最適化 : 半正定値計画を中心に

脇, 隼人
九州大学マス・フォア・インダストリ研究所

出版情報：COE Lecture Note. 46, pp.183-192, 2013-02-28. 九州大学マス・フォア・インダストリ研究所

最適化—半正定値計画を中心に—

脇 隼人

九州大学マス・フォア・インダストリ研究所

1 はじめに

最適化とは, 与えられた条件・制約のもとで, 与えられた関数を最大化または最小化する解やそのときの値を求めることである. 最適化は産業や日常生活の様々な場面で登場する. 金融ではリスク制約のもとで利益を最大化するために, 物流ではコストを最小にする車両配置や在庫管理のために, 製造業ではより良い設計パラメタを求めるため用いられる. 最近さかんにいわれているスマートグリッドでも, 再生エネルギーを効率的に利用するためには最適化が必要となる. また, 大学でも教員に対する授業や教室の割当, 学生の研究室配属などで最適化が利用できる.

最適化問題によって様々なアルゴリズムが提案されている. したがって, 解くべき最適化問題がどのような問題か知ることで, 適切なアルゴリズムやソフトウェアを選択し解を得ることができる. 現実の場面では最適化を行う際, 与えられたデータや現場の状況から最適化問題を作る必要がある. これをモデル化, またはモデリングと呼ぶ. その後, 得られた最適化問題を適切なアルゴリズムで解く, という順になる.

本稿では, その中でも最先端の最適化問題である半正定値計画問題に焦点を合わせて解説する. 半正定値計画問題は 21 世紀の線形計画問題と呼ばれ, 今後も幅広い分野で出現することが期待される最適化問題の一つである.

2 半正定値計画

本節では半正定値計画 (以下 SDP と省略する[1]) について簡単に紹介する. 日本語で紹介されている文献として [3, 6, 17] があげられる. 興味のある読者はこちらも参照してほしい.

SDP は線形計画 (以下 LP と省略する[2]) の行列版ということができる. 実際 SDP は LP の拡張である. 一方で, SDP は非線形な構造を有している, という点が決定的に LP と異なる. 実際, SDP の制約式が構成する領域は曲がった構造をしているのに対して, LP は多面体となっている.

SDP は行列の最適化問題であり, 多くの最適化問題を記述することが可能である. 制御や構造最適化の分野で現れる問題を SDP として記述できる場合がある. 統計分野で現れる相関行

[1] SemiDefinite Programming の略
[2] Linear Programming の略

列を推定する問題や, 最大カット問題などの NP 困難である組合せ最適化問題に対する近似解法で利用されている. 金融や機械学習, 計算量理論の解析でも SDP が現れることがあり, 今後も新しい分野で SDP を利用した研究が行われることが予想される.

LP と同じ様に, より複雑な問題 (例えば 非凸二次最適化問題や多項式最適化問題などの NP 困難な問題) を効率よく解くための手段として用いられることもある. 本稿では, この部分について 3 節で詳しく記述する.

2.1 定式化

$n \times n$ 実対称行列 $\boldsymbol{Z}$ が半正定値であるとは, 任意の $\boldsymbol{x} \in \mathbb{R}^n$ に対して, $\boldsymbol{x}^T \boldsymbol{Z} \boldsymbol{x} \geq 0$ が成り立つことをいう. また, $n \times n$ 実対称行列 $\boldsymbol{Z}$ が正定値であるとは, 任意の $\boldsymbol{x} \in \mathbb{R}^n \setminus \{0\}$ に対して, $\boldsymbol{x}^T \boldsymbol{Z} \boldsymbol{x} > 0$ が成り立つことをいう. $\mathbb{S}^n, \mathbb{S}^n_+, \mathbb{S}^n_{++}$ をそれぞれ $n \times n$ 実対称行列の集合, $n \times n$ 実半正定値対称行列の集合, $n \times n$ 実正定値対称性行列の集合とする. また, $\boldsymbol{Z}, \boldsymbol{W} \in \mathbb{S}^n$ に対して, $\boldsymbol{Z} \bullet \boldsymbol{W} := \sum_{i=1}^n \sum_{j=1}^n Z_{i,j} W_{i,j}$ と定める.

与えられた $\boldsymbol{A}_0, \ldots, \boldsymbol{A}_m \in \mathbb{S}^n$, $b_1, \ldots, b_m \in \mathbb{R}$ に対して, SDP 問題は次の様に定義される[3]:

$$\begin{cases} \text{最小化}: & \boldsymbol{A}_0 \bullet \boldsymbol{X} \\ \text{制約}: & \boldsymbol{A}_j \bullet \boldsymbol{X} = b_j, \ (j = 1, \ldots, m), \ \boldsymbol{X} \in \mathbb{S}^n_+. \end{cases} \tag{1}$$

この SDP では $\boldsymbol{X}$ が変数である. 全ての制約式を満たす $\boldsymbol{X}$ を SDP (1) の実行可能解と呼ぶ.

$\boldsymbol{X} \in \mathbb{S}^n_+$ という制約は, $\boldsymbol{X}$ に対して, 全ての主小行列式が非負の値をとる, ということと等価である. また定義より, $\boldsymbol{X} \in \mathbb{S}^n_+$ は次の式とも等価である:

$$\boldsymbol{v}^T \boldsymbol{X} \boldsymbol{v} \geq 0 \ (\forall \boldsymbol{v} \in \mathbb{R}^n)$$

これは, $\boldsymbol{X}$ が無限本の線形不等式を全て満たさなければならないことを表している. したがって, $\boldsymbol{X} \in \mathbb{S}^n_+$ という制約が非線形な制約ということを表しており, SDP は非線形計画問題の一つと言える. SDP は非線形最適化問題なので, 最小値をとる解が存在しないかもしれないことに注意してほしい. 最小値をとる解が存在しない例を例 2.2 で紹介する[4].

例 2.1 SDP の例題として, 次のような設定を考える (例題は [18] より引用):

$$m = 2, \ n = 3, \ \boldsymbol{A}_0 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \ \boldsymbol{A}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \ \boldsymbol{A}_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \ b_1 = 0, \ b_2 = 2.$$

さらに, 変数行列 X を

$$\boldsymbol{X} = \begin{pmatrix} X_{11} & X_{12} & X_{13} \\ X_{12} & X_{22} & X_{23} \\ X_{13} & X_{23} & X_{33} \end{pmatrix}$$

---

[3] このように記述された最適化問題は次の様に読む: まず "最小化" の横にある数式は, 最小化したい目的関数である. "制約" の横にある 2 つの数式は, 変数行列 $\boldsymbol{X}$ の満たすべき制約式である. したがって, この最適化問題は変数行列 $\boldsymbol{X}$ が半正定値行列で m 本の数式を満たすもののなかで目的関数を最小化する, と読むことができる.

[4] したがって, ここで書いている "最小化" は, min の意味ではなく inf が適切である.

と書くことにする[5]. この時, $\boldsymbol{X} \in \mathbb{S}^3_+$ は全ての主小行列式が非負の値をとることと等価であるので, 7 本の不等式

$$\begin{cases} X_{11}, X_{22}, X_{33} \geq 0, \\ X_{11}X_{22} - X_{12}^2, X_{11}X_{33} - X_{13}^2, X_{22}X_{33} - X_{23}^2 \geq 0, \\ X_{11}X_{22}X_{33} + 2X_{12}X_{23}^2 - X_{22}X_{13}^2 - X_{33}X_{12}^2 - X_{11}X_{23}^2 \geq 0 \end{cases} \tag{2}$$

を満たさなければならない. さらに, $\boldsymbol{A}_0 \bullet \boldsymbol{X}$ は

$$\boldsymbol{A}_0 \bullet \boldsymbol{X} = 0 \cdot X_{11} + 2 \cdot 0 \cdot X_{12} + 2 \cdot 0 \cdot X_{13} + 0 \cdot X_{22} + 2 \cdot 0 \cdot X_{23} + 1 \cdot X_{33} = X_{33}$$

である. 同様に $\boldsymbol{A}_1 \bullet \boldsymbol{X}$, $\boldsymbol{A}_2 \bullet \boldsymbol{X}$ を計算すると, SDP (1) は次の様に書ける:

$$\begin{cases} \text{最小化}: & X_{33} \\ \text{制約}: & X_{11} = 0,\ 2X_{12} + X_{33} = 2,\ (2) \text{にある全ての制約式}. \end{cases} \tag{3}$$

この例題の場合, $X_{11} = 0$ と (2) の 4, 5, 6 番目の不等式から, $X_{12} = X_{13} = 0$ であることがわかる. これを $2X_{12} + X_{33} = 2$ に代入すると, $X_{33} = 2$ である. したがって, SDP (3) の最小値は 1 であり, 最小解は次の様に書ける:

$$\boldsymbol{X} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & X_{22} & X_{23} \\ 0 & X_{23} & 2 \end{pmatrix} \quad (\text{ただし}, 2X_{22} \geq X_{23}^2 \text{ を満たさなければならない})$$

例 2.2 例 2.1 において, $\boldsymbol{A}_0$ と $\boldsymbol{A}_1$ を交換した SDP を考える. つまり,

$$m = 2,\ n = 3,\ \boldsymbol{A}_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},\ \boldsymbol{A}_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},\ \boldsymbol{A}_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix},\ b_1 = 0,\ b_2 = 2,$$

とすると, SDP (1) は, 次の様になる:

$$\begin{cases} \text{最小化}: & X_{11} \\ \text{制約}: & X_{33} = 0,\ 2X_{12} + X_{33} = 2,\ (2) \text{にある全ての制約式}. \end{cases} \tag{4}$$

例 2.1 と同様に, $X_{13} = X_{23} = X_{33} = 0$, $X_{12} = 1$ が言える. 一方, 任意の $\epsilon > 0$ に対して,

$$\boldsymbol{X}_\epsilon := \begin{pmatrix} \epsilon & 1 & 0 \\ 1 & 1/\epsilon & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

とおくと, $\boldsymbol{X}_\epsilon$ は半正定値行列であり, SDP (4) の実行可能解になっている. そのときの目的関数値が ϵ で, このことは任意の $\epsilon > 0$ で成立するので SDP (4) の最小値は 0 であることがわかる. しかしながら, この最小値 0 を達成する解は存在しない. なぜなら, 最小値 0 を達成するためには, $X_{11} = 0$ でなければならないが $X_{12} = 1$ なので, (2) にある不等式 $X_{11}X_{22} - X_{12}^2 \geq 0$ を満たさない. つまり $X_{11} = 0$, $X_{12} = 1$ であるような半正定値行列は存在しない.

[5]変数行列 $\boldsymbol{X}$ は対称行列であるので, ここではそれを考慮して対称行列になる様に要素を記述している

2.2 半正定値計画問題に対するソフトウェア

半正定値計画問題に対するソフトウェアについては, [3] の "半正定値計画問題に対するソルバーの紹介" で 2010 年の状況が詳細に述べられている. 2012 年現在も状況はあまり変化していない. 有償のソフトウェアよりも, 無償のソフトウェアが多い. また, MATLAB で動くものもいくつかある. ここで簡単にあげると, SDPA [14], SeDuMi [16], SDPT3 [15], CSDP [2] 等がある. それぞれのソフトウェアには特徴があり, また, SDP 問題の性質によってどのソフトウェアが有利かは代わることがある. [8] で示されている性能比較を元にソフトウェアを選択するのも一つの手かもしれない.

SDP 問題の規模は通常, 変数行列 $\boldsymbol{X}$ のサイズである n と等式制約の数 m で記述される. しかしながら, これだけで SDP 問題が効率よく解けるかどうか判断するのは難しい. 実際, SDP 問題を解くアルゴリズムである主双対内点法では, 各反復で $m \times m$ の連立方程式を解く必要がある. したがって, あまり m が大きいと高速に解くことが難しくなる. また, $\boldsymbol{X}$ の決定変数は $(n+1)n/2$ 個であり, n が数千くらいが限界である. さらに, 係数行列 $\boldsymbol{A}_j$ の疎性も重要である. できるだけ非零要素が少ない係数行列でモデル化できるなら, その方が高速計算が期待できる.

2.3 主双対内点法

SDP 問題を効率よく解くアルゴリズムとして, 主双対内点法が提案されている. また 2.2 節で記述したソフトウェアには主双対内点法が実装されている[6].

主双対内点法は反復法であり, 双対定理の仮定が成立するもとで収束することが証明されている. 与えられた $\epsilon > 0$ に対して, $O(\sqrt{n}\log(1/\epsilon))$ の反復回数で近似解が得られることが示されている. また, さらに適当な仮定をおくことで, 主双対内点法が超一次収束することも示されている. 一方, 実際に解いてみるとだいたい 20 回から 40 回くらいの反復回数で近似解が得られることが多い. アルゴリズムの詳細は [6, 18] に書かれているので興味のある読者はそちらを参照してほしい.

3 多項式最適化問題に対する半正定値計画緩和

3.1 概要

多項式最適化問題 (以下 POP と省略する[7]) とは, 多項式の不等式で表現される集合上で多項式を最小化する問題のことである. 数式で記述すると, 次のように書ける:

$$\begin{cases} \text{最小化}: & f_0(\boldsymbol{x}) \\ \text{制約}: & f_j(\boldsymbol{x}) \geq 0 \ (j = 1, \ldots, m), \end{cases} \tag{5}$$

[6] 主双対内点法にも様々な種類があり, どのソフトウェアも全く同じアルゴリズムが実装されているわけではない.

[7] Polynomial Optimization Problems の省略

ただし, 変数 $\boldsymbol{x}$ は実 n 次元ベクトルであり, $f_0, f_1, \ldots, f_m$ は n 変数の多項式である. 例えば, 最大カット問題, 最大重み安定集合問題などの組合せ最適化問題や二次計画問題などは POP として記述できる.

Lasserre [9] や Parrilo [12] によって, POP に対する半正定値計画緩和, つまり, SDP を利用して POP の最小値の下界値を求める手法, が提案されている. これ以降, 注目されるようになった理由として,

- 最大カット問題や安定集合問題などの組合せ最適化問題に対して, 既存の手法よりも良い下界値, つまり元の最適化問題の最適値に非常に近い下界値を与える,
- 凸性のない POP であっても, SDP 緩和によって POP の最小値そのものを得ることができる場合がある,
- また, POP によっては SDP 緩和問題の解から POP の最小解を得ることができる,
- 実代数幾何学や関数解析の分野と密接に関連していて, 数学的にも興味深い,

ことがあげられる.

POP に対する SDP 緩和を実装したソフトウェアとして, Gloptipoly [5], SOSTOOLS [13], SparsePOP [20] が公開されている. いずれも MATLAB と SeDuMi を利用する. ただし, SparsePOP に関しては SDPA を使うこともできる.

SparsePOP は, POP が "疎構造" を持っている場合に, それを利用してよりサイズが小さい SDP 緩和問題を構成する手法を組み込んでいる. 疎構造に関しては, 小節 3.4 で簡単に触れるが, 詳細は [19] の 3 節と 4 節を参照してほしい. 一般に, POP に対する SDP 緩和では, POP の変数の数が多いと, 得られる SDP 緩和問題が大規模になる. 疎構造を利用してこの困難を克服しようとしたのが, [19] やそれを実装した SparsePOP である.

3.2 本節で用いる記号について

いくつか記号を導入する. $\mathbb{N}$ は自然数の集合である. $\boldsymbol{x} \in \mathbb{R}^n$, $\alpha \in \mathbb{N}^n$ に対して, 単項式 $x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$ を $\boldsymbol{x}^\alpha$ と書くことにする. ただし, $\boldsymbol{x}^0 = 1$ である. また, 多項式 f を

$$f(\boldsymbol{x}) = \sum_{\alpha \in F} f_\alpha \boldsymbol{x}^\alpha$$

と書くことにする. ここで, $F \subseteq \mathbb{N}^n$ は, 係数が零でない単項式 $\boldsymbol{x}^\alpha$ の指数ベクトル α に対応する. F は有限個の要素からなることに注意する.

$r \in \mathbb{N}$, $\boldsymbol{x} \in \mathbb{R}^n$ に対して, $\boldsymbol{u}_r(\boldsymbol{x}) := (1, x_1, \ldots, x_n, x_1^2, \ldots, x_1^r, \ldots, x_n^r)^T$ と定める. $\boldsymbol{u}_r(\boldsymbol{x})$ は次数 0 から次数 r までの単項式を並べた列ベクトルである. このベクトルの要素数は, $\binom{n+r}{r}$ である. ここで, $N(n, r) := \binom{n+r}{r}$ と書くことにする.

このベクトルに対して, 行列 $\boldsymbol{u}_r(\boldsymbol{x})\boldsymbol{u}_r(\boldsymbol{x})^T$ を考える. 例えば, $n = r = 2$ の場合, 以下の様に

なる:

$$\boldsymbol{u}_2(\boldsymbol{x})\boldsymbol{u}_2(\boldsymbol{x})^T = \begin{pmatrix} 1 & x_1 & x_2 & x_1^2 & x_1x_2 & x_2^2 \\ x_1 & x_1^2 & x_1x_2 & x_1^3 & x_1^2x_2 & x_1x_2^2 \\ x_2 & x_1x_2 & x_2^2 & x_1^2x_2 & x_1x_2^2 & x_2^3 \\ x_1^2 & x_1^3 & x_1^2x_2 & x_1^4 & x_1^2x_2^2 & x_1x_2^3 \\ x_1x_2 & x_1^2x_2 & x_1x_2^2 & x_1^3x_2 & x_1^2x_2^2 & x_1x_2^3 \\ x_2^2 & x_1x_2^2 & x_2^3 & x_1^2x_2^2 & x_1x_2^3 & x_2^4 \end{pmatrix}.$$

行列 $\boldsymbol{u}_r(\boldsymbol{x})\boldsymbol{u}_r(\boldsymbol{x})^T$ は, 半正定値行列である[8]. ここで次の様に定数行列 $\boldsymbol{E}_\alpha$ で記述する:

$$\boldsymbol{u}_r(\boldsymbol{x})\boldsymbol{u}_r(\boldsymbol{x})^T = \boldsymbol{E}_0 + \sum_{\alpha \in \mathbb{N}^n_{2r} \setminus \{0\}} \boldsymbol{E}_\alpha \boldsymbol{x}^\alpha$$

$n = r = 2$ の場合, $\boldsymbol{E}_{(0,0)}$, $\boldsymbol{E}_{(0,2)}$, $\boldsymbol{E}_{(2,2)}$ は実 6×6 対称行列であり次の様になる:

$$\boldsymbol{E}_{(0,0)} = \begin{pmatrix} 1 & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \end{pmatrix}, \ \boldsymbol{E}_{(2,2)} = \begin{pmatrix} & & & & & \\ & & & & & \\ & & & & & \\ & & & & & 1 \\ & & & & 1 & \\ & & & 1 & & \end{pmatrix}, \ \boldsymbol{E}_{(0,2)} = \begin{pmatrix} & & & & & 1 \\ & & & & & \\ & & 1 & & & \\ & & & & & \\ & & & & & \\ 1 & & & & & \end{pmatrix}.$$

ただし, 空欄は 0 を表している.

同様に多項式 f に対して, 行列 $f(\boldsymbol{x})\boldsymbol{u}_r(\boldsymbol{x})\boldsymbol{u}_r(\boldsymbol{x})^T$ を考える. $f(\boldsymbol{x}) \geq 0$ となるベクトル $\boldsymbol{x}$ に対して, この行列は半正定値行列になる. また, 逆に, この行列がある $\boldsymbol{x}$ で半正定値ならば $f(\boldsymbol{x}) \geq 0$ である. なぜなら, $\boldsymbol{u}_r(\boldsymbol{x})\boldsymbol{u}_r(\boldsymbol{x})^T$ の対角要素に 1 を含んでいるからである.

$d = \deg(f)$ とおくと, 同様に定数行列 $\boldsymbol{G}_\alpha$ を使って次の様に書ける:

$$f(\boldsymbol{x})\boldsymbol{u}_r(\boldsymbol{x})\boldsymbol{u}_r(\boldsymbol{x})^T = \boldsymbol{G}_0 + \sum_{\alpha \in \mathbb{N}^n_{2r+d} \setminus \{0\}} \boldsymbol{G}_\alpha \boldsymbol{x}^\alpha.$$

$\mathbb{N}^n_{2r+d}$ あれば, 行列 $f(\boldsymbol{x})\boldsymbol{u}_r(\boldsymbol{x})\boldsymbol{u}_r(\boldsymbol{x})^T$ に現れる全ての単項式が記述できることに注意してほしい. また, いくつかの行列 $\boldsymbol{G}_\alpha$ は零行列かもしれない. 実際, f が定数項を含んでいなければ $\boldsymbol{G}_0$ は零行列になる.

3.3 多項式最適化問題に対する半正定値計画緩和の詳細

POP (5) に対して, $r_0 = \lceil \max\{\deg(f_0), \deg(f_1), \ldots, \deg(f_m)\}/2 \rceil$ とおく. $r \geq r_0$ を満たす $r \in \mathbb{N}$ を選ぶ. また $j = 1, \ldots, m$ に対して, $r_j = r - \lceil \deg(f_j)/2 \rceil$ とおく. これは, (6) で現れる単項式が次数 $2r$ 以下であることを保証するためである. 目的関数 f_0 を次の様に記述する:

$$f_0(\boldsymbol{x}) = \sum_{\alpha \in F_0} (f_0)_\alpha \boldsymbol{x}^\alpha.$$

[8] 任意の $\boldsymbol{v} \in \mathbb{R}^{N(n,r)}$ に対して, $\boldsymbol{v}^T(\boldsymbol{u}_r(\boldsymbol{x})\boldsymbol{u}_r(\boldsymbol{x})^T)\boldsymbol{v} = (\boldsymbol{v}^T\boldsymbol{u}_r(\boldsymbol{x}))^2 \geq 0$ であるので, 半正定値行列であることがわかる.

$(f_0)_\alpha$ は多項式 f_0 の単項式 $\boldsymbol{x}^\alpha$ に対応する係数である.

POP (5) に対して, 次の様に制約を追加する:

$$\left\{\begin{array}{ll} \text{最小化}: & \sum_{\alpha \in F_0} (f_0)_\alpha \boldsymbol{x}^\alpha \\ \text{制約}: & \boldsymbol{G}_{j,0} + \sum_{\alpha \in \mathbb{N}^n_{2r_j} \setminus \{0\}} \boldsymbol{G}_{j,\alpha} \boldsymbol{x}^\alpha \in \mathbb{S}_+^{N(n,r_j)} \ (j = 1, \ldots, m) \\ & \boldsymbol{E}_0 + \sum_{\alpha \in \mathbb{N}^n_{2r} \setminus \{0\}} \boldsymbol{E}_\alpha \boldsymbol{x}^\alpha \in \mathbb{S}_+^{N(n,r)}. \end{array}\right. \tag{6}$$

前の小節で述べた様に, 追加した行列の性質から POP (5) と (6) は等価である. つまり最小解も最小値も変わらない. 次に, 単項式 $\boldsymbol{x}^\alpha$ を y_α という変数で置き換える. これを線形化と呼ぶ:

$$\left\{\begin{array}{ll} \text{最小化}: & \sum_{\alpha \in F_0} (f_0)_\alpha y_\alpha \\ \text{制約}: & \boldsymbol{G}_{j,0} + \sum_{\alpha \in \mathbb{N}^n_{2r_j} \setminus \{0\}} \boldsymbol{G}_{j,\alpha} y_\alpha \in \mathbb{S}_+^{N(n,r_j)} \ (j = 1, \ldots, m) \\ & \boldsymbol{E}_0 + \sum_{\alpha \in \mathbb{N}^n_{2r} \setminus \{0\}} \boldsymbol{E}_\alpha y_\alpha \in \mathbb{S}_+^{N(n,r)}, \\ & \boldsymbol{x}^\alpha = y_\alpha \ (\alpha \in \mathbb{N}^n_{2r} \setminus \{0\}) \end{array}\right. \tag{7}$$

(7) の最後の式より, (7) も (6) も同じ最小値と最小解を持つ. したがって, (7) と (5) も等価である. 最後に, (7) から最後の式を除くことで, POP (5) の SDP 緩和問題を得る.

$$\left\{\begin{array}{ll} \text{最小化}: & \sum_{\alpha \in F_0} (f_0)_\alpha y_\alpha \\ \text{制約}: & \boldsymbol{G}_{j,0} + \sum_{\alpha \in \mathbb{N}^n_{2r_j} \setminus \{0\}} \boldsymbol{G}_{j,\alpha} y_\alpha \in \mathbb{S}_+^{N(n,r_j)} \ (j = 1, \ldots, m) \\ & \boldsymbol{E}_0 + \sum_{\alpha \in \mathbb{N}^n_{2r} \setminus \{0\}} \boldsymbol{E}_\alpha y_\alpha \in \mathbb{S}_+^{N(n,r)}. \end{array}\right. \tag{8}$$

(8) は (7) から制約をいくつか除くことで得られるので, (5), (8) の最小値をそれぞれ f^*, p_r^* とおけば, $p_r^* \leq f^*$ が成立することがわかる. これは, 全ての $r \geq r_0$ で成り立っている. また r に関して単調性があることもわかる. つまり $p_r^* \leq p_{r+1}^* \leq f^*$ が成り立つ. 実は, Lasserre [9] では, ある仮定をおくことで $p_r^* \to f^*$ $(r \to \infty)$ ということを示している. ここで注目してほしいことは, p_r^* は SDP の最小値である, ということである. したがって, 十分大きい r をとって SDP 緩和問題 (8) を構成し, ソフトウェアで解けば f^* に十分近い値が求められるのである.

では, r を十分を大きくして SDP 緩和問題を構成しそれを SDP のソフトウェアで解けば良いのかというと, そうではない. というのも, SDP 緩和問題のサイズ (変数行列のサイズや等式制約の数など) についても r に関して単調増加性を有している. 実際, SDP 緩和問題 (8) において一番大きい行列のサイズは $N(n,r) = \binom{n+r}{r}$ である. したがって, 現在の計算機の能力では r を大きくして解くのは, n が小さくない限り難しい. ただ, 経験的には $r = 2, 3$ 位で (5) の最小値かそれに十分近い値が得られている. 現状では, $n = 25$ で $r = 3$ くらいの SDP 緩和問題を解くのが限界であるといわれている.

3.4 疎構造を持つ多項式最適化問題の簡単な説明

SDP 緩和問題が大規模になりすぎる, という困難を克服しなければならない. [19] では多項式最適化問題 (5) が "疎構造" を持っている場合に, それを利用してよりサイズが小さい SDP

緩和問題を構成する手法を提案している．疎構造に関しては, [19] の 3 節と 4 節を参照してほしいが, 簡単にいうと (5) の中に現れる変数の組合せをいう．例えば, 次の様に変数が 2 つの組に分かれている POP を考えてみる:

$$\begin{cases} \text{最小化：} & f_0(x_1,x_2)+g_0(x_3,x_4) \\ \text{制約：} & f_j(x_1,x_2)\geq 0\ (j=1,\dots,m), \\ & g_j(x_3,x_4)\geq 0\ (j=1,\dots,m) \end{cases} \tag{9}$$

この場合, $\boldsymbol{u}_r(x_1,\dots,x_4)$ を使うよりも, $\boldsymbol{u}_r(x_1,x_2)$ と $\boldsymbol{u}_r(x_3,x_4)$ と分けて SDP 緩和問題を構成しても良さそうな気がする．これにより, SDP 緩和問題のサイズが小さくなることがわかる．実際, この POP は次の様に二つの POP に分けてそれぞれの最小値の和が (9) の最小値になっており, 上記の (9) に対する SDP 緩和は, それぞれを SDP 緩和したことに対応している．

$$\begin{cases} \text{最小化：} & f_0(x_1,x_2) \\ \text{制約：} & f_j(x_1,x_2)\geq 0\ (j=1,\dots,m) \end{cases}, \quad \begin{cases} \text{最小化：} & g_0(x_3,x_4) \\ \text{制約：} & g_j(x_3,x_4)\geq 0\ (j=1,\dots,m) \end{cases}$$

では, 以下の 2 つの POP はどうであろうか:

$$\begin{cases} \text{最小化：} & f_0(x_1,x_2)+g_0(x_2,x_3)+h_0(x_3,x_4) \\ \text{制約：} & f_j(x_1,x_2)\geq 0\ (j=1,\dots,m), \\ & g_j(x_2,x_3)\geq 0\ (j=1,\dots,m), \\ & h_j(x_3,x_4)\geq 0\ (j=1,\dots,m) \end{cases} \tag{10}$$

$$\begin{cases} \text{最小化：} & f_0(x_1,x_2)+g_0(x_1,x_3)+h_0(x_1,x_4) \\ \text{制約：} & f_j(x_1,x_2)\geq 0\ (j=1,\dots,m), \\ & g_j(x_1,x_3)\geq 0\ (j=1,\dots,m), \\ & h_j(x_1,x_4)\geq 0\ (j=1,\dots,m) \end{cases} \tag{11}$$

これらは (9) の様に二つの最適化問題に分離できる構造を持っていない．しかし, (10) では, $\{1,2\}$, $\{2,3\}$, $\{3,4\}$ という変数の組がなんとなく見える．また (11) では, $\{1,2\}$, $\{1,3\}$, $\{1,4\}$ という変数の組がなんとなく見える．

このように, 変数は n 個あるがいくつかのグループに変数を分類できる場合がある．これらのグループの要素数が変数の数 n に比べて少ない時に, POP が疎構造を持っていると呼ぶ[9]. [19] では, (i) 疎構造を持っているかどうか確認する方法と, (ii) 疎構造を持っている場合に, 変数のグループを利用して規模の小さい SDP 緩和問題を構成する方法, を提案している．そして, その機能を実装したのが SparsePOP[20] である．(10) では, $\boldsymbol{u}_r(x_1,x_2)$, $\boldsymbol{u}_r(x_4,x_3)$, $\boldsymbol{u}_r(x_3,x_4)$ を使って, (11) では, $\boldsymbol{u}_r(x_1,x_2)$, $\boldsymbol{u}_r(x_1,x_3)$, $\boldsymbol{u}_r(x_1,x_4)$ を使って SDP 緩和問題を本節の方法で構成することでより小さい SDP 緩和問題を構成している．一方で, $x_1^2+\cdots+x_n^2=1$ のように 1 つの制約式に全ての変数が利用されている場合, 疎構造を有しておらず [19] で提案した手法を適用しても Lasserre の SDP 緩和を適用した場合と大差はない．

[9] もし, グループが 1 つだけの場合, つまり $C_1=\{1,\dots,n\}$ となっている場合, C_1 の要素数は n であり疎構造を有していない, ということになる．

[19] では, 様々な数値実験を行い, 変数が 20 個以上の POP で疎構造を有している場合には, 疎構造を利用した SDP 緩和を利用するのがよいと結論づけている. 例えば [4] にある “alkyl.gms” という POP に対して数値実験を行っている. この POP は 14 個の変数と 37 本の次数 1 から 3 までの多項式等式・不等式からなり, 疎構造を有している. 前節で述べた SDP 緩和では規模が大きくなりすぎて解けないのに対して, 疎構造を利用すると $r = 3$ でこの POP の最小値が数秒で得られている, ということが報告されている.

4　おわりに

SDP やその他の応用に関しては [3, 6, 17, 18] で良く記述されている. 興味を持たれた読者はぜひ読んでみてほしい. また, [10, 11] では POP に対する SDP 緩和についてまとめられたサーベイである. [1] では, SDP に関する最新の結果が豊富に記載されている. こちらもぜひあわせて読んでほしい.

参考文献

[1] M. Anjos and J. B. Lasserre (eds.): Handbook of Semidefinite, Conic and Polynomial Optimization, Springer, New York, (2011)

[2] CSDP, `https://projects.coin-or.org/Csdp`

[3] 藤澤克樹他, “特集　半正定値計画に対するソルバーと応用例”, オペレーションズ・リサーチ, Vol. 55, No. 7, (2010)

[4] GLOBAL Library, `http://www.gamsworld.org/global/globallib.htm` (2005).

[5] D. Henrion and J. B. Lasserre, “GloptiPoly: Global optimization over polynomials with Matlab and SeDuMi”, ACM Transactions Mathematical Software 29: 165–194, 2003, available from `http://homepages.laas.fr/henrion/software/gloptipoly2/`

[6] 小島政和, 土谷隆, 水野眞治, 矢部博, 内点法, 朝倉書店 (2001).

[7] H. Mittelmann, `http://plato.asu.edu/ftp/sparse_sdp.html`

[8] H. Mittelmann, `http://plato.asu.edu/talks/ismp.pdf`

[9] J. B. Lasserre, “Global optimization with polynomials and the problems of moments”, SIAM Journal on Optimization, 11, 796–817 (2001)

[10] J. B. Lasserre, “Moments, Positive Polynomials and Their Applications”, Imperial College Press Optimization Series Vol. 1, Imperial College Press (2009).

[11] M. Laurent, “Sums of squares, moment matrices and optimization over polynomials”. In Emerging Applications of Algebraic Geometry, M. Putinar and S. Sullivant editors, 157–270, Springer (2009).

[12] P. A. Parrilo, “Semidefinite programming relaxations for semialgebraic problems”, Mathematical Programming, 96, 293–320 (2003)

[13] S. Prajna, A. Papachristodoulou, P. Seiler and P. A. Parrilo, "SOSTOOLS (Sums of squares optimization toolbox for MATLAB) User's guide", available from `http://www.cds.caltech.edu/sostools/`

[14] SDPA, `http://sdpa.indsys.chuo-u.ac.jp/sdpa`

[15] SDPT3, `http://www.math.nus.edu.sg/~mattohkc/sdpt3.html`

[16] SeDuMi, `http://sedumi.ie.lehigh.edu/`

[17] 田村明久, 村松正和, 最適化法, 共立出版 (2005).

[18] M. J. Todd, "Semidefinite optimization", Acta Numerica, 10, 515–560 (2001)

[19] H. Waki, S. Kim, M. Kojima and M. Muramatsu, "Sums of Squares and Semidefinite Programming Relaxations for Polynomial Optimization Problems with Structured Sparsity", SIAM Journal on Optimization, 17, 218–242 (2006)

[20] H. Waki, S. Kim, M. Kojima, M. Muramatsu and H. Sugimoto, "SparsePOP: a Sparse SDP Relaxation of Polynomial Optimization Problems", ACM Transactions on Mathematical Software, 35, 2, 15:1–15:13 (2008), available from `http://sourceforge.net/projects/sparsepop/`