1\rangle$. Using these facts and analyzing the proposed algorithm carefully, we can execute the proposed algorithm without computing $\Pi_{\mathrm{Ran}(H)}(\hat{d}_1)$ explicitly. Finally, to solve the linear system (6.6), we can apply a direct method which computes the sparse LU factorization of $\hat{M}$ if it is sparse. Otherwise, we may use an iterative solver, such as BICGSTAB studied in [38].

6.2. Settings for numerical experiments. In this subsection, we present the settings of our numerical experiments. We first set up the stopping criteria for the proposed ALM based on the KKT conditions given in (1.1). We define the following relative KKT residuals,

$$\Delta_1(x_1, y) := \frac{\sqrt{\|Ay - b\|^2 + \|H(x_1 - y)\|^2}}{1 + \|b\| + \|H\|_F}, \quad \Delta_2(y, x_3) := \frac{\|x_3 - \Pi_{\mathcal{K}}(x_3 - y)\|}{1 + \|y\| + \|x_3\|},$$

$$\Delta_3(x_1, x_2, x_3) := \frac{\left\|-Hx_1 + A^\top x_2 + x_3 - c\right\|}{1 + \|c\|},$$

and the relative gap

$$\Delta_4(x_1, x_2, y) := \frac{|\mathrm{pobj} - \mathrm{dobj}|}{1 + |\mathrm{pobj}| + |\mathrm{dobj}|},$$

where $\mathrm{pobj} := \frac{1}{2}\langle x_1,\, Hx_1\rangle - \langle b,\, x_2\rangle$ and $\mathrm{dobj} := -\frac{1}{2}\langle y,\, Hy\rangle - \langle c,\, y\rangle$ are the objective function values for primal and dual problems, respectively. For any given termination tolerance tol which will be specified later, we terminate our ALM solver when

$$(6.7)\quad \Delta^k := \max\left\{\Delta_1(x_1^k, y^k),\ \Delta_2(y^k, x_3^k),\ \Delta_3(x_1^k, x_2^k, x_3^k),\ \Delta_4(x_1^k, x_2^k, y^k)\right\} < \mathrm{tol},$$

where $\{(x_1^k, x_2^k, x_3^k, y^k)\}$ is the sequence generated by the algorithm at the k-th iteration.

In our numerical experiments, we will consider both linear and convex quadratic SOCPs. For linear SOCPs, the solvers that we will benchmark against are the highly powerful commercial solver Mosek[1] (version 9.1.7) and the efficient open source semidefinite-quadratic-linear programs (SQLP) solver SDPT3[2] [45] (version 4.0). For the convex quadratic SOCPs, we apply our ALM solver to the problem with quadratic objective directly, while for Mosek and SDPT3, we solve the reformulated problem (1.2).

For the ALM, we set $\mathrm{tol} = 10^{-8}$ and stop the algorithm whenever it returns a solution such that Δ^k defined in (6.7) is less than tol. Moreover, the maximum number of iterations for the ALM is set to be 100. Since Mosek solves a homogeneous self-dual model which uses different stopping criteria, we use its default settings. The solutions returned by Mosek and SDPT3 under the default settings are then extracted to compute the relative KKT residuals in (6.7). We observe that when the default settings are used, Mosek and SDPT3 provide similar levels of accuracy as ours, in the terms of relative KKT residuals defined in (6.7).

[1] https://www.mosek.com/

[2] https://blog.nus.edu.sg/mattohkc/softwares/sdpt3/

All the computational results are presented in tables. The column under "it" reports the number of iterations for each algorithm. Note that for the column "it(newton)", we report the number of ALM iterations and the total number of Newton systems solved in the ALM. In addition, the column "time" reports the computation time in seconds. For the column "kkt", we report the relative KKT residuals returned by each algorithm.

All experiments are run in MATLAB R2018b on a workstation with Intel Xeon processor E5-2680v3 @2.50GHz (this processor has 12 cores and 24 threads) and 128GB of RAM, equipped with 64-bit Windows 10 OS. Since Mosek can take advantage of multi-threading, we observe that under this operating system, the number of threads used by Mosek is 12, whereas for SDPT3 and our solver, only one thread is observed to be used by MATLAB.

6.3. Application to minimal enclosing ball problems. In this subsection, we consider the *minimal enclosing ball problem* (MEB) whose goal is to compute a ball of smallest radius that encloses a given set of balls (including points). The MEB problem is a member of the family of *minimum containment problems* and it is also known as the *smallest enclosing ball problem* and *minimal bounding sphere problem*, etc. We refer the reader to [47] for an introduction of MEB problems.

Let B_i denote a ball in $\mathbb{R}^d$ with center c_i and radius $r_i \geq 0$, i.e.,

$$B_i = \left\{ z \in \mathbb{R}^d \ : \ \|z - c_i\| \leq r_i \right\}.$$

Given a set of distinct balls $\mathcal{B} = \{B_1, B_2, \cdots, B_m\} \subseteq \mathbb{R}^d$, the MEB problem is equivalent to the following unconstrained convex minimization problem:

$$\min_{z \in \mathbb{R}^d} \ \max_{1 \leq i \leq m} \ \left\{ \|z - c_i\| + r_i \right\}. \tag{6.8}$$

Since the objective function is nonsmooth, the usual gradient-based methods are not applicable. However, if we denote $n = m(d+1)$, $\mathbf{x_2} = (r; z) \in \mathbb{R}^{d+1}$, and

$$\mathbf{x_3} = (t_1; s_1; t_2; s_2; \cdots; t_m; s_m) \in \mathbb{R}^n,$$

problem (6.8) can be reformulated into a linear SOCP problem of the form (6.1) (see e.g., [47] for such reformulation):

$$\text{(MEB)} \quad \max_{\mathbf{x_2}, \boldsymbol{x_3}} \ \left\{ \mathbf{b}^\top \mathbf{x_2} \ \middle| \ \mathbf{A}^\top \mathbf{x_2} + \mathbf{x_3} = \mathbf{c}, \ \mathbf{x_3} \in \mathcal{K} \right\}, \tag{6.9}$$

where

$$\mathbf{b} = -(1; 0; \cdots; 0) \in \mathbb{R}^{d+1}, \quad \mathbf{c} = -(r_1; c_1; r_2; c_2; \cdots; r_m; c_m) \in \mathbb{R}^n,$$

$$\mathbf{A} = -\underbrace{\begin{pmatrix} I_{d+1} & \cdots & I_{d+1} \end{pmatrix}}_{m} \in \mathbb{R}^{(d+1)\times n}, \mathcal{K} = \underbrace{\mathcal{K}^{d+1} \times \cdots \times \mathcal{K}^{d+1}}_{m} \subseteq \mathbb{R}^n.$$

Then, we can apply the proposed ALM to solve the MEB problem. To achieve a fast local convergence rate for the semismooth Newton method when solving the ALM subproblems, we need the constraint nondegeneracy condition. For the MEB problem, by considering its geometrical properties, we are able to show that the constraint nondegeneracy condition holds at any feasible solution of the dual problem of (MEB).

THEOREM 6.1. *Assume that $\mathcal{B} = \{B_1, B_2, \cdots, B_m\} \subset \mathbb{R}^d$ with $m > 1$. Then the constraint non-degeneracy condition holds at any feasible solution $\bar{\mathbf{y}}$ for the dual problem of (MEB), i.e.,*

$$\mathbf{A}(\mathit{lin}(\mathcal{T}_{\mathcal{K}}(\bar{\mathbf{y}}))) = \mathbb{R}^{d+1}, \quad \forall\, \boldsymbol{A}\bar{\boldsymbol{y}} = \boldsymbol{b}.$$

Proof. Let $\bar{\mathbf{y}} = (\bar{\mathbf{y}}_1; \ldots; \bar{\mathbf{y}}_m) \in \mathbb{R}^n$ be any feasible solution, i.e., $\boldsymbol{A}(\bar{\boldsymbol{y}}) = \boldsymbol{b}$ and $\bar{\mathbf{y}}_i = (\alpha_i; v_i) \in \mathcal{K}^{d+1}$ for $i = 1, \ldots, m$. If there exists i such that $\alpha_i > \|v_i\|$, then the conclusion is trivial since $\mathrm{lin}(\mathcal{T}_{\mathcal{K}^{d+1}}(\bar{\mathbf{y}}_i)) = \mathbb{R}^{d+1}$. Assume without loss of generality that for all $1 \le i \le m_0$, $m_0 \le m$, we have that $\alpha_i = \|v_i\| > 0$ and that for all $i > m_0$, $\alpha_i = \|v_i\| = 0$.

We claim that $m_0 \ge 2$. If $m_0 = 1$, then by the feasibility condition, we have that $\alpha_1 = 1$, and $v_1 = 0$, but this is impossible since $\bar{\mathbf{y}}_1 = (\alpha_1; v_1) \in \partial\mathcal{K}^{d+1}$. Thus $m_0 \ge 2$. Next, we show that there exist $1 \le i < j \le m_0$ such that $\bar{\mathbf{y}}_i = (\alpha_i; v_i)$ and $\bar{\mathbf{y}}_j = (\alpha_j; v_j)$ is linearly independent. Suppose that this is not true. Then all the vectors $\bar{\mathbf{y}}_i$, $1 \le i \le m_0$, are parallel, and the feasibility condition implies that $v_i = 0$ for all $1 \le i \le m_0$. The latter contradicts the assumption that $\bar{\mathbf{y}}_i$, $1 \le i \le m_0$, are nonzero vectors on the boundary of $\mathcal{K}^{d+1}$. Now for such i and j, the linearity spaces are given by

$$\mathrm{lin}(\mathcal{T}_{\mathcal{K}^{d+1}}(\alpha_i; v_i)) = (\alpha_i; -v_i)^{\perp}, \quad \mathrm{lin}(\mathcal{T}_{\mathcal{K}^{d+1}}(\alpha_j; v_j)) = (\alpha_j; -v_j)^{\perp}.$$

For the primal constraint non-degeneracy condition to hold, we need to show that

$$\mathrm{lin}(\mathcal{T}_{\mathcal{K}^{d+1}}(\alpha_i; v_i)) + \mathrm{lin}(\mathcal{T}_{\mathcal{K}^{d+1}}(\alpha_j; v_j)) = \mathbb{R}^{d+1}.$$

However, the aforementioned condition is equivalent to

$$\mathrm{span}\left\{(\alpha_i; -v_i)\right\} \cap \mathrm{span}\left\{(\alpha_j; -v_j)\right\} = \{0\},$$

which holds true because of the linear independence of the vectors $\bar{\mathbf{y}}_i$ and $\bar{\mathbf{y}}_j$. This completes the proof. ∎

Next, we evaluate the performance of the proposed ALM against SDPT3 and Mosek. Let $\{\bar{p}_i\}_{i \ge 0}$ denote the following pseudo-random sequence:

$$p_0 = 7, \quad p_{i+1} = (445 p_i + 1) \ \mathbf{mod}\ 4096, \quad \bar{p}_i = \frac{p_i}{40.96}, \quad i = 1, 2, \cdots.$$

Then the elements of c_i, $i = 1, 2, \cdots, m$ are successively set to $\bar{p}_1, \bar{p}_2, \cdots$, in the order

$$r_1, (c_1)_1, \cdots, (c_1)_d, \cdots, r_m, (c_m)_1, \cdots, (c_m)_d.$$

Note that same testing instances were also used in [47]. The associated computational results are presented in Table 1. From these results, we observe that SDPT3, Mosek and the ALM solve all the instances successfully. Our ALM outperforms the other methods in the sense that the computational time is much smaller. Mosek outperforms SDPT3 but becomes less efficient when the problem size is large. Indeed, Mosek is about two times faster than SDPT3 while the ALM is at least two times faster than Mosek when the problem size is large. Thus we can conclude that the proposed ALM is highly efficient and robust for MEB problems.

TABLE 1
Computational results for MEB problems with various value of m and d.

	SDPT3	Mosek	ALM
m, d	it|time|kkt	it|time|kkt	it(newton)|time|kkt
1000, 400	21 | 5.7 | 9.6e-09	13 | 2.7 | 3.5e-09	7(40) | 1.8 | 2.9e-09
1000, 800	22 | 13.1 | 6.5e-09	14 | 5.2 | 1.6e-09	7(44) | 3.0 | 1.9e-09
1000, 1200	21 | 19.8 | 8.0e-09	12 | 8.2 | 1.2e-09	7(42) | 4.1 | 1.3e-09
1000, 1600	21 | 29.7 | 7.3e-09	11 | 11.7 | 7.3e-09	6(39) | 5.3 | 3.2e-09
1000, 2000	19 | 34.3 | 8.1e-09	13 | 17.7 | 4.3e-09	6(37) | 6.0 | 2.3e-09
8000, 100	25 | 13.4 | 7.6e-09	17 | 5.2 | 2.4e-09	7(45) | 3.6 | 9.1e-09
16000, 100	25 | 26.1 | 7.1e-09	19 | 11.0 | 2.2e-09	8(48) | 7.5 | 4.1e-09
32000, 100	25 | 54.6 | 8.4e-09	20 | 23.7 | 1.7e-09	7(44) | 14.1 | 2.2e-09
64000, 100	28 | 119.7 | 6.7e-09	20 | 49.3 | 4.9e-09	7(45) | 28.6 | 1.9e-09
128000, 100	30 | 277.3 | 5.8e-09	18 | 97.9 | 6.4e-09	6(43) | 54.7 | 3.9e-09
256000, 100	30 | 645.3 | 1.0e-08	20 | 377.6 | 1.2e-08	6(45) | 118.4 | 3.6e-09
512000, 100	31 | 1429.2 | 7.0e-09	20 | 1306.5 | 1.0e-08	5(39) | 212.4 | 9.1e-09
3000, 1000	21 | 54.8 | 7.9e-09	14 | 20.1 | 4.4e-09	7(43) | 10.5 | 3.1e-09
3000, 2000	22 | 123.1 | 9.2e-09	15 | 51.9 | 1.9e-10	7(46) | 21.5 | 6.0e-09
3000, 4000	22 | 283.1 | 5.9e-09	11 | 128.8 | 1.5e-10	6(40) | 36.4 | 4.0e-09
3000, 8000	20 | 558.6 | 7.9e-09	12 | 277.2 | 5.0e-09	6(39) | 71.2 | 6.1e-09
3000, 16000	20 | 1334.9 | 5.4e-09	12 | 592.7 | 2.3e-10	6(44) | 164.6 | 1.6e-09

6.4. Application to trust-region subproblems. We consider in this subsection SOCPs arising from the classical trust-region subproblem (TRS):

$$\min_{y\in\mathbb{R}^d} \left\{ \frac{1}{2}\langle y,\, Hy\rangle + \langle c,\, y\rangle \;\middle|\; \|y\| \le 1 \right\}, \tag{6.10}$$

where H is symmetric but not necessarily positive semidefinite. It was proven in [30, Theorem 5] that when $\lambda_H < 0$ (the smallest eigenvalue of H), a tight convex relaxation of the classical TRS (6.10) can be derived and is given by

$$\min_{y\in\mathbb{R}^d} \left\{ \frac{1}{2}\langle y,\, (H-\lambda_H I_d)y\rangle + \langle c,\, y\rangle + \lambda_H \;\middle|\; \|y\| \le 1 \right\}. \tag{6.11}$$

Problem (6.11) can be reformulated (ignoring the constant term λ_H in the objective) to the form of (D):

$$\min_{\boldsymbol{y}} \left\{ \frac{1}{2}\langle \boldsymbol{y},\, \boldsymbol{H}\boldsymbol{y}\rangle + \langle \boldsymbol{c},\, \boldsymbol{y}\rangle \;\middle|\; \boldsymbol{A}\boldsymbol{y} = \boldsymbol{b},\ \boldsymbol{y} \in \mathcal{K}^{d+1} \right\}, \tag{6.12}$$

where $\boldsymbol{y} := (s, y)^\top \in \mathbb{R}^{d+1}$, $\boldsymbol{c} := (0, c)^\top \in \mathbb{R}^{d+1}$, $\boldsymbol{b} := 1$, $\boldsymbol{A} := (1, 0) \in \mathbb{R}^{1\times(d+1)}$, and

$$\boldsymbol{H} := \begin{pmatrix} 0 & 0 \\ 0 & H - \lambda_H I_d \end{pmatrix} \in \mathbb{S}^{d+1}_+.$$

To solve a problem of the form (6.12) by Mosek and SDPT3, we need also to reformulate it as a linear SOCP as we did in the introduction. Specifically, problem (6.12) is equivalent to the following problem with additional affine and rotated quadratic cone constraints:

$$\min_{\bar{\boldsymbol{y}}} \left\{ \langle \bar{\boldsymbol{c}},\, \bar{\boldsymbol{y}}\rangle \mid \bar{\boldsymbol{A}}\bar{\boldsymbol{y}} = \bar{\boldsymbol{b}},\ \bar{\boldsymbol{y}} \in \mathcal{K}^{d+1} \times \mathcal{K}_r^{d+2} \right\},$$

where $\bar{\boldsymbol{y}} := (\boldsymbol{y}, t, q, z)^\top \in \mathbb{R}^{2d+3}$ with $t, q \in \mathbb{R},\ z \in \mathbb{R}^d$, $\bar{\boldsymbol{c}} := (\boldsymbol{c}, 1, 0)^\top \in \mathbb{R}^{2d+3}$, $\bar{\boldsymbol{b}} := (\boldsymbol{b}, 0, 1) \in \mathbb{R}^{d+2}$ and

$$\bar{\boldsymbol{A}} := \begin{pmatrix} \boldsymbol{A} & 0 & 0 & 0 \\ R & 0 & 0 & -I_{d+1} \\ 0 & 0 & 1 & 0 \end{pmatrix} \in \mathbb{R}^{(d+2)\times(2d+3)}$$

with $\boldsymbol{H} = R^\top R$.

Next we compare the performance of our SOCP solver with Mosek on a class of synthetic data. In particular, we randomly generate the input date via the following MATLAB scripts:

```
P = rand(d,d); h = (P*diag(randn(d,1)))*P';
lamh = eigs(h,1,'smallestreal');
H = [zeros(1,d+1);[zeros(d,1),h-lamh*eye(d)]];
c = [0;randn(d,1)]; b = 1; A = [1,zeros(1,d)]; R = H^0.5;
```

The computational results are presented in Table 2. In the table, we also report the minimum eigenvalue of the data matrix H (corresponding to h in the above MATLAB script), which is denoted by the term λ_H. From the table, we can see that our ALM solver outperforms Mosek and SDPT3 in terms of computational time. In most cases, the solution quality returned by our solver is much better than that of Mosek and SDPT3. These results also indicate that dealing with the quadratic objective directly is indeed much more efficient.

TABLE 2
Computational results for trust-region subproblem on synthetic data.

		Mosek	SDPT3	ALM
n	λ_H	it|time|kkt	it|time|kkt	it(newton)|time|kkt
1000	-9.9e+03	3 | 0.9 | 1.1e-09	9 | 2.2 | 1.1e-07	6(13) | 0.1 | 1.7e-09
2000	-1.2e+04	3 | 2.4 | 6.6e-08	8 | 8.3 | 2.6e-06	5(14) | 0.1 | 3.6e-09
3000	-1.3e+04	3 | 6.0 | 1.9e-06	9 | 22.1 | 9.9e-08	6(16) | 0.3 | 6.1e-10
4000	-1.5e+04	3 | 11.9 | 4.0e-09	9 | 40.5 | 1.1e-07	6(15) | 0.6 | 5.9e-11
5000	-1.8e+04	3 | 18.8 | 3.5e-09	9 | 66.6 | 1.2e-07	6(13) | 0.8 | 1.0e-09
6000	-4.1e+04	3 | 29.4 | 9.9e-08	9 | 106.9 | 1.2e-07	6(13) | 1.0 | 3.7e-10
7000	-5.3e+04	3 | 40.7 | 2.4e-07	9 | 149.8 | 1.6e-07	6(13) | 1.4 | 4.6e-11
8000	-6.6e+04	3 | 56.6 | 1.3e-06	9 | 205.8 | 1.0e-07	6(15) | 2.0 | 3.2e-11
9000	-1.4e+05	3 | 72.5 | 1.4e-07	9 | 269.4 | 1.8e-07	5(10) | 1.7 | 4.0e-09
10000	-1.2e+05	3 | 88.1 | 7.5e-10	9 | 292.8 | 5.5e-08	5(11) | 2.1 | 8.1e-10

6.5. Application to square-root Lasso problems. In this experiment, we consider the following square-root Lasso model proposed in [6]:

$$\min_{x \in \mathbb{R}^d} \|Bx - w\| + \lambda \|x\|_1 , \tag{6.13}$$

$B \in \mathbb{R}^{m\times d}$ and $w \in \mathbb{R}^m$ are given data, m is the sample size and d is the dimension of the features.

As explained in [6], the square-root Lasso model is advantageous over the classical Lasso model. When dealing with noise that follows a Gaussian distribution $\mathcal{N}(0, \sigma^2)$, the square-root Lasso model guarantees a near-oracle performance. Moreover, for the square-root Lasso model, one does not need to know an estimate of the standard deviation σ in advance, while such an estimate of σ is needed in the classical Lasso model. However, it is non-trivial to estimate the standard deviation when the dimension of features, d, is much larger than the sample size, m. Therefore, the square-root Lasso model is in some sense more useful.

It is outside the scope of this paper to compare the empirical performance of different models from statistical perspective. Here we focus on the numerical aspects of solving the optimization problem (6.13) by reformulating it into an SOCP of the form (6.1). Hence, we only compare the performance of our proposed ALM against other general SOCP solvers, but not the specialized square-root Lasso solvers such as the one in [42].

As stated in [6, Section 4], problem (6.13) can be equivalently reformulated as a standard SOCP. Indeed, we note that for any real number a, we have $|a| = a_+ + a_-$ and $a = a_+ - a_-$, where a_+ and a_- denote the positive and negative parts of a, respectively. Therefore, we can write $x = p - q$, with $p,\, q \in \mathbb{R}^d_+$ and thus,

$$\|Bx - w\| + \lambda \|x\|_1 = \|Bp - Bq - w\| + \lambda e^\top (p+q), \quad e = \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix}^\top \in \mathbb{R}^d.$$

Now let $z = Bp - Bq - w$. Then (6.13) is equivalent to

$$\text{(6.14)} \quad \min_{(t,z),p,q} \left\{ t + \lambda e^\top (p+q) \;\middle|\; Bp - Bq - z = w,\ (t,z) \in \mathcal{K}^{m+1},\ p,\, q \in \mathbb{R}^d_+ \right\},$$

where $\mathcal{K}^{m+1}$ is the second order cone in $\mathbb{R}^{m+1}$. Denote $\mathbf{y} = (p,q,t,z)^\top \in \mathbb{R}^{2d+m+1}$ and

$$\boldsymbol{b} := w \in \mathbb{R}^m, \quad \mathbf{c} := (\lambda e, \lambda e, 1, 0)^\top \in \mathbb{R}^{2d+m+1}, \quad \mathbf{A} := \begin{pmatrix} B & -B & 0 & -I_m \end{pmatrix} \in \mathbb{R}^{m \times (2d+m+1)}.$$

Then we obtain a standard SOCP in the form of the dual problem of (6.1)

$$\text{(6.15)} \qquad \text{(srLasso)} \quad \min_{\mathbf{y}} \left\{ \mathbf{c}^\top \mathbf{y} \;\middle|\; \mathbf{A}\mathbf{y} = \mathbf{b},\ \mathbf{y} \in \mathcal{K} := \mathbb{R}^d_+ \times \mathbb{R}^d_+ \times \mathcal{K}^{m+1} \right\}.$$

Next, we would test the reformulated problem (6.15) using SDPT3, Mosek and our linear SOCP solver on a collection of UCI dataset[3] which provides the data B and w. For the choice of the regularization parameter, we follow the recent work of Tang et al. [42] where they adopted a tenfold cross validation to estimate the best regularization parameter. In particular, we set the parameter $\lambda = c_0 \Phi^{-1}(1 - \frac{1}{40n})\lambda_c$, with $c_0 = 1.1$.

The choice of λ_c and computational results are both presented in Table 3. From the table, we observe that the three SOCP solvers can successfully solve all the instances. In terms of efficiency, we can see that the ALM has better performance than Mosek while SDPT3 is less efficient.

TABLE 3
Computational results for square-root Lasso problems on UCI dataset.

			SDPT3	Mosek	ALM
problem	λ_c	nnz	it|time|kkt	it|time|kkt	it(newton)|time|kkt
E2006.test (3308,150358)	0.107	1	13 | 102.8 | 8.3e-10	14 | 16.6 | 3.1e-11	4(7) | 3.8 | 2.7e-09
pyrim.scaled.expanded5 (74,201376)	0.619	48	42 | 32.4 | 2.6e-10	26 | 11.8 | 1.9e-09	4(61) | 10.2 | 2.5e-09
abalone.scale.expanded7 (4177,6435)	0.020	32	22 | 83.1 | 2.2e-09	14 | 44.1 | 6.3e-10	12(37) | 23.2 | 5.0e-09
bodyfat.scale.expanded7 (252,116280)	0.067	15	34 | 42.7 | 3.4e-10	20 | 34.1 | 8.5e-09	4(57) | 5.9 | 7.5e-09
housing.scale.expanded7 (506,77520)	0.433	52	30 | 54.9 | 1.2e-09	20 | 47.2 | 4.8e-09	8(52) | 4.6 | 2.1e-09
mpg.scale.expanded7 (392,3432)	0.253	28	23 | 2.0 | 5.5e-10	13 | 1.3 | 1.2e-09	10(35) | 0.7 | 8.4e-09
space.ga.scale.expanded9 (3107,5005)	0.058	16	22 | 42.5 | 3.1e-09	11 | 22.0 | 1.3e-08	6(27) | 7.9 | 2.5e-10

6.6. Numerical experiments on Dimacs Challenge problems. In this subsection, we test each algorithm on the linear SOCPs in DIMACS Implementation Challenge[4]. These instances are commonly used to evaluate the efficiency and accuracy of linear SOCP solvers and they are quite challenging to solve since many of the instances are highly degenerate.

The computational results are presented in Table 4. From the table, we observe that the three methods are able to solve all the instances to the desirable accuracy except for the last few instances. For the computational time, we see that ALM takes longer time than Mosek and SDPT3 for solving many of the instances, especially the last few instances for which the

[3] https://archive.ics.uci.edu/
[4] http://archive.dimacs.rutgers.edu/Challenges/Seventh/Instances/

ALM takes over a thousand Newton iterations to converge. Those instances, as far as we know, are highly degenerate, and it is the degeneracy that causes the slow convergence of the semismooth Newton method. This observation indicates that the ALM may perform poorly on degenerate problems. Finally, based on the presented numerical results, SDPT3 is also observed to be a highly efficient and robust solver for the DIMACS Challenge problems.

TABLE 4
Computational results on DIMACS Challenge problems.

	SDPT3	Mosek	ALM
Problem	it|time|kkt	it|time|kkt	it(newton)|time|kkt
nb	22 | 0.4 | 3.1e-09	10 | 0.4 | 8.1e-09	11(46) | 1.0 | 2.8e-12
nbL1	30 | 4.0 | 1.7e-09	12 | 0.3 | 1.3e-09	22(62) | 4.9 | 1.5e-12
nbL2bessel	20 | 0.4 | 9.6e-10	8 | 0.2 | 2.4e-13	8(15) | 0.3 | 1.9e-09
nbL2	15 | 0.3 | 3.1e-09	8 | 0.3 | 2.2e-10	11(49) | 1.0 | 3.8e-10
nql30new	26 | 1.0 | 4.3e-10	16 | 0.3 | 1.8e-10	38(104) | 1.7 | 9.6e-09
nql60new	27 | 4.5 | 1.9e-10	17 | 1.0 | 9.2e-11	38(109) | 7.7 | 7.9e-09
nql180new	33 | 40.4 | 6.1e-11	21 | 10.4 | 6.8e-10	33(126) | 95.5 | 8.9e-09
qssp30new	20 | 0.7 | 3.9e-10	13 | 0.3 | 5.5e-11	13(42) | 1.0 | 7.7e-09
qssp60new	23 | 3.6 | 4.5e-10	13 | 0.8 | 1.9e-10	21(60) | 7.0 | 6.9e-11
qssp180new	29 | 54.5 | 8.1e-10	19 | 12.8 | 9.3e-10	21(69) | 88.1 | 9.6e-09
sched5050s	27 | 0.9 | 1.4e-09	21 | 0.3 | 2.6e-08	13(44) | 0.5 | 9.2e-09
sched10050s	29 | 1.7 | 7.7e-08	20 | 0.4 | 5.0e-07	67(1381) | 30.1 | 2.5e-09
sched100100s	28 | 4.0 | 9.1e-09	23 | 0.8 | 2.4e-06	100(1597) | 39.3 | 1.7e-06
sched200100s	36 | 12.7 | 1.0e-07	24 | 1.5 | 5.3e-08	52(1119) | 59.5 | 7.5e-09

7. Concluding remarks. In this paper, we have employed the inexact augmented Lagrangian method (ALM) to solve convex quadratic second-order cone programming problems (SOCPs). Under the quadratic growth condition, the KKT residual is shown to possess a R-superlinear convergence rate based on recently developed results in the related topics. We also provide sufficient conditions for the quadratic growth condition to hold. Numerically, a practical SOCP solver is designed and implemented based on the proposed semismooth Newton based ALM. Extensive numerical results on solving various classes of SOCPs demonstrate that our solver is highly efficient and robust. It has comparable performance as the highly powerful commercial solver Mosek and outperforms the well-known open source semidefinite-quadratic-linear programming solver SDPT3 on the tested problems. With fruitful applications of SOCPs in many fields, we believe that our solver could serve as a promising toolbox for solving large-scale SOCPs in real-world applications.

REFERENCES

[1] F. ALIZADEH AND D. GOLDFARB, *Second-order cone programming*, Math. Program., 95 (2003), pp. 3–51.

[2] E. D. ANDERSEN, C. ROOS, AND T. TERLAKY, *On implementing a primal-dual interior point method for conic quadratic optimization*, Math. Program. Ser. B, 95 (2003), pp. 249–277.

[3] F. J. A. ARTACHO AND M. H. GEOFFROY, *Characterization of metric regularity of subdifferentials*, J. Convex Anal., 15 (2008), pp. 365–380.

[4] M. BARADAR, M. R. HESAMZADEH, AND M. GHANDHARI, *Second-order cone programming for optimal power flow in VSC-type AC-DC grids*, IEEE Trans. on Power System, 28 (2013), pp. 4282–4291.

[5] H. H. BAUSCHKE, J. M. BORWEIN, AND M. LI, *Strong conical hull intersection property, bounded linear regularity, Jameson's property (G), and error bounds in convex optimization*, Math. Program., 86 (1999), pp. 135–160.

[6] A. Belloni, V. Chernozhukov, and L. Wang, *Square-root lasso: pivotal recovery of sparse signals via conic programming*, Biometrika., 98 (2011), pp. 791–806.
[7] J. F. Bonnans and A. Shapiro, *Perturbation Analysis of Optimization Problems*, Springer, New York, 2000.
[8] J. F. Bonnans and Héctor Ramírez C., *Perturbation analysis of second-order cone programming problems*, Math. Program., 104 (2005), pp. 205–227.
[9] Z. Cai and K.-C. Toh, *Solving second order cone programming via a reduced augmented system approach*, SIAM J. Optim., 17 (2006), pp. 711-737.
[10] X. D. Chen, D. Sun, and J. Sun, *Complementarity functions and numerical experiments on some smoothing Newton methods for second-order-cone complementarity problems*, Comp. Optim. and Appl., 25 (2003), pp. 39–56.
[11] Y. Cui, C. Ding, and X. Y. Zhao, *Quadratic growth conditions for convex matrix optimization problems associated with spectral functions*, SIAM J. Optim., 27 (2017), pp. 2332–2355.
[12] Y. Cui, D. F. Sun, and K.-C. Toh, *On the R-superlinear convergence of the KKT residuals generated by the augmented Lagrangian method for convex composite conic programming*, Math. Program, 178 (2019), pp. 381–415.
[13] H. Federer, *Geometric measure theory*, Springer, 2014.
[14] R. W. Freund and N. M. Nachtigal, *A new Krylov-subspace method for symmetric indefinite linear system*, in Proceedings of the 14th IMACS World Congress on Computational and Applied Mathematics, Atlanta, USA, W.F. Ames ed., 1994, pp. 1253-1256.
[15] M. Fukushima, Z. Q. Luo, and P. Tseng, *Smoothing functions for second-order cone complementarity problems*, SIAM J. Optim., 12 (2001), pp. 436–460.
[16] D. Goldfarb and W. Yin, *Second order cone programming methods for total variation-based image restoration*, SIAM J. on Sci. Comp., 27 (2005), pp. 622–645.
[17] N. T. V. Hang, B. S. Mordukhovich, and M. E. Sarabi, *Second-order variational analysis in second-order cone programming*, Math. Program., 180 (2018), pp. 75–116.
[18] N. T. V. Hang, B. S. Mordukhovich, and M. E. Sarabi, *Augmented Lagrangian method for second-order cone programs under second-order sufficiency*, arXiv preprint: arXiv:2005.04182, (2020).
[19] S. Hayashi, N. Yamashita, and M. Fukushima, *A combined smoothing and regularization method for monotone second-order cone complementarity problems*, SIAM. J. Optim., 15 (2005), pp. 593–615.
[20] A. J. Hoffman, *On approximate solutions of systems of linear inequalities*, J. Res. Nat. Bur. Stand., 49 (1952), pp. 263–265.
[21] C. Kanzow and M. Fukushima, *Semismooth methods for linear and nonlinear second-order cone programs*, Inst. of Math., 2006.
[22] Y. J. Kuo and H. D. Mittelmann, *Interior Point Methods for Second-Order Cone Programming and OR Applications*, Comput. Optim. & Appl., 28 (2004), pp. 255–285.
[23] X. D. Li, D. F. Sun, and K.-C. Toh, *QSDPNAL: A two-phase augmented Lagrangian method for convex quadratic semidefinite programming*, Math. Program. Comp., 10 (2018), pp. 703–743.
[24] Y. J. Liu and L. W. 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.
[25] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, *Applications of second-order cone programming*, Linear Alg. Appl., 284 (1998), pp. 193–228.
[26] F. J. Luque, *Asymptotic convergence analysis of the proximal point algorithm*, SIAM J. Control and Optim., 22 (1984), pp. 277–293.
[27] A. Makrodimopoulos and C. M. Matin, *Upper bound limit analysis using simplex strain elements and second-order cone programming*, Intern. J. for Numer.and Analy.

Methods in Geomechanics, 31 (2007), pp. 835–865.

[28] O. L. MANGASARIAN, *A simple characterization of solution sets of convex programs*, Oper. Res. Lett., 7 (1988), pp. 21–26.

[29] R. D. C. MONTEIRO AND T. TSUCHIYA, *Polynomial convergence of primal-dual algorithms for the second-order cone program based on the MZ-family of directions*, Math. Program., 72 (2000), pp. 61–83.

[30] H.-N. NAM AND K.-K FATAM, *A second-order cone based approach for solving the trust-region subproblem and its variants*. SIAM J. Optim., 27 (2017), pp. 1485–1512.

[31] A. NEMIROVSKI AND K. SCHEINBERG, *Extension of Karmarkar's algorithm onto convex quadratically constrained quadratic programming*, Math. Progam., 72 (1996), pp. 273-289.

[32] Y. NESTEROV AND A. NEMIROVSKI, *Interior point polynomial methods in convex programming: theory and applications*, Soc. for Ind. and App. Math., SIAM, Philadelphia, 2014.

[33] J. V. OUTRATA AND D. F. SUN, *On the coderivative of the projection operator onto the second-order cone. Set-Valued Analysis*, 16 (2008), pp. 999–1014.

[34] J. S. PANG, D. F. SUN, AND J. SUN, *Semismooth homeomorphisms and strong stability of semidefinite and Lorentz cone complementarity problems*, Math. Oper. Res., 28 (2003), pp. 39–63.

[35] R. T. ROCKAFELLAR, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Math. Oper. Res., 1 (1976), pp. 97–116.

[36] R. T. ROCKAFELLAR, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 16 (1976), pp. 397–407.

[37] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, 1970.

[38] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, 2003.

[39] R. K. SHIVASWAMY, C. BHATTACHARYYA, AND A. J. SMOLA, *Second order cone programming approaches for handling missing and uncertain data*, JMLR, 7 (2006), pp. 1283–1314.

[40] J. SUN, *On monotropic piecewise quadratic programming*, PhD thesis, University of Washington, Seattle, 1986.

[41] Y. SUN, S. H. PAN, AND S. J. BI, *Metric subregularity and/or calmness of the normal cone mapping to the p-order conic constraint system*, Optim. Letters, 13 (2019), pp. 1095–1110.

[42] P. P. TANG, C. J. WANG, D. F. SUN, AND K.-C. TOH, *A sparse semismooth Newton based proximal majorization-minimization algorithm for nonconvex square-root-loss regression problems*, arXiv preprint: arXiv:1903.11460, 2019.

[43] P. TSENG, *Second-order cone programming relaxation of sensor network localization*, SIAM J. Optim., 18 (2007), pp. 156–185.

[44] T. TSHCHIYA, *A convergence analysis of the scaling-invariant primal-dual path-following algorithms for second-order cone programming*, Optim. Methods and Soft., 11 (1999), pp. 141-182.

[45] R. H. TUTUNCU, K.-C. TOH, AND M. J. TODD, *Solving semidefinite-quadratic-linear programs using SDPT3*, Math. Program. Ser. B, 95 (2003), pp. 189-217.

[46] X. Y. ZHAO, D. F. SUN, AND K.-C. TOH, *A Newton-CG augmented Lagrangian method for semidefinite programming*, SIAM J. Optim., 20 (2020), pp. 1737–1765.

[47] G. L. ZHOU, K.-C TOH, AND J. SUN, *Efficient algorithms for the smallest enclosing ball problem*, Comp. Opt. and Appl., 30 (2005), pp. 147-160.