数理解析研究所講究録
第 1544 巻 2007 年 77-84

二次錐制約をもつ半無限計画問題の解法

京都大学・情報学研究科　林 俊介 (Shunsuke Hayashi)
Graduate School of Informatics, Kyoto University

国立成功大学（台湾）・数学系　呉 順益 (Soon-Yi Wu)
Department of Mathematics, National Cheng Kung University

1 はじめに

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を半無限計画問題 (Semi-infinite programming problem: SIP) [2, 8, 9, 10, 12] といい，以下の形で定式化できる．

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

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 の要素が有限個で，$X = \Re^n$ もしくは $X = \Re^n_+$ であれば，問題 (1.1) は普通の有限制約の非線形計画問題である．

本稿では，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．

$$
\begin{aligned}
&\text{Minimize} \quad \boldsymbol{c}^T \boldsymbol{x} \\
&\text{subject to} \quad \boldsymbol{x} \in \mathcal{K}^n, \quad \boldsymbol{a}(t)^T \boldsymbol{x} - b(t) \geq 0 \ \ (\forall t \in T),
\end{aligned} \tag{1.2}
$$

ここで，$\boldsymbol{c} \in \Re^n$ は与えられた定数ベクトル，$b : T \to \Re$, $\boldsymbol{a} : T \to \Re^n$ は任意の $x \in \Re^n$ に対して $\min_{t \in T}\{\boldsymbol{a}(t)^T \boldsymbol{x} - b(t)\}$ が存在するような与えられた関数である．また，$\mathcal{K}^n \subset \Re^n$ は以下で定義される n 次元の二次錐 (Second-order cone: SOC) である．

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

ここで，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．また，これ以後 $\|\cdot\|$ はユークリッドノルムを表すものとする．SIP (1.2) 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．

近年，二次錐を含む問題に対する研究が盛んになされている．その中でも最もよく知られているのが二次錐計画問題 (Second-order cone programming: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, 11]．また，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 [3, 4] を二次錐制約へと拡張した問題である．文献 [6] 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，[6] 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 [7] 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．なお，(1.2) のような二次錐制約を含む半無限計画問題の応用として，二次錐制約を含むロバスト最適化などが考えられる．

本研究の目的は，二次錐制約を含む SIP (1.2) 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 [10] 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，[10] 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 (1.2) 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．

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 [5] 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．

二次元以上の任意のベクトル $\boldsymbol{x} := (x_1, \tilde{\boldsymbol{x}}) \in \Re \times \Re^{n-1}$ 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 $\boldsymbol{v}_1(\boldsymbol{x})$, $\boldsymbol{v}_2(\boldsymbol{x}) \in \Re^n$ は以下で定義される．

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

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

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

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．

2.2 二次錐相補性条件

本節では，後述するアルゴリズムの解析に重要な役割を果たす二次錐相補性条件を説明する．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．

$$\boldsymbol{x} \in \mathcal{K}^n,\ \boldsymbol{z} \in \mathcal{K}^n,\ \boldsymbol{x}^T\boldsymbol{z} = 0. \qquad (2.2)$$

このような性質を満たすベクトルの組に対して以下が成り立つ．

命題 2.1 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 (2.2) 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．

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

また，各々のスペクトルベクトルは以下の性質を満たす[1]．

$$\boldsymbol{v}_1(\boldsymbol{x})=\boldsymbol{v}_2(\boldsymbol{z}) \ \textit{and} \ \boldsymbol{v}_2(\boldsymbol{x})=\boldsymbol{v}_1(\boldsymbol{z}). \tag{2.3}$$

式 (2.3) より，$\boldsymbol{x}$ と $\boldsymbol{z}$ が二次錐相補性条件を満たすならば，

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

とおくことができ，それぞれ以下のように書き換えることができる．

$$\boldsymbol{x}=\hat{x}_1\hat{\boldsymbol{e}}_1+\hat{x}_2\hat{\boldsymbol{e}}_2, \quad \boldsymbol{z}=\hat{z}_1\hat{\boldsymbol{e}}_1+\hat{z}_2\hat{\boldsymbol{e}}_2. \tag{2.4}$$

さらに，以下が成り立つ．

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

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

上記の関係式 (b) と (2.4) 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：

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

を満たすならば，すべての i に対して，$x_i\ge0$, $z_i\ge0$, $\min\{x_i,z_i\}=0$ 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，(2.4) 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 $\{(x_1,\tilde{\boldsymbol{x}})\in\Re\times\Re^{n-1}\,|\,\tilde{\boldsymbol{x}}=\boldsymbol{0}\}$ 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．

3 アルゴリズムと収束性

本節では，SIP (1.2) を解くための手法を提案し，その収束定理を与える．そのために，まず SIP (1.2) に対する緩和問題を定義する．集合 T から有限個の要素 $t_1,\ldots,t_m$ を取ってきて

[1] $\tilde{\boldsymbol{x}}=\boldsymbol{0}$ もしくは $\tilde{\boldsymbol{z}}=\boldsymbol{0}$ のときは $\boldsymbol{v}_i$ $(i=1,2)$ の定義において w を適当に選べばよい．

それを E とする．(i.e., $E := \{t_1, \ldots, t_m\} \subset T$.) そのとき，緩和問題 LSOCP($E$) を以下で定義する．

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

この問題は有限個の制約をもつ線形な SOCP なので，既存の手法で解くことができる．また，制約の数が減少しているため，制約領域は元の SIP (1.2) よりも大きくなっている．LSOCP(E) の双対問題 DLSOCP(E) は以下のように書ける．

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

上記の緩和問題を用いて，以下のようなアルゴリズムを提案する．なお，この手法では，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有限の値に抑えることができるのが最も大きな特長の一つである．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 $\{\nu_{k+1}(t) \,|\, t \in \overline{E}^{k+1}\}$ を得る．

Step 3

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

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，後述する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 $\boldsymbol{a}(t)^T\boldsymbol{x}^{k+1} - b(t) = 0$ であるため，$\boldsymbol{x}^{k+1}$ における非有効制約が取り除かれている．

アルゴリズムの収束を示す前に，仮定と記号の定義をいくつか導入する．まず，次の仮定が成り立つものとする．

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 $\boldsymbol{x}^k, \nu_k$ をもち，(ii) 両者の最適値は等しい．

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．このとき，以下の相補スラック条件が成り立つ．

$$\begin{aligned} &\nu_k(t) \geq 0, \quad y^k(t) \geq 0, \quad \nu^k(t)\, y^k(t) = 0, \quad (\forall t \in \overline{E}^k) \\ &\boldsymbol{x}^k \in \mathcal{K}^n, \quad \boldsymbol{z}^k \in \mathcal{K}^n, \quad (\boldsymbol{x}^k)^T \boldsymbol{z}^k = 0. \end{aligned} \tag{3.1}$$

ただし，

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

である．ところで，(3.1) は二次錐相補性条件にほかならないので，(2.4) と同様の方法で分解することができる．

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

ただし，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．さらに，次のような仮定を考える．

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 $k \geq 1$ に対して以下の (a)–(e) 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 $\delta \leq \max\{\hat{x}_i^k, \hat{z}_i^k\} \leq M$ 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，$\boldsymbol{a}(t)^T \hat{\boldsymbol{e}}_2^k \notin (-\delta, 0]$ となるような $t \in E^k$ が存在する．

(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 $E^k := \{t_1^k, \ldots, t_{m_k}^k\}$ を用いて次のように定義されるものとする．

$$H_k := \left(\boldsymbol{a}(t_1^k), \ldots, \boldsymbol{a}(t_{m_k}^k)\right) \in \Re^{n \times m_k}. \tag{3.2}$$

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．

$$\hat{x}_i^k = 0 \Longleftrightarrow \hat{z}_i^k \geq \delta, \quad \hat{x}_i^k \geq \delta \Longleftrightarrow \hat{z}_i^k = 0.$$

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は成立する．以上の仮定の下で次の定理を得る．

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 (1.2) の解となる．

本定理より，以下の系が成り立つことが容易に分かる．

系 3.1 仮定 A および B が成立するとする．また，SIP (1.2) の解が唯一存在するとする．このとき．アルゴリズム 3.1 で生成される点列 $\{\boldsymbol{x}^k\}$ は SIP (1.2) の解に収束する．

4 数値実験

本節では，提案したアルゴリズムの効能を確かめるために行った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^k_{new} 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 LSOCP($\overline{E}^{k+1}$) と DLSOCP($\overline{E}^{k+1}$) を [7] 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 のスペック上で動かした．

最初の実験では，SIP (1.2) 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，♯ite は反復回数を，cpu(s) は CPU 時間を表す．なお，♯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．

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

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

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．実際，問題 (4.3) は SIP (1.2) の形にはなっていないが，$\boldsymbol{c} =$

表 1: Obtained results for SIP with (4.1)

c	$\lambda_1(\boldsymbol{x}^*)$	$\lambda_2(\boldsymbol{x}^*)$	♯ite	cpu(s)
c_1	0	0	0	0.010
c_2	0	1.495	2.45	0.209
c_3	0.900	1.139	9.94	1.010

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

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

$(1,0,\ldots,0)^T \in \Re^8$, $T = [0,1] \cup [2,3]$,

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

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

とすることにより SIP (1.2) に帰着できる．得られた結果を表 2 に示す．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，これは T の端点である．一方，SIP (4.3) に対しては，反復回数が 4 回もしくは 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

本研究では，二次錐制約と線形関数を含む半無限計画問題に対して Lai and Wu の explicit algorithm を拡張した手法を提案し，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座標表記が重要な役割を果たした．また，提案したアルゴリズムが効率的に解を見つけることを数値実験で示した．今後の課題としては，提案手法を二次錐の直積制約を含む SIP など，より一般的な形のものへ拡張することや，収束率の解析などが挙げられる．

参考文献

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

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

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

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

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

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

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

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

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

[10] ——, *On linear semi-infinite programming problems: an algorithm*, Numerical Functional Analysis and Optimization, 13 (1992), pp. 287–304.

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

[12] S.-Y. Wu, D. H. Li, L. Qi, and G. Zhou, *An iterative method for solving KKT system of the semi-infinite programming*, Optimization Methods and Software, 20 (2005), pp. 629–643.