

Smoothing penalty approach for solving second-order cone complementarity problems

Chieu Thanh Nguyen ¹
Department of Mathematics
Faculty of Information Technology
Vietnam National University of Agriculture
Hanoi 131000, Vietnam

Jan Harold Alcantara ²
Center for Advanced Intelligence Project
RIKEN, Tokyo, 103-0027, Japan

Zijun Hao ³
School of Mathematics and Information Science
North Minzu University
Yinchuan 750021, China

Jein-Shan Chen ⁴
Department of Mathematics
National Taiwan Normal University
Taipei 116059, Taiwan

May 27, 2024

Abstract In this paper, we propose a smoothing penalty approach for solving the second-order cone complementarity problem (SOCCP). The SOCCP is approximated by a smooth nonlinear equation with penalization parameter. We show that any solution sequence of the approximating equations converges to the solution of the SOCCP under the assumption that the associated function of the SOCCP satisfies a uniform Cartesian-type property. We present a corresponding algorithm for solving the SOCCP based on

¹E-mail: ntchieu@vnua.edu.vn.

²E-mail: janharold.alcantara@riken.jp.

³E-mail: zijunhao@126.com. The author's work is supported by the Natural Science Fund of Ningxia (No. 2022AAC03235).

⁴Corresponding author. E-mail: jschen@math.ntnu.edu.tw. The author's work is supported by the National Science and Technology Council, Taiwan.

this smoothing penalty approach, and we demonstrate the efficiency of our method for solving linear, nonlinear and tensor complementarity problems in the second-order cone setting.

Keywords Second-order cone; nonlinear complementarity problem; penalty method

MC codes 90C25; 90C30; 90C33

1 Introduction

Let \mathcal{K} be the Cartesian product of second-order cones (SOCs), also called Lorentz cones [9, 16], described by

$$\mathcal{K} := \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_r} \quad (1)$$

with $r, n_1, \dots, n_r \geq 1$, $n_1 + \cdots + n_r = n$ and

$$\mathcal{K}^{n_i} := \{(x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n_i-1} \mid \|x_2\| \leq x_1\},$$

where $\|\cdot\|$ denotes the Euclidean norm and $(x_1, x_2) := (x_1, x_2^T)^T$. Note that \mathcal{K}^1 denotes the set of nonnegative real numbers \mathbb{R}_+ . In this paper, we consider the second-order cone complementarity problem (SOCCP), which involves finding a vector $x \in \mathbb{R}^n$ such that

$$x \in \mathcal{K}, \quad F(x) \in \mathcal{K}, \quad \langle x, F(x) \rangle = 0, \quad (2)$$

where $\langle \cdot, \cdot \rangle$ is the Euclidean inner product and $F = (F_1, F_2, \dots, F_r)$ with continuously differentiable functions $F_i : \mathbb{R}^n \rightarrow \mathbb{R}^{n_i}$, $i = 1, 2, \dots, r$. In particular, when F is affine, SOCCP (2) reduces to the second-order cone linear complementarity problem (SOCLCP).

The SOCCP (2) is an extension of the nonlinear complementarity problem (NCP), which corresponds to the case where $\mathcal{K} = \mathbb{R}_+^n$, while the special case of SOCLCP with $\mathcal{K} = \mathbb{R}_+^n$ is a generalization of the standard linear complementarity problem (LCP). These problems have a broad range of applications in economics, engineering problems and robust Nash equilibria (see [5, 17, 24] and the references therein). Moreover, they can also be obtained from the KKT optimality conditions of the nonlinear second-order cone programming (SOCP):

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & Ax = b, \quad x \in \mathcal{K}, \end{aligned}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a convex twice continuously differentiable function, A is an $m \times n$ matrix with $m \leq n$, $\text{rank } A = m$, $b \in \mathbb{R}^m$.

There are various methods for solving SOCCP including the interior-point method [3, 28, 30], the smoothing Newton method [13, 18, 26], the smoothing-regularization

approach [22], the semismooth Newton method [27, 31], the merit function approach [7, 8, 11], and the matrix splitting method [21, 37], among others. Recently, power penalty methods for linear and nonlinear complementarity problems (that is, $\mathcal{K} = \mathbb{R}_+^n$) were proposed in [36, 25], and this approach is extended to the general SOCCP problem (2) by Hao et al. in [20]. In this approach, the complementarity problem is approximated by an equation of the form

$$F(x) - \alpha[x]_-^\sigma = 0, \quad (3)$$

where $\sigma > 0$ is a power parameter, $\alpha > 0$ is the penalty parameter, and $[x]_-$ denotes projection of the vector $-x$ onto the cone \mathcal{K} . By letting the penalty parameter go to infinity, it was shown in the aforementioned works that the corresponding solution sequence of the approximate nonlinear equations converges to the solution of the complementarity problem under certain monotonicity assumptions on F . While theoretically appealing, the main hurdle lies in the development of solution methods due to the nonsmoothness of the projection operator in the equation (3).

To deal with the nonsmoothness that prohibits the use of available efficient numerical methods for solving (3), the recent work [19] focused on the SOCLCP case (i.e., F is affine) and used a novel smoothing function approach to approximate the projection $[x]_-$ in (3) for the SOCLCP problem. Motivated by this approach, we extend the framework to the general SOCCP (2), where the function F may be nonlinear, and we also propose a practical *smoothing power penalty algorithm* by utilizing the smoothing functions in [6]. We provide theoretical guarantees under the assumption that F is a uniform ξ -P function, a class of functions introduced in [34], which is larger than the one considered in [20]. In particular, we show that the smoothing approximation of equation (3) has a unique solution, and we derive an error bound of order $O(\alpha^{-1/\xi\sigma})$ between the solution sequence of the approximating penalty equations and the solution of the SOCCP (2). Therefore, our main contributions are threefold: First, we significantly generalize and improve the algorithm proposed in [19] that only considers affine functions F with positive definite gradient. Second, our theoretical framework is applicable to uniform ξ -P functions which subsumes the class of monotonic functions in [20]. Finally, as opposed to [20], our algorithm has a practical implementation due to the smoothing strategy that permits the use of derivative-based algorithms for solving systems of equations. Indeed, we demonstrate the applicability and efficiency of our proposed algorithm through numerical experiments involving affine and nonlinear functions F , as well as applications to second-order cone tensor complementarity problems.

This paper is organized as follows: In Section 2, we briefly introduce some properties related to the second-order cone that will be useful in our subsequent analysis. We also present smoothing approximations for the function $[x]_-^\sigma$. In Section 3, we prove our main convergence results under the uniform ξ -P assumption. Extensive numerical experiments are presented in Section 4, and concluding remarks are given in Section 5.

2 Preliminaries

2.1 The second-order cone

We review some definitions and properties for the case of single block second-order cone $\mathcal{K} = \mathcal{K}^n$; most of the materials herein can be found in [9, 16, 18]. For any $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$, $y = (y_1, y_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$, their *Jordan product* is defined as $x \circ y = (\langle x, y \rangle, y_1 x_2 + x_1 y_2)$. The spectral decomposition of x with respect to the SOC is given by

$$x = \lambda_1(x)u_x^{(1)} + \lambda_2(x)u_x^{(2)},$$

where for $i = 1, 2$,

$$\lambda_i(x) = x_1 + (-1)^i \|x_2\|, \quad u_x^{(i)} = \begin{cases} \frac{1}{2}(1, (-1)^i \frac{x_2}{\|x_2\|}) & \text{if } \|x_2\| \neq 0, \\ \frac{1}{2}(1, (-1)^i w) & \text{if } \|x_2\| = 0, \end{cases} \quad (4)$$

with $w \in \mathbb{R}^{n-1}$ being any unit vector. The two scalars $\lambda_1(x)$ and $\lambda_2(x)$ are called spectral values of x , while the two vectors $u_x^{(1)}$ and $u_x^{(2)}$ are called the spectral vectors of x .

For any $x \in \mathbb{R}^n$, let $[x]_+$ denote the projection of x onto \mathcal{K}^n , and $[x]_-$ be the projection of $-x$ onto \mathcal{K}^n , that is, $[x]_- = [-x]_+$. It is well-known that $x = [x]_+ - [x]_-$, and we also have the following useful formulas:

$$[x]_+ = [\lambda_1(x)]_+ u_x^{(1)} + [\lambda_2(x)]_+ u_x^{(2)}, \quad [x]_- = [\lambda_1(x)]_- u_x^{(1)} + [\lambda_2(x)]_- u_x^{(2)}, \quad (5)$$

and

$$[x]_+^\sigma = [\lambda_1(x)]_+^\sigma u_x^{(1)} + [\lambda_2(x)]_+^\sigma u_x^{(2)}, \quad [x]_-^\sigma = [\lambda_1(x)]_-^\sigma u_x^{(1)} + [\lambda_2(x)]_-^\sigma u_x^{(2)},$$

where $\sigma > 0$, $[t]_+ = \max\{0, t\}$ and $[t]_- = \max\{0, -t\}$ for $t \in \mathbb{R}$ (see [9], for instance). We also have that $[x]_+, [x]_- \in \mathcal{K}^n$ and $[x]_+ \circ [x]_- = 0$.

The following results play an important role in our convergence analysis.

Proposition 2.1. [9, 16, 18] *For any $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$ with the spectral values $\lambda_1(x), \lambda_2(x)$ and spectral vectors $u_x^{(1)}, u_x^{(2)}$ given as (4), we have:*

- (a) $u_x^{(1)} \circ u_x^{(2)} = 0$ and $u_x^{(i)} \circ u_x^{(i)} = u_x^{(i)}$, $\|u_x^{(i)}\|^2 = 1/2$ for $i = 1, 2$.
- (b) $\lambda_1(x), \lambda_2(x)$ are nonnegative (positive) if and only if $x \in \mathcal{K}^n$ ($x \in \text{int}(\mathcal{K}^n)$), *where $\text{int}(\mathcal{K}^n)$ denotes the interior of \mathcal{K}^n .*
- (c) For any $x \in \mathbb{R}^n$, $x \in \mathcal{K}^n$ if and only if $\langle x, y \rangle \geq 0$ for all $y \in \mathcal{K}^n$.
- (d) The norm of x can be expressed in terms of $\lambda_1(x)$ and $\lambda_2(x)$ as follows:

$$\|x\|^2 = \frac{1}{2}(\lambda_1^2(x) + \lambda_2^2(x)).$$

Lemma 2.2. For any $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$, $y = (y_1, y_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$. $\|x_1 y_2 + y_1 x_2\| \leq \|x\| \|y\|$

Proof. By expanding the terms of $\|x_1 y_2 + y_1 x_2\|^2 - \|x\|^2 \|y\|^2$ and using Cauchy-Schwarz inequality, the result immediately follows. \square

Lemma 2.3. For any $x = (x_1, x_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$, $y = (y_1, y_2) \in \mathbb{R} \times \mathbb{R}^{n-1}$. Then the following results hold:

(a) If $y \in \mathcal{K}^n$ then $\langle x, y \rangle \leq \langle [x]_+, y \rangle$.

(b) $\|x \circ y\| \leq \sqrt{2} \|x\| \|y\|$ and $\lambda_2(x \circ y) = 2 \langle x \circ y, u_{x \circ y}^{(2)} \rangle \leq 2 \|x\| \|y\|$.

Proof. (a) holds by [4, Lemma 3.1], while (b) is a direct consequence of Lemma 2.2. \square

2.2 Smoothing approximation of the projection function with power parameter

From (5), we see that the nonsmoothness of the operator $[x]_-$ comes from the nonsmoothness of the real-valued function $[t]_- = \max(0, -t)$, which in turn renders the equation (3) to be nonsmooth as well. To avoid this problem, we use smooth approximations. We say that $\phi^- : \mathbb{R}_{++} \times \mathbb{R} \rightarrow \mathbb{R}$ is a smoothing function of $[t]_- = \max(0, -t)$ if it satisfies the following conditions:

(i) ϕ^- is continuously differentiable at $(\mu, t) \in \mathbb{R}_{++} \times \mathbb{R}$;

(ii) $\lim_{\mu \downarrow 0} \phi^-(\mu, t) = [t]_-$ for any $t \in \mathbb{R}$.

In [6], a systematic framework of generating smoothing functions for the plus function $[t]_+ = \max(0, t)$ is developed using density functions. By using the property that $[t]_- = [-t]_+$, we may then obtain smoothing functions for $[t]_-$, and consequently for $[x]_-$ via (5). Indeed, some specific smoothing functions of $[t]_-$ are also provided in [19].

Naturally, the function $(\phi^-)^\sigma$ is a candidate smoothing function of $[t]_-^\sigma$, which is indeed the case provided that $\sigma \geq 1$. For the sake of generality, we make the following assumptions given in Assumption 2.1 so that our framework applies to the general case $\sigma > 0$. Nevertheless, Assumption 2.1 is not required when $\sigma \geq 1$. We also note that there are numerous smoothing functions satisfying Assumption 2.1. For instance, we may take $\phi^-(\mu, t) := \psi(\mu, -t)$, where ψ is a strictly positive smoothing function of the plus function generated in [2].

Assumption 2.1.

(a) $\phi^-(\mu, t)$ is a strictly positive smoothing function of $[t]_-$, i.e., $\phi^-(\mu, t) > 0$ for all $\mu > 0$ and $t \in \mathbb{R}$.

(b) Fixed $\mu > 0$, $(\phi^-(\mu, t))'$ is a monotonically decreasing function for all $t \in \mathbb{R}$.

Lemma 2.4. Suppose that $\phi^-(\mu, t)$ satisfies Assumption 2.1(a). Then, for any $\sigma > 0$, $\phi^-(\mu, t)^\sigma$ is a smoothing function of $[t]_-^\sigma$.

Proof. The proof is straightforward. \square

Using these smoothing functions, we now construct a smoothing function for $[x]_-$ associated with the general SOC (1). Denote $x = (x^1, x^2, \dots, x^r) \in \mathbb{R}^n$ so that $[x]_- = ([x^1]_-, [x^2]_-, \dots, [x^r]_-) \in \mathcal{K}$. Assume that $\phi^-(\mu, t)$ satisfies Assumption 2.1, then for any $\sigma > 0$, we define the vector-valued functions

$$\Phi^-(\mu, x) := (\Phi_1^-(\mu, x^1), \Phi_2^-(\mu, x^2), \dots, \Phi_r^-(\mu, x^r)), \quad (6)$$

$$\Phi^-(\mu, x)^\sigma := (\Phi_1^-(\mu, x^1)^\sigma, \Phi_2^-(\mu, x^2)^\sigma, \dots, \Phi_r^-(\mu, x^r)^\sigma), \quad (7)$$

with $\Phi_\nu^- : \mathbb{R}_{++} \times \mathbb{R}^{n_\nu} \rightarrow \mathbb{R}^{n_\nu}$, $\nu \in \{1, 2, \dots, r\}$ given by

$$\Phi_\nu^-(\mu, x^\nu) := \phi^-(\mu, \lambda_1(x^\nu)) u_{x^\nu}^{(1)} + \phi^-(\mu, \lambda_2(x^\nu)) u_{x^\nu}^{(2)}, \quad (8)$$

$$\Phi_\nu^-(\mu, x^\nu)^\sigma := \phi^-(\mu, \lambda_1(x^\nu))^\sigma u_{x^\nu}^{(1)} + \phi^-(\mu, \lambda_2(x^\nu))^\sigma u_{x^\nu}^{(2)}, \quad (9)$$

where $\mu \in \mathbb{R}_{++}$ is a parameter, $\lambda_1(x^\nu)$, $\lambda_2(x^\nu)$ are the spectral values, and $u_{x^\nu}^{(1)}$, $u_{x^\nu}^{(2)}$ are the spectral vectors of x^ν as shown in (4).

As shown in the following lemma, the functions (6) and (7) serve as smooth approximations of $[x]_-$ and $[x]_-^\sigma$, respectively. That is,

$$\lim_{\mu \rightarrow 0^+} \Phi^-(\mu, x) = [x]_-,$$

and

$$\lim_{\mu \rightarrow 0^+} \Phi^-(\mu, x)^\sigma = [x]_-^\sigma.$$

Lemma 2.5. Suppose that $\Phi_\nu^-(\mu, x^\nu)$, $\Phi_\nu^-(\mu, x^\nu)^\sigma$ are defined in (8), (9), respectively. Then $\Phi_\nu^-(\mu, x^\nu)$, $\Phi_\nu^-(\mu, x^\nu)^\sigma$ are smooth on $\mathbb{R}_{++} \times \mathbb{R}^{n_\nu}$. Moreover,

$$\lim_{\mu \rightarrow 0^+} \Phi_\nu^-(\mu, x^\nu) = [\lambda_1(x^\nu)]_- u_{x^\nu}^{(1)} + [\lambda_2(x^\nu)]_- u_{x^\nu}^{(2)} = [x^\nu]_-.$$

$$\lim_{\mu \rightarrow 0^+} \Phi_\nu^-(\mu, x^\nu)^\sigma = [\lambda_1(x^\nu)]_-^\sigma u_{x^\nu}^{(1)} + [\lambda_2(x^\nu)]_-^\sigma u_{x^\nu}^{(2)} = [x^\nu]_-^\sigma.$$

Proof. The proof follows from [10, Proposition 5] and Lemma 2.4. \square

The following result will later be important in our analysis.

Lemma 2.6. Suppose that $\Phi_\nu^-(\mu, x^\nu)$, $\Phi_\nu^-(\mu, x^\nu)^\sigma$ are defined in (8), (9), respectively. Then, the following hold.

(a) $\Phi_\nu^-(\mu, x^\nu) \in \mathcal{K}^{n_\nu}$ and $\Phi^-(\mu, x) \in \mathcal{K}$; and

(b) $\Phi_\nu^-(\mu, x^\nu)^\sigma \in \mathcal{K}^{n_\nu}$ and $\Phi^-(\mu, x)^\sigma \in \mathcal{K}$.

Proof. The proof is analogous to that of [19, Lemma 3.2]. \square

3 Smoothing power penalty algorithm

3.1 Convergence analysis

Inspired by Lemma 2.5, we propose a penalty approach for solving SOCCP (2). More precisely, we consider the approximate penalty equations (APEs): find $x \in \mathbb{R}^n$ such that

$$H_{\mu,\alpha}^\sigma(x) := F(x) - \alpha\Phi^-(\mu, x)^\sigma = 0, \quad (10)$$

where $\sigma > 0$ is a given power parameter, $\alpha > 0$ is a penalty parameter and $\Phi^-(\mu, x)^\sigma$ is defined in (7). We denote a solution of (10) by $x_{\mu,\alpha}$. Since the penalty term $\alpha\Phi^-(\mu, x)^\sigma$ penalizes the negative part of x , the equation (10) is a penalized equation associated with SOCCP (2). Moreover, from Lemma 2.6 and (10), it is easy to verify that $F(x_{\mu,\alpha}) \in \mathcal{K}$. Hence, intuitively, a sequence of solutions $\{x_{\mu,\alpha}\}$ of (10) includes points that satisfy the second feasibility condition in (2), while the penalization forces $x_{\mu,\alpha}$ to be in the cone \mathcal{K} to satisfy the first condition, and orthogonality is obtained in the limit by pre-multiplying (10) by $x_{\mu,\alpha}^T$.

Our main goal is to show that indeed, any solution sequence $\{x_{\mu,\alpha}\}$ converges to a solution of the SOCCP (2) when $\alpha \rightarrow +\infty$ and $\mu \rightarrow 0^+$. To this end, we consider the following class of functions introduced in [34].

Definition 3.1. [34] *F is a uniform ξ -P function for some $\xi > 1$, i.e., there exist $\rho > 0$ and an index $\nu \in \{1, 2, \dots, r\}$ such that*

$$\langle x^\nu - y^\nu, F_\nu(x) - F_\nu(y) \rangle \geq \rho \|x - y\|^\xi \quad \forall x, y \in \mathbb{R}^n.$$

An important property of the SOCCP (2) is that it has a unique solution when F is a uniform ξ -P function, as shown in the following result.

Lemma 3.1. *Suppose that F is a uniform ξ -P function for some $\xi > 1$. Then, the SOCCP (2) has a unique solution.*

Proof. The proof follows by using the same arguments as Proposition 2.12 and Proposition 2.13 in [34] when we consider a closed convex cone $\mathcal{K} \subset \mathbb{R}^n$. \square

We will now show that the APE (10) also has a unique solution when F is a uniform ξ -P function for some $\xi > 1$.

Lemma 3.2. *Suppose that F is a uniform ξ -P function for some $\xi > 1$. Then for any $\mu, \alpha, \sigma > 0$, the function $H_{\mu,\alpha}^\sigma(x) := F(x) - \alpha\Phi^-(\mu, x)^\sigma$ is a uniform ξ -P function.*

Proof. For simplicity, we denote $\Phi_\nu^-(\cdot) \equiv \Phi_\nu^-(\mu, \cdot)$ where $\nu \in \{1, 2, \dots, r\}$ is an index such that Definition 3.1 is satisfied. We also denote $\phi^-(\cdot) \equiv \phi^-(\mu, \cdot)$. First, note that

$$\begin{aligned} & \left\langle (x^\nu - y^\nu) \circ (\Phi_\nu^-(x)^\sigma - \Phi_\nu^-(y)^\sigma), u_{y^\nu}^{(1)} \right\rangle \\ &= \left\langle x^\nu \circ \Phi_\nu^-(x)^\sigma, u_{y^\nu}^{(1)} \right\rangle - \left\langle x^\nu \circ \Phi_\nu^-(y)^\sigma, u_{y^\nu}^{(1)} \right\rangle - \left\langle y^\nu \circ \Phi_\nu^-(x)^\sigma, u_{y^\nu}^{(1)} \right\rangle + \left\langle y^\nu \circ \Phi_\nu^-(y)^\sigma, u_{y^\nu}^{(1)} \right\rangle \\ &= \left\langle x^\nu \circ \Phi_\nu^-(x)^\sigma, u_{y^\nu}^{(1)} \right\rangle - \left\langle x^\nu, \Phi_\nu^-(y)^\sigma \circ u_{y^\nu}^{(1)} \right\rangle - \left\langle \Phi_\nu^-(x)^\sigma, y^\nu \circ u_{y^\nu}^{(1)} \right\rangle + \left\langle y^\nu \circ \Phi_\nu^-(y)^\sigma, u_{y^\nu}^{(1)} \right\rangle. \end{aligned}$$

We have

$$\begin{aligned} \left\langle x^\nu \circ \Phi_\nu^-(x)^\sigma, u_{y^\nu}^{(1)} \right\rangle &= \left\langle \lambda_1(x^\nu) \phi^-(\lambda_1(x^\nu)) u_{x^\nu}^{(1)} + \lambda_2(x^\nu) \phi^-(\lambda_2(x^\nu)) u_{x^\nu}^{(2)}, u_{y^\nu}^{(1)} \right\rangle \\ &= \lambda_1(x^\nu) \phi^-(\lambda_1(x^\nu)) \left\langle u_{x^\nu}^{(1)}, u_{y^\nu}^{(1)} \right\rangle + \lambda_2(x^\nu) \phi^-(\lambda_2(x^\nu)) \left\langle u_{x^\nu}^{(2)}, u_{y^\nu}^{(1)} \right\rangle. \end{aligned}$$

$$\begin{aligned} \left\langle x^\nu, \Phi_\nu^-(y)^\sigma \circ u_{y^\nu}^{(1)} \right\rangle &= \left\langle \lambda_1(x^\nu) u_{x^\nu}^{(1)} + \lambda_2(x^\nu) u_{x^\nu}^{(2)}, \phi^-(\lambda_1(y^\nu)) u_{y^\nu}^{(1)} \right\rangle \\ &= \lambda_1(x^\nu) \phi^-(\lambda_1(y^\nu)) \left\langle u_{x^\nu}^{(1)}, u_{y^\nu}^{(1)} \right\rangle + \lambda_2(x^\nu) \phi^-(\lambda_1(y^\nu)) \left\langle u_{x^\nu}^{(2)}, u_{y^\nu}^{(1)} \right\rangle. \end{aligned}$$

$$\begin{aligned} \left\langle \Phi_\nu^-(x)^\sigma, y^\nu \circ u_{y^\nu}^{(1)} \right\rangle &= \left\langle \phi^-(\lambda_1(x^\nu)) u_{x^\nu}^{(1)} + \phi^-(\lambda_2(x^\nu)) u_{x^\nu}^{(2)}, \lambda_1(y^\nu) u_{y^\nu}^{(1)} \right\rangle \\ &= \lambda_1(y^\nu) \phi^-(\lambda_1(x^\nu)) \left\langle u_{x^\nu}^{(1)}, u_{y^\nu}^{(1)} \right\rangle + \lambda_1(y^\nu) \phi^-(\lambda_2(x^\nu)) \left\langle u_{x^\nu}^{(2)}, u_{y^\nu}^{(1)} \right\rangle. \end{aligned}$$

$$\begin{aligned} \left\langle y^\nu \circ \Phi_\nu^-(y)^\sigma, u_{y^\nu}^{(1)} \right\rangle &= \left\langle \lambda_1(y^\nu) \phi^-(\lambda_1(y^\nu)) u_{y^\nu}^{(1)} + \lambda_2(y^\nu) \phi^-(\lambda_2(y^\nu)) u_{y^\nu}^{(2)}, u_{y^\nu}^{(1)} \right\rangle \\ &= \frac{1}{2} \lambda_1(y^\nu) \phi^-(\lambda_1(y^\nu)). \end{aligned}$$

From the above equations, we get

$$\begin{aligned} & \left\langle (x^\nu - y^\nu) \circ (\Phi_\nu^-(x)^\sigma - \Phi_\nu^-(y)^\sigma), u_{y^\nu}^{(1)} \right\rangle \\ &= (\lambda_1(x^\nu) \phi^-(\lambda_1(x^\nu)) - \lambda_1(x^\nu) \phi^-(\lambda_1(y^\nu)) - \lambda_1(y^\nu) \phi^-(\lambda_1(x^\nu))) \left\langle u_{x^\nu}^{(1)}, u_{y^\nu}^{(1)} \right\rangle \\ &+ (\lambda_2(x^\nu) \phi^-(\lambda_2(x^\nu)) - \lambda_2(x^\nu) \phi^-(\lambda_1(y^\nu)) - \lambda_1(y^\nu) \phi^-(\lambda_2(x^\nu))) \left\langle u_{x^\nu}^{(2)}, u_{y^\nu}^{(1)} \right\rangle \\ &+ \frac{1}{2} \lambda_1(y^\nu) \phi^-(\lambda_1(y^\nu)) \\ &= \frac{1}{4} (\lambda_1(x^\nu) \phi^-(\lambda_1(x^\nu)) - \lambda_1(x^\nu) \phi^-(\lambda_1(y^\nu)) - \lambda_1(y^\nu) \phi^-(\lambda_1(x^\nu))) \left(1 + \frac{x_2^T y_2}{\|x_2\| \|y_2\|} \right) \\ &+ \frac{1}{4} (\lambda_2(x^\nu) \phi^-(\lambda_2(x^\nu)) - \lambda_2(x^\nu) \phi^-(\lambda_1(y^\nu)) - \lambda_1(y^\nu) \phi^-(\lambda_2(x^\nu))) \left(1 - \frac{x_2^T y_2}{\|x_2\| \|y_2\|} \right) \\ &+ \frac{1}{2} \lambda_1(y^\nu) \phi^-(\lambda_1(y^\nu)). \end{aligned}$$

Similarly, we also have

$$\begin{aligned}
& \left\langle (x^\nu - y^\nu) \circ (\Phi_\nu^-(x)^\sigma - \Phi_\nu^-(y)^\sigma), u_{y^\nu}^{(2)} \right\rangle \\
&= (\lambda_1(x^\nu)\phi^-(\lambda_1(x^\nu)) - \lambda_1(x^\nu)\phi^-(\lambda_2(y^\nu)) - \lambda_2(y^\nu)\phi^-(\lambda_1(x^\nu))) \left\langle u_{x^\nu}^{(1)}, u_{y^\nu}^{(2)} \right\rangle \\
&+ (\lambda_2(x^\nu)\phi^-(\lambda_2(x^\nu)) - \lambda_2(x^\nu)\phi^-(\lambda_2(y^\nu)) - \lambda_2(y^\nu)\phi^-(\lambda_2(x^\nu))) \left\langle u_{x^\nu}^{(2)}, u_{y^\nu}^{(2)} \right\rangle \\
&+ \frac{1}{2}\lambda_2(y^\nu)\phi^-(\lambda_2(y^\nu)) \\
&= \frac{1}{4} (\lambda_1(x^\nu)\phi^-(\lambda_1(x^\nu)) - \lambda_1(x^\nu)\phi^-(\lambda_2(y^\nu)) - \lambda_2(y^\nu)\phi^-(\lambda_1(x^\nu))) \left(1 - \frac{x_2^T y_2}{\|x_2\| \|y_2\|} \right) \\
&+ \frac{1}{4} (\lambda_2(x^\nu)\phi^-(\lambda_2(x^\nu)) - \lambda_2(x^\nu)\phi^-(\lambda_2(y^\nu)) - \lambda_2(y^\nu)\phi^-(\lambda_2(x^\nu))) \left(1 + \frac{x_2^T y_2}{\|x_2\| \|y_2\|} \right) \\
&+ \frac{1}{2}\lambda_2(y^\nu)\phi^-(\lambda_2(y^\nu)).
\end{aligned}$$

Then

$$\begin{aligned}
& \left\langle (x^\nu - y^\nu) \circ (\Phi_\nu^-(x)^\sigma - \Phi_\nu^-(y)^\sigma), u_{y^\nu}^{(1)} + u_{y^\nu}^{(2)} \right\rangle \\
&= A + B \frac{x_2^T y_2}{\|x_2\| \|y_2\|} + \frac{1}{2}\lambda_1(y^\nu)\phi^-(\lambda_1(y^\nu)) + \frac{1}{2}\lambda_2(y^\nu)\phi^-(\lambda_2(y^\nu)),
\end{aligned}$$

where

$$\begin{aligned}
A &:= \frac{1}{4} (\lambda_1(x^\nu)\phi^-(\lambda_1(x^\nu)) - \lambda_1(x^\nu)\phi^-(\lambda_1(y^\nu)) - \lambda_1(y^\nu)\phi^-(\lambda_1(x^\nu))) \\
&+ \frac{1}{4} (\lambda_2(x^\nu)\phi^-(\lambda_2(x^\nu)) - \lambda_2(x^\nu)\phi^-(\lambda_1(y^\nu)) - \lambda_1(y^\nu)\phi^-(\lambda_2(x^\nu))) \\
&+ \frac{1}{4} (\lambda_1(x^\nu)\phi^-(\lambda_1(x^\nu)) - \lambda_1(x^\nu)\phi^-(\lambda_2(y^\nu)) - \lambda_2(y^\nu)\phi^-(\lambda_1(x^\nu))) \\
&+ \frac{1}{4} (\lambda_2(x^\nu)\phi^-(\lambda_2(x^\nu)) - \lambda_2(x^\nu)\phi^-(\lambda_2(y^\nu)) - \lambda_2(y^\nu)\phi^-(\lambda_2(x^\nu)))
\end{aligned}$$

and

$$\begin{aligned}
B &:= \frac{1}{4} (\lambda_1(x^\nu)\phi^-(\lambda_1(x^\nu)) - \lambda_1(x^\nu)\phi^-(\lambda_1(y^\nu)) - \lambda_1(y^\nu)\phi^-(\lambda_1(x^\nu))) \\
&- \frac{1}{4} (\lambda_2(x^\nu)\phi^-(\lambda_2(x^\nu)) - \lambda_2(x^\nu)\phi^-(\lambda_1(y^\nu)) - \lambda_1(y^\nu)\phi^-(\lambda_2(x^\nu))) \\
&- \frac{1}{4} (\lambda_1(x^\nu)\phi^-(\lambda_1(x^\nu)) - \lambda_1(x^\nu)\phi^-(\lambda_2(y^\nu)) - \lambda_2(y^\nu)\phi^-(\lambda_1(x^\nu))) \\
&+ \frac{1}{4} (\lambda_2(x^\nu)\phi^-(\lambda_2(x^\nu)) - \lambda_2(x^\nu)\phi^-(\lambda_2(y^\nu)) - \lambda_2(y^\nu)\phi^-(\lambda_2(x^\nu)))
\end{aligned}$$

Since $\phi^-(\cdot)$ is the monotonically decreasing by Assumption 2.1(b) and $\lambda_2(\cdot) \geq \lambda_1(\cdot)$, we have

$$\begin{aligned}
B &= \frac{1}{4} (\phi^-(\lambda_1(y^\nu)) - \phi^-(\lambda_2(y^\nu))) (\lambda_2(x^\nu) - \lambda_1(x^\nu)) \\
&+ \frac{1}{4} (\phi^-(\lambda_1(x^\nu)) - \phi^-(\lambda_2(x^\nu))) (\lambda_2(y^\nu) - \lambda_1(y^\nu)) \\
&\geq 0.
\end{aligned}$$

It follows that

$$\begin{aligned}
& \left\langle (x^\nu - y^\nu) \circ (\Phi_\nu^-(x)^\sigma - \Phi_\nu^-(y)^\sigma), u_{y^\nu}^{(1)} + u_{y^\nu}^{(2)} \right\rangle \\
& \leq A + B + \frac{1}{2} \lambda_1(y^\nu) \phi^-(\lambda_1(y^\nu)) + \frac{1}{2} \lambda_2(y^\nu) \phi^-(\lambda_2(y^\nu)) \\
& = \frac{1}{2} (\phi^-(\lambda_1(x^\nu)) - \phi^-(\lambda_1(y^\nu))) (\lambda_1(x^\nu) - \lambda_1(y^\nu)) \\
& \quad + \frac{1}{2} (\phi^-(\lambda_2(x^\nu)) - \phi^-(\lambda_2(y^\nu))) (\lambda_2(x^\nu) - \lambda_2(y^\nu)) \\
& \leq 0
\end{aligned}$$

which implies that

$$\langle x^\nu - y^\nu, \Phi_\nu^-(\mu, x)^\sigma - \Phi_\nu^-(\mu, y)^\sigma \rangle \leq 0.$$

This, together with the fact that F is a uniform ξ -P function, implies that

$$\begin{aligned}
\langle x^\nu - y^\nu, (H_{\mu, \alpha}^\sigma)_\nu(x) - (H_{\mu, \alpha}^\sigma)_\nu(y) \rangle &= \langle x^\nu - y^\nu, F_\nu(x) - F_\nu(y) - \alpha(\Phi_\nu^-(\mu, x)^\sigma - \Phi_\nu^-(\mu, y)^\sigma) \rangle \\
&\geq \rho \|x - y\|^\xi.
\end{aligned}$$

The proof is complete. \square

The following proposition shows that if we start from a sufficiently small smoothing parameter μ , the sequence of solutions of (10) for a fixed power parameter σ and penalty parameter α is bounded, and therefore has accumulation points.

Proposition 3.3. *Suppose that F is a uniform ξ -P function for some $\xi > 1$, and let $\sigma > 0$ and $\alpha > 0$ be given. Then, there exists $\delta > 0$ such that the set $\{x_{\mu, \alpha} \in \mathbb{R}^n : x_{\mu, \alpha} \text{ solves the APE (10) and } \mu \in (0, \delta]\}$ is bounded.*

Proof. Since F is a uniform ξ -P function, there exist $\rho > 0$ and an index $\nu \in \{1, 2, \dots, r\}$ such that

$$\begin{aligned}
\rho \|x_{\mu, \alpha}\|^\xi &\leq \langle x_{\mu, \alpha}^\nu - 0, F_\nu(x_{\mu, \alpha}) - F_\nu(0) \rangle \\
&= \langle x_{\mu, \alpha}^\nu, F_\nu(x_{\mu, \alpha}) \rangle - \langle x_{\mu, \alpha}^\nu, F_\nu(0) \rangle.
\end{aligned} \tag{11}$$

To proceed, we consider three cases. For convenience, we denote $\lambda_1 \equiv \lambda_1(x_{\mu, \alpha}^\nu)$ and $\lambda_2 \equiv \lambda_2(x_{\mu, \alpha}^\nu)$.

Case 1: Suppose $x_{\mu, \alpha}^\nu \in \mathcal{K}^{n_\nu}$. From (10), (11) and using Cauchy-Schwarz inequality, we have

$$\begin{aligned}
\rho \|x_{\mu, \alpha}\|^\xi &\leq \|x_{\mu, \alpha}^\nu\| (\alpha \|\Phi_\nu^-(\mu, x_{\mu, \alpha}^\nu)^\sigma\| + \|F_\nu(0)\|) \\
&\leq \|x_{\mu, \alpha}^\nu\| (\alpha \|\phi^-(\mu, \lambda_1)^\sigma u^{(1)} + \phi^-(\mu, \lambda_2)^\sigma u^{(2)}\| + \|F_\nu(0)\|) \\
&\leq \|x_{\mu, \alpha}\| (\alpha \|\sqrt{2} \phi^-(\mu, 0)^\sigma\| + \|F_\nu(0)\|),
\end{aligned} \tag{12}$$

where the last inequality holds by the triangle inequality and the monotone decreasing property of $\phi^-(\mu, t)$ with respect to t , since $0 \leq \lambda_1 \leq \lambda_2$ in this case. From Lemma 2.4 we have that $\lim_{\mu \rightarrow 0^+} \phi^-(\mu, 0)^\sigma = 0$. Hence, there exists a positive real number δ , such that

$$\sqrt{2}\alpha \|\phi^-(\mu, 0)^\sigma\| \leq 1, \quad \forall \mu \in (0, \delta]. \quad (13)$$

It follows from (12) and (13) that

$$\|x_{\mu, \alpha}\|^{\xi-1} \leq \frac{1}{\rho}(1 + \|F_\nu(0)\|).$$

Thus, $\|x_{\mu, \alpha}\| \leq \left(\frac{1 + \|F_\nu(0)\|}{\rho}\right)^{\frac{1}{\xi-1}}$.

Case 2: Suppose $x_{\mu, \alpha}^\nu \in -\mathcal{K}^{n_\nu}$, that is $\lambda_1 \leq \lambda_2 \leq 0$. Using Assumption 2.1(b), we have that

$$\langle x_{\mu, \alpha}^\nu, \Phi_\nu^-(\mu, x_{\mu, \alpha}^\nu)^\sigma \rangle = \frac{1}{2} (\lambda_1 \phi^-(\mu, \lambda_1)^\sigma + \lambda_2 \phi^-(\mu, \lambda_2)^\sigma) \leq 0.$$

This, together with (10) and (11), gives

$$\rho \|x_{\mu, \alpha}\|^\xi \leq -\langle x_{\mu, \alpha}^\nu, F_\nu(0) \rangle \leq \|x_{\mu, \alpha}^\nu\| \|F_\nu(0)\| \leq \|x_{\mu, \alpha}\| \|F_\nu(0)\|.$$

It follows that $\|x_{\mu, \alpha}\| \leq \left(\frac{\|F_\nu(0)\|}{\rho}\right)^{\frac{1}{\xi-1}}$.

Case 3: Suppose $x_{\mu, \alpha}^\nu \notin \mathcal{K}^{n_\nu} \cup -\mathcal{K}^{n_\nu}$. Under this case, we know that $\lambda_1 < 0 < \lambda_2$, and therefore $[x_{\mu, \alpha}^\nu]_+ = \lambda_2 u^{(2)}$ and $\lambda_1 \phi^-(\mu, \lambda_1)^\sigma < 0 < \lambda_2 \phi^-(\mu, \lambda_2)^\sigma$. In turn,

$$\begin{aligned} \alpha \langle x_{\mu, \alpha}^\nu, \Phi_\nu^-(\mu, x_{\mu, \alpha}^\nu)^\sigma \rangle &= \frac{1}{2} \alpha (\lambda_1 \phi^-(\mu, \lambda_1)^\sigma + \lambda_2 \phi^-(\mu, \lambda_2)^\sigma) \\ &\leq \frac{\sqrt{2}}{2} \left(\frac{\sqrt{2}}{2} \lambda_2\right) \alpha \phi^-(\mu, \lambda_2)^\sigma \\ &\leq \frac{\sqrt{2}}{2} \|x_{\mu, \alpha}^\nu\| \alpha \phi^-(\mu, \lambda_2)^\sigma \\ &\leq \sqrt{2} \|x_{\mu, \alpha}^\nu\| \alpha \phi^-(\mu, 0)^\sigma, \end{aligned} \quad (14)$$

where the second inequality holds due to $\frac{\sqrt{2}}{2} \lambda_2 = \|[x_{\mu, \alpha}^\nu]_+\| \leq \|x_{\mu, \alpha}^\nu\|$. It follows from (10), (11), (13) and (14) that

$$\begin{aligned} \rho \|x_{\mu, \alpha}\|^\xi &\leq \|x_{\mu, \alpha}^\nu\| (\sqrt{2}\alpha \phi^-(\mu, 0)^\sigma + \|F_\nu(0)\|) \\ &\leq \|x_{\mu, \alpha}\| (\sqrt{2}\alpha \phi^-(\mu, 0)^\sigma + \|F_\nu(0)\|) \\ &\leq \|x_{\mu, \alpha}\| (1 + \|F_\nu(0)\|). \end{aligned}$$

Therefore,

$$\|x_{\mu, \alpha}\| \leq \left(\frac{1 + \|F_\nu(0)\|}{\rho}\right)^{\frac{1}{\xi-1}}.$$

Setting $M := \left(\frac{1+\|F_\nu(0)\|}{\rho}\right)^{\frac{1}{\xi-1}}$. From the above three cases, we obtain $\|x_{\mu,\alpha}\| \leq M$. Thus, the proof is complete. \square

From Proposition 3.3 and the continuity of F , we have that for a fixed $\sigma > 0$ and $\alpha > 0$, the set $\{\|F(x_{\mu,\alpha})\| : x_{\mu,\alpha} \text{ solves APE (10) and } \mu \in (0, \delta]\}$ is bounded, provided that δ is sufficiently small. Using the same techniques as in [20, Proposition 4.4], we can obtain the following proposition.

Proposition 3.4. *Suppose that F is a uniform ξ - P function for some $\xi > 1$. Let $\sigma > 0$, $\alpha > 0$ and $\nu \in \{1, 2, \dots, r\}$. Then, there exist positive constants C^ν and $\delta > 0$ such that*

$$\|\Phi_\nu^-(\mu, x_{\mu,\alpha})\| \leq \frac{C^\nu}{\alpha^{1/\sigma}},$$

and

$$\|[x_{\mu,\alpha}^\nu]_-\| \leq \frac{C^\nu}{\alpha^{1/\sigma}}, \quad (15)$$

for all $\mu \in (0, \delta]$.

We now prove our main result.

Theorem 3.5. *Suppose that F is a uniform ξ - P function for some $\xi > 1$. Let $\sigma > 0$, $\alpha > 0$, and x_* be the solution of SOCCP (2). Then, there exist positive constants C and $\delta > 0$ such that*

$$\|x_* - x_{\mu,\alpha}\| \leq \frac{C}{\alpha^{1/\xi\sigma}}, \quad \forall \mu \in (0, \delta]. \quad (16)$$

Proof. Since x_* solves the SOCCP (2), then we have that $x_*^\nu \in \mathcal{K}^{n_\nu}$, $F_\nu(x_*) \in \mathcal{K}^{n_\nu}$, and $\langle x_*^\nu, F_\nu(x_*) \rangle = 0$. From Lemma 2.3 and using the Cauchy-Schwarz inequality, we have

$$\begin{aligned} \rho \|x_* - x_{\mu,\alpha}\|^\xi &\leq \langle x_*^\nu - x_{\mu,\alpha}^\nu, F_\nu(x_*) - F_\nu(x_{\mu,\alpha}) \rangle \\ &= \langle x_{\mu,\alpha}^\nu, F_\nu(x_{\mu,\alpha}) \rangle + \langle x_*^\nu, -F_\nu(x_{\mu,\alpha}) \rangle + \langle -x_{\mu,\alpha}^\nu, F_\nu(x_*) \rangle \\ &\leq \lambda_2(x_{\mu,\alpha}^\nu \circ F_\nu(x_{\mu,\alpha})) + \langle [-F_\nu(x_{\mu,\alpha})]_+, x_*^\nu \rangle + \langle [-x_{\mu,\alpha}^\nu]_+, F_\nu(x_*) \rangle \\ &\leq \lambda_2([x_{\mu,\alpha}^\nu \circ F_\nu(x_{\mu,\alpha})]_+) + \|[-F_\nu(x_{\mu,\alpha})]_+\| \|x_*^\nu\| + \|[-x_{\mu,\alpha}^\nu]_+\| \|F_\nu(x_*)\| \\ &\leq \sqrt{2} \|[x_{\mu,\alpha}^\nu \circ F_\nu(x_{\mu,\alpha})]_+\| + \|[F_\nu(x_{\mu,\alpha})]_-\| \|x_*^\nu\| + \|[x_{\mu,\alpha}^\nu]_-\| \|F_\nu(x_*)\|. \end{aligned}$$

Since

$$x_{\mu,\alpha}^\nu \circ F_\nu(x_{\mu,\alpha}) = \alpha[x_{\mu,\alpha}^\nu \circ \Phi_\nu^-(\mu, x_{\mu,\alpha})^\sigma] = \alpha[\lambda_1 \phi^-(\mu, \lambda_1)^\sigma u_{x_{\mu,\alpha}^\nu}^{(1)} + \lambda_2 \phi^-(\mu, \lambda_2)^\sigma u_{x_{\mu,\alpha}^\nu}^{(2)}],$$

it follows that

$$[F_\nu(x_{\mu,\alpha}) \circ x_{\mu,\alpha}^\nu]_+ = \begin{cases} \alpha[\lambda_1 \phi^-(\mu, \lambda_1)^\sigma u_{x_{\mu,\alpha}^\nu}^{(1)} + \lambda_2 \phi^-(\mu, \lambda_2)^\sigma u_{x_{\mu,\alpha}^\nu}^{(2)}] & \text{if } x_{\mu,\alpha}^\nu \in \mathcal{K}^{n_\nu}, \\ 0 & \text{if } x_{\mu,\alpha}^\nu \in -\mathcal{K}^{n_\nu}, \\ \alpha \lambda_2 \phi^-(\mu, \lambda_2)^\sigma u_{x_{\mu,\alpha}^\nu}^{(2)} & \text{otherwise.} \end{cases}$$

If $x^\nu \in \mathcal{K}^{n_\nu}$, then using the Cauchy-Schwarz inequality and Proposition 2.1(d), we have that

$$\begin{aligned} \alpha[\lambda_1\phi^-(\mu, \lambda_1)^\sigma u_{x_{\mu,\alpha}^\nu}^{(1)} + \lambda_2\phi^-(\mu, \lambda_2)^\sigma u_{x_{\mu,\alpha}^\nu}^{(2)}] &\leq \frac{\alpha}{\sqrt{2}} (\lambda_1\phi^-(\mu, \lambda_1)^\sigma + \lambda_2\phi^-(\mu, \lambda_2)^\sigma) \\ &\leq \frac{\alpha}{\sqrt{2}} \sqrt{\lambda_1^2 + \lambda_2^2} \sqrt{\phi^-(\mu, \lambda_1)^{2\sigma} + \phi^-(\mu, \lambda_2)^{2\sigma}} \\ &\leq \sqrt{2}\alpha \|x^\nu\| \phi^-(\mu, 0)^\sigma \\ &\leq \sqrt{2}\alpha \|x_{\mu,\alpha}\| \phi^-(\mu, 0)^\sigma \\ &\leq \sqrt{2}\alpha M \phi^-(\mu, 0)^\sigma, \end{aligned}$$

where M is the constant stipulated in the proof of Proposition 3.3, and satisfies $\|x_{\mu,\alpha}\| \leq M$.

It follows from Case 3 of Proposition 3.3 that $\|\alpha\lambda_2\phi^-(\mu, \lambda_2)^\sigma u_{x_{\mu,\alpha}^\nu}^{(2)}\| \leq \sqrt{2}\alpha M \phi^-(\mu, 0)^\sigma$. Hence, we obtain

$$\|[F_\nu(x_{\mu,\alpha}) \circ x_{\mu,\alpha}^\nu]_+\| \leq \sqrt{2}\alpha M \phi^-(\mu, 0)^\sigma.$$

Since $\lim_{\mu \rightarrow 0^+} \phi^-(\mu, 0)^\sigma = 0$, for any $\alpha > 0$, there exists a positive real number δ such that

$$\sqrt{2}\alpha M \|\phi^-(\mu, 0)^\sigma\| \leq \frac{1}{\alpha^{1/\sigma}}, \quad \forall \mu \in (0, \delta].$$

Therefore,

$$\|[F_\nu(x_{\mu,\alpha}) \circ x_{\mu,\alpha}^\nu]_+\| \leq \frac{1}{\alpha^{1/\sigma}}.$$

On the other hand, since $F_\nu(x_{\mu,\alpha}) \in \mathcal{K}^{n_\nu}$, $\nu = 1, 2, \dots, r$, we have $\|[F_\nu(x_{\mu,\alpha})]_-\| = 0$. All of these in conjunction with (15) lead to

$$\rho \|x^* - x_{\mu,\alpha}\|^\xi \leq \frac{1}{\alpha^{1/\sigma}} + \frac{C^\nu \|F_\nu(x_*)\|}{\alpha^{1/\sigma}}.$$

Therefore, (16) holds with $C := \left(\frac{1+C^\nu \|F_\nu(x_*)\|}{\rho}\right)^{\frac{1}{\xi}}$, thus completing the proof. \square

3.2 Implementation of the algorithm

In light of Theorem 3.5, we develop an algorithm that solves a sequence of APEs (10) with $\alpha \rightarrow +\infty$ and $\mu \rightarrow 0^+$ in order to obtain the solution of the SOCCP (2). One important aspect towards practical implementation of the algorithm is the development of a procedure to increase α and decrease μ . To this end, we propose a scheme to manage these parameters based on Proposition 3.6 below.

Proposition 3.6. *Let $\alpha > 0$ and $\sigma > 0$, and suppose that F_ν has no zeros for some $\nu \in \{1, 2, \dots, r\}$, that is, $F_\nu(x) \neq 0$ for any $x \in \mathbb{R}^n$. Then, there exists $\delta > 0$ such that the ν^{th} block of $x_{\mu,\alpha}$ does not belong to \mathcal{K}^{n_ν} for all $\mu \in (0, \delta]$, that is, $x_{\mu,\alpha}^\nu \notin \mathcal{K}^{n_\nu}$.*

Proof. Suppose otherwise. Then there exists a sequence $\{\mu_k\}$ with $\mu_k \rightarrow 0$ such that $x_{\mu_k, \alpha}^\nu$ belongs to \mathcal{K}^{n_ν} for all k . From (10), we have

$$\|F_\nu(x_{\mu_k, \alpha})\| = \alpha \|\Phi_\nu^-(\mu_k, x_{\mu_k, \alpha}^\nu)^\sigma\| = \alpha \|\phi^-(\mu_k, \eta_1)^\sigma v^{(1)} + \phi^-(\mu_k, \eta_2)^\sigma v^{(2)}\| \leq \sqrt{2}\alpha \phi^-(\mu_k, 0)^\sigma.$$

Since $\lim_{\mu \rightarrow 0^+} \phi^-(\mu, 0)^\sigma = 0$ by Lemma 2.4, we have that

$$\lim_{k \rightarrow \infty} \|F_\nu(x_{\mu_k, \alpha})\| = 0. \quad (17)$$

Meanwhile, a consequence of Proposition 3.3 is that the sequence $\{x_{\mu_k, \alpha}^\nu\}$ is bounded, and therefore has an accumulation point, say x_* . Since \mathcal{K}^{n_ν} is closed, we also have that $x_* \in \mathcal{K}^{n_\nu}$. By continuity of F_ν , together with (17), we conclude that $F_\nu(x_*) = 0$. As this contradicts our hypothesis on F_ν , thus completing the proof. \square

We now summarize in Algorithm 1 our smoothing power penalty approach based on Theorem 3.5 and Proposition 3.6. Naturally, we increase the penalty parameter when the constraints of SOCCP (2) are violated, which is the essence of penalty-based methods. However, note that an exact solution of (20) yields $F(x_{\mu, \alpha}) \in \mathcal{K}$, and therefore we only need to keep track of whether or not the first constraint in the SOCCP (2) holds. Indeed, this condition is satisfied if and only if $\lambda_1(x^\nu) \geq 0$ for all $\nu = 1, \dots, r$, that is,

$$\beta(x) := \max\{(-\lambda_1(x^1))_+, \dots, (-\lambda_1(x^r))_+\} = 0. \quad (18)$$

When this condition is not satisfied up to a certain tolerance, we increase α as indicated in (19), and in this case, we also reset the value of the smoothing parameter μ to the initial estimate μ_0 . On the other hand, we decrease μ if the current point nearly satisfies the first feasibility condition $x \in \mathcal{K}$, since Proposition 3.6 implies that satisfaction of this feasibility constraint is suggestive that the smoothing parameter is not yet sufficiently small. **In summary, our implementation's update schemes for the penalty and smoothing parameters are based on the satisfaction of the feasibility conditions of the SOCCP, as inspired by Theorem 3.5 and Proposition 3.6. We note that another novel implementation that has been used in the literature of smoothing techniques for complementarity problems involves the introduction of a merit function that incorporates the smoothing parameter as a variable (for instance, see [1]). In this approach, the update on the smoothing parameter is dictated by the descent on the formulated merit function. For nonlinear complementarity problems, this kind of implementation along with a smoothing power penalty approach was used in [35].**

Remark 3.7. Although our SPPA algorithm is applicable in general for continuous functions F , the assumptions of most algorithms that are efficient for solving nonlinear equations (e.g., Newton method) are quite prohibitive. Consequently, in Section 4, we only consider differentiable functions F similar to the existing works in SOCCP literature [13, 23]. In turn, this allows us to take advantage of our smoothing approach as the left-hand side of (20) is indeed a differentiable function, therefore permitting the use of

Algorithm 1: Smoothing Power Penalty Algorithm for SOCCP (SPPA)

Choose an initial point $x^0 \in \mathbb{R}^n$, an initial penalty parameter $\alpha_0 > 0$, an initial smoothing parameter $\mu_0 \in (0, 1)$, and a termination parameter ϵ . Choose parameters $c_1 > 1$ and $c_2 \in (0, 1)$, a power parameter $\sigma > 0$, and feasibility tolerance parameter $\tau \in (0, 1)$. Set $\bar{\mu} = \mu_0$ and $k = 0$.

Step 1. Terminate the algorithm if $\text{RESIDUAL}(x^k) < \epsilon$, where RESIDUAL is given in (21). Otherwise, let $\beta_k = \beta(x^k)$, where β is defined by (18), and set

$$(\mu_{k+1}, \alpha_{k+1}) = \begin{cases} (\bar{\mu}, c_1 \alpha_k) & \text{if } \beta_k > \tau, \\ (c_2 \mu_k, \alpha_k) & \text{otherwise,} \end{cases} \quad (19)$$

then go to Step 2.

Step 2. Obtain a solution of the nonlinear equation

$$F(x) - \alpha_k \Phi^-(\mu_k, x)^\sigma = 0, \quad (20)$$

and denote it by x^{k+1} . Set $k \leftarrow k + 1$ and go to Step 1.

Newton method for finding its zeros. As for the power parameter, Theorem 3.5 suggests that a lower value of σ theoretically provides tighter error bounds, and is therefore desirable to obtain faster convergence. However, setting $\sigma \in (0, 1)$ renders equation (20) more difficult to solve numerically, and so we set $\sigma = 1$ in our simulations.

4 Numerical experiments

In this section, we illustrate through numerous experiments the applicability and efficiency of our proposed SPPA algorithm for solving second-order cone linear complementarity problems (SOCLCP) and nonlinear complementarity problems (SOCNCP) in Examples 4.1 and 4.2, respectively. We also test our approach for a special class of SOCNCP, namely tensor complementarity problems (SOCTCP) in Example 4.3.

For the parameters of our algorithm, we universally set $\alpha_0 = 10^6$, $\mu_0 = 10^{-6}$, $c_1 = 10$, $c_2 = 0.9$, $\sigma = 1$ and $\tau = 10^{-6}$ all throughout our experiments. As indicated in Remark 3.7, we consider smooth instances of F , and therefore we employ Newton's method for approximately solving the smoothed penalized equation (20). For the smoothing function ϕ^- , we use $\phi_1^-(\mu, t) = -t + \mu \ln(1 + \exp(t/\mu))$ for problems involving functions F that has the uniform Cartesian property. Otherwise, we use $\phi_2^-(\mu, t) = 0.5(\sqrt{4\mu^2 + t^2} - t)$ which has a better numerical behavior for general problems (for instance, when F is only

a P_0 function)⁵. Comparisons with two well-known algorithms, namely the combined smoothing and regularized method called ReSNA⁶ in [23] and the smoothing Newton Fischer-Burmeister (SNFB) algorithm in [13] are conducted. We used the default setting of parameters of ReSNA and SNFB. These algorithms are similar in nature to ours, in the sense that a Newton direction is computed at **every iteration**. To obtain a fair comparison, we terminate the algorithms when the residual given by

$$\text{RESIDUAL}(x) := \max_{\nu=1,\dots,r} \{\|x^\nu - [x^\nu - F(x^\nu)]_+\|\} \quad (21)$$

is less than $\epsilon = 10^{-6}$ at the current iteration $x = x^k$ (see [15, Proposition 1.5.8]), or when the maximum number of iterations is reached, which we set to 100 for SPPA and SNFB, and to 50 for ReSNA (default value).

All tests are implemented in MATLAB R2021a in a laptop with 8GB RAM and Intel Core i7 at 2.6 GHz. The results of our experiments are summarized in Tables 1-5. **NI** denotes the number of Newton iterations conducted by the algorithm, while **CPU** is the time (in seconds) needed to obtain a point x^* satisfying $\text{RESIDUAL}(x^*) < 10^{-6}$. An entry “–” indicates that the corresponding algorithm for the given initial point x^0 did not reach the required residual value. The vectors **0** and **1** denote the zero vector and the vector consisting of ones, respectively, while $\mathbf{e} = (e^1, e^2, \dots, e^r)$, where e^ν is the identity element in \mathcal{K}^{n_ν} .

For small dimensional linear, nonlinear, and tensor complementarity problems presented in Tables 1, 3, and 4, respectively, our approach is quite competitive: It obtains an SOCCP solution in at most one-hundredth or one-thousandth of a second, and often outperforms ReSNA and SNFB. Moreover, our SPPA is relatively more consistent in solving the SOCCPs, whereas ReSNA and SNFB do not always arrive at an SOCCP solution,

On the other hand, we can see from Tables 2 and 5 that our algorithm can significantly outperform both ReSNA and SNFB for larger dimensional problems. For instance, observe from Table 5 that for $n = 100$, SPPA solved SOCTCP3 for all initial points considered, whereas ReSNA failed to solve the problem when $x^0 = 10 \cdot \mathbf{1}$. For the initial points $x^0 \in \{\mathbf{e}, 10 \cdot \mathbf{1}\}$, SNFB is marginally faster than our algorithm, but for the initial point $x^0 = \pm \mathbf{1}$, our algorithm significantly outperforms SNFB.

Example 4.1 (SOCLCP test problems). *We test different functions $F(x) = Ax - b$ and cones \mathcal{K} described below.*

⁵In our numerical experiments, we only used ϕ_2^- for the test problem SOCTCP3.

⁶The codes for ReSNA are publicly available and can be downloaded from <http://optima.ws.hosei.ac.jp/hayashi/ReSNA/>.

Table 1: Comparison of algorithms for solving SOCLCPs using the smoothing function ϕ_1^- .

| Test Problem | Algorithm | $x^0 = \mathbf{0}$ | | $x^0 = \mathbf{1}$ | | $x^0 = \mathbf{e}$ | | $x^0 = -\mathbf{1}$ | |
|--------------|-----------|--------------------|--------|--------------------|--------|--------------------|--------|---------------------|--------|
| | | NI | CPU | NI | CPU | NI | CPU | NI | CPU |
| SOCLCP1 | SPPA | 9 | 0.0002 | 11 | 0.0002 | 10 | 0.0002 | 11 | 0.0003 |
| | ReSNA | 7 | 0.0009 | 8 | 0.0008 | 6 | 0.0006 | 6 | 0.0005 |
| | SNFB | – | – | 9 | 0.0004 | – | – | – | – |
| SOCLCP2 | SPPA | 17 | 0.0003 | 14 | 0.0003 | 15 | 0.0009 | 12 | 0.0002 |
| | ReSNA | 6 | 0.0006 | 4 | 0.0004 | 7 | 0.0007 | 7 | 0.0006 |
| | SNFB | 7 | 0.0004 | – | – | – | – | – | – |
| SOCLCP3 | SPPA | 32 | 0.0008 | 32 | 0.0007 | 30 | 0.0009 | 34 | 0.0011 |
| | ReSNA | 6 | 0.0010 | 6 | 0.0007 | 5 | 0.0007 | 6 | 0.0008 |
| | SNFB | 9 | 0.0008 | 8 | 0.0005 | – | – | 9 | 0.0006 |
| SOCLCP4 | SPPA | 33 | 0.0009 | 32 | 0.0008 | 31 | 0.0008 | 36 | 0.0009 |
| | ReSNA | 8 | 0.0009 | 8 | 0.0009 | 7 | 0.0007 | 6 | 0.0007 |
| | SNFB | 8 | 0.0007 | 8 | 0.0007 | 6 | 0.0004 | 8 | 0.0005 |

SOCLCP1 ([12]). $A = \begin{pmatrix} 15 & -5 & -1 & 4 & -5 \\ 0 & 5 & 0 & 0 & 1 \\ 1 & -3 & 8 & 2 & -3 \\ 2 & -4 & 2 & 9 & -4 \\ 0 & -5 & 0 & 0 & 10 \end{pmatrix}$, $b = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$, and $\mathcal{K} = \mathcal{K}^5$

SOCLCP2 ([12]). $A = \begin{pmatrix} 21 & -9 & 18 \\ -9 & 4 & -7 \\ 18 & -7 & 19 \end{pmatrix}$, $b = \begin{pmatrix} -3 \\ -7 \\ -1 \end{pmatrix}$, and $\mathcal{K} = \mathcal{K}^3$.

SOCLCP3 ([19]). A as given in SOCLCP2, $b = (3, 0, 2, 2, 5)^T$, and $\mathcal{K} = \mathcal{K}^3 \times \mathcal{K}^2$.

SOCLCP4 ([19]).

$$A = \begin{pmatrix} 3.9475 & 1.1370 & -0.3462 & -0.1258 & -1.2034 & -0.4979 & -1.0337 \\ 1.1370 & 3.5593 & -1.2955 & -0.4391 & -0.3009 & -0.6016 & -0.0404 \\ -0.3462 & -1.2955 & 5.0908 & -1.1187 & -0.6652 & -1.5541 & -1.0419 \\ -0.1258 & -0.4391 & -1.1187 & 3.5778 & -0.4033 & -0.1402 & -0.1991 \\ -1.2034 & -0.3009 & -0.6652 & -0.4033 & 2.9766 & 0.3725 & 0.0995 \\ -0.4979 & -0.6016 & -1.5541 & -0.1402 & 0.3725 & 4.8431 & -0.5048 \\ -1.0337 & -0.0404 & -1.0419 & -0.1991 & 0.0995 & -0.5048 & 4.0049 \end{pmatrix}$$

$$b = (2, -1, 3, -2, 4, -1, 3)^T,$$

and $\mathcal{K} = \mathcal{K}^3 \times \mathcal{K}^4$.

Table 2: Comparison of algorithms for solving SOCLCP5 with $n = 2000$ for different values of r using the smoothing function ϕ_1^- . We generated 10 random SOCLCP5 test problems, solve each generated problem with four initial points, and get the average number of Newton iterations (NI) and CPU time (CPU) for successful simulations. PS denotes the number of problems solved by the algorithm.

| Rank (r) | Algorithm | $x^0 = \mathbf{0}$ | | | $x^0 = \mathbf{1}$ | | | $x^0 = \mathbf{e}$ | | | $x^0 = -\mathbf{1}$ | | |
|--------------|-----------|--------------------|------|-----------|--------------------|-------|------------|--------------------|------|------------|---------------------|---------|------------|
| | | PS | NI | CPU | PS | NI | CPU | PS | NI | CPU | PS | NI | CPU |
| $r = 200$ | SPPA | 10 | 18.1 | 2.55±0.38 | 10 | 25.8 | 3.86±0.89 | 10 | 26 | 3.90±1.13 | 10 | 24.3 | 3.61±0.67 |
| | ReSNA | 10 | 9.3 | 6.54±1.86 | 10 | 11.5 | 8.10±3.77 | 10 | 6.8 | 4.83±1.67 | 10 | 10.8 | 7.61±2.82 |
| | SNFB | 9 | 8.56 | 5.48±1.82 | 9 | 14.44 | 9.51±6.62 | 8 | 8.38 | 5.40±2.36 | 9 | 13.8889 | 9.18±7.06 |
| $r = 500$ | SPPA | 10 | 22.9 | 3.43±0.63 | 10 | 22.9 | 3.45±0.45 | 10 | 23.4 | 3.55±0.56 | 10 | 23.4 | 3.55±0.44 |
| | ReSNA | 10 | 10.1 | 7.20±3.51 | 10 | 10.9 | 7.81±2.84 | 10 | 7.7 | 5.52±3.35 | 10 | 10.7 | 7.66±2.49 |
| | SNFB | 10 | 8.5 | 5.48±0.62 | 10 | 15.2 | 10.18±9.84 | 10 | 8.1 | 5.28±0.97 | 10 | 13.9 | 9.21±7.76 |
| $r = 1000$ | SPPA | 10 | 25.4 | 3.79±0.86 | 10 | 26.2 | 4.00±0.92 | 10 | 22.4 | 3.34±0.62 | 10 | 24.3 | 3.65±0.42 |
| | ReSNA | 10 | 10.9 | 7.77±2.47 | 10 | 11.5 | 8.17±2.58 | 10 | 11 | 7.96±4.54 | 10 | 11.7 | 8.35±2.71 |
| | SNFB | 10 | 8.6 | 5.55±0.65 | 10 | 12.2 | 7.99±3.63 | 10 | 8 | 5.22±0.65 | 10 | 11.6 | 7.57±3.02 |
| $r = 1500$ | SPPA | 10 | 24 | 3.50±0.79 | 10 | 25.2 | 3.79±0.61 | 10 | 24.3 | 3.67±0.73 | 10 | 26.2 | 3.92±0.89 |
| | ReSNA | 10 | 14 | 9.92±2.54 | 10 | 12.3 | 8.75±2.04 | 10 | 20.5 | 14.56±5.97 | 10 | 12.4 | 8.80±2.03 |
| | SNFB | 10 | 10.1 | 6.52±0.95 | 10 | 16.8 | 11.16±4.48 | 10 | 9.3 | 6.09±1.02 | 10 | 16.3 | 10.65±4.08 |
| $r = 2000$ | SPPA | 10 | 20.9 | 2.95±0.16 | 10 | 20.9 | 2.99±0.53 | 10 | 19.9 | 2.83±0.52 | 10 | 20.9 | 2.99±0.53 |
| | ReSNA | 10 | 11.5 | 7.91±2.02 | 10 | 11.7 | 8.00±3.21 | 10 | 16.9 | 11.63±5.91 | 10 | 11.6 | 7.95±3.08 |
| | SNFB | 10 | 9.6 | 5.99±0.89 | 10 | 17.3 | 11.20±6.86 | 10 | 9.2 | 5.85±1.07 | 9 | 15.67 | 10.15±6.98 |

SOCLCP5 ([23]). A is randomly generated by setting $A = BB^T$, where $B \in \mathbb{R}^{n \times r}$ with entries uniformly sampled from $[-1, 1]$. The vector b is set to $b = Ae - 10^\alpha n^{0.5} p$, where $p = \frac{1}{\sqrt{2}} \left(\cos \theta \begin{pmatrix} 1 \\ w \end{pmatrix} + \sin \theta \begin{pmatrix} 1 \\ -w \end{pmatrix} \right) \in \text{int}\mathcal{K}^n$, θ randomly chosen from $(0, \pi/2)$, $w \in \mathbb{R}^{n-1}$ with entries randomly chosen from $[-1, 1]$, and α randomly chosen from $[-1, 1]$. The cone is $\mathcal{K} = \mathcal{K}^n$.

Table 3: Comparison of algorithms for solving SOCNCPs using the smoothing function ϕ_1^- .

| Test Problem | Algorithm | $x^0 = \mathbf{0}$ | | $x^0 = \mathbf{1}$ | | $x^0 = \mathbf{e}$ | | $x^0 = -\mathbf{1}$ | |
|--------------|-----------|--------------------|--------|--------------------|--------|--------------------|--------|---------------------|--------|
| | | NI | CPU | NI | CPU | NI | CPU | NI | CPU |
| SOCNCP1 | SPPA | 41 | 0.0009 | 19 | 0.0005 | – | – | 34 | 0.0010 |
| | ReSNA | 8 | 0.0012 | 7 | 0.0008 | 8 | 0.0010 | 7 | 0.0009 |
| | SNFB | 8 | 0.0006 | 8 | 0.0005 | 8 | 0.0004 | 7 | 0.0004 |
| SOCNCP2 | SPPA | 46 | 0.0021 | 46 | 0.0016 | 48 | 0.0018 | 45 | 0.0031 |
| | ReSNA | 8 | 0.0031 | 27 | 0.0092 | – | – | 12 | 0.0031 |
| | SNFB | 15 | 0.0052 | 10 | 0.0016 | 10 | 0.0018 | 13 | 0.0022 |
| SOCNCP3 | SPPA | 46 | 0.0047 | 62 | 0.0079 | – | – | 465 | 0.0165 |
| | ReSNA | – | – | – | – | – | – | – | – |
| | SNFB | 8 | 0.0007 | 9 | 0.0007 | 9 | 0.0007 | 13 | 0.0023 |
| SOCNCP4 | SPPA | 23 | 0.0006 | 24 | 0.0005 | 25 | 0.0005 | 23 | 0.0005 |
| | ReSNA | 9 | 0.0010 | 9 | 0.0010 | 8 | 0.0011 | 10 | 0.0012 |
| | SNFB | – | – | 12 | 0.0013 | 12 | 0.0016 | – | – |
| SOCNCP5 | SPPA | 15 | 0.0007 | 15 | 0.0006 | 14 | 0.0006 | 15 | 0.0006 |
| | ReSNA | 5 | 0.0008 | 6 | 0.0010 | 5 | 0.0008 | 6 | 0.0010 |
| | SNFB | 9 | 0.0012 | 10 | 0.0015 | – | – | – | – |

Example 4.2 (SOCNCP test problems). We test different nonlinear functions F and cones \mathcal{K} as follows.

SOCNCP1 ([12, 20, 23]). $F(x) = \begin{pmatrix} 0.07x_1^3 - 4 \\ 0.04x_2^3 - 3.93 \\ 0.03x_3^3 - 5.72 \end{pmatrix}$ and $\mathcal{K} = \mathcal{K}^3$.

SOCNCP2 ([12, 33]).

$$F(x) = \begin{pmatrix} 24(2x_1 - x_2)^3 + e^{x_1 - x_3} - 4x_4 + x_5 \\ -12(2x_1 - x_2)^3 + 3(3x_2 + 5x_3)/\sqrt{1 + (3x_2 + 5x_3)^2} - 6x_4 - 7x_5 \\ -e^{x_1 - x_3} + 5(3x_2 + 5x_3)/\sqrt{1 + (3x_2 + 5x_3)^2} - 3x_4 + 5x_5 \\ 4x_1 + 6x_2 + 3x_3 - 1 \\ -x_1 + 7x_2 - 5x_3 + 2 \end{pmatrix}$$

and $\mathcal{K} = \mathcal{K}^3 \times \mathcal{K}^2$.

$$\text{SOCNCP3 ([20])}. F(x) = \begin{pmatrix} \frac{e^{a^T x^1}}{1+e^{a^T x^1}} a + \frac{c^T x^1}{\sqrt{3+(c^T x^1)^2}} c + d - A^T x^2 \\ Ax^1 + b \end{pmatrix}, \text{ where}$$

$$\begin{aligned} a &= (10, 5, -4, -8)^T, & b &= (1, 0, 0, 0)^T, & c &= (6, 2, -3, -5)^T \\ d &= (6, 3.5, -7.5, -3.5)^T, & A &= \text{diag}(5/3, 1, -4, 2), \end{aligned}$$

and $\mathcal{K} = \mathcal{K}^4 \times \mathcal{K}^4$.

$$\text{SOCNCP4 ([29])}. F(x) = \begin{pmatrix} 2x_1 + 2x_2 - 10 + x_3 + 2(x_1 + 1)x_4 \\ 2x_1 + 4x_2 - 12 - 3x_3 + 2(x_2 - 1)x_4 \\ 8 - x_1 + 3x_2 \\ 3 - x_1^2 - 2x_1 + 2x_2 - x_2^2 \end{pmatrix}$$

and $\mathcal{K} = \mathcal{K}^2 \times \mathcal{K}^2$.

SOCNCP5 ([33]).

$$F(x) = \begin{pmatrix} 2x_1 + x_2 + 1 \\ x_1 + 6x_2 - x_3 - 2 \\ -x_2 + 3x_3 - \frac{6}{5}x_4 + 3 \\ -\frac{6}{5}x_3 + 2x_4 + \frac{1}{2} \sin x_4 \cos x_5 \sin x_6 + 6 \\ \frac{1}{2} \cos x_4 \sin x_5 \sin x_6 + 2x_5 - \frac{5}{2} \\ -\frac{1}{2} \cos x_4 \cos x_5 \cos x_6 + 2x_6 + \frac{1}{4} \cos x_6 \sin x_7 \cos x_8 + 1 \\ \frac{1}{4} \sin x_6 \cos x_7 \cos x_8 + 4x_7 - 2 \\ -\frac{1}{4} \sin x_6 \sin x_7 \sin x_8 + 2x_8 + \frac{1}{2} \end{pmatrix}$$

and $\mathcal{K} = \mathcal{K}^3 \times \mathcal{K}^3 \times \mathcal{K}^2$.

Example 4.3 (SOCTCP test problems). *We consider the second-order cone complementarity problem which involves a tensor $\mathcal{A} = (a_{i_1 i_2 \dots i_m}) \in \mathbb{T}^m(\mathbb{R}^n)$ ($i_j \in \{1, \dots, n\}$), a function $F(x) = \mathcal{A}x^{m-1} - b$ where $\mathcal{A}x^{m-1} \in \mathbb{R}^n$ with components*

$$(\mathcal{A}x^{m-1})_i := \sum_{i_2, i_3, \dots, i_m=1}^m a_{i i_2 \dots i_m} x_{i_2} x_{i_3} \cdots x_{i_m}, \quad \forall i \in \{1, \dots, n\},$$

and $\mathcal{K} = \mathcal{K}^n$. We test different tensors \mathcal{A} from the indicated references.

SOCTCP1 ([32]) *The entries of \mathcal{A} are listed below*

$$\begin{aligned}\mathcal{A}(:, :, 1) &= \begin{pmatrix} 0.4333 & 0.4278 & 0.4140 \\ 0.8154 & 0.0199 & 0.5598 \\ 0.0643 & 0.3815 & 0.8834 \end{pmatrix} \\ \mathcal{A}(:, :, 2) &= \begin{pmatrix} 0.4866 & 0.8087 & 0.2073 \\ 0.7641 & 0.9924 & 0.8752 \\ 0.6708 & 0.8296 & 0.1325 \end{pmatrix} \\ \mathcal{A}(:, :, 3) &= \begin{pmatrix} 0.3871 & 0.0769 & 0.3151 \\ 0.1355 & 0.7727 & 0.4089 \\ 0.9715 & 0.7726 & 0.5526 \end{pmatrix}\end{aligned}$$

and we set $b = (-4, -3, 1)^T$.

SOCTCP2 $\mathcal{A} \in \mathbb{T}^4(\mathbb{R}^2)$ with nonzero entries $\mathcal{A}_{1ij1} = 1$ and $\mathcal{A}_{2ij2} = -2$ for all $i, j \in \{1, 2\}$ and we set $b = (-1, 1)^T$.

SOCTCP3 ([14]) $\mathcal{A} \in \mathbb{T}^4(\mathbb{R}^n)$ such that $\mathcal{A}_{i_1 i_2 i_3 i_4} = \arctan(i_1 i_2^2 i_3^3 i_4^4)$, and we set $b = -e$.

Table 4: Comparison of algorithms for solving SOCTCP1 and SOCTCP2 using the smoothing function ϕ_1^- . Note that $\mathbf{0}$ is a solution of SOCTCP1 and SOCTCP2.

| Test Problem | Algorithm | $x^0 = \mathbf{1}$ | | $x^0 = \mathbf{e}$ | | $x^0 = -\mathbf{1}$ | | $x^0 = 10 \cdot \mathbf{1}$ | |
|--------------|-----------|--------------------|--------|--------------------|--------|---------------------|--------|-----------------------------|--------|
| | | NI | CPU | NI | CPU | NI | CPU | NI | CPU |
| SOCTCP1 | SPPA | 29 | 0.0014 | 29 | 0.0011 | 28 | 0.0010 | 48 | 0.0028 |
| | ReSNA | – | – | – | – | 5 | 0.0010 | – | – |
| | SNFB | – | – | – | – | – | – | 12 | 0.0014 |
| SOCTCP2 | SPPA | 18 | 0.0010 | 18 | 0.0007 | 13 | 0.0006 | 24 | 0.0009 |
| | ReSNA | 16 | 0.0027 | 16 | 0.0028 | – | – | 4 | 0.0007 |
| | SNFB | 13 | 0.0023 | 13 | 0.0023 | – | – | 19 | 0.0020 |

5 Conclusions

In this work, we proposed a smoothing power penalty approach for solving the SOCCP (2), wherein the SOCCP (2) is approximated by a nonlinear equation with power penalty and smoothing parameters. Under the assumption that the function involved has the uniform ξ -P property, we provided a theoretical guarantee that the solution sequence of the APE (10) converges to the unique solution of the SOCCP (2). Our proposed algorithm yields promising performance as compared with two existing well-known methods in the

Table 5: Comparison of algorithms for solving SOCTCP3 for different values of n using the smoothing function ϕ_2^- . Note that $\mathbf{0}$ is a solution of SOCTCP3 for any n .

| Dimension | Algorithm | $x^0 = \mathbf{1}$ | | $x^0 = \mathbf{e}$ | | $x^0 = -\mathbf{1}$ | | $x^0 = 10 \cdot \mathbf{1}$ | |
|-----------|-----------|--------------------|---------|--------------------|---------|---------------------|---------|-----------------------------|---------|
| | | NI | CPU | NI | CPU | NI | CPU | NI | CPU |
| $n = 5$ | SPPA | 14 | 0.0068 | 11 | 0.0019 | 29 | 0.0051 | 36 | 0.0075 |
| | ReSNA | 14 | 0.0148 | 6 | 0.0036 | 11 | 0.0056 | 6 | 0.0028 |
| | SNFB | 14 | 0.0078 | 8 | 0.0027 | 12 | 0.0026 | 20 | 0.0113 |
| $n = 10$ | SPPA | 27 | 0.0351 | 15 | 0.0125 | 34 | 0.0278 | 18 | 0.0147 |
| | ReSNA | 15 | 0.0262 | 6 | 0.0134 | 5 | 0.0072 | 13 | 0.0260 |
| | SNFB | 14 | 0.0201 | 9 | 0.0114 | 14 | 0.0099 | 19 | 0.0150 |
| $n = 20$ | SPPA | 20 | 0.0700 | 33 | 0.0994 | 16 | 0.0441 | 43 | 0.1076 |
| | ReSNA | 5 | 0.0264 | 6 | 0.0326 | 13 | 0.0760 | 21 | 0.1012 |
| | SNFB | 16 | 0.0704 | 9 | 0.0606 | 16 | 0.0681 | 22 | 0.0988 |
| $n = 50$ | SPPA | 18 | 0.7924 | 28 | 1.2155 | 24 | 1.0365 | 32 | 1.3768 |
| | ReSNA | 15 | 1.3777 | 6 | 0.5939 | 16 | 1.4744 | 26 | 2.4700 |
| | SNFB | 21 | 1.5373 | 9 | 0.3724 | 20 | 1.4494 | 24 | 0.9822 |
| $n = 100$ | SPPA | 36 | 23.7014 | 23 | 15.1790 | 15 | 9.9174 | 39 | 25.6518 |
| | ReSNA | 22 | 29.3993 | 9 | 14.5196 | 10 | 13.8577 | – | – |
| | SNFB | 27 | 74.8485 | 8 | 5.2614 | 29 | 84.3143 | 26 | 16.8180 |

literature, as we have shown in our extensive numerical experiments involving second-order cone linear, nonlinear, and tensor complementarity problems.

References

- [1] ALCANTARA, J. H., AND CHEN, J.-S. A new class of neural networks for ncps using smooth perturbations of the natural residual function. *Journal of Computational and Applied Mathematics* 407 (2022), 114092.
- [2] ALCANTARA, J. H., NGUYEN, C. T., OKUNO, T., TAKEDA, A., AND CHEN, J.-S. Unified smoothing approach for best hyperparameter selection problem using a bilevel optimization strategy, 2021. arXiv:2110.12630.
- [3] ALIZADEH, F., AND GOLDFARB, D. Second-order cone programming. *Mathematical Programming* 95 (2003), 3–51.
- [4] AUSLENDER, A. Variational inequalities over the cone of semidefinite positive matrices and over the lorentz cone. *Optimization Methods and Software* 18 (2003), 359–376.

- [5] BILLUPS, S. C., AND MURFY, K. G. Complementarity problems. *Journal of Computational and Applied Mathematics* 124 (2000), 303–318.
- [6] CHEN, C., AND MANGASARIAN, O. L. A class of smoothing functions for nonlinear and mixed complementarity problems. *Computational Optimization and Applications* 5 (1996), 97–138.
- [7] CHEN, J. Two classes of merit functions for the second-order cone complementarity problem. *Mathematical Methods of Operations Research* 64, 3 (2006), 495–519.
- [8] CHEN, J., AND PAN, S. A descent method for a reformulation of the second-order cone complementarity problem. *Journal of Computational and Applied Mathematics* 213, 2 (2008), 547–558.
- [9] CHEN, J.-S. *SOC Functions and their Applications*, vol. 143. Springer, Singapore, 2019.
- [10] CHEN, J.-S., CHEN, X., AND TSENG, P. Analysis of nonsmooth vector-valued functions associated with second-order cones. *Mathematical Programming* 101 (2004), 95–117.
- [11] CHEN, J.-S., AND TSENG, P. An unconstrained smooth minimization reformulation of the second-order cone complementarity problem. *Mathematical Programming* 104 (2005), 297–327.
- [12] CHEN, L., AND MA, C. A modified smoothing and regularized Newton method for monotone second-order cone complementarity problems. *Computers and Mathematics with Applications* 61 (2011), 1407–1418.
- [13] CHEN, X., SUN, D., AND SUN, J. Complementarity functions and numerical experiments on some smoothing Newton methods for second-order-cone complementarity problems. *Computational Optimization and Applications* 25 (2003), 39–56.
- [14] CHENG, L., AND ZHANG, X. A semidefinite relaxation method for second-order cone polynomial complementarity problems. *Computational Optimization and Applications* 75 (2020), 629–647.
- [15] FACCHINEI, F., AND PANG, J.-S. *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer-Verlag, New York, NY, 2003.
- [16] FARAUT, J., AND KORÁNYI, A. *Analysis on Symmetric Cones*. Oxford Mathematical Monographs. Clarendon Press, 1994.
- [17] FERRIS, M. C., AND PANG, J. S. Engineering and economic applications of complementarity problems. *SIAM Review* 39, 4 (1997), 669–713.

- [18] FUKUSHIMA, M., LUO, Z.-Q., AND TSENG, P. Smoothing functions for second-order cone complementarity problems. *SIAM Journal on Optimization* 12 (2002), 436–460.
- [19] HAO, Z., NGUYEN, C., AND CHEN, J.-S. An approximate lower order penalty approach for solving second-order cone linear complementarity problems. *Journal of Global Optimization* 83 (2022), 671–697.
- [20] HAO, Z., WAN, Z., CHI, X., AND CHEN, J. A power penalty method for second-order cone nonlinear complementarity problems. *Journal of Computational and Applied Mathematics* 290 (2015), 136–149.
- [21] HAYASHI, S., YAMAGUCHI, T., YAMASHITA, N., AND FUKUSHIMA, M. A matrix-splitting method for symmetric affine second-order cone complementarity problems. *Journal of Computational and Applied Mathematics* 175, 2 (2005), 335–353.
- [22] HAYASHI, S., YAMASHITA, N., AND FUKUSHIMA, M. A combined smoothing and regularization method for monotone second-order cone complementarity problems. *SIAM Journal on Optimization* 15, 2 (2005), 563–615.
- [23] HAYASHI, S., YAMASHITA, N., AND FUKUSHIMA, M. A combined smoothing and regularization method for monotone second-order cone complementarity problems. *SIAM Journal on Optimization* 15, 2 (2005), 563–615.
- [24] HAYASHI, S., YAMASHITA, N., AND FUKUSHIMA, M. Robust nash equilibria and second-order cone complementarity problems. *Journal of Nonlinear and Convex Analysis* 6, 2 (01 2005), 283–296.
- [25] HUANG, C., AND WANG, S. A power penalty approach to a nonlinear complementarity problem. *Operations Research Letters* 38 (2010), 72–76.
- [26] HUANG, Z.-H., AND NI, T. Smoothing algorithms for complementarity problems over symmetric cones. *Computational Optimization and Applications* 45 (04 2010), 557–579.
- [27] KANZOW, C., FERENCZI, I., AND FUKUSHIMA, M. On the local convergence of semismooth newton methods for linear and nonlinear second-order cone programs without strict complementarity. *SIAM Journal on Optimization* 20, 1 (2009), 297–320.
- [28] LOBO, M. S., VANDENBERGHE, L., BOYD, S., AND LEBRET, H. Applications of second-order cone programming. *Linear Algebra and its Applications* 284, 1 (1998), 193–228.

- [29] MIAO, X., CHEN, J.-S., AND KO, C.-H. A smoothed NR neural network for solving nonlinear convex programs with second-order cone constraints. *Information Sciences* 268 (2014), 255–270.
- [30] MONTEIRO, R., AND TSUCHIYA, T. Polynomial convergence of primal-dual algorithms for the second-order cone program based on the MZ-family of directions. *Mathematical Programming, Series B* 88 (2000), 61–83.
- [31] PAN, S., AND CHEN, J. A regularization method for the second-order cone complementarity problem with the Cartesian P_0 -property. *Nonlinear Analysis, Theory, Methods and Applications* 70, 4 (2009), 1475–1491.
- [32] QI, L. The best rank-one approximation ratio of a tensor space. *SIAM Journal on Matrix Analysis and Applications* 32, 2 (2011), 430–442.
- [33] SUN, J., FU, W., ALCANTARA, J. H., AND CHEN, J.-S. A neural network based on the metric projector for solving SOCCVI problem. *IEEE Transactions on Neural Networks and Learning Systems* 32, 7 (2021), 2886–2900.
- [34] TIAN, B., HU, Y., AND YANG, X. A box-constrained differentiable penalty method for nonlinear complementarity problems. *Journal of Global Optimization* 62 (2015), 724–747.
- [35] TIAN, B., AND YANG, X. Smoothing power penalty method for nonlinear complementarity problems. *Pacific Journal of Optimization* 12, 2 (2016), 461–484.
- [36] WANG, S., AND YANG, X. A power penalty method for linear complementarity problems. *Operations Research Letters* 36, 2 (2008), 211–214.
- [37] ZHANG, L.-H., AND YANG, W. H. An efficient matrix splitting method for the second-order cone complementarity problem. *SIAM Journal on Optimization* 24, 3 (2014), 1178–1205.