Sparsifying Global Constraints in Polynomial Optimization Problems （多項式最適化問題における大域的成約の疎性化）

数理情報学専攻 48–196218　當眞 ジェイソン翔
指導教員　谷川 眞一 准教授

1　はじめに

n 変数実係数多項式 $f, g_1, g_2, \ldots, g_k$ を用いて

$$\begin{aligned} &\inf_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \\ &\text{subject to } g_k(\boldsymbol{x}) \geq 0 \quad (k=1,2,\ldots,m), \end{aligned} \tag{1}$$

という形式で書ける形の最適化問題を多項式最適化問題 (Polynomial Optimization Problem; POP) という．POP は多くの分野に現れる重要な最適化問題である．一般に POP は NP 困難な問題である [5]．そのため，POP の下界を得るための緩和手法として SOS (Sum of Squares) 緩和という手法が広く用いられている [2]．また，元の POP に疎性がある場合，より効率の良い Sparse SOS 緩和という緩和手法が提案されている [7, 3]．これらの手法によって得られた緩和問題は，半正定値計画問題 (SDP) として定式化できる．

本研究では，球面制約や一次の最適性条件など，これまでの Sparse SOS 緩和の枠組みでは取り扱えなかった大域的制約を，疎性を満たすように分解するアルゴリズムを提案する．アルゴリズムの性能は，数値実験を通して評価した．

2　既存研究

(1) は

$$\begin{aligned} &\sup_{(\boldsymbol{x},\gamma)\in\mathbb{R}^{n+1}} \gamma \\ &\text{subject to } f(\boldsymbol{x}) - \gamma \geq 0, \\ &\qquad\qquad g_k(\boldsymbol{x}) \geq 0 \quad (k=1,2,\ldots,m), \end{aligned} \tag{2}$$

と等価である．(2) の不等式制約を以下のように置き換えた最適化問題

$$(\mathrm{P}_d) \quad \begin{aligned} &\sup_{(\boldsymbol{x},\gamma)\in\mathbb{R}^{n+1}} \gamma \\ &\text{subject to } f(\boldsymbol{x}) - \gamma \in Q_d(g), \end{aligned} \tag{3}$$

を考え，その最適値を γ_d とする．[2] によって，緩い仮定をおいた上で $d \to \infty$ で γ_d が (1) の最適値に収束することが示された．ここで，$Q_d(g)$ は

$$Q_d(g) \coloneqq \left\{ \sigma_0 + \sum_{i=1}^{m} \sigma_i g_i \;\middle|\; \begin{aligned} &\sigma_i \in \Sigma[\boldsymbol{x}] \\ &\deg(\sigma_0) \leq 2d \\ &\deg(\sigma_i g_i) \leq 2d \end{aligned} \right\}, \tag{4}$$

という形の集合であり，$\Sigma[\boldsymbol{x}]$ は SOS 多項式 (他の多項式の二乗和で書ける多項式) 全体の集合である．(P_d) は，半正定値変数のサイズが $\mathrm{O}(n^d)$，線形制約の個数が $\mathrm{O}(n^{2d})$ の半正定値計画問題として定式化できるということが知られている．しかしながら，現在のノート PC で解ける SDP のサイズは行列のサイズが $5,000$ 以下，線形制約の個数が $20,000$ 以下程度のものであるとされている [6]．この基準に照らし合わせると，SOS 緩和問題 (3) は n, d の値が大きくなるにつれて計算機の制約ゆえに解くのが困難になってしまう．

[7] では，POP に疎性が成り立つときに SOS 緩和にもその疎性を遺伝させる手法を提案している．以下，彼らの手法を Sparse SOS 緩和と呼ぶ．彼らの提案の段階では，Sparse SOS には収束保証がなかったが，[3] によって収束保証がつけられた．ここで，POP (1) に集合 $I_1, I_2, \ldots, I_p \subseteq \{x_1, x_2, \ldots, x_n\}$ による疎性が成り立つとは，以下の 3 つの条件をすべて満たす場合のことを言う．

(1) $f \in \sum_{k=1}^{p} \mathbb{R}[I_k]$.
(2) 任意の k に対して，ある ℓ が存在し，$g_k \in \mathbb{R}[I_\ell]$.
(3) $I_1, I_2, \ldots, I_p$ を頂点とする木分解が存在する．

ここで，$\mathbb{R}[I_k]$ は I_k に属する変数のみからなる多項式全体の集合を指す．$V = \{I_1, I_2, \ldots, I_p\}$ を頂点とする木分解とは，頂点集合が V である無向木であり，任意の頂点対 (I_s, I_t) に対し，「I_u が I_s, I_t を結ぶ唯一のパス上にある頂点ならば $I_s \cap I_t \subseteq I_u$」という条件を満たすもののことをいう．このとき，次に示す 2 つの条件を満たす SDP の列 $(\mathrm{P}_d^{\mathrm{sparse}}), d = 1, 2, \ldots$ が存在する．

(1) $(\mathrm{P}_d^{\mathrm{sparse}})$ は半正定値変数のサイズが $\mathrm{O}(|\Delta|^d)$ であり，半正定値制約の個数が $\mathrm{O}(p|\Delta|^{2d})$ である．ここで，$|\Delta|$ は $I_1, I_2, \ldots I_p$ のうち要素数が最大のものの要素数を指す．
(2) $\gamma_d^{\mathrm{sparse}}$ を $(\mathrm{P}_d^{\mathrm{sparse}})$ の最適値とすると，緩い仮定のもと，

$$\lim_{d\to\infty} \gamma_d^{\mathrm{sparse}} = \gamma^*, \tag{5}$$

が成り立つ．

3 本研究で提案するアルゴリズム

POP (1) に $I_1, I_2, \ldots, I_p$ による疎性が成り立つと仮定する．本研究では，この疎性の枠組みに，球面制約

$$x_1^2 + x_2^2 + \cdots + x_n^2 = R, \tag{6}$$

などの $\sum_{k=1}^{p} \mathbb{R}[I_k]$ の元となっている大域的制約を加える手法を提案する．

Algorithm 1 疎性化のアルゴリズム (大枠)

Input : $g \in \sum_{k=1}^{p} \mathbb{R}[I_k]$.
Output: $g'_k \in \mathbb{R}[I'_k], k = 1, 2, \ldots, p$.

$g = \sum_{k=1}^{p} g_k, g_k \in \mathbb{R}[I_k]$ となるように g を分割
$s_2, s_3, \ldots, s_p \leftarrow$ 新しく追加する変数
$\mathcal{T} \leftarrow I_1$ を根とする木分解
for $k = 2$ **to** p **do**
    $g'_k \leftarrow s_k - g_k - \sum_{I_\ell : I_k の子} s_\ell$
$g'_1 \leftarrow g_1 + \sum_{I_\ell : I_1 の子} s_\ell$

$\boldsymbol{x}' = (x_1, x_2, \ldots, x_n, s_2, s_3, \ldots, s_p)^\top$ とする．上のアルゴリズムで，$g'_1(\boldsymbol{x}') \geq 0$ と $g'_k(\boldsymbol{x}') = 0$ が任意の k で成り立つならば，$g(\boldsymbol{x}) \geq 0$ となることがわかる．すなわち $g'_1, g'_2, \ldots, g'_p$ は g を疎性化した多項式である．上記のアルゴリズムの大枠を元にすることで，以下の結果を得る．

主結果． POP (1) が，集合 $I_1, I_2, \ldots, I_p$ に対して以下を満たすとする．

(1) $f \in \sum_{k=1}^{p} \mathbb{R}[I_k]$.
(2) $g_k \in \sum_{\ell=1}^{p} \mathbb{R}[I_\ell], \quad k = 1, 2, \ldots, m$.
(3) $I_1, I_2, \ldots, I_p$ を頂点とする木分解が存在する．

このとき，各 I_k に変数を高々 $3m$ 個追加することで，POP (1) と等価な POP で，疎性を持つものを得るアルゴリズムが存在する． □

4 数値実験

いくつかの POP に疎性化アルゴリズムを適用し，実行時間と解の精度を確認することでアルゴリズムの性能を評価した．実験に使用したマシンの OS は Ubuntu 16.04 である．マシンの CPU は Intel® Xeon® Gold 5222 (3.80 GHz) で，物理メモリは 768 GiB である．

ここでは，Broyden banded function [1, 4]

$$f(x) = \sum_{i=1}^{n} \left(x_i(2 + 5x_i^2) + 1 - \sum_{j \in J_i} (1 + x_j)x_j \right)^2, \tag{7}$$

$(J_i = \{j \mid j \neq i, \max(1, i-5) \leq j \leq \min(n, i+1)\})$ を球面制約 $\sum_{i=1}^{n} x_i^2 = 1$ 下で最小化する問題で，球面制約に疎性化アルゴリズムを適用した場合の問題サイズや，POP ソルバ CS-TSSOS [8] による実行時間を示す．CS-TSSOS の緩和次数は 3，疎性次数は 1 とした．球面制約を疎性化した場合，$n = 1000$ 変数に対する SDP にはサイズが 8 の半正定値変数が 1488 個，サイズが 7 以下の半正定値変数が 499 個存在し，線形制約の個数は 218234 個となっていた．制約の個数は多いが，半正定値変数のサイズが小さいため，$5,249$ CPU 秒で SDP の最適値が求まった．球面制約を疎性化しない場合，$n = 60$ の問題を解くのに $8,706$ CPU 秒かかるため，疎性化の効果が明確に出ていることがわかる．

5 まとめ

疎性化のアルゴリズムを提案し，いくつかの問題で性能の評価を行った．疎性化する際に追加した変数による誤差の精査が今後の展望である．

参考文献

[1] C. G. Broyden. The Convergence of an Algorithm for Solving Sparse Nonlinear Systems. *Mathematics of Computation*, 25(114):285–294, 1971.
[2] J. B. Lasserre. Global Optimization with Polynomials and the Problem of Moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.
[3] J. B. Lasserre. Convergent SDP-relaxations in Polynomial Optimization with Sparsity. *SIAM Journal on Optimization*, 17(3):822–843, 2006.
[4] J. J. Moré, B. S. Garbow, and K. E. Hillstrom. Testing Unconstrained Optimization Software. *ACM Transactions on Mathematical Software*, 7(1):1741, Mar. 1981.
[5] K. G. Murty and S. N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Mathematical Programming*, 39:117–129, 1987.
[6] K. C. Toh. Some numerical issues in the development of SDP algorithms. *INFORMS OS Today*, 8(2):7–20, 2018.
[7] H. Waki, S. Kim, M. Kojima, and M. Muramatsu. Sums of Squares and Semidefinite Program Relaxations for Polynomial Optimization Problems with Structured Sparsity. *SIAM Journal on Optimization*, 17:218–242, 01 2006.
[8] J. Wang, V. Magron, J. B. Lasserre, and N. A. Mai. CS-TSSOS: Correlative and term sparsity for large-scale polynomial optimization. *ArXiv*, abs/2005.02828, 2020.