二次錐制約をもつ半無限計画問題に対する切除平面アプローチ
A cutting plane approach for semi-infinite programming problems with second-order cone constraints

林　俊介
Shunsuke Hayashi

京都大学大学院情報学研究科数理工学専攻
Department of Applied Mathematics and Physics
Graduate School of Informatics
Kyoto University

Kyoto 606-8501, Japan
hayashi@amp.i.kyoto-u.ac.jp

概要

半無限計画問題とは，有限次元の決定変数と無限個の制約式で特徴付けられる最適化問題であり，幅広い分野に応用が可能なことから，これまで盛んに研究がなされてきた．しかし，既存の研究の殆どは等式制約および不等式制約のみで表されるような問題を対象としており，二次錐制約を含むような半無限計画問題に対する研究はこれまで殆どなされていなかった．そこで，本研究では，Lai and Wu の提案した切除平面アプローチを二次錐制約を含む半無限計画問題に拡張した手法を提案する．また，アルゴリズムにおける部分問題とその双対問題の解が満たすべき二次錐相補性条件に対して，Jordan 代数に基づいた解析を行うことにより，提案手法の収束性を示す．さらに，アルゴリズムの性能を調べるために行ったいくつかの数値実験の結果も合わせて報告する．

Keywords: 半無限計画問題，二次錐，切除平面アプローチ，explicit algorithm

1　はじめに

半無限計画問題 (Semi-infinite programming problem: SIP) [2, 13, 14, 15, 24] とは，有限次元のベクトル引数をもつ関数を無限個の制約の下で最小化する問題であり，以下のように記述される．

$$\begin{array}{ll} \text{Minimize} & f(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{x} \in X, \quad g(\boldsymbol{x}, t) \geq 0 \ \ (\forall t \in T). \end{array} \tag{1.1}$$

ここで，$X \subseteq \Re^n$ は与えられた凸集合，T はコンパクトな Hausdorff 空間，$f : \Re^n \to \Re$ および $g : \Re^n \times T \to \Re$ は与えられた関数である．もし，T が有限個の要素をもつような集合ならば，問題 (1.1) は有限個の制約をもつ普通の非線形計画問題である．

SIP の解を求めるため，これまで多くの手法が研究されてきた．離散化法 (discretization method) [19, 22] はその中でも重要な手法の一つである．この手法では，各反復において，集

合 T を有限集合 $T_k \subset T$ で近似した部分問題を解き，反復が進むにつれて $|T_k|$ を大きくしていく．より詳細にいうと，各反復で以下のような非線形計画問題:

$$\text{Minimize } f(\boldsymbol{x}) \quad \text{subject to } \boldsymbol{x} \in X,\ \ g(\boldsymbol{x}, t) \geq 0 \ \ (\forall t \in T_k) \tag{1.2}$$

を解き，T_k の密度を高くしていくことによって，$\boldsymbol{x}^k$ を SIP (1.1) の解に収束させていくものである．しかしながら，この手法では反復が進むにつれて T_k の要素数が膨大になり，その結果，計算コストがとても大きくなってしまうという欠点がある．離散化法以外に，置換法 (exchange method) [13, 23] や逐次 KKT 法 [24] なども有効な手法として知られている．除去法 (reduction method) は [9, 12, 13] などの文献で提案された手法であり，集合 T が二回微分可能な関数 $c_j : \Re^m \to \Re,\ j = 1, \ldots, r$ によって $T = \{t \,|\, c_j(t) \leq 0,\ j = 1, \ldots, r\}$ と表される SIP を対象としている．典型的な除去法およびそれに基づいた手法では，各反復 k において，最適化問題

$$\text{Minimize } g(\boldsymbol{x}^k, t) \quad \text{subject to } t \in T$$

を解き，さらに，その解集合 S_k に対して以下の問題を解く．

$$\text{Minimize } f(\boldsymbol{x}) \quad \text{subject to } g(\boldsymbol{x}, t) \geq 0,\ \forall t \in S_k$$

そして，次の反復点 $\boldsymbol{x}^{k+1}$ を得る．ある種の正則性の仮定 [19] の下で集合 S_k が有限，すなわち部分問題が普通の非線形計画問題に帰着することが証明されている．除去 SQP 法，離散化 SQP 法，信頼領域法タイプの手法もまた何人かの研究者によって研究されてきた [4, 16, 21]．これらの手法は適当な条件の下で大域的収束性を示すが，すべての局所最適解を見つけるのは非常に困難で計算コストがかかる．文献 [20] では，問題 (1.2) のすべての局所最適解を探し出すことがいかに困難であるかについて言及している．Lai and Wu [15] は，$X = \Re^n_+$ であり，関数 g が t を固定したときに $\boldsymbol{x}$ に対して線形であるような SIP に対して，切除平面アプローチの一種である *explicit algorithm* を提案した．この手法では，離散化法と同様，各反復で T を有限集合 T_k でおきかえた部分問題を解くが，各反復で有効な制約だけを保持するため，$|T_k| \leq n$ という関係が常に保証され，計算コストを大幅に節約できるというメリットがある．

本論文では，以下のような二次錐制約と無限個の線形不等式を含むような SIP を考える．

$$\begin{aligned} \text{Minimize} \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{x} \in \mathcal{K}^n, \quad \boldsymbol{a}(t)^T \boldsymbol{x} - b(t) \geq 0 \ \ (\forall t \in T), \end{aligned} \tag{1.3}$$

ここで，ベクトル $\boldsymbol{c} \in \Re^n$ および関数 $b : T \to \Re$, $\boldsymbol{a} : T \to \Re^n$ は与えられており，$\mathcal{K}^n \subset \Re^n$ は以下で定義される n 次元の二次錐 (Second-order cone: SOC) である．

$$\mathcal{K}^n := \left\{ \boldsymbol{x} = (x_1, \tilde{\boldsymbol{x}}) \in \Re \times \Re^{n-1} \,\middle|\, x_1 \geq \|\tilde{\boldsymbol{x}}\| \right\}.$$

本論文を通して，$\|\cdot\|$ はユークリッドノルムを表すものとする．また，ベクトル $\boldsymbol{x} \in \Re^n$ は $\boldsymbol{x} := (x_1, \tilde{\boldsymbol{x}}) \in \Re \times \Re^{n-1}$ のようにも表記され，$\tilde{\boldsymbol{x}} := (x_2, \ldots, x_n)^T \in \Re^{n-1}$ である．ただし，$n = 1$ の場合は $\mathcal{K}^1$ は非負実数の集合，すなわち $\mathcal{K}^1 = \Re_+$ である．SIP (1.3) において，制約 $\boldsymbol{x} \in \mathcal{K}^n$ の代わりに $\boldsymbol{x} \in \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_m}$（ただし $n_1 + \cdots + n_m = n$）という直積制約を考

えた方がより一般的ではあるが，本稿では簡単のため一つの二次錐に対する制約のみを考える．また，T が有限個の要素をもつような集合ならば，問題 (1.3) は有限個の制約をもつ，いわゆる二次錐計画問題 (Second-order cone programming: SOCP) [1] に帰着されることに注意する．

近年，二次錐を含む問題に対する研究が多くの注目を集めてきた．その中でも最もよく知られているのが二次錐計画問題 (SOCP) である．この問題は凸二次計画問題をサブクラスとして含み，現在のところ主双対内点法が有効な手段として確立されている [1, 17]．また，最近では，線形計画問題の単体法に基づいたアプローチも提案されている [18]．一方，非線形な関数を含む SOCP はより複雑であり，研究もそれほど多くはない．文献 [25] では，著者らはバリア関数とポテンシャル関数によって構成されるような主双対メリット関数を含むアルゴリズム提案しており，その大域的収束性を示している．二次錐相補性問題 (Second-order cone complementarity problem: SOCCP) は線形相補性問題や非線形相補性問題 [5, 6] を二次錐制約へと拡張した問題である．文献 [8] では，著者らは Jordan 代数を SOCCP に導入し，アルゴリズムを構築する上で必要ないくつかの平滑化関数に対する性質を分析した．また，[8] の分析に基づき，平滑化法と正則化法を組み合わせたアルゴリズムが [11] で提案されており，ある種の単調性の仮定の下で大域的収束性と二次収束性を有することが証明されている．その他にも，Jordan 代数に基づいたベクトル値関数の性質に関する研究 [3] や大規模問題に対する行列分割法 [10] なども最近の SOCCP の研究として挙げられる．

本研究の目的は，二次錐制約を含む SIP (1.3) を解くためのアルゴリズムを構築し，その収束性を示すことである．実際，本研究で提案するアルゴリズムは，Lai and Wu が非負ベクトル制約と無限個の線形関数を含む SIP に対して提案した explicit algorithm [15] を拡張したものである．しかしながら，その解析の方法は全く異なる．実際，[15] では相補スラック条件を成分毎に分解した上で収束解析を行っているが，SIP (1.3) に対する相補スラック条件は二次錐に特化した相補性条件を含むため，成分毎の解析は本質的に意味をなさない．そこで，本研究では Jordan 代数におけるスペクトル分解を導入し，それに基づくある種の座標系を定義して，その座標系の下でアルゴリズムの収束性を示す．

本稿の構成は次のようになる．2 節では，まず二次錐に対するスペクトル分解の概念を紹介する．また，SIP (1.3) に対する双対定理と相補スラック条件を記述する．3 節では SIP (1.3) に対する explicit algorithm を提案し，適当な仮定の下で収束性を示す．4 節では数値実験の結果を示す．5 節ではまとめと今後の課題を述べる．

2　準備

2.1　二次錐に対するスペクトル分解

まず，二次錐に対するスペクトル分解を導入する．これは，Jordan 代数 [7] におけるトピックの一つであり，アルゴリズムにおける反復点と二次錐との関係を記述するのに重要な役割を演ずる．

$n \geq 2$ であるような任意のベクトル $\boldsymbol{x} := (x_1, \tilde{\boldsymbol{x}}) \in \Re \times \Re^{n-1}$ に対して，そのスペクトル

値 $\lambda_1(\boldsymbol{x}),\ \lambda_2(\boldsymbol{x}) \in \Re$, およびスペクトルベクトル $\boldsymbol{v}_1(\boldsymbol{x}),\ \boldsymbol{v}_2(\boldsymbol{x}) \in \Re^n$ を以下で定義する．

$$\begin{aligned}
\lambda_i(\boldsymbol{x}) &:= x_1 + (-1)^i \|\tilde{\boldsymbol{x}}\|, \\
\boldsymbol{v}_i(\boldsymbol{x}) &:= \begin{cases} \dfrac{1}{2}\left(1,\ (-1)^i \dfrac{\tilde{\boldsymbol{x}}}{\|\tilde{\boldsymbol{x}}\|}\right) & (\tilde{\boldsymbol{x}} \neq \boldsymbol{0}), \\ \dfrac{1}{2}\left(1,\ (-1)^i \boldsymbol{w}\right) & (\tilde{\boldsymbol{x}} = \boldsymbol{0}), \end{cases} \qquad (i = 1, 2)
\end{aligned}$$

ここで，$\boldsymbol{w} \in \Re^{n-1}$ は $\|\boldsymbol{w}\| = 1$ であるような任意のベクトルである．このとき，$\boldsymbol{x}$ に対するスペクトル分解が

$$\boldsymbol{x} = \lambda_1(\boldsymbol{x})\boldsymbol{v}_1(\boldsymbol{x}) + \lambda_2(\boldsymbol{x})\boldsymbol{v}_2(\boldsymbol{x}). \tag{2.1}$$

で定義される．ここで，$\lambda_1(\boldsymbol{x}) \leq \lambda_2(\boldsymbol{x})$ が常に成立し，$\lambda_1(\boldsymbol{x}) \geq 0$ ならばそのときに限り $\boldsymbol{x} \in \mathcal{K}^n$ であることに注意する．さらに，$\lambda_1(\boldsymbol{x}) = 0$ であることと $\boldsymbol{x} \in \mathrm{bd}\,\mathcal{K}^n$ ($\mathcal{K}^n$ の境界) が同値であり，$\lambda_1(\boldsymbol{x}) > 0$ であることと $\boldsymbol{x} \in \mathrm{int}\,\mathcal{K}^n$ ($\mathcal{K}^n$ の内部) であることが同値であることにも注意する．また，$\|\boldsymbol{v}_1(\boldsymbol{x})\| = \|\boldsymbol{v}_2(\boldsymbol{x})\| = 1/\sqrt{2}$ および $\boldsymbol{v}_1(\boldsymbol{x})^T\boldsymbol{v}_2(\boldsymbol{x}) = 0$ が常に成り立つ．

2.2 双対定理と相補スラック条件

次に，SIP (1.3) に対する双対定理と相補スラック条件を紹介する．$M(T)$ を T におけるすべての有界な正則ボレル測度を含む空間とし，$M_+(T)$ を $M(T)$ に含まれる非負錐とする．このとき，SIP (1.3) の双対問題は以下で与えられる．

$$\begin{aligned}
&\text{Maximize} \quad \int_T b(t)d\mu(t) \\
&\text{subject to} \quad \mu \in M_+(T), \quad \boldsymbol{c} - \int_T \boldsymbol{a}(t)d\mu(t) \in \mathcal{K}^n.
\end{aligned} \tag{2.2}$$

ただし，

$$\int_T \boldsymbol{a}(t)d\mu(t) := \left(\int_T a_1(t)d\mu(t), \ldots, \int_T a_n(t)d\mu(t)\right)^T \in \Re^n$$

である．任意の主双対の実行可能解の組 $(\overline{\boldsymbol{x}}, \overline{\mu})$, に対して弱双対定理が成り立つ．

$$\boldsymbol{c}^T\overline{\boldsymbol{x}} \geq \int_T b(t)d\overline{\mu}(t).$$

さらに，ある条件の下で強双対定理も成り立つ．

命題 2.1 (強双対定理)

(a) 主問題 (1.3) が最適解 $\boldsymbol{x}^*$ を持つとする．また，任意の $t \in T$ に対してなるような実行可能点 $\overline{\boldsymbol{x}} \in \mathcal{K}^n$ が存在するとする．このとき，双対問題 (2.2) は最適解 $\mu^* \in M_+(T)$ をもち，$\boldsymbol{c}^T\boldsymbol{x}^* = \int_T b(t)d\mu^*(t)$ を満たす．

(b) 双対問題 (2.2) が最適解 μ^* を持つとする．また，任意の $t \in T$ に対して $\boldsymbol{c} - \int_T \boldsymbol{a}(t)d\overline{\mu}(t) \in \mathrm{int}\,\mathcal{K}^n$ となるような実行可能点 $\overline{\mu} \in M_+(T)$ が存在するとする．このとき，主問題 (1.3) は最適解 $\boldsymbol{x}^*$ をもち，$\boldsymbol{c}^T\boldsymbol{x}^* = \int_T b(t)d\mu^*(t)$ を満たす．

強双対性定理を用いて，SIP (1.3) に対する相補スラック条件を導くことができる．

命題 2.2 (相補スラック条件) $\boldsymbol{x}^* \in \Re^n$ と $\mu^* \in M(T)$ をそれぞれ (1.3) と (2.2) の実行可能解とする．また，命題 2.1 (a) か (b) の仮定のいずれかが成立するものとする．このとき，$\boldsymbol{x}^*$ と μ^* がいずれも最適解であることと，以下の関係が成り立つことは等価である．

$$\begin{aligned}&\mu^* \in M_+(T),\ \boldsymbol{a}(t)^T\boldsymbol{x}^* - b(t) \geq 0\ (\forall t \in T),\ \int_T \Big(\boldsymbol{a}(t)^T\boldsymbol{x}^* - b(t)\Big)\, d\mu^*(t) = 0,\\ &\boldsymbol{x}^* \in \mathcal{K}^n,\ \boldsymbol{c} - \int_T \boldsymbol{a}(t) d\mu^*(t) \in \mathcal{K}^n,\ (\boldsymbol{x}^*)^T \left\{\boldsymbol{c} - \int_T \boldsymbol{a}(t) d\mu^*(t)\right\} = 0.\end{aligned} \tag{2.3}$$

ベクトル $\boldsymbol{x} \in \Re^n$ と $\boldsymbol{z} \in \Re^n$ が以下の関係を満たすとき，それらは二次錐相補性条件を満たすという．

$$\boldsymbol{x} \in \mathcal{K}^n,\ \boldsymbol{z} \in \mathcal{K}^n,\ \boldsymbol{x}^T\boldsymbol{z} = 0. \tag{2.4}$$

式 (2.3) において二つのベクトル $\boldsymbol{x}^*$ と $\boldsymbol{c} - \int_T \boldsymbol{a}(t)d\mu^*(t)$ は二次錐相補性条件を満たしている．さらに，二次錐相補性条件を満たすベクトルの組に対して以下の命題が成り立つ．

命題 2.3 二つの n 次元実ベクトル $\boldsymbol{x}$ と $\boldsymbol{z}$ が二次錐相補性条件 (2.4) を満たしているとする．このとき，$\boldsymbol{x}$ と $\boldsymbol{z}$ のスペクトル値に関して以下の三つのいずれかが成り立つ．

$$\begin{aligned}&\text{(a)}\quad 0 = \lambda_1(\boldsymbol{x}) = \lambda_2(\boldsymbol{x})\ \textit{and}\ 0 \leq \lambda_1(\boldsymbol{z}) \leq \lambda_2(\boldsymbol{z})\quad (\textit{i.e.},\ \boldsymbol{x} = \boldsymbol{0}\ \textit{and}\ \boldsymbol{z} \in \mathcal{K}^n),\\ &\text{(b)}\quad 0 \leq \lambda_1(\boldsymbol{x}) \leq \lambda_2(\boldsymbol{x})\ \textit{and}\ 0 = \lambda_1(\boldsymbol{z}) = \lambda_2(\boldsymbol{z})\quad (\textit{i.e.},\ \boldsymbol{x} \in \mathcal{K}^n\ \textit{and}\ \boldsymbol{z} = \boldsymbol{0}),\\ &\text{(c)}\quad 0 = \lambda_1(\boldsymbol{x}) \leq \lambda_2(\boldsymbol{x})\ \textit{and}\ 0 = \lambda_1(\boldsymbol{z}) \leq \lambda_2(\boldsymbol{z})\quad (\textit{i.e.},\ \boldsymbol{x} \in \mathrm{bd}\,\mathcal{K}^n,\\ &\qquad\qquad \boldsymbol{z} \in \mathrm{bd}\,\mathcal{K}^n,\ \textit{and}\ \boldsymbol{x} \perp \boldsymbol{z}).\end{aligned}$$

また，スペクトルベクトルは必ず以下を満たすように選ぶことができる．

$$\boldsymbol{v}_1(\boldsymbol{x}) = \boldsymbol{v}_2(\boldsymbol{z})\ \textit{and}\ \boldsymbol{v}_2(\boldsymbol{x}) = \boldsymbol{v}_1(\boldsymbol{z}). \tag{2.5}$$

この命題より，二次錐相補性条件を満たすような $\boldsymbol{x}$ と $\boldsymbol{z}$ に対するスペクトル分解が次のように書き換えられることが分かる．

$$\boldsymbol{x} = \hat{x}_1\hat{\boldsymbol{e}}_1 + \hat{x}_2\hat{\boldsymbol{e}}_2, \quad \boldsymbol{z} = \hat{z}_1\hat{\boldsymbol{e}}_1 + \hat{z}_2\hat{\boldsymbol{e}}_2. \tag{2.6}$$

ここで，

$$\begin{gathered}\hat{x}_1 = \lambda_1(\boldsymbol{x})/\sqrt{2}, \quad \hat{x}_2 = \lambda_2(\boldsymbol{x})/\sqrt{2}, \quad \hat{z}_1 = \lambda_2(\boldsymbol{z})/\sqrt{2}, \quad \hat{z}_2 = \lambda_1(\boldsymbol{z})/\sqrt{2}\\ \hat{\boldsymbol{e}}_1 = \sqrt{2}\boldsymbol{v}_1(\boldsymbol{x}) = \sqrt{2}\boldsymbol{v}_2(\boldsymbol{z}), \quad \hat{\boldsymbol{e}}_2 = \sqrt{2}\boldsymbol{v}_2(\boldsymbol{x}) = \sqrt{2}\boldsymbol{v}_1(\boldsymbol{z}).\end{gathered}$$

である．さらに，以下が成り立つ．

(a) $\hat{\boldsymbol{e}}_1, \hat{\boldsymbol{e}}_2 \in \mathrm{bd}\,\mathcal{K}^n$, $\|\hat{\boldsymbol{e}}_1\| = \|\hat{\boldsymbol{e}}_2\| = 1$, $(\hat{\boldsymbol{e}}_1)^T\hat{\boldsymbol{e}}_2 = 0$, $\hat{\boldsymbol{e}}_1 + \hat{\boldsymbol{e}}_2 = (\sqrt{2}, \boldsymbol{0})^T$.

(b) $0 \leq \hat{x}_1 \leq \hat{x}_2$, $0 \leq \hat{z}_2 \leq \hat{z}_1$, $\min\{\hat{x}_1, \hat{z}_1\} = \min\{\hat{x}_2, \hat{z}_2\} = 0$.

上記の関係式 (b) と (2.6) は，普通の n 次元ベクトル空間における成分毎の相補性条件と正規直交系の関係によく似ている．実際，二つのベクトル $\boldsymbol{x}$ と $\boldsymbol{z}$ が普通の相補性条件：

$$\boldsymbol{x} \geq \boldsymbol{0},\ \boldsymbol{z} \geq \boldsymbol{0},\ \boldsymbol{x}^T\boldsymbol{z} = 0,$$

を満たすならば，すべての i に対して，$x_i \geq 0,\, z_i \geq 0,\, \min\{x_i, z_i\} = 0$ が成り立つ．このような場合，$\boldsymbol{x}$ と $\boldsymbol{z}$ は，i 番目の成分だけが 1 で他の成分が 0 であるようなベクトル $\boldsymbol{e}_i$ を用いて $\boldsymbol{x} = \sum_{i=1}^n x_i\boldsymbol{e}_i,\, \boldsymbol{z} = \sum_{i=1}^n z_i\boldsymbol{e}_i$ とできる．一方，(2.6) における $\hat{\boldsymbol{e}}_1$ と $\hat{\boldsymbol{e}}_2$ は，ベクトル $\boldsymbol{x}, \boldsymbol{z}$ および二次錐の軸 $\{(x_1, \tilde{\boldsymbol{x}}) \in \Re \times \Re^{n-1} \mid \tilde{\boldsymbol{x}} = \boldsymbol{0}\}$ を含む二次元部分空間に対する正規直交基底になっている．しかしながら，$\boldsymbol{e}_i$ は固定されたベクトルであるのに対して，$\hat{\boldsymbol{e}}_1$ と $\hat{\boldsymbol{e}}_2$ は $(\boldsymbol{x}, \boldsymbol{z})$ に依存することに注意する．

3 Explicit algorithm

本節では，SIP (1.3) を解くために，切除平面アプローチの一つである explicit algorithm を提案し，その収束定理を与える．

本研究で提案するアルゴリズムでは，各反復において部分問題として次のような有限個の制約をもつ SOCP を考える．有限個の要素をもつ集合 $E := \{t_1, \ldots, t_m\} \subset T$ に対して線形な SOCP（以降 LSOCP(E) と略す）を以下で定義する．

$$\begin{array}{ll} \text{Minimize} & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} & \boldsymbol{x} \in \mathcal{K}^n, \quad \boldsymbol{a}(t_j)^T\boldsymbol{x} - b(t_j) \geq 0 \quad (j = 1, \ldots, m). \end{array}$$

このとき，双対問題 DLSOCP(E) は以下のように書ける．

$$\begin{array}{ll} \text{Maximize} & \displaystyle\sum_{j=1}^m b(t_j)\nu(t_j) \\ \text{subject to} & \nu(t_j) \geq 0, \quad \boldsymbol{c} - \displaystyle\sum_{j=1}^m \boldsymbol{a}(t_j)\nu(t_j) \in \mathcal{K}^n. \end{array}$$

ここで双対変数 $\{\nu(t_j)\}_{j=1}^m$ は m 次元の実ベクトルと見なせることに注意する．さらに，主問題と双対問題のそれぞれの最適値を $V(\mathrm{LSOCP}(E))$ および $V(\mathrm{DLSOCP}(E))$ と表す．

SIP (1.3) を解く切除平面アプローチの一つとして，まず implicit algorithm が考えられる．このアルゴリズムは次のような手順で解を求める． (1) 最初に有限集合 $E^0 \subset T$ を選ぶ．(2) 各反復で LSOCP(E^k) を解く．(3) $t^k_{\mathrm{new}} := \mathrm{argmin}_{t\in T}\{\boldsymbol{a}(t)^T\boldsymbol{x}^k - b(t)\}$ とし，E^{k+1} を $E^{k+1} := E^k \cup \{t^k_{\mathrm{new}}\}$ となるように更新．(4) 適当な終了条件を満たせば反復を終了．さもなくば，$k := k + 1$ として (2) に戻る． 実際，implicit algorithm は直感的に理解しやすく収束解析も比較的容易であるが，反復回数 k が大きくなればなるほど各部分問題の制約の数が増えていくため，部分問題を解くのにかかる計算コストが大きくなってしまう．このような欠点を避けるため，本研究では，[15] のアプローチに基いた explicit algorithm を提案する．このアルゴリズムでは，各反復において有効でない制約を取り除いて行くため，部分問題の制約の数をある程度以下に抑えることができる．より詳細なステップは以下のようになる．

アルゴリズム 3.1

Step 0 有限個の点 $E^0 := \{t_1^0, \ldots, t_{m_0}^0\} \subset T$ を選ぶ．また，LSOCP(E^0) を解き，最適解 $\boldsymbol{x}^0$ を得る．$k := 0$ とする．

Step 1 もし，$\min_{t \in T}\{\boldsymbol{a}(t)^T \boldsymbol{x}^k - b(t)\} \geq 0$ であるならば，反復を終了する．そうでなければ，次のように値を更新する．

$$t_{\text{new}}^k := \operatorname*{argmin}_{t \in T} \left\{\boldsymbol{a}(t)^T \boldsymbol{x}^k - b(t)\right\} \quad \textit{and} \quad \overline{E}^{k+1} := E^k \cup \{t_{\text{new}}^k\}.$$

Step 2 LSOCP($\overline{E}^{k+1}$) と DLSOCP($\overline{E}^{k+1}$) を解き，最適解 $\boldsymbol{x}^{k+1}$ および $\{\nu_{k+1}(t) \mid t \in \overline{E}^{k+1}\}$ を得る．

Step 3

$$E^{k+1} := \{t \in \overline{E}^{k+1} \mid \nu_{k+1}(t) > 0\}$$

とする．$k := k+1$ とおき *Step* 1 に戻る．

Step 2 では LSOCP($\overline{E}^{k+1}$) と DLSOCP($\overline{E}^{k+1}$) は既存の手法を用いて同時に解くことができる．さらに，$\boldsymbol{x}^{k+1}$ は LSOCP(E^{k+1}) の解にもなっていることに注意する．Step 3 では，相補スラック条件より $\nu_{k+1}(t) > 0$ であるような任意の t に対して $\boldsymbol{a}(t)^T \boldsymbol{x} - b(t) = 0$ であるため，$\boldsymbol{x}^k$ における非有効制約が取り除かれている．このステップが implicit algorithm との最も大きな違いである．実際，Step 3 における更新規則が $E^{k+1} := \overline{E}^{k+1}$ で置き換えられたら，上のアルゴリズムは implicit algorithm に他ならない．

アルゴリズムの収束を論ずる前に，いくつか記号を定義する．$\{\boldsymbol{x}^k\}$ と $\{\nu_k\}$ をアルゴリズム 3.1 で生成される点列とする．このとき，

$$Z^k := \{t \in \overline{E}^k \mid \nu_k(t) = 0\}, \;\; y^k(t) := \boldsymbol{a}(t)^T \boldsymbol{x}^k - b(t), \;\; \boldsymbol{z}^k := \boldsymbol{c} - \sum_{t \in E^k} \boldsymbol{a}(t)\nu_k(t)$$

とおく．明らかに $E^k \cup Z^k = \overline{E}^k$ と $E^k \cap Z^k = \emptyset$ が成り立つ．さらに，相補スラック条件より以下が従う．

$$\begin{aligned} &\nu_k(t) \geq 0, \quad y^k(t) \geq 0, \quad \nu^k(t)\, y^k(t) = 0, \qquad (\forall t \in \overline{E}^k) \\ &\boldsymbol{x}^k \in \mathcal{K}^n, \quad \boldsymbol{z}^k \in \mathcal{K}^n, \quad (\boldsymbol{x}^k)^T \boldsymbol{z}^k = 0. \end{aligned}$$

そこで，各反復点 $\boldsymbol{x}^k$ および $\boldsymbol{z}^k$ を (2.6) と同様の方法で次のように分解する．

$$\boldsymbol{x}^k = \hat{x}_1^k \hat{\boldsymbol{e}}_1^k + \hat{x}_2^k \hat{\boldsymbol{e}}_2^k, \quad \boldsymbol{z}^k = \hat{z}_1^k \hat{\boldsymbol{e}}_1^k + \hat{z}_2^k \hat{\boldsymbol{e}}_2^k.$$

一般的に $(\hat{\boldsymbol{e}}_1^k, \hat{\boldsymbol{e}}_2^k) \neq (\hat{\boldsymbol{e}}_1^{k+1}, \hat{\boldsymbol{e}}_2^{k+1})$ であることに注意する．さらに，以下のような添字集合を定義する．

$$U_+^k := \{i \in \{1,2\} \mid \hat{x}_i^k > 0\}, \quad U_0^k := \{i \in \{1,2\} \mid \hat{x}_i^k = 0\}.$$

各 $i = 1, 2$ に対して，$\min\{\hat{x}_i^k, \hat{z}_i^k\} = 0$ であることから，$i \in U_+^k$ ならば $\hat{z}_i^k = 0$ であることに注意する．また，$\hat{x}_1^k \leq \hat{x}_2^k$ であるので，以下の 3 つのケースしか存在しないことが分かる．

$$\text{(i) } U_0^k = \emptyset,\, U_+^k = \{1,2\}, \;\text{(ii) } U_0^k = \{1\},\, U_+^k = \{2\}, \;\text{(iii) } U_0^k = \{1,2\},\, U_+^k = \emptyset.$$

次にアルゴリズム 3.1 の収束を証明するのに必要な仮定と補題を与える．まず，以下の仮定を考える．

仮定 A すべての k に対して，(i) LSOCP$(\overline{E}^k)$ および DLSOCP$(\overline{E}^k)$ は唯一の解をもち，(ii) $V(\mathrm{LSOCP}(\overline{E}^k)) = V(\mathrm{DLSOCP}(\overline{E}^k))$ を満たす．

LSOCP$(\overline{E}^k)$ および DLSOCP$(\overline{E}^k)$ の実行可能領域は，多面体と二次錐（もしくはそのアフィン変換）との共通集合であるため，(i) は通常成り立つ．また，(ii) は言うまでもなく強双対性のことであり，これも多くの場合で成立する．以上の仮定の下で，次の補題が成り立つ．

補題 3.1 仮定 A が成り立つとする．このとき，すべての $k \geq 0$ に対して以下が従う．

(a) $V(\mathrm{LSOCP}(E^k)) < V(\mathrm{LSOCP}(E^{k+1}))$,

(b) $t^k_{\mathrm{new}} \in E^{k+1}$.

ここで，(a) は部分問題の最適値が単調増加していることにほかならないが，すべての k に対して $V(\mathrm{LSOCP}(E^k))$ は SIP (1.3) の最適値 $V(\mathrm{SIP})$ 以下であるため，以下の関係が成り立つ．

$$V_\infty := \lim_{k\to\infty} V(\mathrm{LSOCP}(E^k)) \leq V(\mathrm{SIP}). \tag{3.1}$$

また，(b) は $t = t^k_{\mathrm{new}}$ に対応する不等式制約が $\boldsymbol{x}^k$ において有効であることを意味している．次の補題は，部分問題の最適値のステップ毎の増分を評価したものである．

補題 3.2 仮定 A が成立するとする．このとき，以下の等式が成り立つ．

$$\begin{aligned} V(\mathrm{LSOCP}(E^{k+1})) - V(\mathrm{LSOCP}(E^k)) &= \sum_{t\in Z^{k+1}} y^{k+1}(t)\nu_k(t) + (\boldsymbol{z}^k)^T \boldsymbol{x}^{k+1} \\ &= -(\boldsymbol{z}^{k+1})^T \boldsymbol{x}^k - y^k(t^k_{\mathrm{new}})\nu_{k+1}(t^k_{\mathrm{new}}). \end{aligned}$$

次に，以下のような仮定をさらに導入する．

仮定 B ある十分大きい数 $M > 0$ および十分小さい数 $\delta > 0$ が存在し，任意の $k \geq 1$ に対して以下が成り立つ．

(a) $\|\boldsymbol{x}^k\| \leq M$, $\|\boldsymbol{z}^k\| \leq M$.

(b) 任意の $t \in E^k$ に対して $\nu_k(t) \geq \delta$ である．

(c) 各 $i = 1, 2$ に対して $\delta \leq \max\{\hat{x}^k_i, \hat{z}^k_i\} \leq M$ が成り立つ．

(d) もし $\boldsymbol{x}^k \in \mathrm{bd}\,\mathcal{K}^n$ かつ $\boldsymbol{x}^k \neq \boldsymbol{0}$ ならば，ある $t \in E^k$ が存在して $\boldsymbol{a}(t)^T \hat{\boldsymbol{e}}^k_2 \leq -\delta$ もしくは $\boldsymbol{a}(t)^T \hat{\boldsymbol{e}}^k_2 > 0$ が成り立つ．

(e) もし $\boldsymbol{x}^k \in \mathrm{int}\,\mathcal{K}^n$ ならば，$\lambda_{\min}(H_k H_k^T) \geq \delta$ が成り立つ．ここで，$\lambda_{\min}$ は最小固有値を意味し，行列 H_k は次のように定義されるものとする．

$$H_k := \Big(\boldsymbol{a}(t^k_1), \ldots, \boldsymbol{a}(t^k_{m_k})\Big) \in \Re^{n\times m_k} \tag{3.2}$$

ただし，$E^k = \{t^k_1, \ldots, t^k_{m_k}\}$ である．

仮定 B (a) は生成された点列が有界であることを，仮定 B (b) および (c) は各部分問題の解に対して相補性条件が十分狭義に成り立つことを意味している．さらに，仮定 B (c) および $\min\{\hat{x}_i^k, \hat{z}_i^k\} = 0$ より以下が成り立つことに注意する．

$$
\begin{aligned}
i \in U_0^k &\iff \hat{x}_i^k = 0,\ \hat{z}_i^k \geq \delta, \\
i \in U_+^k &\iff \hat{x}_i^k \geq \delta,\ \hat{z}_i^k = 0.
\end{aligned}
$$

仮定 B (d) および (e) はある種の正則性を主張しており，多くの場合でこれらの仮定は成立する．次の補題は，仮定 B がさらに成り立つとき，添字集合 U_+^k および U_0^k が十分大きな任意の k に対して不変であることを主張している．

補題 3.3 アルゴリズム 3.1 で生成される点列に対して仮定 A および B が成立するとする．このとき，ある自然数 $\bar{k}$ が存在して，任意の $k \geq \bar{k}$ に対して以下を満たす．

$$
U_+^k = U_+^{k+1}, \quad U_0^k = U_0^{k+1}.
$$

この補題を用いて，本節で最も重要な定理を導く．

定理 3.1 仮定 A および B が成立するとする．このとき．アルゴリズム 3.1 で生成される点列 $\{\boldsymbol{x}^k\}$ の任意の集積点は SIP (1.3) の解となる．

式 (3.1) に注意すると，この定理は各部分問題の最適値の列 $\{V(\mathrm{LSOCP}(E^k))\}$ が SIP (1.3) の最適値 $V(\mathrm{SIP})$ に収束することを含んでいることが分かる．

4 数値実験

本節では，数値実験の結果をいくつか報告する．アルゴリズム 3.1 を計算機に実装するにあたって，パラメータを次のように定めた．Step 0 では E^0 の要素数は $n+1$ とした．また，各要素は集合 T の中からランダムに選んだ．Step 1 では t_{new}^k を二分割法とニュートン法を組み合わせることにより求めた．また，終了条件は $\min_{t \in T}\{\boldsymbol{a}(t)^T \boldsymbol{x}^k - b(t)\} \geq -10^{-8}$ と緩和した．Step 2 では $\mathrm{LSOCP}(\overline{E}^{k+1})$ と $\mathrm{DLSOCP}(\overline{E}^{k+1})$ を [11] で提案されている SOCCP に対するアルゴリズムを用いて解いた．Step 3 では条件 $\nu_{k+1}(t) > 0$ を $\nu_{k+1} > 10^{-8}$ に緩和した．また，アルゴリズムは MATLAB 7.0 で記述し，Intel(R) Xeon(TM) CPU 3.60GHz および 2GB RAM のスペックをもつ計算機上で動かした．

最初の実験では，SIP (1.3) に対して，

$$
\boldsymbol{a}(t) := \begin{pmatrix} -(2t-1.13)^2 - 1.03 \\ -(2t-0.98)^3 \\ (2t-1.05)^2 - 0.9 \end{pmatrix}, \quad b(t) := -(2t-1.08)^2 - 1.1, \tag{4.1}
$$

$T = [0,1]$ とした．さらに，目的関数におけるベクトル $\boldsymbol{c}$ の候補として，$\boldsymbol{c}_1 = (1,0,0)^T$, $\boldsymbol{c}_2 = (-0.88, 0.23, -0.98)^T$, $\boldsymbol{c}_3 = (-0.79, -0.35, -0.03)^T$ の 3 つを選んだ．得られた結果を表 1 に示す．ここで，$\lambda_1(\boldsymbol{x}^*)$ および $\lambda_2(\boldsymbol{x}^*)$ は最適解 $\boldsymbol{x}^*$ におけるスペクトル値を，$\sharp$ite は反

表 1: Obtained results for SIP with (4.1)

$\boldsymbol{c}$	$\lambda_1(\boldsymbol{x}^*)$	$\lambda_2(\boldsymbol{x}^*)$	$\sharp$ite	cpu(s)
$\boldsymbol{c}_1$	0	0	0	0.010
$\boldsymbol{c}_2$	0	1.495	2.45	0.209
$\boldsymbol{c}_3$	0.900	1.139	9.94	1.010

復回数を，cpu(s) は CPU 時間を表す．なお，$\sharp$ite と cpu(s) の値は，各 $\boldsymbol{c}_i$ $(i = 1, 2, 3)$ に対して，集合 E^0 の要素と部分問題を解くために用いた初期点をランダムに変えて行った 100 回の試行に対する平均値である．

$\boldsymbol{c} = \boldsymbol{c}_1$ のときは最適解 $\boldsymbol{x}^*$ が原点と一致し，任意の $t \in T$ に対して $\boldsymbol{a}(t)^T\boldsymbol{x}^* - b(t) > 0$ である．さらに，100 回すべての試行において，最初に集合 E^0 を選んだ時点で終了条件が満たされてしまうことが分かる．$\boldsymbol{c} = \boldsymbol{c}_2$ のときは，$\boldsymbol{x}^* = (0.747, -0.654, 0.361)^T$ であり，スペクトル値は $0 = \lambda_1(\boldsymbol{x}^*) < \lambda_2(\boldsymbol{x}^*)$ を満たす．すなわち，最適解 $\boldsymbol{x}^*$ は二次錐の境界に位置する．さらに，$\boldsymbol{x}^*$ において不等式制約 $\boldsymbol{a}(t)^T\boldsymbol{x} - b(t) \geq 0$ は幾つかの $t \in T$ で等式を満たす，すなわち有効である．また，反復回数や CPU 時間が比較的少ないことも見てとれる．$\boldsymbol{c} = \boldsymbol{c}_3$ のときは，最適解は $\boldsymbol{x}^* = (1.019, 0.118, -0.020)^T$ であり，スペクトル値は $0 < \lambda_1(\boldsymbol{x}^*) < \lambda_2(\boldsymbol{x}^*)$ である．このことは，$\boldsymbol{x}^*$ が二次錐の内部に位置し，二次錐制約 $\boldsymbol{x} \in \mathcal{K}^n$ が効いていないことを意味する．この場合，反復回数や CPU 時間が $\boldsymbol{c} = \boldsymbol{c}_2$ の場合に比べてかなり大きくなっている．

次の実験では，以下の二つの SIP を解いた．

$$\begin{aligned} &\text{Minimize} && \sum_{i=1}^{7} \frac{x_i}{i} \\ &\text{subject to} && \boldsymbol{x} \in \mathcal{K}^7, \ \ \sum_{i=1}^{7} t^{i-1}x_i \geq \sum_{i=0}^{4} t^{2i} \quad (\forall\, t \in [0, 1]). \end{aligned} \tag{4.2}$$

$$\begin{aligned} &\text{Minimize} && h \\ &\text{subject to} && \begin{pmatrix} h \\ \boldsymbol{x} \end{pmatrix} \in \mathcal{K}^8, \ \ h \geq \left| \sum_{i=1}^{7} t^{i-1}x_i - \sin\left(\frac{5\pi t}{6}\right) \right| \quad (\forall\, t \in [0, 1]). \end{aligned} \tag{4.3}$$

なお，問題 (4.3) は関数 $f(\boldsymbol{x}) := \max_{t\in[0,1]}\{\|\boldsymbol{x}\|, |\sum_{i=1}^{7} t^{i-1}x_i - \sin(5\pi t/6)|\}$ に対する制約無し最適化問題と等価であることに注意されたい．実際，問題 (4.3) は SIP (1.3) の形にはなっていないが，$\boldsymbol{c} = (1, 0, \ldots, 0)^T \in \Re^8$, $T = [0, 1] \cup [2, 3]$,

$$\boldsymbol{a}(t) := \begin{cases} \left(1,\, 1,\, t,\, t^2, \ldots,\, t^6\right)^T & \text{if } t \in [0, 1] \\ \left(1, -1, -(t-2), -(t-2)^2, \ldots, -(t-2)^6\right)^T & \text{if } t \in [2, 3] \\ \boldsymbol{0} & \text{otherwise}, \end{cases}$$

$$b(t) := \begin{cases} \sin\left(\dfrac{5\pi t}{6}\right) & \text{if } t \in [0, 1] \\ -\sin\left(\dfrac{5\pi(t-2)}{6}\right) & \text{if } t \in [2, 3] \\ -\infty & \text{otherwise}. \end{cases}$$

とすることにより SIP (1.3) に帰着できる．得られた結果を表 2 に示す．各列の表記は表 1 と同様である．まず，すべての試行に対して，SIP (4.2) が 1 回の反復のみで解を得ていること

表 2: Obtained results for SIPs (4.2) and (4.3)

SIP	$\lambda_1(\boldsymbol{x}^*)$	$\lambda_2(\boldsymbol{x}^*)$	$\sharp$ite	cpu(s)
(4.2)	0	3.275	1	0.259
(4.3)	0	0.903	4.09	0.603

が観察できる．実際，解 $\boldsymbol{x}^*$ において集合 $E^* := \{t \in T \mid \boldsymbol{a}(t)^T\boldsymbol{x}^* - b(t) = 0\}$ は $E^* = \{1\}$ となるが，これが T の端点であることが1回の反復で解が得られる大きな原因ではないかと予想される．一方，SIP (4.3) に対しては，反復回数が4～5回となり，これは最初の有限集合 E^0 に依存する．さらに，解 $\boldsymbol{x}^*$ において $E^* = \{0.540\}$ となり，これは T の端点ではない．

5　まとめと今後の課題

本論文では，二次錐制約と線形関数を含む半無限計画問題に対して切除平面アプローチの一つである explicit algorithm を提案した．さらに，その収束性を適当な仮定の下で示した．特に，収束解析にあたって，スペクトル分解に基づいたある種の座標系を導入した．また，数値実験では，提案したアルゴリズムが効率的に解を見つけることが分かった．今後の課題として，提案手法を二次錐の直積制約をもつ SIP へ拡張することが考えられる．実際，本論文では $\boldsymbol{x} \in \mathcal{K}^n$ という制約を考えたが，実際の状況では，$\boldsymbol{x} \in \mathcal{K}^{n_1} \times \mathcal{K}^{n_2} \times \cdots \times \mathcal{K}^{n_m}$ といった直積制約を考慮することがしばしば必要となってくる．その場合，アルゴリズム自体は同様に構築できるが，その解析はより複雑になることが予想される．

参考文献

[1] Alizadeh, F. and Goldfarb, D.: Second-order cone programming, *Mathematical Programming*, Vol. 95 (2003), 3–51.

[2] Anderson, E. J. and Wu, S.-Y.: The continuous complementarity problem, *Optimization*, Vol. 22 (1991), 419–426.

[3] Chen, J.-S., Chen, X. and Tseng, P.: Analysis of nonsmooth vector-valued functions associated with second-order cones, *Mathematical Programming*, Vol. 101 (2004), 95–117.

[4] Coope, I. D. and Price, C. J.: Exact penalty function methods for nonlinear semi-infinite programming, in Reemtsen, R. and Rückmann, J. eds., *Semi-Inifinte Programming*, Kluwer Academic Publishers, Boston, 1998, 137–157.

[5] Cottle, R. W., Pang, J.-S. and Stone, R. E.: *The Linear Complementarity Problem*, Academic Press, San Diego, 1992.

[6] Facchinei, F. and Pang, J.-S.: *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Springer-Verlag, New York, 2003.

[7] Faraut, J. and Korányi, A.: *Analysis on Symmetric Cones*, Clarendon Press, New York, 1994.

[8] Fukushima, M., Luo, Z.-Q. and Tseng, P.: Smoothing functions for second-order cone complementarity problems, *SIAM Journal on Optimization*, Vol. 12 (2001), 436–460.

[9] Gramlich, G., Hettich, R. and Sachs, E.: SQP-methods in semi-infinite programming, *SIAM Journal on Optimization*, Vol. 5 (1995), 641–658.

[10] 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*, Vol. 175 (2005), 335–353.

[11] Hayashi, S., Yamashita, N. and Fukushima, M.: A combined smoothing and regularization method for monotone second-order cone complementarity problems, *SIAM Journal on Optimization*, Vol. 15 (2005), 593–615.

[12] Hettich, R. and Jongen, H. T.: Semi-infinite programming: conditions of optimality and applications, *Lecture Notes in Control and Information Sciences*, Vol. 7 (1978), 1–11.

[13] Hettich, R. and Kortanek, K. O.: Semi-infinite programming: theory, methods, and applications, *SIAM Review*, Vol. 35 (1993), 380–429.

[14] Lai, H. C. and Wu, S.-Y.: Extremal points and optimal solutions for general capacity problems, *Mathematical Programming*, Vol. 54 (1992), 87–113.

[15] Lai, H. C. and Wu, S.-Y.: On linear semi-infinite programming problems: an algorithm, *Numerical Functional Analysis and Optimization*, Vol. 13 (1992), 287–304.

[16] Lawrence, C. T. and Tits, A. L.: Feasible sequential quadratic programming for finely discretized problems from SIP, in Reemtsen, R. and Rückmann, J. eds., *Semi-Inifinte Programming*, Kluwer Academic Publishers, Boston, 1998, 159–193.

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

[18] Muramatsu, M.: A Pivoting Procedure for a Class of Second-Order Cone Programming, *Optimization Methods and Software*, Vol. 21 (2006), 295–314.

[19] Reemsten, R. and Rückmann, J.: *Semi-Infinite Programming*, Kluwer Academic Publishers, Boston, 1998.

[20] Reemtsen, R. and Görner, S.: Numerical methods for semi-infinite programming: a survey, in Reemtsen, R. and Rückmann, J. eds., *Semi-Inifinte Programming*, Kluwer Academic Publishers, Boston, 1998, 195–275.

[21] Tanaka, Y., Fukushima, M. and Ibaraki, T.: A globally convergent SQP method for semi-infinite nonlinear optimization, *Journal of Computational and Applied Mathematics*, Vol. 23 (1988), 141–153.

[22] Teo, K. L., Yang, X. Q. and Jennings, L. S.: Computational discretization algorithms for functional inequality constrained optimization, *Annals of Operations Research*, Vol. 98 (2000), 215–234.

[23] Wu, S.-Y., Fang, S. C. and Lin, C. J.: Relaxed cutting plane method for solving semi-infinite programming problems, *Journal of Optimization Theory and Applications*, Vol. 99 (1998), 759–779.

[24] Wu, S.-Y., Li, D. H., Qi, L. and Zhou, G.: An Iterative Method for Solving KKT System of the Semi-Infinite Programming, *Optimization Methods and Software*, Vol. 20 (2005), 629–643.

[25] Yamashita, H. and Yabe, H.: A primal-dual interior point method for nonlinear optimization over second order cones, manuscript, Mathematical Systems Inc., 2005.