Journal of Nonlinear and Convex Analysis
Volume 17, Number 2, 2016, 225–241

A SELF-CONCORDANT INTERIOR POINT ALGORITHM FOR NONSYMMETRIC CIRCULAR CONE PROGRAMMING

PENGFEI MA, YANQIN BAI*, AND JEIN-SHAN CHEN

Abstract. In this paper, we consider a particular conic optimization problem over nonsymmetric circular cone. This class of optimization problem has been found useful in optimal grasping manipulation problems for multi-fingered robots. We first introduce a pair of logarithmically homogeneous self-concordant barrier function for circular cone and its dual cone. Then, based on these two logarithmically homogeneous self-concordant barrier functions and their related properties, we present an interior point algorithm for circular cone optimization problem. Furthermore, we derive the iteration bound for this interior point algorithm. Finally, we show some numerical tests to demonstrate the performance of the proposed algorithm.

1. Introduction

Nonsymmetric circular cone programming problems are convex programming problems because their objectives are linear functions and their feasible sets are the intersection of an affine space with the Cartesian product of a finite number of circular cones. The circular cone [10] is defined as

$$\begin{aligned} \mathcal{C}_\theta^n : &= \{(x_1, x_{2:n})^T \in \mathcal{R} \times \mathcal{R}^{n-1} | \cos\theta \|x\| \le x_1\} \\ &= \{(x_1, x_{2:n})^T \in \mathcal{R} \times \mathcal{R}^{n-1} | \|x_{2:n}\| \le x_1 \tan\theta\}. \end{aligned} \tag{1.1}$$

where $\theta \in (0, \frac{\pi}{2})$ is called rotation angle, $\|\cdot\|$ denotes the Euclidean norm and $(\mathcal{C}_\theta^n)^*$ is the dual cone of $\mathcal{C}_\theta^n$. It is easy to verify that

$$(\mathcal{C}_\theta^n)^* = \mathcal{C}_{\frac{\pi}{2}-\theta}^n = \{(x_1, x_{2:n})^T \in \mathcal{R} \times \mathcal{R}^{n-1} \mid \sin\theta \|x\|_2 \le x_1\}. \tag{1.2}$$

The geometric illustration of a circular cone, its dual cone, and a second order cone are depicted in Figure 1.

A nonsymmetric circular cone programming problem is an optimization problem with the following form:

$$\begin{aligned} \min_x \quad & c^T x \\ \text{s.t.} \quad & Ax = b, \\ & x \in \mathcal{K}, \end{aligned} \tag{1.3}$$

where $\mathcal{K} \subset \mathcal{R}^n$ is the Cartesian product of several circular cones, i.e.,

$$\mathcal{K} = \mathcal{C}_{\theta_1}^{n_1} \times \mathcal{C}_{\theta_2}^{n_2} \times \cdots \times \mathcal{C}_{\theta_N}^{n_N}, \tag{1.4}$$

2010 *Mathematics Subject Classification.* 90C25, 90C51, 90C60.

Key words and phrases. Conic optimization problem, interior point methods (IPMs), self-concordant barriers.

*Corresponding author. This research was supported by a grant from the National Natural Science Foundation of China (No.11371242).

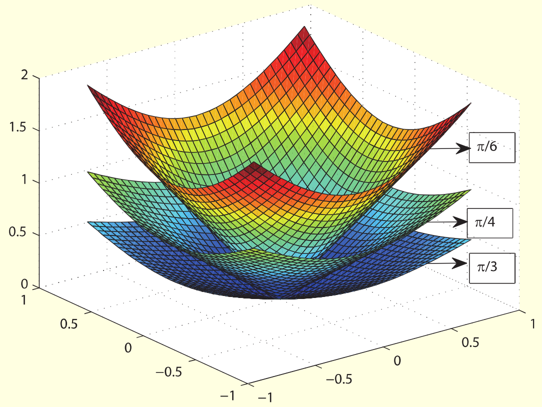

FIGURE 1. The graph of circular cone in $\mathcal{R}^3$

with $n = \sum_{j=1}^{N} n_j$ and $\theta_j \in (0, \frac{\pi}{2})$ for $j = 1, 2, \ldots, N$. Furthermore, we partition the vectors x, c and matrix A as $x = (x^1; x^2; \ldots; x^N)$ with $x^j \in \mathcal{C}^{n_j}_{\theta_j}$, $c = (c^1; c^2; \ldots; c^N)$ with $c^j \in \mathcal{R}^{n_j}$, and $A = (A^1, A^2, \ldots, A^N)$ with $A^j \in \mathcal{R}^{m \times n_j}$, and $b \in \mathcal{R}^m$. Without loss of generality, we assume that the matrix A has full row rank, i.e., rank(A)=m. Obviously, the problem (1.3) is also expressed as

$$
\begin{aligned}
\min_{x} \quad & c^T x \\
\text{s.t.} \quad & Ax = b, \\
& f_i(x) \le 0, \ i = 1, 2, \ldots, N,
\end{aligned} \tag{1.5}
$$

where $f_i(x) = \|x^i_{2:n}\|^2 - (x^i_1)^2 \cdot \tan^2 \theta_i$ for $i = 1, 2, \ldots, N$.

The problem (1.5) is a second order cone programming (SOCO) problem if $\theta_j = \frac{\pi}{4}$ for $j = 1, 2, \ldots, N$. It is well-known that second order cone programming problems have had widely applications (see, e.g., [1, 13]). Moreover, the circular cone described by (1.1) with $\theta \neq \frac{\pi}{4}$ naturally arises in many real-life engineering problems [6, 7, 11, 12]. One example is to formulate optimal grasping manipulation for multi-fingered robots. The grasping force of the i-th finger can be expressed in the local coordinate frame n_i, o_i, t_i by $f_i = (f_{n_i}, f_{o_i}, f_{t_i})^T$ where f_{n_i}, f_{o_i} and f_{t_i} are the components of f_i along n_i, o_i and t_i, respectively. To ensure no slipping at a contact point, the components of the contact force f_i must satisfy the contact constraint

$$
\|(f_{o_i}, f_{t_i})\| \le \mu f_{n_i}, \tag{1.6}
$$

where μ is the static friction coefficient of the substrate. In fact, (1.6) geometrically represent a circular cone with rotation angle $\beta = \tan^{-1} \mu$ (see Figure 2).

By (1.1) and (1.2), as long as rotation angle $\theta = \frac{\pi}{4}$, the circular cone and its dual cone reduce to the well-known second order cone (also known as the Lorentz cone and the ice-cream cone) given by

$$
\mathcal{L}^n := \{(x_1, x_{2:n})^T \in \mathcal{R} \times \mathcal{R}^{n-1} | \|x_{2:n}\| \le x_1\}. \tag{1.7}
$$

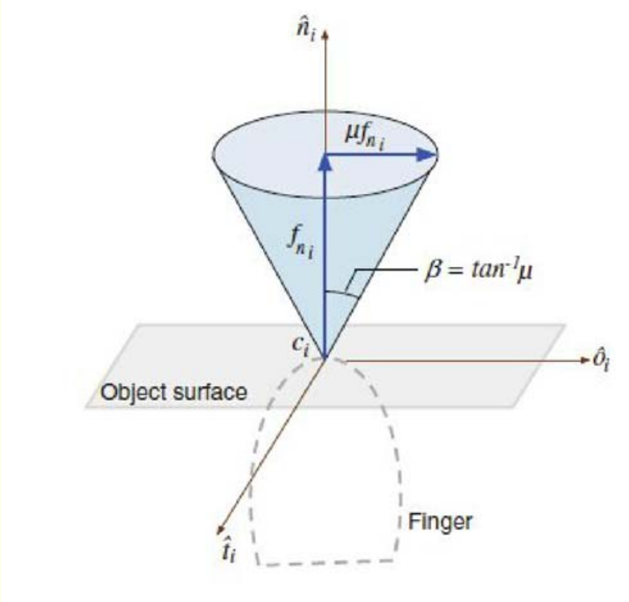

FIGURE 2. Circular cone at a grasp point

It is clear that the second order cone is a symmetric cone. However, the circular cone is a nonsymmetric cone because it is not self-dual, i.e., $\mathcal{K} \neq \mathcal{K}^*$ unless $\theta = \frac{\pi}{4}$. A main difference between circular cone constraints and most of the other cone constraints [2, 5, 22] is that the circular cone is nonsymmetric, which makes the problem (1.5) more challenging.

As mentioned in [25], there is a close relation between $\mathcal{C}_\theta^n$ and $\mathcal{L}^n$ as below

$$\mathcal{L}^n = A_\theta \mathcal{C}_\theta^n \quad \text{where} \quad A_\theta := \begin{bmatrix} \tan\theta & 0 \\ 0 & E_{n-1} \end{bmatrix}.$$

We point out that, with the above transformation, it is possible to construct a new inner product which ensures the circular cone $\mathcal{C}_\theta^n$ to be self-dual. More precisely, we define an inner product associated with A_θ as $\langle x, y\rangle_{A_\theta} := \langle A_\theta x, A_\theta y\rangle$. Then, we have

$$\begin{aligned} (\mathcal{C}_\theta^n)^* &= \{x \mid \langle x, y\rangle_{A_\theta} \geq 0,\ \forall y \in \mathcal{C}_\theta^n\} = \{x \mid \langle A_\theta x, A_\theta y\rangle \geq 0,\ \forall y \in A_\theta^{-1}\mathcal{L}^n\} \\ &= \{x \mid \langle A_\theta x, y\rangle \geq 0,\ \forall y \in \mathcal{L}^n\} = \{x \mid A_\theta x \in \mathcal{L}^n\} \\ &= A_\theta^{-1}\mathcal{L}^n = \mathcal{C}_\theta^n. \end{aligned}$$

However, under this new inner product the second-order cone is not self-dual, because

$$\begin{aligned} (\mathcal{L}^n)^* &= \{x \mid \langle x, y\rangle_{A_\theta} \geq 0,\ \forall y \in \mathcal{L}^n\} = \{x \mid \langle A_\theta x, Ay\rangle \geq 0,\ \forall y \in \mathcal{L}^n\} \\ &= \{x \mid \langle A_\theta^2 x, y\rangle \geq 0,\ \forall y \in \mathcal{L}^n\} = \{x \mid A_\theta^2 x \in \mathcal{L}^n\} = A_\theta^{-2}\mathcal{L}^n. \end{aligned}$$

Since we cannot find an inner product such that the circular cone and second-order cone are both self-dual simultaneously, we must choose an inner product from the standard inner product or the new inner product associated with A_θ. In view of the well-known properties regarding second-order cone and second-order cone programming (in which many results are based on the Jordan algebra and second-order cones are considered as self-dual cones), we adopt the standard inner product in this paper.

Some researchers have investigated circular cones and nonsymmetric circular cone programming problems. In [9, 24–26], Chen et al. paid a lot attentions to study some properties of circular cone and vector-valued functions associated with circular cones. In [27], Zhou et al. established complete characterizations of full and tilt stability of locally optimal solutions to parameterized circular cone programming problems. Moreover, Bai et al. considered kernel function-based interior point algorithm for solving the problem (1.3) or (1.5) in [3]. They conclude that the problem (1.5) is polynomial-time solvable. At the same time, Bai et al. also investigated kernel function-based interior point algorithm for convex quadratic circular cone programming problems in [4].

Recently, Nesterov [21] proposed a new interior point algorithm that is based on an extension of the ideas of self-scaled optimization to the nonsymmetric conic optimization. The author developed a $4n$-self-concordant barrier for an n-dimensional p-cone, which is a special case of nonsymmetric cone. Matsukawa and Yoshise [14] proposed a primal barrier function phase I algorithm for solving conic optimization problems over doubly nonnegative cone. Skajaa and Ye [23] designed a homogeneous interior point algorithm for nonsymmetric convex conic optimization. All these IPMs are designed based on self-concordant barrier functions for its corresponding cone.

Self-concordant barrier functions are presented by Nesterov and Nemirovski [15]. They play an important role in the powerful polynomial-time IPMs for convex programming. Several classes of interior point algorithms for linear programming are extended to nonlinear setting in terms of self-concordant barrier functions for convex region. Following the work of Nesterov and Nemirovski, many articles have issued using this type of function to construct barrier functions for IPMs [16–18]. In [20] and [8], the authors also presented 3-self-concordant barriers for the nonsymmetric power cone and the exponential cone, respectively. Therefore, conic programming problems with the power cone or the exponential cone constraints can be solved by efficient interior point algorithm [23].

Inspirited by the nice properties of self-concordant barrier functions, in this paper, we consider a particular conic optimization problem over nonsymmetric circular cone, which has been found useful application in optimal grasping manipulation problems for multi-fingered robots. We first introduce a pair of logarithmically homogeneous self-concordant barrier function for circular cone and its dual cone. Then, based on these two logarithmically homogeneous self-concordant barrier functions and their related properties, we present an interior point algorithm for circular cone optimization problem. Furthermore, we derive the iteration bound for this interior point algorithm. Finally, we show some numerical tests to demonstrate the performance of the proposed algorithm.

The paper is organized as follows. In Section 2, we recall basic concepts and properties on self-concordant barrier functions. In Section 3, we introduce a pair of logarithmically homogeneous self-concordant barrier functions for circular cone and its dual cone, respectively. In Section 4, we discuss optimality conditions and central paths of nonsymmetric circular cone programming problems. In Section 5, based on logarithmically homogeneous self-concordant barrier function for circular cone and its dual cone, we present an interior point algorithm for nonsymmetric circular cone programming. In Section 6, we implement our algorithm by several random examples to show the performance of the algorithm. Finally, we conclude and give further research in Section 7.

2. Preliminaries

As mentioned in the Introduction, self-concordant barrier functions are crucial to IPMs. In order to proceed with our discussion, we recall some basic concepts and properties of self-concordant barrier functions which will be used in this paper. The materials can be found in [15] and [19], we here omit their proofs.

Given a closed convex function $f(x)(\text{dom } f)$ with open domain and fix a point $x \in \text{dom} f$ and a direction $u \in \mathcal{R}^n$, we consider the function

$$\phi(t) = f(x + tu),$$

depending on the variable $t \in \text{dom}\phi(x;\cdot) \subseteq \mathcal{R}$. Then, we denote

$$\begin{aligned} Df(x)[u] &= \phi'(t) = \langle \nabla f(x), u\rangle, \\ D^2 f(x)[u,u] &= \phi''(t) = \langle \nabla^2 f(x)u, u\rangle, \\ D^3 f(x)[u,u,u] &= \phi'''(t). \end{aligned}$$

With these notations, self-concordant function and self-concordant barrier function are defined as follows.

Definition 2.1. A closed convex function $F \in C^3$ (three times continuously differentiable) with open domain C is called self-concordant if

$$|D^3 F(x)[h,h,h]| \leq 2(D^2 F(x)[h,h])^{3/2}, \tag{2.1}$$

for all $x \in \text{dom}F$ and for all $h \in \mathcal{R}^n$.

Definition 2.2. A self-concordant function F is a ν-self-concordant barrier for a closed convex set $\mathcal{K}$ if

$$\nabla F(x)^T (\nabla^2 F(x))^{-1} \nabla F(x) \leq \nu, \ \forall x \in \mathcal{K}^\circ.$$

The value ν is called the parameter of the barrier F and $\mathcal{K}^\circ$ is interior of the set $\mathcal{K}$.

In order to prove our Theorem 3.3, we need a property regarding ν-self-concordant barrier under an affine transformation.

Lemma 2.3. *Let $F: \ C^\circ \subseteq \mathcal{R}^n \to \mathcal{R}$ be a ν-self-concordant barrier, $\mathcal{A}: \mathcal{R}^p \to \mathcal{R}^n$ such that $\mathcal{A}(y) = By + b$ for $B \in \mathcal{R}^{n\times p}$ and $b \in \mathcal{R}^n$. Assume $\mathcal{A}(\mathcal{R}^p) \cap C \neq \emptyset$. Define $C^+ := \mathcal{A}^{-1}(C) = \{y \in \mathcal{R}^p : \mathcal{A}(y) \in C\} \subseteq \mathcal{R}^p$. Then $\tilde{F}: C^+ \to \mathcal{R}$ defined as*

$$\tilde{F}(y) = F(\mathcal{A}(y))$$

is a ν-self-concordant barrier for C^+.

For proper cones, Nesterov and Nemirovski have presented a special class of barriers in [15] and [19]. The definition is stated as follows.

Definition 2.4. Let $\mathcal{K}$ be a proper cone, $F : \mathcal{K}^\circ \to \mathcal{R}$ a twice continuously differentiable, convex barrier function. F is called ν-logarithmically homogeneous for $\mathcal{K}$ if

$$F(tx) = F(x) - \nu \ln t \tag{2.2}$$

for any $x \in \mathcal{K}$ and any $t > 0$.

In order to introduce logarithmically homogeneous barrier of dual cone of the circular cone, we employ the following Definition 2.5 and Lemma 2.6.

Definition 2.5. Let F be a ν-logarithmically homogeneous barrier for $\mathcal{K}$. Its conjugate is defined as

$$F_*(s) = \sup_{x \in \mathcal{K}^\circ} \{-s^T x - F(x)\}.$$

Lemma 2.6. *Let $F : \mathcal{K}^\circ \to \mathcal{R}$ be a ν-self-concordant barrier for $\mathcal{K}$. Then $F_*(s)$ is a ν-self-concordant barrier for $\mathcal{K}^*$.*

For the positive orthant, the second order cone, and the cone of positive semidefinite matrices, we list their self-concordant barriers in Table 1. Note that the three cones lead to linear programming problems, second order cone programming problems, and semidefinite programming problems, respectively. In all these examples, the cones are symmetric and the barriers are self-scaled [17]. However, in general, this cannot be true.

TABLE 1. self-concordant barriers over some convex cones

Cones	Self-concordant barriers	Parameters	Conjugate functions
$\mathcal{R}^n_+ = \{x \in \mathcal{R}^n : x \geq 0\}$	$F(x) = -\sum_{i=1}^{n} \ln(x_i)$	$\nu = n$	$F_*(s) = F(s) - n$
$\{(x_1, x_{2:n})^T \in \mathcal{R} \times \mathcal{R}^{n-1} \| \|x_{2:n}\| \leq x_1\}$	$F(x_1, x_{2:n}) = -\ln(x_1^2 - \|x_{2:n}\|^2)$	$\nu = 2$	$F_*(s) = F(s) + 2\ln 2 - 2$
$\mathcal{S}^+_n = \{X \in S^n : X \succeq 0\}$	$F(X) = -\ln\det(X)$	$\nu = n$	$F_*(s) = F(s) - n$

3. THE BARRIER FUNCTION FOR CIRCULAR CONE

In this section we introduce logarithmically homogeneous self-concordant barrier functions for the circular cone and its dual cone.

First, we recall an important Lemma in [3, Theorem 2.3]. The Lemma is critical to our subsequent analysis. Given a rotation angle θ, let

$$A_\theta := \begin{bmatrix} \tan\theta & 0 \\ 0 & E_{n-1} \end{bmatrix} \tag{3.1}$$

where E_{n-1} is an $n-1$ dimensional unit matrix. It is straightforward to verify that

$$A_\theta^{-1} = \begin{bmatrix} \cot\theta & 0 \\ 0 & E_{n-1} \end{bmatrix}. \tag{3.2}$$

Lemma 3.1. *For any $x \in \mathcal{C}^n_\theta$ and $s \in (\mathcal{C}^n_\theta)^*$, there exist $\tilde{x} \in \mathcal{L}^n$ and $\tilde{s} \in \mathcal{L}^n$ such that*

(1) *$\tilde{x} = A_\theta x$ and $\tilde{s} = A_\theta^{-1} s$.*
(2) *$\tilde{x}^T \tilde{s} = 0$ if and only if $x^T s = 0$.*

Then, we prove the following Lemma 3.2 to obtain a self-concordant barrier of dual cone.

Lemma 3.2. *Let h be a convex function on $\mathcal{R}^n$, and let*

$$f(x) = h(Ax + b) + a^T x + \alpha,$$

where A is a one-to-one linear transformation from $\mathcal{R}^n$ to $\mathcal{R}^n$, a and b are vectors in $\mathcal{R}^n$, and $\alpha \in \mathcal{R}$. Then

$$f_*(s) = h_*\left((A^{-1})^T(a + s)\right) + (A^{-1}b)^T s + \alpha^*,$$

where $\alpha^* = -\alpha + (A^{-1}b)^T a$.

Proof. The substitution $y = Ax + b$ enables us to calculate f_* as follows

$$\begin{aligned} f_*(s) &= \sup_x \{-s^T x - h(Ax + b) - a^T x - \alpha\} \\ &= \sup_y \{-s^T \left(A^{-1}(y - b)\right) - h(y) - \left(A^{-1}(y - b)\right)^T a - \alpha\} \\ &= \sup_y \{-(A^{-1}y)^T(s + a) - h(y)\} + (A^{-1}b)^T(s + a) - \alpha \\ &= \sup_y \{-y^T(A^{-1})^T(s + a) - h(y)\} + (A^{-1}b)^T s + (A^{-1}b)^T a - \alpha \\ &= h_*\left((A^{-1})^T(s + a)\right) + (A^{-1}b)^T s + \alpha^*. \end{aligned}$$

□

Based on the self-concordant barrier for the second order cone, Lemma 2.3, Lemma 2.6, Lemma 3.1 and Lemma 3.2, we introduce self-concordant barriers for the circular cone and its dual cone.

Theorem 3.3. *The function*

$$F_\theta(x) = -\ln(x_1^2 \cdot \tan^2\theta - \|x_{2:n}\|^2) \tag{3.3}$$

is a 2-self-concordant barrier for $x \in \mathcal{C}_\theta^n$ and the function

$$(F_\theta)_*(s) = -\ln(s_1^2 \cdot \cot^2\theta - \|s_{2:n}\|^2) + 2\ln 2 - 2 \tag{3.4}$$

is a 2-self-concordant barrier for $s \in (\mathcal{C}_\theta^n)^*$. Furthermore,

$$F_{\mathcal{K}}(x) = -\sum_{j=1}^{N} \ln\left((x_1^j)^2 \cdot \tan^2\theta_j - \|x_{2:n}^j\|^2\right) \tag{3.5}$$

is $2N$-self-concordant barrier for $\mathcal{K}$.

Proof. Obviously, the function

$$F(x) = -\ln(x_1^2 - \|x_{2:n}\|^2)$$

is 2-self-concordant barrier for the second order cone and its conjugate function is

$$F_*(s) = F(s) + 2\ln 2 - 2.$$

Using Lemma 2.3 and Lemma 3.2, one has

$$F_\theta(x) = F\left(\begin{bmatrix} \tan\theta & 0 \\ 0 & E_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_{2:n} \end{bmatrix}\right) = -\ln(x_1^2 \cdot \tan^2\theta - \|x_{2:n}\|^2)$$

is a 2-self-concordant barrier for $\mathcal{C}_\theta^n$. According to Lemma 3.1, Lemma 2.6 and Lemma 3.2, we obtain

$$(F_\theta)_*(s) = F_*\left(\begin{bmatrix} \cot\theta & 0 \\ 0 & E_{n-1} \end{bmatrix}\begin{bmatrix} s_1 \\ s_{2:n} \end{bmatrix}\right) = -\ln(s_1^2 \cdot \cot^2\theta - \|s_{2:n}\|^2) + 2\ln 2 - 2$$

is a 2-self-concordant barrier for $(\mathcal{C}_\theta^n)^*$. By using [19, Theorem 4.2.2], we complete the proof. □

Suppose that F_θ is a 2-self-concordant barrier of the circular cone. It is clear that the Hessian of F_θ is a positive definite matrices. Using the Hessian of F_θ for any $x \in (\mathcal{C}_\theta^n)^\circ$, we can define the following local norms on $\mathcal{C}_\theta^n$ and $(\mathcal{C}_\theta^n)^*$:

$$\|h\|_x = \sqrt{h^T(\nabla^2 F_\theta(x))h}, \ \ for \ \ h \in \mathcal{C}_\theta^n, \tag{3.6}$$

$$\|s\|_x^* = \sqrt{s^T(\nabla^2 {F_\theta}^{-1}(x))s}, \ \ for \ \ s \in (\mathcal{C}_\theta^n)^*. \tag{3.7}$$

Other properties of F_θ is refer to [17, 18].

4. Self-concordant barrier and central path

In this section, we define central path in terms of $F_\mathcal{K}(x)$. We assume that the problem (1.5) is strictly feasible, i.e., there exists x_0 such that $Ax_0 = b$ and $f_i(x_0) < 0$ for $i = 1, \ldots, N$. This means that Slater's constraint qualification holds, so there exits dual optimal $\lambda^* = (\lambda_1^*, \lambda_1^*, \ldots, \lambda_N^*) \in \mathcal{R}^N$, $v^* \in \mathcal{R}^m$, which together with optimal solution x^* satisfy the KKT conditions

$$\begin{aligned} Ax^* = b, \ \ f_i(x^*) \le 0, \ i = 1, 2, \ldots, N \\ \lambda_i^* \ge 0, \ i = 1, 2, \ldots, N \\ c + \sum_{i=1}^N \lambda_i^* \nabla f_i(x^*) + A^T v^* = 0, \\ \lambda_i^* f_i(x^*) = 0, \ i = 1, 2, \ldots, N. \end{aligned} \tag{4.1}$$

We refer to a pair $(\lambda^*, \ v^*)$ with $\lambda_i^* \ge 0, \ i = 1, 2, \ldots, N$ and $c + \sum_{i=1}^N \lambda_i^* \nabla f_i(x^*) + A^T v^* = 0$ as dual feasible.

Based on $F_\mathcal{K}(x)$ and the basic idea of barrier function, a suitable equality constrained problem is given by $t > 0$:

$$\min_x \{f_t(x) = t\langle c, x\rangle + F_\mathcal{K}(x) : Ax = b\}, \tag{4.2}$$

which is a penalty problem as nonsymmetric circular cone programming (1.5). The problem (4.2) is an equality constrained problem to which Newton's method can be applied. By using strongly convex of $t\langle c, x\rangle + F_\mathcal{K}(x)$, the problem (4.2) has a unique solution for each $t > 0$.

For any $t > 0$, we define $x(t)$ as the solution of (4.2). The central path associated with problem (1.3) is defined as the set of points $x(t)$ for $t > 0$. The points on the central path are characterized by the following necessary and sufficient conditions:

$$\begin{cases} tc + \nabla F_\mathcal{K}(x(t)) + A^T v(t) = 0 \\ Ax(t) = b, x(t) \in \mathcal{K}^\circ. \end{cases} \tag{4.3}$$

By (4.3), we have

$$
\begin{aligned}
0 &= tc + \nabla F_{\mathcal{K}}(x(t)) + A^T v \\
&= tc + \sum_{i=1}^{N} \frac{1}{-f_i(x(t))} \nabla f_i(x(t)) + A^T v.
\end{aligned} \tag{4.4}
$$

From (4.4), we can yield a dual feasible point

$$
\lambda_i(t) = -\frac{1}{-t f_i(x(t))},\ i = 1, 2, \ldots, N,\ v(t) = \frac{v}{t}. \tag{4.5}
$$

It is clear that $\lambda_i(t) > 0$ because $f_i(x(t)) < 0,\ i = 1, 2, \ldots, N$. Moreover, by (4.4), we have

$$
c + \sum_{i=1}^{N} \lambda_i(t) \nabla f_i(x(t)) + A^T v(t) = 0.
$$

We see that $x(t)$ minimizes the Lagrangian

$$
L(x, \lambda, v) = c^T x + \sum_{i=1}^{N} \lambda_i f_i(x) + v^T (b - Ax),
$$

for $\lambda = \lambda(t) = (\lambda_1(t), \lambda_2(t), \ldots, \lambda_N(t)),\ v = v(t)$. Therefore the dual function

$$
g\left(\lambda(t), v(t)\right) = \inf_z L\left(z, \lambda(t), v(t)\right)
$$

is finite, and

$$
\begin{aligned}
g\left(\lambda(t), v(t)\right) &= c^T x(t) + \sum_{i=1}^{N} \lambda_i(t) f_i(x(t)) + v(t)^T (b - Ax(t)) \\
&= c^T x(t) - \frac{N}{t}.
\end{aligned}
$$

In particular, the duality gap associated with $x(t)$ and the dual feasible point $\lambda(t),\ v(t)$ is $\frac{N}{t}$. As an important consequence, we have

$$
c^T x(t) - c^T x^* \leq \frac{N}{t}. \tag{4.6}
$$

By the above inequality, it implies that $x(t)$ converges to optimal point x^* as $t \to \infty$.

5. Interior point algorithm for nonsymmetric circular cone programming

In this section, we discuss the search direction from Newton-type system of (4.3). Then, based on the search direction, we describe the scheme of our algorithm. Moreover, the iteration bound of the algorithm is computed.

5.1. **The search direction.** We will go along Newton direction towards the minimizer of (4.2). To compute the Newton direction, we use the gradient and the Hessian of the objective function $f_t(x)$, which is given by

$$\nabla f_t(x) = tc + \nabla F_{\mathcal{K}}(x), \quad \nabla^2 f_t(x) = \nabla^2 F_{\mathcal{K}}(x).$$

The Newton direction $\triangle x(t)$ is then defined as the direction of linear system

$$\begin{bmatrix} \nabla^2 F_{\mathcal{K}}(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x(t) \\ \Delta v(t) \end{bmatrix} = \begin{bmatrix} -(tc + \nabla F_{\mathcal{K}}(x)) \\ 0 \end{bmatrix} \tag{5.1}$$

By (5.1), we have $\Delta x(t) = -(\nabla^2 f_t(x))^{-1}(\nabla f_t(x) + A^T \Delta v(t))$. We denote the Newton decrement for (4.2) at the point x by

$$\delta_{x,t} = \|\Delta x(t)\|_x = \sqrt{\Delta x(t)^T \nabla^2 f_t(x) \Delta x(t)}.$$

Obviously, we have $\delta_{x,t} = \|\nabla f_t(x) + A^T \Delta v(t)\|_x^*$.

To complete the following Theorem 5.2, we need the following technical Lemma.

Lemma 5.1 ([8, Theorem 2.3.4]). *Let $F(x)$ be a self-concordant function, $A \in \mathcal{R}^{m\times n}$, $b \in \mathcal{R}^m$ and $x \in domF$ such that $Ax = b$ and $\delta_x < 1$. Then*

$$\omega(\delta_x) \le F(x) - F(x^*) \le \omega_*(\delta_x), \tag{5.2}$$

$$\omega'(\delta_x) \le \|x - x^*\|_x \le {\omega_*}'(\delta_x), \tag{5.3}$$

where x^* denotes an optimal solution for the following problem

$$\min_x \{F(x) : Ax = b\}, \tag{5.4}$$

δ_x denotes Newton decrement for (5.4) at the point x, $\omega(t) = t - \ln(1+t)$, $t > -1$ and $\omega_*(t) = -t - \ln(1-t)$, $t < 1$.

Theorem 5.2. *For any $t > 0$ and $\delta_{x,t} \le \beta < 1$, then*

$$c^T(x - x^*) \le \frac{1}{t}\kappa(\beta, N).$$

where $\kappa(\beta, N) = \frac{(\beta+\sqrt{2N})\beta}{1-\beta} + N$.

Proof. By using $tc = \nabla f_t(x) - \nabla F_{\mathcal{K}}(x)$, we have

$$\begin{aligned} tc^T(x - x(t)) &= (\nabla f_t(x) - \nabla F_{\mathcal{K}}(x))^T(x - x(t)) \\ &= (\nabla f_t(x) + A^T v(t) - \nabla F_{\mathcal{K}}(x) - A^T v(t))^T(x - x(t)) \\ &= (\nabla f_t(x) + A^T v(t) - \nabla F_{\mathcal{K}}(x))^T(x - x(t)) - v(t)^T A(x - x(t)) \\ &\le (\|\nabla f_t(x) + A^T v(t)\|_x^* + \|\nabla F_{\mathcal{K}}(x)\|_x^*) \cdot \|x - x(t)\|_x \\ &\le (\delta_{x,t} + \sqrt{2N}) \cdot \omega_*'(\delta_{x,t}) \\ &\le (\delta_{x,t} + \sqrt{2N}) \frac{\delta_{x,t}}{1 - \delta_{x,t}} \\ &\le \frac{(\beta + \sqrt{2N})\beta}{1-\beta}, \end{aligned}$$

We use Lemma 5.1 on the above second inequality. From (4.6) and the above inequality, we immediately yield the desired result. □

5.2. **Interior point algorithm.** In this subsection, we describe our algorithm. First, we denote by $\Delta x^{i}_{t_k}$ the Newton direction at the point $x^{(i)}$ towards the target point $x(t_k)$ on the central path. Furthermore, $\delta_{t_k}(x^{(i)}) = \|\Delta x^{(i)}_{t_k}\|_{x^{(i)}}$ is the Newton decrement of $\Delta x^{(i)}_{t_k}$ with respect to the current iterate $x^{(i)}$. The algorithm is as follows.

Interior point algorithm for nonsymmetric circular cone programming

Input: $A \in \mathcal{R}^{m\times n}$ with full row rank, $b \in \mathcal{R}^m$, $c \in \mathcal{R}^n$, θ_i for $i = 1, 2, \dots, N$ and $F_{\mathcal{K}}(x)$.
Parameter: Choose $\epsilon > 0$, $0 < \beta \le \frac{1}{2}$, $\mu > 1$ and define $\kappa(\beta, N) = \frac{(\beta+\sqrt{2N})\beta}{1-\beta} + N$.
Initialize: $k = 0$, $i = 0$, $t_0 > 0$ and x_0 satisfies $Ax_0 = b$, $x_0 \in \mathcal{K}$ and $\delta_{t_0}(x^{(0)}) \le \beta$.
while $\epsilon \cdot t_k < \kappa(\beta, N)$ **do**
1) compute Newton direction $\Delta x^{(i)}_{t_k}$ from (5.1)
2) compute Newton decrement $\delta_{t_k}(x^{(i)}) = \|\Delta x^{(i)}_{t_k}\|_{x^{(i)}}$
while $\delta_{t_k}(x^{(i)}) > \beta$ **do**
a) $x^{(i+1)} = x^{(i)} + \frac{1}{1+\delta_{t_k}(x^{(i)})} \cdot \Delta x^{(i)}_{t_k}$
b) $i = i + 1$
c) compute Newton direction $\Delta x^{(i)}_{t_k}$ from (5.1)
d) compute Newton decrement $\delta_{t_k}(x^{(i)}) = \|\Delta x^{(i)}_{t_k}\|_{x^{(i)}}$
end while
3) update $t_{k+1} = \mu \cdot t_k$
4) $k = k + 1$
end while

FIGURE 3. Interior point algorithm for nonsymmetric circular cone programming.

5.3. **The iteration bound.** In this subsection, we analyze the iteration bound for algorithm 5.2. To proceed, two technical Lemmas are needed.

Lemma 5.3 ([8, Theorem 2.3.6])**.** *Let $F(x)$ be a self-concordant function, $A \in \mathcal{R}^{m\times n}$, $b \in \mathcal{R}^m$ and $x \in domF$ such that $Ax = b$ and we define the new iterate*

$$x^+ = x + \frac{1}{1+\delta_x} \cdot \Delta x.$$

Then $x^+ \in domF$ and $Ax^+ = b$. Moreover, we have

$$F(x^+) \le F(x) - \omega(\delta_x),$$

where Δx denote the Newton direction for (5.4) *at the point x and δ_x denotes Newton decrement.*

By using the above Lemma 5.3, an upper bound on the functional difference is given in the process of inner iterations. From the upper bound, we can yeild an upper bound on the number of iterations from x to $x(\mu t)$

Lemma 5.4. *Let $x \in \mathcal{K}$, $Ax = b$ and $\delta_{x,t} \le \beta$. If we update t to μt and impose additionally that $\beta \le \frac{1}{2}$, then we have the following bound on the functional difference:*

$$f_{\mu t}(x) - f_{\mu t}(x(\mu t)) \le \mu(N + 2(\sqrt{2N} + 1)). \tag{5.5}$$

Proof. If we denote $\rho = f_{\mu t}(x) - f_{\mu t}(x(\mu t))$, $\rho_1 = f_{\mu t}(x) - f_{\mu t}(x(t))$ and $\rho_2 = f_{\mu t}(x(t)) - f_{\mu t}(x(\mu t))$, then

$$\rho = \rho_1 + \rho_2.$$

The upper bound on ρ is obtained by adding upper bounds for ρ_1 and ρ_2.

First, we drive an upper bound on ρ_1. By convexity of $F_{\mathcal{K}}(x)$ on x, one has

$$\begin{aligned}
\rho_1 &= f_{\mu t}(x) - f_{\mu t}(x(t)) \\
&= \mu t c^T x + F_{\mathcal{K}}(x) - \mu t c^T x(t) - F_{\mathcal{K}}(x(t)) \\
&= \mu t c^T (x - x(t)) + (F_{\mathcal{K}}(x) - F_{\mathcal{K}}(x(t))) \\
&\le \mu t c^T (x - x(t)) + \langle \nabla F_{\mathcal{K}}(x), x - x(t) \rangle \\
&\le \mu t c^T (x - x(t)) + \|\nabla F_{\mathcal{K}}(x)\|_x^* \cdot \|x - x(t)\|_x \\
&\le \mu \frac{(\beta + \sqrt{2N})\beta}{1-\beta} + \sqrt{2N} \cdot \frac{\delta_{x,t}}{1 - \delta_{x,t}} \\
&\le \mu \frac{(\beta + \sqrt{2N})\beta}{1-\beta} + \sqrt{2N} \cdot \frac{\beta}{1 - \beta} \\
&\le (\mu + 1)(\sqrt{2N} + 1) \\
&\le 2\mu(\sqrt{2N} + 1).
\end{aligned}$$

Then, we drive an upper bound on ρ_2 as follows:

$$\begin{aligned}
\rho_2 &= f_{\mu t}\left(x(t)\right) - f_{\mu t}\left(x(\mu t)\right) \\
&= \mu t c^T x(t) + F_{\mathcal{K}}\left(x(t)\right) - \mu t c^T x(\mu t) - F_{\mathcal{K}}\left(x(\mu t)\right) \\
&= \mu t c^T \left(x(t) - x(\mu t)\right) + \sum_{i=1}^{N} \Big[\ln \Big(-\mu t \lambda_i(t) f_i\left(x(\mu t)\right) \Big) \\
&\quad - \ln \Big(-\mu t \lambda_i(t) f_i\left(x(t)\right) \Big) \Big] \\
&= \mu t c^T (x(t) - x(\mu t)) + \sum_{i=1}^{N} \ln \Big(-\mu t \lambda_i(t) f_i(x(\mu t)) \Big) - N \ln \mu \\
&\le \mu t c^T (x(t) - x(\mu t)) - \mu t \sum_{i=1}^{N} \lambda_i(t) f_i(x(\mu t)) - N - N \ln \mu \\
&= \mu t c^T x(t) - \mu t \Big[c^T x(\mu t) + \sum_{i=1}^{N} \lambda_i(t) f_i(x(\mu t))
\end{aligned}$$

$$
\begin{aligned}
&+ v(t)^T(b - Ax(\mu t))\big] - N - N\ln\mu \\
&\le \mu t c^T x(t) - \mu t g(\lambda(t), v(t)) - N - N\ln\mu \\
&= N(\mu - 1 - \ln\mu) \\
&\le N\mu.
\end{aligned}
$$

To obtain the fourth equality from the third, we use $\lambda_i(t) = -\dfrac{1}{tf_i(t)}$. In the first inequality we use the fact that $\ln y \le y - 1$ for $y > 0$. To obtain fifth equality from the first inequality, we use $Ax(\mu t) = b$. The second inequality follows from the definition of dual function:

$$
\begin{aligned}
g(\lambda(t), v(t)) &= \inf_z \left(c^T z + \sum_{i=1}^N \lambda_i(t) f_i(z) + v(t)^T(b - Az) \right) \\
&\le c^T x(\mu t) + \sum_{i=1}^N \lambda_i(t) f_i(x(\mu t)) + v(t)^T(b - Ax(\mu t)).
\end{aligned}
$$

The last equality follows from $g(\lambda(t), v(t)) = c^T x(t) - N/t$.

In a word, we have $\rho \le \mu(N + 2(\sqrt{2N} + 1))$. □

Next, we state our main result as follows.

Theorem 5.5. *Let $\beta \le \frac{1}{2}$. Then, the algorithm 5.2 terminates after at most $k \le O(N \ln \dfrac{N}{t_0\epsilon})$ iterations with a point x_k such that*

$$\langle c, x_k - x^* \rangle \le \epsilon.$$

Proof. According to Theorem 5.2, we have

$$c^T(x - x^*) \le \frac{1}{t}\kappa(\beta, N).$$

If we desire $c^T(x - x^*) \le \epsilon$, then this is guaranteed by finding a point x such that $\delta_t(x) \le \beta$ and $\frac{1}{t}\kappa(\beta, N) \le \epsilon$. The latter condition is satisfied if $\frac{1}{\mu^{k_1} t_0}\kappa(\beta, N) \le \epsilon$, where k_1 is the number of outer iterations. Then, k_1 is no more than

$$\frac{\ln\kappa(\beta, N) - \ln(t_0\epsilon)}{\ln\mu} = O\Big(\ln\frac{N}{t_0\epsilon}\Big).$$

By Lemma 5.3, we have

$$\omega(\delta_x) \le F(x) - F(x^+).$$

As long as $\delta_x > \beta$, we can reduce the function $f_{\mu t}(x)$ at least $\omega(\beta)$ by $\omega(\delta_x) > \omega(\beta)$. Using Lemma 5.4, that means the optimality gap $f_{\mu t}(x) - f_{\mu t}(x(\mu t))$ will be reduced at most

$$\frac{f_{\mu t}(x) - f_{\mu t}(x(\mu t))}{\omega(\beta)} \le \frac{\mu(N + 2(\sqrt{2N} + 1))}{\omega(\beta)} = O(N)$$

times before $\delta_x \le \beta$. We complete the proof. □

6. Numerical results

In this section, we give some numerical examples to illustrate the performance of the proposed algorithm for solving nonsymmetric circular cone programming problems described in Section 5.

Numerical examples are generated randomly and the number of constraints is set as half of dimension for its corresponding problem. For these circular cones, we choose the rotation angles as $\theta = \frac{\pi}{12}, \frac{\pi}{6}, \frac{\pi}{4}, \frac{\pi}{3}, \frac{5\pi}{12}$, respectively. The test problems are divided into three categories for every fixed rotation angle. In the first group, the problems' dimension is between 10 and 90, and the problems' dimension is between 100 and 900 in the second group, whereas it is between 1000 and 2000 in the third group.

The numerical experiments are implemented by using MATLAB R2008b, and on a PC with Intel 2.20 GHz CPU, 2 GB RAM. The parameters were selected as: $\beta = 0.2925$ and $\epsilon = 1 \times 10^{-5}$.

The numerical results are listed in Table 2, Table 3 and Table 4. The following notations are used: iter, number of iterations; CPU(s), CPU time (in seconds).

Table 2. The numerical results on circular cone programming (small size)

θ	$\frac{\pi}{12}$		$\frac{\pi}{6}$		$\frac{\pi}{4}$		$\frac{\pi}{3}$		$\frac{5\pi}{12}$	
(n, m)	iter	CPU(s)	iter	CPU(s)	iter	CPU(s)	iter	CPU(s)	iter	CPU(s)
(10,5)	28	0.085950	23	0.007586	24	0.064620	23	0.009108	23	0.008762
(20,10)	28	0.014009	25	0.027221	25	0.029289	24	0.116081	23	0.127255
(30,15)	29	0.018051	26	0.036856	25	0.103724	24	0.015807	24	0.084869
(40,20)	29	0.089729	27	0.145478	26	0.048549	25	0.074739	24	0.021951
(50,25)	30	0.117831	27	0.133267	26	0.076434	26	0.072427	26	0.034333
(60,30)	30	0.071203	27	0.102295	26	0.101482	26	0.045275	25	0.099405
(70,35)	30	0.139458	28	0.064778	27	0.122284	26	0.117338	26	0.114639
(80,40)	31	0.277781	29	0.157181	27	0.101396	26	0.120880	26	0.151339
(90,45)	30	0.110020	29	0.264767	27	0.123550	26	0.097459	26	0.185536

Table 3. The numerical results on circular cone programming (medium size)

θ	$\frac{\pi}{12}$		$\frac{\pi}{6}$		$\frac{\pi}{4}$		$\frac{\pi}{3}$		$\frac{5\pi}{12}$	
(n, m)	iter	CPU(s)	iter	CPU(s)	iter	CPU(s)	iter	CPU(s)	iter	CPU(s)
(100,50)	31	0.191587	29	0.122548	27	0.150536	26	0.112776	26	0.119702
(200,100)	32	0.718956	30	0.699293	28	0.661747	28	0.876931	27	0.556061
(300,150)	33	2.237525	30	1.822652	29	1.553632	29	3.204657	28	1.749023
(400,200)	33	4.153084	31	3.754879	30	3.025065	29	3.204657	29	3.734951
(500,250)	33	6.700060	31	6.510338	30	5.288580	29	6.166231	29	5.865587
(600,300)	34	10.678081	32	10.461737	30	8.000719	29	9.698626	29	8.716132
(700,350)	34	15.733862	32	14.290476	30	11.815231	29	13.848642	29	13.128566
(800,400)	35	20.898460	32	20.421453	30	16.041597	30	19.409672	29	17.971985
(900,450)	35	29.560119	32	27.392799	31	23.314514	30	25.578036	30	24.746241

From Table 2, Table 3 and Table 4, we see that the iterative number of our algorithm ranges from 25 to 40. In particular, when the rotation angle is getting

Table 4. The numerical results on circular cone programming (large size)

θ	$\frac{\pi}{12}$		$\frac{\pi}{6}$		$\frac{\pi}{4}$		$\frac{\pi}{3}$		$\frac{5\pi}{12}$	
(n, m)	iter	CPU(s)	iter	CPU(s)	iter	CPU(s)	iter	CPU(s)	iter	CPU(s)
(1000,500)	35	37.090044	32	35.099251	31	27.448155	30	32.902582	30	33.633529
(1100,550)	35	44.537521	33	45.148683	31	35.335342	30	43.566852	30	42.799847
(1200,600)	34	54.978862	33	56.331029	31	43.534430	31	52.749817	30	51.741102
(1300,650)	35	62.850570	33	70.953161	32	53.073101	30	64.660031	31	61.763158
(1400,700)	36	82.304524	33	81.420084	32	64.888901	31	77.799324	31	75.506688
(1500,750)	35	94.100085	33	91.887821	32	77.277800	31	92.013727	31	88.459140
(1600,800)	36	111.545756	34	112.266897	31	86.991472	31	107.168125	31	104.693379
(1700,850)	36	128.554900	33	136.690869	32	102.955529	31	122.958628	31	123.637996
(1800,900)	35	141.416163	34	156.262517	31	116.271943	31	138.461658	31	136.811174
(1900,950)	36	166.054804	33	178.260190	32	139.689364	31	159.19593	31	154.884698
(2000,1000)	36	189.208820	33	198.598245	32	163.310439	32	202.917734	31	182.404055

smaller, the iterations become larger. The computing time for $\theta = \frac{\pi}{4}$ is always less, no matter what size the problem is. Another phenomenon is that, with the increase of the dimension, the iterations of our algorithm become more and the computing time gets longer.

7. Conclusions

In this paper, based on the algebraic relationship between the second cone and the circular cone, we introduce a logarithmically homogeneous self-concordant barrier functions for circular cone and its dual cone. By using logarithmically homogeneous self-concordant barrier function of circular cone, we investigate an interior point algorithm to solve nonsymmetric circular cone programming and derive the iteration bound. Finally, The numerical results show the effectiveness of the proposed algorithm.

At last, we point out that the proposed algorithm may be extended as infeasible initial point for nonsymmetric circular cone programming. Moreover, we can explore how to solve large scale problems. We leave them as our future research work.

References

[1] F. Alizadeh and D. Goldfarb, *Second-order cone programming*, Math. Program. **95** (2003), 3–31.

[2] Y. Q. Bai, *Kernel Function-based Interior-point Algorithms for Conic Optimization*, Science Press, Beijing, China, 2010.

[3] Y. Q. Bai, P. F. Ma and J. Zhang, *A polynomial-time interior-point method for circular cone programming based on kernel functions*, J. Indust. Manage. Optim. **12** (2016), 739–756.

[4] Y. Q. Bai, X. R. Gao and G. Q. Wang, *Primal-dual interior-point algorithms for convex quadratic circular cone optimization*, Numer. Alge. Control Optim. **5** (2015), 211–231.

[5] A. Ben-Tal and A.S. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms and Engineering Applications,* SIAM, Philadelphia, 2001.

[6] P. H. Borgstrom, M. A. Batalin, G. Sukhatme, et al. *Weighted barrier functions for computation of force distributions with friction cone constraints*, in:

Robotics and Automation (ICRA), 2010 IEEE International Conference on, IEEE, 2010, pp. 785–792.

[7] S. Boyd and B. Wegbreit, *Fast computation of optimal contact forces*, IEEE Transactions on Robotics **23** (2007), 1117–1132.

[8] R. Chares, *Cones and Interior-Point Algorithms for Structured Convex Optimization involving Powers and Exponentials,* PhD thesis, Université Catholique de Louvain, Louvain-la-Neuve, 2008.

[9] Y.-L. Chang, C.-Y. Yang and J.-S. Chen, *Smooth and nonsmooth analyses of vector-valued functions associated with circular cones*, Nonlinear Anal. Theory, Methods Appl. **85** (2013), 160–173.

[10] J. Dattorro, *Convex Optimization and Euclidean Distance Geometry,* Meboo Publishing, California, 2005.

[11] C.-H. Ko and J.-S. Chen, *Optimal grasping manipulation for multifingered robots using semismooth Newton method,* Math. Problems in Engineer. **2013**, Article ID 681710, 9 pages,(2013) to appear.

[12] B. León, A. Morales and J. Sancho-Bru, *Robot grasping foundations*, in: From Robot to Human Grasping Simulation, Springer International Publishing, Switzerland, 2014, pp. 15–31.

[13] M. S. Lobo, L. Vandenberghe, S. E. Boyd and H. Lebret, *Applications of second-order cone programming*, Linear Algebra and its Applications **284** (1998), 193–228.

[14] Y. Matsukawa and A. Yoshise, *A primal barrier function phase I algorithm for nonsymmetric conic optimization problems*, Japan J. Industrial Applied Math. **29** (2012), 499–517.

[15] Y. Nesterov and A. Nemirovskii, *Interior Point Polynomial Time Methods in Convex Programming,* SIAM, Philadelphia, 1994.

[16] Y. Nesterov, *Long-step strategies in interior-point primal-dual methods,* Math. Program. **76** (1996), 47–94.

[17] Y. Nesterov and M. J. Todd, *Self-scaled barriers and interior-point methods for convex programming,* Math. Operation Research **22** (1997), 1–42.

[18] Y. Nesterov and M. J. Todd, Primal-dual interior-point methods for self-scaled cones, SIAM J. Optimization **22** (1998), 324–364.

[19] Y. Nesterov, *Introductory Lectures on Convex Optimization,* Kluwer, Boston, 2004.

[20] Y. Nesterov, *Constructing self-concordant barriers for convex cones,* CORE Discussion Paper, 2006.

[21] Y. Nesterov, *Towards nonsymmetric conic optimization,* Optim. Methods Software **27** (2012), 893–917.

[22] A. Nemirovskii, Advanced in Convex Optimization: Conic Optimization, *A plenary speak in ICM2006*, 2006.

[23] A. Skajaa and Y. Ye, *Homogeneous interior-point algorithm for nonsymmetric convex conic optimization*, Math. Program., 2014 to appear.

[24] J. Zhou, J.-S. Chen and H.-F. Hung, *Circular cone convexity and some inequalities associated with circular cones*, J. Inequalities Appl. **2013** (2013), 1–17.

[25] J. C. Zhou and J.-S. Chen, *Properties of circular cone and spectral factorization associated with circular cone,* J. Nonlinear Convex Anal. **14** (2013), 807–816.

[26] J. C. Zhou and, J.-S. Chen, *On the vector-valued functions associated with circular cones*, Abstract and Applied Analysis, Article ID 603542, 2014.

[27] J. C. Zhou, J.-S. Chen, B. S. Mordukhovich, *Variational analysis of circular cone programs*, Optimization **64** (2015), 113–147.

Manuscript received March 14, 2015
revised April 10, 2015

Pengfei Ma
Department of Mathematics, Shanghai University, Shanghai, 200444, China
E-mail address: mathpengfeima@126.com

Y. Q. Bai
Department of Mathematics, Shanghai University, Shanghai, 200444, China
E-mail address: yqbai@shu.edu.cn

Jein-Shan Chen
Department of Mathematics, National Taiwan Normal University, Taipei, 11677, Taiwan
E-mail address: jschen@math.ntnu.edu.tw