

A Cone Constrained Convex Program: Structure and Algorithms

**Liqun Qi, Yi Xu, Ya-Xiang Yuan &
Xinzhen Zhang**

**Journal of the Operations Research
Society of China**

ISSN 2194-668X
Volume 1
Number 1

J. Oper. Res. Soc. China (2013) 1:37-53
DOI 10.1007/s40305-013-0005-z



Your article is protected by copyright and all rights are held exclusively by Operations Research Society of China, Periodicals Agency of Shanghai University, and Springer-Verlag Berlin Heidelberg. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your work, please use the accepted author's version for posting to your own website or your institution's repository. You may further deposit the accepted author's version on a funder's repository at a funder's request, provided it is not made publicly available until 12 months after publication.

A Cone Constrained Convex Program: Structure and Algorithms

Liquan Qi · Yi Xu · Ya-Xiang Yuan · Xinzhen Zhang

Received: 25 October 2012 / Revised: 25 February 2013 / Accepted: 26 February 2013 /
Published online: 14 March 2013
© Operations Research Society of China, Periodicals Agency of Shanghai University, and
Springer-Verlag Berlin Heidelberg 2013

Abstract In this paper, we consider the positive semi-definite space tensor cone constrained convex program, its structure and algorithms. We study defining functions, defining sequences and polyhedral outer approximations for this positive semi-definite space tensor cone, give an error bound for the polyhedral outer approximation approach, and thus establish convergence of three polyhedral outer approximation algorithms for solving this problem. We then study some other approaches for solving this structured convex program. These include the conic linear programming approach, the nonsmooth convex program approach and the bi-level program approach. Some numerical examples are presented.

L. Qi was supported by the Hong Kong Research Grant Council (Grant Nos. PolyU 501909, 502510, 502111 and 501212).

Y.-X. Yuan was supported by National Natural Science Foundation of China (Grant Nos. 10831006 and 11021101) and by CAS grant (No. kjc-x-yw-s7).

X. Zhang was supported by the National Natural Science Foundation of China (Grant Nos. 11101303 and 11171180).

L. Qi (✉) · Y. Xu

Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China

e-mail: maqilq@polyu.edu.hk

Y. Xu

e-mail: yi.xu1983@gmail.com

Y.-X. Yuan

State Key Laboratory of Scientific/Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, AMSS, Chinese Academy of Sciences, Zhong Guan Cun Donglu 55, Beijing, 100190, P.R. China

e-mail: yyx@lsec.cc.ac.cn

X. Zhang

Department of Mathematics, School of Science, Tianjin University, Tianjin, 300072, China

e-mail: xzzhang@tju.edu.cn

Keywords Convex program · Space tensor · Positive semi-definiteness · Cone · Algorithms

1 Introduction

Let m be a positive even integer. A symmetric space tensor of order m is a physical entity in the three-dimensional space. For example, a higher-order diffusion tensor in higher order diffusion magnetic resonance imaging (MRI) model is such a symmetric tensor [13]. Let $n = \frac{1}{2}(m + 1)(m + 2)$. According to [16], a symmetric space tensor \mathcal{A} of order m under a rectangular co-ordinate system can be represented by an n -dimensional vector $x = (x_{ij}) \in \mathfrak{R}^n$, where $i = 0, \dots, m, j = 0, \dots, m - i$. Then for $y \in \mathfrak{R}^3$, we have

$$\mathcal{A}y^m = x(y) = \sum_{i=0}^m \sum_{j=0}^{m-i} x_{ij} y_1^i y_2^j y_3^{m-i-j}. \tag{1}$$

In the MRI models, the value of $\mathcal{A}y^m = x(y)$ is related with the apparent diffusion coefficient of the tensor \mathcal{A} at direction y . This value should be nonnegative physically. The tensor \mathcal{A} and the vector $x \in \mathfrak{R}^n$ are called positive semi-definite if for all $y \in \mathfrak{R}^3$, $\mathcal{A}y^m = x(y) \geq 0$. The tensor \mathcal{A} and the vector $x \in \mathfrak{R}^n$ are called positive definite if for all $y \in \mathfrak{R}^3$ and $y \neq 0$, $\mathcal{A}y^m = x(y) > 0$. In the MRI research, the vector x and the tensor \mathcal{A} are calculated by the least squares method. Let \bar{x} be the solution of the least squares problem. Because of noise, it is not guaranteed that the vector \bar{x} and hence the tensor \mathcal{A} , obtained by the least squares method, are positive semi-definite. MRI researchers have made efforts to correct this noise effect. Some approaches had been proposed to preserve positive semi-definiteness of a diffusion tensor of second order or fourth order [1, 2, 5, 7]. None of them can work for arbitrary high order diffusion tensors. In [17], a comprehensive model, called PSDT (positive semi-definite tensor), was proposed to approximate the diffusivity function by a positive semi-definite diffusion tensor of either second or higher order. A nonnegative diffusion orientation distribution function model was further proposed in [18].

Denote the set of all positive semi-definite vectors in \mathfrak{R}^n as $\mathcal{S} = \mathcal{S}(m) = \{x \in \mathfrak{R}^n : x(y) \geq 0, \forall y \in \mathfrak{R}^3\}$. It was proved in [17] that \mathcal{S} is a closed convex cone in \mathfrak{R}^n . The PSDT minimization problem was proposed there:

$$\min\{P(x) \equiv (x - \bar{x})^\top Q(x - \bar{x}) : x \in \mathcal{S}\}, \tag{2}$$

where \bar{x} is the solution of the original least squares problem mentioned above, Q is an $n \times n$ positive semi-definite matrix.

In this paper, we study algorithms for solving the following structured convex program:

$$\min\{f(x) : g(x) \leq 0, x \in \mathcal{S}\}, \tag{3}$$

where $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$ and $g : \mathfrak{R}^n \rightarrow \mathfrak{R}^p$ are twice continuously differentiable convex functions. Clearly, the PSDT problem (2) is a special case of (3).

In this paper, we use $\|\cdot\|$ to denote the 2-norm.

2 Structure and Polyhedral Outer Approximation Algorithms

2.1 Two Assumptions

We now make two assumptions on problem (3).

Assumption A Problem (3) has an interior point \hat{x} .

This assumption is needed for many algorithms [8, 9]. In the case of (2), it holds naturally, as by [16], any positive definite vector x in \mathfrak{N}^n is an interior point of \mathcal{S} .

Assumption B Problem (3) has an optimal solution x^* in a simple compact region R , such as a ball or a cube.

This assumption is also needed for many algorithms [8, 9]. If this assumption holds, then (3) is equivalent to the bounded convex program

$$\min\{f(x) : g(x) \leq 0, x \in R \cap \mathcal{S}\}. \tag{4}$$

Proposition 1 *Suppose that Assumption A holds and that function f is strongly convex, i.e., there is a positive constant c such that for any $x, \tilde{x} \in \mathfrak{N}^n$, we have*

$$f(x) \geq f(\tilde{x}) + \nabla f(\tilde{x})^\top (x - \tilde{x}) + \frac{c}{2} \|x - \tilde{x}\|^2.$$

Then Assumption B holds with

$$R = \left\{x \in \mathfrak{N}^n : \|x - \hat{x}\| \leq \frac{2}{c} \|\nabla f(\hat{x})\|\right\}.$$

Proof Suppose that $x \in \mathfrak{N}^n$ and $f(x) \leq f(\hat{x})$. Then

$$f(\hat{x}) \geq f(x) \geq f(\hat{x}) + \nabla f(\hat{x})^\top (x - \hat{x}) + \frac{c}{2} \|x - \hat{x}\|^2.$$

This implies that $x \in R$. As R is a compact region, the conclusion follows. □

Note that for problem (2), the objective function is strongly convex if Q is positive definite.

2.2 Defining Functions

Suppose that $\phi : \mathfrak{N}^n \rightarrow \mathfrak{R}$ is a concave function and

$$\mathcal{S} = \{x \in \mathfrak{N}^n : \phi(x) \geq 0\}.$$

Then we call ϕ a defining function of the cone \mathcal{S} and we may rewrite (3) as

$$\min\{f(x) : g(x) \leq 0, \phi(x) \geq 0\}. \tag{5}$$

Let $\bar{\Omega}$ be a compact convex set in \mathfrak{N}^n and the origin is an interior point of $\bar{\Omega}$. Let $\bar{\Omega}$ be the boundary surface of $\bar{\Omega}$. Then

$$\phi_{\bar{\Omega}}(x) = \min\{x(y) : y \in \bar{\Omega}\}$$

and

$$\phi_{\Omega}(x) = \min\{x(y) : y \in \Omega\}$$

are concave functions as they are defined by minimization problems. Whenever x is positive semi-definite, $\phi_{\bar{\Omega}}(x) = 0$. On the other hand, if x is positive definite, then $\phi_{\Omega}(x) > 0$; if x is positive semi-definite but not positive definite, then $\phi_{\Omega}(x) = 0$. Hence, both $\phi_{\bar{\Omega}}$ and ϕ_{Ω} are defining functions of \mathcal{S} , but ϕ_{Ω} presents more information than $\phi_{\bar{\Omega}}$. Furthermore, $\bar{\Omega}$ is three-dimensional, while Ω is two-dimensional. Thus, we only use ϕ_{Ω} in the following discussion. We call Ω a defining surface of \mathcal{S} .

Let $\bar{\Omega} = \bar{B}$ be the unit ball in \mathfrak{R}^3 . Then

$$\phi_B(x) = \min\{x(y) : y_1^2 + y_2^2 + y_3^2 = 1\}$$

is the smallest Z -eigenvalue function discussed in [14–17].

Let $\bar{\Omega} = \bar{C}$ be the unit cube in \mathfrak{R}^3 . Let $C_1 = \{(1, t_1, t_2) : -1 \leq t_1, t_2, \leq 1\}$, $C_2 = \{(t_1, 1, t_2) : -1 \leq t_1, t_2, \leq 1\}$, $C_3 = \{(t_1, t_2, 1) : -1 \leq t_1, t_2, \leq 1\}$ and $D = C_1 \cup C_2 \cup C_3$. Let

$$\phi_D = \min\{x(y) : y \in D\}.$$

Then we see that ϕ_D is also a defining function of \mathcal{S} , though D is only about a half of C , as we have $x(-y) = x(y)$ since m is even. Thus, we also call D a defining surface of \mathcal{S} . Then we have

$$\phi_D(x) = \min\{\phi_k(x) : k = 1, 2, 3\},$$

where for $k = 1, 2, 3$,

$$\phi_k(x) = \min\{d_k(t) : -1 \leq t_1, t_2 \leq 1\}, \tag{6}$$

$t \in \mathfrak{R}^2$, $d_1(t) = x(1, t_1, t_2)$, $d_2(t) = x(t_1, 1, t_2)$ and $d_3(t) = x(t_1, t_2, 1)$. As the minimization problem for defining ϕ_k only involves a two-variable vector t in the unit square, computationally, the defining function ϕ_D may be better than the defining function ϕ_B .

The defining functions ϕ_B and ϕ_D are both nonsmooth. A question is if there exists a smooth defining function or a set of smooth defining functions of \mathcal{S} .

2.3 Polyhedral Outer Approximations

Let $F \subset \mathfrak{R}^3$ be a finite set. Let $y \in F$. By the definition of \mathcal{S} , for any $x \in \mathcal{S}$, we have

$$x(y) = \sum_{i=0}^m \sum_{j=0}^{m-i} x_{ij} y_1^i y_2^j y_3^{m-i-j} \geq 0.$$

Note that this is a linear constraint with respect to x . Let

$$\mathcal{S}(F) = \{x \in \mathfrak{R}^n : x(y) \geq 0, \forall y \in F\}.$$

Then $\mathcal{S}(F)$ is a polyhedral cone and $\mathcal{S} \subset \mathcal{S}(F)$. We call $\mathcal{S}(F)$ an polyhedral out approximation of \mathcal{S} , generated by the finite set F . We may use $\mathcal{S}(F)$ to relax the nonsmooth constraints of (3). We have the following twice continuously differentiable convex program:

$$\min\{f(x) : g(x) \leq 0, x \in \mathcal{S}(F)\}, \tag{7}$$

The twice continuously differentiable convex program (7) is a relaxation of the non-smooth convex program (3). We may solve (7) by a conventional method. Suppose that x_F is a solution of (7). If $x_F \in \mathcal{S}$, then x_F is also an optimal solution of (3). Otherwise, $f(x_F)$ is a lower bound of (3). Furthermore, we have the following error bound result.

Proposition 2 *Suppose that Assumption A holds and ϕ is a defining function of \mathcal{S} . Assume that (7) has an optimal solution x_F . If $\phi(x_F) < 0$, then we have*

$$0 \leq f^* - f(x_F) \leq \sigma [f(\hat{x}) - f(x_F)],$$

where

$$f^* = \inf\{f(x) : g(x) \leq 0, x \in \mathcal{S}\},$$

$$\sigma = \frac{-\phi(x_F)}{\phi(\hat{x}) - \phi(x_F)},$$

and $0 < \sigma < 1$.

Proof Since (7) is a relaxation of (3), we have $0 \leq f^* - f(x_F)$. Since $\hat{x} \in \mathcal{S}$ and $\phi(x_F) < 0$, we have $0 < \sigma < 1$. Then

$$\phi((1 - \sigma)x_F + \sigma\hat{x}) \geq (1 - \sigma)\phi(x_F) + \sigma\phi(\hat{x}) = 0.$$

Thus, $(1 - \sigma)x_F + \sigma\hat{x} \in \mathcal{S}$. Since $g(x_F) \leq 0$ and $g(\hat{x}) \leq 0$, we have $g((1 - \sigma)x_F + \sigma\hat{x}) \leq 0$. This implies that $(1 - \sigma)x_F + \sigma\hat{x}$ is a feasible point of (3). We have $f^* \leq f((1 - \sigma)x_F + \sigma\hat{x})$. Thus, we have

$$\begin{aligned} 0 &\leq f^* - f(x_F) \\ &\leq f((1 - \sigma)x_F + \sigma\hat{x}) - f(x_F) \\ &\leq \sigma [f(\hat{x}) - f(x_F)]. \end{aligned}$$

This proves the proposition. □

This proposition says that if $-\phi(x_F)$ is small, then $f(x_F)$ is close to f^* . In the next theorem, we will show that if F is denser on a compact defining surface Ω , then $f(x_F)$ is closer to f^* . We now need Assumption B. Instead of solving (7), we now solve

$$\min\{f(x) : g(x) \leq 0, x \in R \cap \mathcal{S}(F)\}. \tag{8}$$

Theorem 1 *Suppose that Assumptions A and B hold. Let $F \subset \Omega$, where Ω is a compact defining surface of \mathcal{S} . Let $\phi = \phi_\Omega$ be the defining function associated with Ω . Then (8) has an optimal solution x_F . If $\phi(x_F) \geq 0$, then x_F is an optimal solution of (4) and (3). If $\phi(x_F) < 0$, then we have*

$$-\phi(x_F) \leq M\rho(F)$$

and

$$0 \leq f(x^*) - f(x_F) \leq \bar{\sigma} [f(\hat{x}) - f(x_F)],$$

where x^* is an optimal solution of (4),

$$\bar{\sigma} = \frac{M\rho(F)}{\phi(\hat{x}) + M\rho(F)},$$

$0 < \bar{\sigma} < 1$, M is a constant depending on R and Ω only, and

$$\rho(F) = \max\{\text{dist}(y, F) : y \in \Omega\}.$$

Proof By Assumptions A and B, (8) has an optimal solution. Denote this optimal solution as x_F . If $\phi(x_F) \geq 0$, then the conclusions hold obviously. Suppose that $\phi(x_F) < 0$. Suppose that y_F is an optimal solution of

$$\phi(x_F) = \min\{x_F(y) : y \in \Omega\}.$$

Then $\phi(x_F) = x_F(y_F)$. Let \bar{y} be the closest point in F to y_F . Then $x_F(\bar{y}) \geq 0$ and

$$\|y_F - \bar{y}\| \leq \rho(F).$$

Let M be the Lipschitz constant of the function $L(x, y) \equiv x(y)$ on the compact set $R \times \Omega$. Then

$$-\phi(x_F) = -x_F(y_F) \leq x_F(\bar{y}) - x_F(y_F) \leq M\rho(F).$$

The remaining conclusions now follow from Proposition 2. □

2.4 The Basic Polyhedral Outer Approximation Algorithm

By Theorem 1, if we let $F_k \subset \Omega$ such that $\rho(F_k) \rightarrow 0$ as $k \rightarrow \infty$, and denote $x^{(k)} = x_{F_k}$, where x_{F_k} is an optimal solution of (8) with $F = F_k$, then we have a polyhedral outer approximation algorithm. To distinguish this algorithm from its improved version in next subsection, we call this algorithm the basic polyhedral outer approximation algorithm.

Theorem 2 *Under Assumptions A and B, we have*

$$\lim_{k \rightarrow \infty} f(x^{(k)}) = f^*,$$

where f^* is the optimal objective function value of (4), and any accumulation point of $\{x^{(k)}\}$ is an optimal solution of (4).

The proof of Theorem 2 follows the proof of Theorem 1. We omit it. We now discuss possible schemes to construct F_k . Let $\Omega = D$, the closed half surface of the unit cube. Let F_0 be the set of the seven vertices of D . Let F_k be the set of grid points on D , with the grid length as $\frac{1}{2^{k-1}}$, for $k = 0, 1, \dots$. Then $\rho(F_k) = \frac{\sqrt{2}}{2^k}$. Furthermore, we have $F_k \subset F_{k+1}$ and hence $f(x^{(k)}) \leq f(x^{(k+1)})$ for all k . Denote $S_k = S(F_k)$ for $k = 0, 1, \dots$. Then we have

$$S_0 = \left\{ x \in \mathfrak{R}^n : \sum_{i=0}^m \sum_{j=0}^{m-i} x_{ij} \geq 0, \sum_{i=0}^m \sum_{j=0}^{m-i} x_{ij} (-1)^{i+j} \geq 0, \right. \\ \left. \sum_{i=0}^m \sum_{j=0}^{m-i} x_{ij} (-1)^j \geq 0, \sum_{i=0}^m \sum_{j=0}^{m-i} x_{ij} (-1)^i \geq 0 \right\}$$

and

$$S_1 = \left\{ x \in S_0 : x_{m0} \geq 0, x_{0m} \geq 0, x_{00} \geq 0, \sum_{i=0}^m x_{i,m-i} \geq 0, \sum_{i=0}^m x_{i,m-i}(-1)^i \geq 0, \right. \\ \left. \sum_{i=0}^m x_{i0} \geq 0, \sum_{i=0}^m x_{i0}(-1)^i \geq 0, \sum_{i=0}^m x_{0i} \geq 0, \sum_{i=0}^m x_{0i}(-1)^i \geq 0 \right\}.$$

2.5 Defining Sequences and an Iterative Polyhedral Outer Approximation Algorithm

In the basic polyhedral outer approximation algorithm, the number of constraints in (8) is huge when k is big. This is not practical. Furthermore, the sequence $\{x^{(k)} : k = 0, 1, \dots\}$ is not iterative. The point $x^{(k+1)}$ is obtained without using the knowledge of $x^{(k)}$. We now explore an improved version of that algorithm. We call a sequence in \mathfrak{R}^3 a defining sequence of S if for any $x \in \mathfrak{R}^n$, x is positive definite if and only if $x(y) \geq 0$ for any y in that sequence. We see that the sequence consisting of all points in F_k for all k is such a defining sequence. We now give a general formula for this sequence. As $x(\cdot)$ is an even function, for grid points y and $-y$, we only need to include one of them. Thus, in F_0 , we only need to include four vertices:

$$G_0 = \{(1, 1, 1), (1, 1, -1), (1, -1, 1), (-1, 1, 1)\}.$$

We see that $S_0 = S(G_0)$. Then for any $x \in \mathfrak{R}^n$, x is positive semi-definite if and only if $x(y) \geq 0$ for any $y \in \Gamma \equiv G_0 \cup D_1 \cup D_2 \cup D_3$, where

$$D_1 = \{(1, t_1, t_2) : -1 \leq t_1 \leq 1, -1 < t_2 < 1\}, \\ D_2 = \{(t_2, 1, t_1) : -1 \leq t_1 \leq 1, -1 < t_2 < 1\}, \\ D_3 = \{(t_1, t_2, 1) : -1 \leq t_1 \leq 1, -1 < t_2 < 1\}.$$

We see that G_0, D_1, D_2 and D_3 are disjoint each other, and if $y \in \Gamma$ then $-y \notin \Gamma$. Hence, we may take all the grid points from Γ . Let

$$G_k = (F_k \setminus F_{k-1}) \cap \Gamma$$

for $k = 1, 2, \dots$. Then the set

$$G = G_0 \cup G_1 \cup G_2 \cup \dots$$

is what we want. We may write out the general formula for points in G_k as: $(t_1, t_2, 1), (t_2, 1, t_1), (t_1, t_2, 1)$ for $t_1 = -1 + \frac{i}{2^{k-1}}$ and $t_2 = -1 + \frac{2^j-1}{2^{k-1}}$, for $i = 0, \dots, 2^k$ and $j = 1, \dots, 2^{k-1}$, and $(t_1, t_2, 1), (t_2, 1, t_1), (t_1, t_2, 1)$ for $t_1 = -1 + \frac{2i-1}{2^{k-1}}$ and $t_2 = -1 + \frac{j}{2^{k-2}}$, for $i = 1, \dots, 2^{k-1}$ and $j = 1, \dots, 2^{k-1} - 1$. Then $|G_k| = 9 \times 2^{2k-2}$.

We now give an iterative polyhedral outer approximation algorithm. Suppose that Assumptions A and B hold.

Algorithm 1 (An Iterative Polyhedral Outer Approximation Algorithm)

Step 0 Let $k = 0, \mu_{-1} = 1$ and $F = G_0$. Let N be a positive integer.

Step 1 Let $\mu_k = \mu_{k-1}$. Compute an optimal solution of (8) and denote it as $x^{(k)}$.

Step 2 If $\mu_k \geq N$, stop. Otherwise, let $H_k = \{y \in G_{\mu_k} : x^{(k)}(y) < 0\}$.

Step 3 If $H_k = \emptyset$, let $\mu_k = \mu_k + 1$ and go to Step 2. Otherwise, let $F = H_k \cup F$ and $k = k + 1$. Go to Step 1.

Theorem 3 *Suppose that Assumptions A and B hold. Then we have $f(x^{(k)}) \leq f(x^{(k+1)}) \leq f(x^*)$ for all k . If $N \rightarrow \infty$, then we have a sequence $\{x^{(k)} : k = 0, 1, \dots\} \subset R$. If x^{**} is an accumulation point of $\{x^{(k)}\}$, then x^{**} is an optimal solution of (3).*

Proof At the k th iteration, $x^{(k)}$ is an optimal solution of (8). On the other hand, $x^{(k+1)}$ is an optimal solution of

$$\min\{f(x) : g(x) \leq 0, x \in R \cap \mathcal{S}(H_k \cup F)\}, \tag{9}$$

which is a restriction of (8). Hence, $f(x^{(k)}) \leq f(x^{(k+1)})$. As (8) and (9) are relaxations of (4), we have $f(x^{(k)}) \leq f(x^{(k+1)}) \leq f(x^*)$ for all k . This proves the first conclusion.

Suppose that $N \rightarrow \infty$ and $x^{(k_i)} \rightarrow x^{**}$, where $\{k_i : i = 0, 1, \dots\}$ is a subsequence of $\{0, 1, \dots\}$. Assume that $\phi(x^{**}) = -\epsilon < 0$. Then there is $y^{**} \in D$ such that $x^{**}(y^{**}) = \phi(x^{**}) = -\epsilon < 0$. Because of the structure of G_μ , there is $K_1 > 0$ and $\delta > 0$ such that for all $\mu \geq K_1$, there is a $y_\mu \in G_\mu$ such that $y_\mu \in N(y^{**}; \delta) \equiv \{y \in D : \|y - y^{**}\| \leq \delta\}$ and $x^{**}(y_\mu) \leq -\frac{2}{3}\epsilon$. Since $x^{(k_i)} \rightarrow x^{**}$, there is a $K_2 \geq K_1$ such that for all $k_i \geq K_2$ and $y \in N(y^{**}; \delta)$,

$$|x^{(k_i)}(y) - x^{**}(y)| \leq \frac{1}{3}\epsilon. \tag{10}$$

As $\mu_{k_i} \geq k_i$, we have $y_{\mu_{k_i}} \in N(y^{**}; \delta)$ such that $x^{**}(y_{\mu_{k_i}}) \leq -\frac{2}{3}\epsilon$. Combining it with (10), we have

$$x^{(k_i)}(y_{\mu_{k_i}}) \leq -\frac{1}{3}\epsilon < 0 \tag{11}$$

and

$$x^{(k_{i+1})}(y_{\mu_{k_i}}) \leq -\frac{1}{3}\epsilon < 0. \tag{12}$$

According to (11), we have $y_{\mu_{k_i}} \in H_k$. This implies that $y_{\mu_{k_i}} \in F$ forever after the k th iteration. Thus

$$x^{(k_{i+1})}(y_{\mu_{k_i}}) \geq 0,$$

as $x^{(k_{i+1})}$ is an optimal solution of (8), with $y_{\mu_{k_i}} \in F$ now. This contradicts (12). Hence, we have $\phi(x^{**}) \geq 0$. This shows that x^{**} is an optimal solution of (3). The proof of the theorem is completed. \square

2.6 Another Iterative Polyhedral Outer Approximation Algorithm

In Algorithm 1, the set F expands steadily as k increases. Hence, the number of constraints in (8), though is much less than the number of constraints in (7) in the basic polyhedral outer approximation algorithm, is still somewhat large when k is big. We now give another iterative polyhedral outer approximation algorithm to improve this. Suppose that Assumptions A and B hold.

Algorithm 2 (Another Iterative Polyhedral Outer Approximation Algorithm)

- Step 0* Let $k = 0$, $\mu_{-1} = 1$ and $F = G_0$. Let N be a positive integer.
- Step 1* Let $\mu_k = \mu_{k-1}$. Compute an optimal solution of (8) and denote it as $x^{(k)}$.
- Step 2* If $\mu_k \geq N$, stop. Otherwise, let $H_k = \{y \in G_{\mu_k} : x^{(k)}(y) < 0\}$.
- Step 3* If $H_k = \emptyset$, let $\mu_k = \mu_k + 1$ and go to Step 2. Otherwise, let $\bar{F} = \{y \in F : x^{(k)}(y) = 0\}$. Let $F = H_k \cup \bar{F}$ and $k = k + 1$. Go to Step 1.

Theorem 4 *Suppose that Assumptions A and B hold. Then we have $f(x^{(k)}) \leq f(x^{(k+1)}) \leq f(x^*)$ for all k . If $N \rightarrow \infty$, then we have a sequence $\{x^{(k)} : k = 0, 1, \dots\} \subset R$. If furthermore $\{x^{(k)}\}$ converges to a point x^{**} , then x^{**} is an optimal solution of (3).*

The proof of this theorem is similar to the proof of Theorem 3. We omit it.

Remark We see that the number of constraints in (8) is stable in Algorithm 2. A cost of this improvement is that the second conclusion of Theorem 4 is a little weaker than the second conclusion of Theorem 3.

3 The Other Approaches

3.1 The Conic Linear Program Approach

If f is a convex quadratic function and g is an affine function, then (3) can be converted to a conic linear program (CLP) via a second-order cone transformation [16]. Theoretically, CLP problems can be solved by polynomial-time interior point algorithms [11, 12]. In this approach, a self-concordant function needs to be established. The smallest Z-eigenvalue function $\phi_B(x)$ has some properties of a self-concordant function. But it is nonsmooth and thus does not satisfy the differentiability requirements of a self-concordant function. Further investigation on possible smooth defining functions of the positive semi-definite space tensor cone \mathcal{S} is needed to establish a self-concordant function.

A closed convex cone is a symmetric cone if it is a self-dual and homogeneous cone [6]. However, unlike the semi-definite problem (SDP) and the second-order cone problem (SDCP), where the semi-definite matrix cone and the second-order cone are self-dual, the positive semi-definite space tensor cone \mathcal{S} is not self-dual [16]. We may check if the positive semi-definite space tensor cone \mathcal{S} is a homogeneous cone or not. A closed convex cone \mathcal{K} with nonempty interior is homogeneous if for any two interior points u and v of \mathcal{K} , there exists an invertible linear mapping T such that $T(\mathcal{K}) = \mathcal{K}$ and $T(u) = v$, i.e., the group of automorphisms of \mathcal{K} acts transitively on the interior of \mathcal{K} . If the positive semi-definite space tensor cone \mathcal{S} is a homogeneous cone, then we may use the techniques on homogeneous cones [10] to construct interior point algorithms to solve the CLP with the conic constraint $x \in \mathcal{S}$. If the positive semi-definite space tensor cone \mathcal{S} is not a homogeneous cone, then we may further check if it is a hyperbolic cone [10, 19] or not. Hyperbolic cones contain homogeneous cones as a subclass [10, 19]. If the positive semi-definite space tensor cone \mathcal{S}

is a hyperbolic cone, then we may use the techniques on hyperbolic cones [10, 19] to construct interior point algorithms to solve the concerned CLP. Finally, it is possible that the positive semi-definite space tensor cone \mathcal{S} is not a hyperbolic cone. Then we will analyze the properties of the positive semi-definite space tensor cone further, to see what kind of interior point algorithms are suitable for solving the concerned CLP.

Therefore, the conic linear program approach for solving this problem is not ready for practical use at this moment.

3.2 The Nonsmooth Convex Program Approach

We now discuss algorithms for solving the nonsmooth convex program (3). Under Assumptions A and B, we may convert it to a standard convex feasibility problem [8, 9]. A convex feasibility problem is to compute a point in a convex set $\bar{\mathcal{S}}$, where $\bar{\mathcal{S}}$ is contained in a compact set R , and is assumed to contain an interior [8, 9]. Then we need an oracle which for every point $\bar{x} \in R$ returns either a statement that \bar{x} is feasible, or a cutting plane to separate \bar{x} from the feasible set. With such an oracle, we may apply the analytic center cutting plane method in [8, 9], and obtain a convergence estimate in $O(n(\log 1/\epsilon)^2)$ calls to the oracle. As f and g are twice smooth, it is easy to handle them. The key part of this oracle is to solve

$$\phi_D(x) = \min\{x(y) : y \in D\}.$$

We have the following proposition.

Proposition 3 *Given $x \in \mathfrak{R}^n$. Then $x \in \mathcal{S}$ if and only if the following three conditions are satisfied:*

- (i) $x \in \mathcal{S}_0$;
- (ii) *for any stationary point s of ψ_k for $-1 \leq s \leq 1$ and $k = 1, \dots, 6$, we have $\psi_k(s) \geq 0$, where $\psi_1(s) = x(1, 1, s)$, $\psi_2(s) = x(1, -1, s)$, $\psi_3(s) = x(1, s, 1)$, $\psi_4(s) = x(1, s, -1)$, $\psi_5(s) = x(s, 1, 1)$ and $\psi_6(s) = x(s, 1, -1)$;*
- (iii) *for any stationary point t of d_k , where $-1 \leq t_1, t_2 \leq 1$ and $k = 1, 2, 3$, we have $d_k(t) \geq 0$.*

Proof The “only if” part follows from the definition of positive semi-definiteness of x . The “if” part follows from (6) and the fact that ϕ_D is a defining function of \mathcal{S} . Note that C has 12 edges and D contains 9 edges. Because the fact that $x(y)$ is an even function for y , we only need to consider six edges in (ii). □

The condition (i) is easy to check. It is not difficult to find stationary points of ψ_k for $k = 1, 2, 3$. To find stationary points of d_k , we need to solve

$$\begin{cases} \frac{\partial}{\partial t_1} d_k(t) = 0, \\ \frac{\partial}{\partial t_2} d_k(t) = 0. \end{cases}$$

This is a two-variable polynomial system. We may use the Sylvester formula [4] to solve it.

3.3 The Bi-level Program Approach

We can write our problem (3) into the following bi-level program problem:

$$\min f(x) \tag{13}$$

$$\text{s.t. } g(x) \leq 0, \tag{14}$$

$$\min_{y \in \mathfrak{N}^3} x(y) \geq 0. \tag{15}$$

Due to the harmonic property of the function $x(y)$, the above problem is equivalent to

$$\min f(x) \tag{16}$$

$$\text{s.t. } g(x) \leq 0, \tag{17}$$

$$\phi(x) = \min_{y \in \mathfrak{N}^3, \|y\|=1} x(y) \geq 0. \tag{18}$$

As discussed in Sect. 2.3, $\phi(x)$ is a concave function. Thus problem (16)–(18) is a convex program. The dual problem can be written as

$$\max_{\tau \in \mathfrak{N}^p, \rho \in \mathfrak{R}} L(\tau, \rho) \tag{19}$$

$$\text{s.t. } \tau \geq 0, \rho \geq 0, \tag{20}$$

where

$$L(\tau, \rho) = \min_x \{ f(x) + \tau^T g(x) - \rho \phi(x) \}. \tag{21}$$

Thus we can apply a dual algorithm for solving (19)–(20).

Algorithm 3 (Dual Algorithm for Structured Convex Problems)

Step 1 $\tau_0 = 0, \rho_0 = 0, k = 0$.

Step 2 Compute

$$z^{(k)} := \nabla L(\tau_k, \rho_k) = \begin{pmatrix} \nabla_{\tau} L(\tau_k, \rho_k) \\ \nabla_{\rho} L(\tau_k, \rho_k) \end{pmatrix}. \tag{22}$$

Step 3 if $z^{(k)} \leq 0$ and

$$(z^{(k)})^T \begin{pmatrix} \tau_k \\ \rho_k \end{pmatrix} = 0,$$

then stop.

Carry out a line search, namely computing $\alpha_k > 0$ and let

$$\begin{pmatrix} \tau_{k+1} \\ \rho_{k+1} \end{pmatrix} := \left[\begin{pmatrix} \tau_k \\ \rho_k \end{pmatrix} + \alpha_k z^{(k)} \right]_+,$$

$k := k + 1$, and go to Step 2.

It is easy to show that the above algorithm is convergent if certain line search conditions are satisfied. Indeed, the above algorithm is a truncated gradient method for the dual problem. The analysis of Calamai and Moré [3] can be used. Namely, we can try

$$\alpha_k = \min[2\alpha_{k-1}, \gamma]$$

first, and if necessary, we reduce α_k by a fractor of 2 until the line search condition

$$L(\tau_{k+1}, \rho_{k+1}) \geq L(\tau_k, \rho_k) + c_1 (\nabla L(\tau_k, \rho_k))^T \begin{pmatrix} \tau_{k+1} - \tau_k \\ \rho_{k+1} - \rho_k \end{pmatrix}$$

is satisfied, where $\gamma > 0$ and $c_1 \in (0, 0.5)$ are constants. More details can be found in Chap. 11 of Sun and Yuan [20].

Now we consider the specific problem (2), which can be rewritten as:

$$\min_x (x - \bar{x})^T Q(x - \bar{x}) \tag{23}$$

subject to

$$\lambda_{min}(x) \geq 0. \tag{24}$$

Define

$$\psi(y) = \{\psi_{ij} | \psi_{ij} = y_1^i y_2^j y_3^{m-i-j}\}.$$

Then we have

$$\lambda_{min}(x) = \min_{\|y\|=1} x^T \psi(y) = x^T \psi(y(x)),$$

for some $y(x)$ satisfying $\|y(x)\| = 1$. It is easy to show that

$$\nabla_x \lambda_{min}(x) = \psi(y(x)). \tag{25}$$

The dual problem is now

$$\max_{\rho > 0} L(\rho), \tag{26}$$

where

$$\begin{aligned} L(\rho) &= \min_x \{(x - \bar{x})^T Q(x - \bar{x}) - \rho \lambda_{min}(x)\} \\ &= (x(\rho) - \bar{x})^T Q(x(\rho) - \bar{x}) - \rho \lambda_{min}(x(\rho)) \end{aligned} \tag{27}$$

with $x(\rho)$ being the solution of

$$2Q(x - \bar{x}) - \rho \psi(y(x)) = 0. \tag{28}$$

Hence, we have that

$$L'(\rho) = [Q(x(\rho) - \bar{x}) - \rho \psi(y(x(\rho)))]^T x'(\rho) - \lambda_{min}(x(\rho)) = -\lambda_{min}(x(\rho)). \tag{29}$$

The fact that $\mathbf{x}(\rho)$ satisfies (28) implies that

$$2Qx'_\rho(\rho) - \psi(y(x(\rho))) - \rho \psi'_\rho(y(x(\rho))) = 0, \tag{30}$$

which gives that

$$x'_\rho(\rho) = \frac{1}{2} Q^+ [\psi(y(x(\rho))) + \rho \psi'_\rho(y(x(\rho)))]. \tag{31}$$

Therefore it follows from (25), (29) and (31) that

$$\begin{aligned} L''_{\rho\rho}(\rho) &= -\frac{d}{d\rho} [\lambda_{\min}(x(\rho))] = -(\psi(y(x(\rho))))^T x'_\rho(\rho) \\ &= -\frac{1}{2} (\psi(y(x(\rho))))^T Q^+ [\psi(y(x(\rho))) + \rho \psi'_\rho(y(x(\rho)))] \\ &\approx -\frac{1}{2} (\psi(y(x(\rho))))^T Q^+ \psi(y(x(\rho))). \end{aligned} \tag{32}$$

Thus, it follows (29) and (32) that an approximation to the Newton's method for the dual problem is

$$\rho_{k+1} = \rho_k - \frac{2\lambda_{\min}(x(\rho_k))}{(\psi(y(x(\rho_k))))^T Q^+ \psi(y(x(\rho_k)))}. \tag{33}$$

Now (28) implies that

$$2Q(x(\rho_{k+1}) - x(\rho_k)) = \rho_{k+1} \psi(y(x(\rho_{k+1}))) - \rho_k \psi(y(x(\rho_k))).$$

As $y(x(\rho_{k+1}))$ is not known before x_{k+1} is computed. Thus, we define the next iterate x_{k+1} by

$$2Q(x_{k+1} - x_k) = (\rho_{k+1} - \rho_k) \psi(y(x_k)). \tag{34}$$

Relations (33) and (34) give that the iterate formula

$$x_{k+1} = x_k - \frac{\lambda_{\min}(x_k)}{(\psi(y(x_k))))^T Q^+ \psi(y(x_k))} Q^+ \psi(y(x_k)).$$

Hence we have the following dual algorithm for the special problem (2).

Algorithm 4 (Dual Algorithm)

Step 1 $k := 0, x^{(k)} = \bar{x}$.

Step 2 Find $y^{(k)} = (y_1^{(k)}, y_2^{(k)}, y_3^{(k)})^T$ such that $\|y^{(k)}\| = 1$ and

$$(x^{(k)})^T \psi(y^{(k)}) = \lambda_{\min}(x^{(k)})$$

$$\psi = \psi(y^{(k)}).$$

Step 3 If $\lambda_{\min}(x^{(k)}) \geq 0$ then stop.

Step 4 Let $\Delta\rho := \frac{-\lambda_{\min}(x^{(k)})}{\psi^T Q^+ \psi}$

$$x^{(k+1)} = x^{(k)} + \Delta\rho Q^+ \psi$$

$k := k + 1$, go to Step 2.

For the above algorithm, we can establish the following convergence result.

Lemma 1 *If Q is positive definite, and let $x^{(k)}$ be generated by the above algorithm. Then either $\lambda_{\min}(\bar{x}) \geq 0$ or any accumulation point \hat{x} of $\{x^{(k)}\}$ satisfies that $\lambda_{\min}(\hat{x}) = 0$.*

Proof If Q is positive, we have that

$$x^{(k+1)} = x^{(k)} - \frac{(x^{(k)})^T \psi(y^{(k)})}{(\psi(y^{(k)}))^T Q^{-1} \psi(y^{(k)})} Q^{-1} \psi(y^{(k)}) = \left(I - \frac{Q^{-1} \psi_k \psi_k^T}{\psi_k^T Q^{-1} \psi_k} \right) x^{(k)},$$

where $\psi_k = \psi(y^{(k)})$. Thus we have that

$$\|x^{(k+1)}\|_Q^2 = \|x^{(k)}\|_Q^2 - \frac{(\psi_k^T x^{(k)})^2}{\|\psi_k\|_{Q^{-1}}^2},$$

where $\|x\|_Q^2 = x^T Q x$. The above relations show that the iterate points $\{x^{(k)}\}$ generated by the algorithm are in the bounded set $\{x | x^T Q x \leq \bar{x}^T Q \bar{x}\}$. Thus ψ_k are also bounded. Hence

$$\sum_{k=1}^{\infty} [\lambda_{\min}(x^{(k)})]^2 = \sum_{k=1}^{\infty} [\psi_k^T x_k]^2 < \infty.$$

The above relation shows that any accumulation point \hat{x} of $\{x_k\}$ satisfies $\lambda_{\min}(\hat{x}) = 0$. This completes our proof. □

The above result shows that $\{x^{(k)}\}$ converge to the feasible set $\{x | \lambda_{\min}(x) \geq 0\}$, but it does not imply the sequence $\{x^{(k)}\}$ convergence to a fix point. Indeed, it does not even ensure the optimality conditions at an accumulate point.

Thus, we modify the algorithm if

$$|\psi_k^T x_k| < 0.5 |\psi_k^T \bar{x}|. \tag{35}$$

Namely, when (35) holds, we define the next iterate by the following formulae:

$$x_{k+1} = \bar{d} - \frac{\psi_k^T \bar{x}}{\psi_k^T Q^+ \psi_k} Q^+ \psi_k = \left[I - \frac{Q^+ \psi_k^T \psi_k}{\psi_k^T Q^+ \psi_k} \right] \bar{x}. \tag{36}$$

Formulae (36) is an SQP step of to problem (23)–(24), thus fast local convergence is expected.

4 Numerical Experiments

We report some numerical experiments for Algorithms 1, 2 and 4. The experiments are implemented in Matlab by using the a dual-core 2.4 GHz cpu 2GB RAM personal computer.

For Algorithms 1 and 2, we take $N = 9$. When k is equal to N , there are 9×2^{16} points in G_k and $\rho(F_k)$ is about 2^{-8} . The problems we test for Algorithms 1 and 2 are:

Example 1

$$\min \left\{ \frac{1}{2} (x - \bar{x})^T Q (x - \bar{x}) : Ax \leq b, x \in S \right\}$$

Table 1 Example 1 for Algorithms 1 and 2 in the small order case

$dim(x)$	$num(G)$	m	Algorithm 1			Algorithm 2		
			Time (s)	$num(Con)$	$\lambda_{min}(x^{(k)})$	Time (s)	$num(Con)$	$\lambda_{min}(x^{(k)})$
15	9×2^{16}	4	0.1187	7.4	-1.79×10^{-4}	0.1156	7.6	-1.79×10^{-4}
45	9×2^{16}	8	0.947	11.8	-2.75×10^{-4}	0.778	7.6	-1.85×10^{-4}
91	9×2^{16}	12	1.861	15.2	-1.36×10^{-3}	1.713	12	-6.1×10^{-5}
153	9×2^{16}	16	6.685	31	-1.31×10^{-4}	4.77	17.6	-7.57×10^{-5}

Table 2 Example 2 for Algorithms 1 and 2 in the small order case

$dim(x)$	$num(G)$	m	Algorithm 1			Algorithm 2		
			Time (s)	$num(Con)$	$\lambda_{min}(x^{(k)})$	Time (s)	$num(Con)$	$\lambda_{min}(x^{(k)})$
15	9×2^{16}	4	0.0844	5.4	-1.46×10^{-4}	0.1344	5.8	-2.6×10^{-4}
45	9×2^{16}	8	0.6438	10	-1.69×10^{-4}	0.8218	10.2	-3.2×10^{-4}
91	9×2^{16}	12	1.437	14.4	-6.6×10^{-5}	2.034	18.6	-1.42×10^{-4}
153	9×2^{16}	16	6.929	17	-1.13×10^{-4}	5.809	27	-1.16×10^{-4}

Example 2

$$\min \left\{ \frac{1}{2}(x - \bar{x})^\top Q(x - \bar{x}) : x \in S \right\}$$

In Examples 1 and 2, $Q = PDP^\top$ is a positive definite matrix, where the orthogonal matrix P and the diagonal matrix D as well as \bar{x} are all generated randomly. To make it easy to compare, we set $A = (1, 1 \dots, 1) \in R^{1 \times \frac{(m+1)(m+1)}{2}}$, $b = \frac{(m+1)(m+2)}{2}$. Example 2 is the PSDT model in [17].

We test 7 different orders for every algorithms, 4 small orders and 3 lager orders: $m = 4$; $m = 8$; $m = 12$; $m = 16$; $m = 20$; $m = 30$; $m = 40$, the dimensions of x are 15, 45, 91, 153, 231, 496, 861, respectively.

The $num(Con)$, $num(G)$, $\lambda_{min}(x^{(k)})$ and k in Tables 1–5 mean the number of the constraints in the relaxation program, the number of points in G_k , the minimum Z-eigenvalue of $x^{(k)}$ and the number of iteration when Algorithm stops. We use the method in [17] to calculate all the eigenvalues and eigenvectors of $x^{(k)}$, and take the least one as $\lambda_{min}(x^{(k)})$. Failure means that the time Algorithm cost is too long or the data overflows from the Ram. At each order we test 5 examples, and get the averages of $time$ (s), $num(Con)$ and $\lambda_{min}(x^{(k)})$.

The tested example for Algorithm 4 is

$$\min \{ (x - \bar{x})^\top (x - \bar{x}) : x \in S \}$$

where \bar{x} is generated randomly. It is the projection problem to S .

We test 4 different orders for Algorithm 4: $m = 4$; $m = 6$; $m = 8$; $m = 10$, and set the stop rule for Algorithm 4 to be $\lambda_{min}(x_k) > -10^{-5}$.

Table 3 Example 1 for Algorithms 1 and 2 in the lager order case

$dim(x)$	$num(G)$	m	Algorithm 1		Algorithm 2	
			Time (s)	$num(Con)$	Time (s)	$num(Con)$
231	9×2^{16}	20	13.2	38.4	13.5	38
496	9×2^{16}	30	152	64	144.1	34
861	9×2^{16}	40	Failure		Failure	

Table 4 Example 2 for Algorithms 1 and 2 in the lager order case

$dim(x)$	$num(G)$	m	Algorithm 1		Algorithm 2	
			Time (s)	$num(Con)$	Time (s)	$num(Con)$
231	9×2^{16}	20	13.9	40	14.1	38.4
496	9×2^{16}	30	136.3	66.2	141.1	33.4
861	9×2^{16}	40	665.9	81	695.1	129

Table 5 The tested examples for Algorithm 4

$dim(x)$	m	Time (s)	$\lambda_{min}(x^{(k)})$	$iter$
15	4	2.8	-1.3×10^{-17}	3.2
28	6	9.3	-1.27×10^{-16}	4
45	8	21	-8.6×10^{-17}	4
66	10	65.2	-2.5×10^{-6}	4.8

The $\lambda_{min}(x^{(k)})$ and $iter$ in the following table mean the minimum Z-eigenvalue and the number of iterations when Algorithm 4 stops. We test 5 examples for each order and get the average of $\lambda_{min}(x^{(k)})$ and $iter$.

We see that Algorithms 1 and 2 can be used to solve some large scale problems. The failure in Table 3 was caused by the storage problem. It is worth improving the storage use in these two algorithms. On the other hand, Algorithm 1 is efficient for solving some small order problems for $Q = I$. We have not tested the analytic center cutting plane approach discussed in Sect. 3.2.

Acknowledgements The authors would like to thank two referees for their helpful comments.

References

- [1] Barmpoutis, A., Hwang, M.S., Howland, D., Forder, J.R., Vemuri, B.C.: Regularized positive-definite fourth order tensor filed estimation from DW-MRI. *Neuroimage* **45**, 5153–5162 (2009)
- [2] Barmpoutis, A., Jian, B., Vemuri, B.C., Shepherd, T.M.: Symmetric positive 4th order tensors & their estimation from diffusion weighted MRI. In: Karssemeijer, M., Lelieveldt, B. (eds.) *Information Processing and Medical Imaging*, pp. 308–319. Springer, Berlin (2007)
- [3] Calamai, P.H., Moré, J.J.: Projected gradient methods for linearly constrained problems. *Math. Program.* **39**, 93–116 (1987)
- [4] Cox, D., Little, J., O’Shea, D.: *Using Algebraic Geometry*. Springer, New York (1998)
- [5] Descoteaux, M., Angelino, E., Fitzgibbons, S., Deriche, R.: Apparent diffusion coefficients from high angular diffusion imaging: estimation and applications. *Magn. Reson. Med.* **56**, 395–410 (2006)
- [6] Faraut, J., Korányi, A.: *Analysis on Symmetric Cones*. Oxford University Press, New York (1994)

- [7] Ghosh, A., Descoteaux, M., Deriche, R.: Riemannian framework for estimating symmetric positive definite 4th order diffusion tensors. In: Metaxas, D., Axel, L., Fichtinger, G., Székeley, G. (eds.) *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2008*, pp. 858–865. Springer, Berlin (2008)
- [8] Goffin, J.L., Luo, Z.Q., Ye, Y.: Complex analysis of an interior cutting plane method for convex feasibility problems. *SIAM J. Control Optim.* **6**, 638–652 (1996)
- [9] Goffin, J.L., Vial, J.P.: Convex nondifferentiable optimization: a survey focused on the analytical center cutting plane method. *Optim. Methods Softw.* **17**, 805–867 (2002)
- [10] Güler, O., Tuncel, L.: Characterization of the barrier parameter of homogeneous convex cones. *Math. Program.* **81**, 55–76 (1998)
- [11] Nemirovski, A.: Advances in convex optimization: conic programming. In: Sanz-Sol, M., Soria, J., Varona, J.L., Verdera, J. (eds.) *Proceedings of the International Congress of Mathematicians, 2006*, vol. 1, pp. 413–444. EMS, Zürich (2007)
- [12] Nesterov, Yu., Nemirovski, A.: *Interior Point Polynomial Time Methods in Convex Programming*. SIAM, Philadelphia (1994)
- [13] Ozarslan, E., Mareci, T.H.: Generalized diffusion tensor imaging and analytical relationships between diffusion tensor imaging and high angular resolution diffusion imaging. *Magn. Reson. Med.* **50**, 955–965 (2003)
- [14] Qi, L.: Eigenvalues of a real supersymmetric tensor. *J. Symb. Comput.* **40**, 1302–1324 (2005)
- [15] Qi, L., Wang, F., Wang, Y.: Z-eigenvalue methods for a global polynomial optimization problem. *Math. Program.* **118**, 301–316 (2009)
- [16] Qi, L., Ye, Y.: Space tensor conic programming. Department of Applied Mathematics, The Hong Kong Polytechnic University, Revised in June 2012
- [17] Qi, L., Yu, G., Wu, E.X.: Higher order positive semi-definite diffusion tensor imaging. *SIAM J. Imaging Sci.* **3**, 416–433 (2010)
- [18] Qi, L., Yu, G., Xu, Y.: Nonnegative diffusion orientation distribution function. *J. Math. Imaging Vis.* **45**, 103–113 (2013)
- [19] Renegar, J.: Hyperbolic programs, and their derivative relaxations. *Found. Comput. Math.* **6**, 59–79 (2006)
- [20] Sun, W., Yuan, Y.: *Optimization Theory and Methods*. Springer, Berlin (2006)