Vol.2017-MPS-112 No.1
2017/2/27

非線形半正定値計画問題の定式化の比較

加藤 拓海[1,a)]　ロウレンソ ブルノ フィゲラ[1,b)]　池上 敦子[1,c)]

概要：非線形半正定値計画問題と，この問題を 2 乗スラック変数法を用いて再定式化した問題の求解速度や精度を比較する．非線形半正定値計画問題は非線形計画問題を含んでおり，半正定値の性質を使うことにより扱える問題の幅が広がるというメリットがある．現在では非線形半正定値計画問題を解くソルバーが少ない．しかし，非線形半正定値計画問題は 2 乗スラック変数法を用いて再定式化すると，多くのソルバーが実装されている非線形計画問題の形にできる．このことは求解の手段が増えるメリットを生むが，2 乗スラック変数法で再定式化した問題は変数の数が多くなるので，求解速度や精度に影響が出る可能性がある．2 つの定式化の求解速度や精度を比較し，現在のソルバーの性能では，どちらの定式化が効率的かを分析する

キーワード：非線形半正定値計画問題，2 乗スラック変数法，非線形計画問題，最適化

1. はじめに

数理最適化問題には様々な形がある．古典的な問題の 1 つとして，非線形計画問題 (**Nonlinear Programming, NLP**) がある [2], [6]. 非線形計画問題は以下のような問題である．

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \qquad (\text{NLP}) \\ \text{subject to} \quad & h(x) = 0 \\ & g(x) \in \mathbb{R}^m_+ \end{aligned}$$

ここで，f, g, h は 2 回連続的微分可能関数である．$f : \mathbb{R}^n \to \mathbb{R}$，$h : \mathbb{R}^n \to \mathbb{R}^l$, $g : \mathbb{R}^n \to \mathbb{R}^m$ である．$\mathbb{R}^m_+$ は $\mathbb{R}^m$ における非負象限を表す．

近年，非線形半正定値形画問題 (**Nonlinear Semidefinite Programming, NSDP**) という問題クラスは注目されている．NSDP は以下の問題である．

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \qquad (\text{NSDP}) \\ \text{subject to} \quad & G(x) \in S^m_+ \end{aligned}$$

f, G は 2 回連続的微分可能関数である．$f : \mathbb{R}^n \to \mathbb{R}$, $G : \mathbb{R}^n \to S^m$ である．記号 S^m は $m \times m$ 対称行列空間，S^m_+ は半正定値行列集合を表す．

[1] 成蹊大学
Seikei University, Musashino-shi, Tokyo 180–8633, Japan
[a)] us132033@cc.seikei.ac.jp
[b)] lourenco@st.seikei.ac.jp
[c)] atsuko@st.seikei.ac.jp

半正定値制約のために，NSDP の形で定式化出来る問題は NLP より多い．しかし，NSDP に対するアルゴリズムが多く開発されいている [7] のに対し，NSDP を取り扱える汎用数理計画ソフトはまだ少ない．我々の調査では，2 つしか存在しない：イギリスの PENLAB/PENNON[3] と日本の Numerical Optimizer[9].

しかし，ある NSDP を解く必要がある場合，NSDP ソルバーが手に入らなくとも，工夫の余地がある．半正定値性を利用して，同値の NLP の形として再定式化できる．

半正定値行列の集合は以下の性質がある [1].

$$S^m_+ = \{Y \circ Y \mid Y \in S^m\}$$

ここでの記号 $\circ$ は対称行列に関するジョルダン積，対称行列 W，Z に対して，$W \circ Z = (WZ + ZW)/2$ で定義される 2 項演算を表している．この性質を利用して (NSDP) 問題を 2 乗スラック変数法で再定式化した問題を以下に示す．

$$\begin{aligned} \underset{x,Y}{\text{minimize}} \quad & f(x) \qquad (\text{NSDP-SLACK}) \\ \text{subject to} \quad & G(x) - Y \circ Y = 0 \end{aligned}$$

NSDP を変形し，再定式化することによって，局所的最適解が同値の別の形にすることが出来る．したがって NLP のソルバーを用いて解くことが出来るようになる．本研究では 2 乗スラック変数法を用いて NSDP を NLP の形にする．2 乗スラック変数法は最適化の分野では求解の安定性が欠けたり，速度が遅くなったりと悪影響が出る可能性が

あると考えられている．ここでは NSDP の場合それは事実かどうかを調べる．

2. 先行研究

2 次錐計画問題 (**Second-Order Cone Programming, SOCP**) に対して 2 乗スラック変数法により NLP の形にする研究があり [4], [10]，その研究と同じことを NSDP にも適用出来るかどうかを試した研究もある [5]．しかし，その研究では PENLAB/PENNON を使用して実験したので，アルゴリズムとしては，その中に実装されている Augmented Lagrangian Method でしか試せていない．したがって，本研究では複数のアルゴリズムを実装している Numerical Optimizer で実験を行う．

3. 実験

本実験では 3 つの問題を扱うが，ここでは 2 つの問題に対する結果のみ報告する．使用する汎用数理計画ソフトは Numerical Optimizer(ver. 18.1.0)，Intel(R) Core(TM) i7-5930K CPU @ 3.50GHz，メモリ 32.0GB である．

Numerical Optimizer は暫定解による KKT 条件の外れ量が $1.0e^{-8}$ 以下になった際に停止する．アルゴリズムの反復上限回数のデフォルト値は 150 回である．

本実験ではアルゴリズムの反復上限の回数以内に停止条件を満たす場合を最適化に成功した（Success) とする．

実験で使用する NLP のアルゴリズムは下記のものである [8].

- bfgs : 準ニュートン法.
- lsqp : 直線探索法に基づく逐次二次計画法.
- slpsqp : 逐次線形二次計画法.
- tipm : (新版) 信頼領域内点法.
- tsqp : 信頼領域法での逐次二次計画法.

NSDP のアルゴリズムは以下のものである.

- lmsdp : Levenberg-Marquardt 法での非線形半正定値計画問題に対する主双対内点法.
- qnsdp : 準ニュートン法での非線形半正定値計画問題に対する主双対内点法.
- trsdp : 信頼領域法での非線形半正定値計画問題に対する主双対内点法.

3.1 実験 2

ベンチマーク問題である相関行列取得問題を扱う．この問題は行列 H にもっとも近い相関行列を見つける問題である．この問題を (Q1) とする．

$$\begin{aligned} \underset{X}{\text{minimize}} \quad & \langle X-H, X-H\rangle && \text{(Q1)} \\ \text{subject to} \quad & X_{ii}=1, \quad i\in\{1,\dots,m\} \\ & X\in S_+^m \end{aligned}$$

行列 H は $m\times m$ 対称行列，H の対角要素の値は 1，H の対角要素以外の値は [-1,1] である．(Q1) 問題を 2 乗スラック変数法を用いて再定式化した問題を (Q2) とする．

$$\begin{aligned} \underset{X}{\text{minimize}} \quad & \langle (X\circ X)-H, (X\circ X)-H\rangle && \text{(Q2)} \\ \text{subject to} \quad & (X\circ X)_{ii}=1, \quad i\in\{1,\dots,m\} \\ & X\in S^m \end{aligned}$$

これらの問題に対して，対角要素以外の値を [-1,1] の乱数を用いて 100 個の H を生成し実験を行う．行列 H の大きさは $m=5,10,15,20,30,50$ とする．初期解として単位行列を与える．(Q1) 問題と (Q2) 問題で H は同値のものを使用する．

3.1.1 実験結果と分析

表 1，表 2 は求解速度の最大値 (Max)，最小値 (Min)，平均値 (Mean)，最適化成功率 (Success) をアルゴリズ毎にまとめたものである．最大値，最小値，平均値を算出する際には最適化に失敗しているデータ (問題例) は除いている．問題に対する各アルゴリズムの中で最も安定性が高い，すなわち最適化成功率が一番高いアルゴリズムを比較することにした．最適化成功率が同値の場合は平均値が低い (良い) 方を比較することにした．

$m=30,50$ の場合は問題の規模が大きくなるので，アルゴリズムの反復回数の上限を 1500 回にする．

図 1 は全ての H に対しての求解速度を比較しているグラフである．縦軸が求解速度 (秒)，横軸が何番目の H かを表している．赤いマーカーは最適化に失敗していることを表している．(Q2) 問題の求解速度について昇順にソートしたグラフになっている．

表 1 では bfgs と lsqp は少ないながら最適化に成功している，だが表 2 では lsqp は最適化成功率が 100%になり，bfgs は 0%であった．

図 1 では全てのデータで (Q1) 問題の求解速度が上だった．全ての図で (Q1) 問題の求解速度においてはばらつきが少ない，(Q2) 問題の求解速度はばらつきが多い．これは (Q2) 問題の変数の数が (Q1)　問題よりも多いので，H の値による求解速度の変動が大きいと考えられる．m の値が大きくなると (Q2) 問題の求解速度の幅が大きくなっている．

3.2 実験 3

以下に (R1) 問題を示す.

$$\begin{aligned} \underset{X,z}{\text{minimize}} \quad & \langle zX-H, zX-H\rangle && \text{(R1)} \\ \text{subject to} \quad & zX_{ii}=1, \quad i\in\{1,\dots,m\} \\ & I_m\preceq X\preceq kI_m \end{aligned}$$

k は 1 より大き正の数である．$X\succeq kI_m$ は $X-kI_m\in S_+^m$

を表す．行列 H は $m \times m$ 対称行列，H の対角要素の値は 1，H の対角要素以外の値は [-1,1] である．

(R1) 問題を 2 乗スラック変数法を用いて再定式化した問題を (R2) とする．

$$\begin{aligned} \underset{X,z,Y_1,Y_2}{\text{minimize}} \quad & \langle zX - H, zX - H \rangle && \text{(R2)} \\ \text{subject to} \quad & zX_{ii} = 1, && i \in \{1, \ldots, m\} \\ & kI_m - X = Y_1 \circ Y_1 \\ & X - I_m = Y_2 \circ Y_2 \end{aligned}$$

これらの問題に対して，$k = 10$ とし，対角要素以外の値を [-1,1] の乱数を用いて 100 個の H を生成し実験を行う．行列 H の大きさは $m = 5, 10, 15, 20$ とする．初期解として単位行列を与える．(R1) 問題と (R2) 問題では H は同値のものを使用する．

3.2.1 実験結果と分析

表 3，表 4 は求解速度の最大値 (Max)，最小値 (Min)，平均値 (Mean)，最適化成功率 (Success) をアルゴリズム毎にまとめたものである．最大値，最小値，平均値を算出する際には最適化に失敗したデータは除いている．問題に対する各アルゴリズムの中で最も安定性が高い，すなわち最適化成功率が一番高いアルゴリズムを比較することにした．最適化成功率が同値の場合は平均値が低い (良い) 方を比較することにした．

図 2 は全 H に対しての求解速度を比較しているグラフである．縦軸が求解速度 (秒)，横軸が何番目の H かを表している．赤いマーカーは最適化に失敗していることを表している．(R2) 問題の求解速度について昇順にソートしたグラフになっている．

1 回目に行った $m = 20$ の実験では (R1) 問題の最適化成功率が低かった．最適化に失敗している理由はアルゴリズムの反復回数の上限を超えてしまうことがことが多かったからである．そこで，より正確な実験結果のために上限回数を 500 回に設定してもう再実験を行った．

表 3($m = 5$) では qnsdp が最適化に失敗しデータがあるのに対して，より規模の大きい問題の表 4($m = 20$) では全てのデータが最適化に成功している．qnsdp 以外のアルゴリズムは問題の規模が大きくなると最適化成功率が等しいか下がるのに対し，qnsdo は上がっている．

図 2($m = 20$) では求解速度が最も遅いデータが最適化に失敗している．

3.3 分析

実験 2 と実験 3 の各アルゴリズムの最適化成功率を m の値毎にまとめたものが表 5 である．この表から実験 2 では全ての m に対して，NSDP の最適化成功率が 100%となるアルゴリズムが 2 つあり，NLP の最適化成功率が 95%以上となるアルゴリズムが 3 つあった．実験 3 では NSDP は

表 1 求解速度と安定性の比較表 ($m = 5$)，反復回数 150 回

	Algorithm	Max (s)	Min (s)	Mean (s)	Success rate
(Q1)	lmsdp	0.018	0.006	0.00736	100
	qnsdp	0.032	0.013	0.01831	99
	trsdp	0.022	0.008	0.00993	100
(Q2)	bfgs	0.019	0.011	0.01337	100
	lsqp	0.017	0.006	0.00769	98
	slpsqp	0.022	0.011	0.01388	99
	tipm	0.024	0.007	0.00897	100
	tsqp	0.022	0.006	0.00927	100

表 2 求解速度と安定性の比較表 ($m = 50$)，反復回数 1500 回

	Algorithm	Max (s)	Min (s)	Mean (s)	Success rate
(Q1)	lmsdp	11.495	10.416	10.71274	100
	qnsdp	49.704	20.532	28.15894	100
	trsdp	12.706	11.48	12.00278	100
(Q2)	bfgs	////	////	////	0
	lsqp	198.757	147.996	158.60320	100
	slpsqp	2954.226	278.657	1416.37065	100
	tipm	749.147	134.293	327.58003	100
	tsqp	6904.198	201.220	621.44399	96

表 3 求解速度と安定性の比較表 ($m = 5$，反復回数 500 回)

	Algorithm	Max (s)	Min (s)	Mean (s)	Success (%)
(R1)	lmsdp	0.037	0.011	0.01442	100
	qnsdp	0.081	0.021	0.03731	99
	trsdp	0.027	0.013	0.01575	100
(R2)	bfgs	0.15	0.035	0.04698	100
	lsqp	0.197	0.018	0.03189	99
	slpsqp	1.33	0.052	0.19634	56
	tipm	0.173	0.038	0.06762	100
	tsqp	0.156	0.042	0.06914	98

表 4 求解速度と安定性の比較表 ($m = 20$，反復回数 500 回)

	Algorithm	Max (s)	Min (s)	Mean (s)	Success (%)
(R1)	lmsdp	1.063	0.79	0.91076	100
	qnsdp	11.628	1.39	2.74977	100
	trsdp	0.252	0.21	0.22988	100
(R2)	bfgs	76.981	34.44	40.64131	96
	lsqp	166.354	29.27	48.00567	89
	slpsqp	342.134	29.33	102.04708	25
	tipm	32.246	4.78	9.31657	76
	tsqp	160.911	17.99	40.66889	66

表 5　各アルゴリズムの最適化成功率

	m	lmsdp	qnsdp	trsdp	bfgs	lsqp	slpsqp	tipm	tsqp
実験 2	5	100	99	100	100	98	99	100	100
	10	100	85	100	99	100	99	100	100
	15	100	91	100	75	100	100	99	97
	20	100	91	100	0	0	99	100	97
	30	100	100	100	0	100	100	100	97
	50	100	100	100	0	100	100	100	96
実験 3	5	100	99	100	100	99	56	100	98
	10	100	99	100	100	99	32	92	93
	15	100	100	100	96	94	28	89	90
	20	100	100	100	96	89	25	76	66

実験 2 と同様の結果であったが，NLP の最適化成功率が 95%以上となるアルゴリズムは 1 つしかなかった．

求解の安定性と速度に関しては，規模が小さい問題であれば同程度であったが，規模が大きくなれば NSDP の方が優れていた．NLP が最適化に失敗する理由はアルゴリズムの反復上限の回数を超える場合が多かったので，時間を考慮しなければ上限を大きく設定して解くことが出来るかもしれない．

4. おわりに

本研究での実験では規模が小さい問題に関しては，非線形計画問題の求解速度が非線形半正定値計画問題よりも速い場合もあったが，基本的には非線形半正定値計画問題の方が求解速度が速かった．

求解の安定性の面に関しても，非線形半正定値計画問題では規模が大きい問題に対しても最適化成功率が 100%のアルゴリズムが存在したが，非線形計画問題では 100%のアルゴリズムはなかった．

非線形半正定値ソルバーを持たない汎用数理計画ソフトでも非線形半正定値計画問題と局所的最適解が同値となる非線形計画問題を解くことが出来るが，問題の規模が大きくなると求解時間が膨大になってしまうことがある．

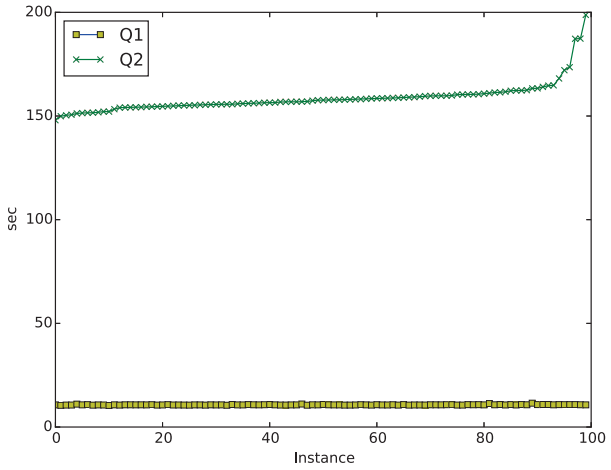

図 1　lmsdp と lsqp の求解速度の比較グラフ ($m = 50$)，反復回数 1500 回

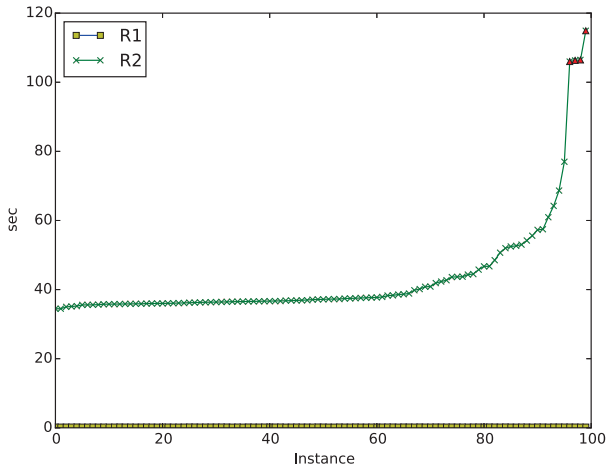

図 2　trsdp と bfgs の求解速度の比較グラフ ($m = 20$，反復回数 500 回)

参考文献

[1] Faybusovich, L.: Several Jordan-algebraic aspects of optimization, *Optimization*, Vol. 57, No. 3, pp. 379–393 (2008).
[2] Fiacco, A. and McCormick, G.: *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Classics in Applied Mathematics, SIAM (1990).
[3] Fiala, J., Kočvara, M. and Stingl, M.: PENLAB: A MATLAB solver for nonlinear semidefinite optimization, *ArXiv e-prints* (2013).
[4] Fukuda, E. H. and Fukushima, M.: The Use of Squared Slack Variables in Nonlinear Second-Order Cone Programming, *Journal of Optimization Theory and Applications*, Vol. 170, No. 2, pp. 394–418 (2016).
[5] Lourenço, B. F., Fukuda, E. H. and Fukushima, M.: Optimality conditions for nonlinear semidefinite programming via squared slack variables, *To appear in Mathematical Programming*, pp. 1–24 (2016).
[6] Luenberger, D.: *Linear and Nonlinear Programming*, Addison-Wesley (1973).
[7] Yamashita, H. and Yabe, H.: A survey of numerical methods for nonlinear semidefinite programming, *Journal of the Operations Research Society of Japan*, Vol. 58, No. 1, pp. 24–60 (2015).
[8] 株式会社 NTT データ数理システム：14.2 アルゴリズム一覧，`https://www.msi.co.jp/nuopt/docs/v18/manual/html/14-02-00.html` 2016/7/20 参照.
[9] 株式会社 NTT データ数理システム：数理計画法 パッケージ Numerical Optimizer，`https://www.msi.co.jp/nuopt/` 2016/7/20 参照.
[10] エレン秀美福田，雅夫福島：2 次錐計画と 2 乗スラック変数法，オペレーションズ・リサーチ: 経営の科学，Vol. 59, No. 12, pp. 707–715（オンライン），入手先 ⟨http://ci.nii.ac.jp/naid/110009893081/⟩ (2014).