情報処理学会論文誌　数理モデル化と応用　Vol.10　No.3　1–9 (Dec. 2017)

非線形半正定値計画問題の異なる定式化と複数アルゴリズムによる数値的比較

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

受付日 2017年1月30日，再受付日 2017年3月28日 / 2017年5月2日，
採録日 2017年5月11日

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非線形計画問題について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．現在のところ非線形半正定値計画問題を直接扱うソルバーは少ない．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

Computational Comparison of Different Formulations for Nonlinear Semidefinite Programming Problems Using Several Algorithms

Takumi Kato[1,a)]　Bruno F. Lourenço[1,b)]　Atsuko Ikegami[1,c)]

Received: January 30, 2017, Revised: March 28, 2017/May 2, 2017,
Accepted: May 11, 2017

Abstract: Semidefinite programming is a far-reaching class of optimization problems with many applications in engineering, statistics and computer science. Here, our main object of interest is the so-called *nonlinear semidefinite programming problem* (NSDP), where we wish to minimize an arbitrary differentiable function subject to positive semidefinitene constraints. In this work, we examine the computational prospects of transforming an NSDP into a classical nonlinear program through the usage of squared slack variables, and we perform three computational experiments.

Keywords: nonlinear semidefinite programming, square slack variables, nonlinear programming, optimization

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], [7]．非線形計画問題は以下のような問題である．

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

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 Semidefi-**

[1] 成蹊大学
Seikei University, Musashino, Tokyo 180–8633, Japan
[a)] takumi@cleo.ci.seikei.ac.jp
[b)] lourenco@st.seikei.ac.jp
[c)] atsuko@st.seikei.ac.jp

nite Programming，NSDP）という問題クラスが注目されている．NSDP は以下の問題である．

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

ここで 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．

半正定値制約のために，NSDP の形で定式化できる問題は 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 [12] 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 [14] のみである．

NSDP の中で，半正定値計画問題（**Semidefinite Programming，SDP**）という問題クラスがある．SDP の場合，f と G は両方とも線形関数である．SDP とのその応用はいくつかのサーベイに述べられている [9], [11]．SDP を取り扱えるソルバーが複数ある [4], [8], [10] とはいえ，一般の NSDP を直接に取り扱えるソルバーは非常に少ない．

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，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非線形計画問題 (NLP) を以下に示す．

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

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できる．したがって NLP のソルバーを用いて解くことができるようになる．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 の場合にどうなるかを調べる．

2. 研究の目的

本研究では NSDP の定式化と再定式化した問題との求解速度や精度，安定性を比較する．そして，再定式化した問題は安定的に最適解を求めることができるかどうか，問題の規模によってはどの程度の時間を要するのかを分析する．比較するには同じ環境での実験が必要不可欠であるので，非線形半正定値計画ソルバーを実装している Numerical Optimizer を使用する．Numerical Optimizer に NSDP，NLP に対するアルゴリズムが複数あるが本研究では比較を行う際に異なる手法のアルゴリズムどうしを比較することもある．

2.1 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る [5], [15]．SOCP と同じことを NSDP にも適用できるかどうかを試した研究もすでに存在し [6]，この研究では PENLAB/PENNON を使用して実験を行っている．PENLAB/PENNON は 1 つのアルゴリズム（Augmented Lagrangian Method）のみで NSDP と NSDP-SLACK を解くことができるが，これとは別のアルゴリズムを使用した場合，2 乗スラック変数法が求解にどのような影響を与えるかが，先行研究からだけでは予想しにくい．これに対し，本研究では Numerical Optimizer を利用して実験を行う．Numerical Optimizer では Augmented Lagrangian Method とは異なるアルゴリズムを適用できる．NSDP に対して 3 つのアルゴリズム，NSDP-SLACK には 5 つのアルゴリズムで解くことができる．複数アルゴリズムを実装していると，より細かい実験を行えると同時に，NSDP での最良のアルゴリズムと NSDP-SLACK での最良のアルゴリズムとの間で求解速度や精度を比較することができる．

本研究では，先行研究に加えて，2 乗スラック変数法が求解にどのような影響を与えているかを確かめることを目的とする．

3. 実験

本実験では 3 つの問題を扱う．1 つ目の問題は規模が小さく，2 つ目，3 つ目の問題は 1 つ目より規模の大きい問題である．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 @ 3.50 GHz，メモリ 32.0 GB である．

Numerical Optimizer は暫定解による KKT 条件の外れ量が 10^{-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した場合を最適化に成功した（Success）とする．

実験で使用する NLP（非線形計画法）のアルゴリズムは下記のものである [13]．

- bfgs：準ニュートン法．
- lsqp：直線探索法に基づく逐次 2 次計画法．
- slpsqp：逐次線形 2 次計画法．
- tipm：（新版）信頼領域内点法．

- tsqp：信頼領域法での逐次 2 次計画法.

NSDP（非線形半正定値計画法）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 1

2 乗スラック変数法で解けるかを調べるために，最初に規模の小さい問題を解いた．以下の問題は規模の小さい問題だが，目的関数と制約式が非凸なので，NSDP の汎用ソルバーに対して適したテストとして考えられる．特に，PENOPT/PENLAB には，この問題が搭載されている．ここでは，この問題を (NSDP1) とする.

$$\begin{aligned}
\underset{x_1,x_2,x_3,x_4,x_5,x_6}{\text{minimize}} \quad & x_1x_4(x_1+x_2+x_3)+x_3 && \text{(NSDP1)}\\
\text{subject to} \quad & x_1x_2x_2x_3x_4-x_5-25=0\\
& x_1^2+x_2^2+x_3^2+x_4^2-x_6-40=0\\
& \begin{pmatrix} x_1 & x_2 & 0 & 0\\ x_2 & x_4 & x_2+x_3 & 0\\ 0 & x_2+x_3 & x_4 & x_3\\ 0 & 0 & x_3 & x_1 \end{pmatrix} \in S_+^4\\
& 1\le x_i\le 5, \qquad i=1,2,3,4\\
& x_i\ge 0, \qquad i=5,6
\end{aligned}$$

問題 (NSDP1) を 2 乗スラック変数法を用いて再定式化した問題を (NLP1) とする.

$$\begin{aligned}
\underset{x_1,x_2,x_3,x_4,x_5,x_6,Y}{\text{minimize}} \quad & x_1x_4(x_1+x_2+x_3)+x_3 && \text{(NLP1)}\\
\text{subject to} \quad & x_1x_2x_2x_3x_4-x_5-25=0\\
& x_1^2+x_2^2+x_3^2+x_4^2-x_6-40=0\\
& \begin{pmatrix} x_1 & x_2 & 0 & 0\\ x_2 & x_4 & x_2+x_3 & 0\\ 0 & x_2+x_3 & x_4 & x_3\\ 0 & 0 & x_3 & x_1 \end{pmatrix} - Y\circ Y=0\\
& 1\le x_i\le 5, \qquad i=1,2,3,4\\
& x_i\ge 0, \qquad i=5,6
\end{aligned}$$

3.1.1　実験結果と分析

表 1 は最適化に成功したアルゴリズムの目的関数値と求解速度の表である.

問題 (NSDP1) の目的関数値と問題 (NLP1) の目的関数値が近い数値が出ているので，実験は成功した．この実験は規模が一定で小さいものなので，さらに規模の大きい実験をしていく.

表 1　目的関数値と求解速度の比較表
Table 1　Comparison of objective values and running times.

	Algorithm	Value of Objective	Time (s)
(NSDP1)	trsdp	87.71098	0.02
(NLP1)	bfgs	89.23832	0.08
	tipm	87.71049	0.06

3.2　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 に最も近い相関行列を見つける問題である．この問題を (NSDP2) とする.

$$\begin{aligned}
\underset{X}{\text{minimize}} \quad & \langle X-H, X-H\rangle && \text{(NSDP2)}\\
\text{subject to} \quad & X_{ii}=1, \qquad i\in\{1,\dots,m\}\\
& X\in S_+^m
\end{aligned}$$

行列 $H\in S^m$ 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.

問題 (NSDP2) を 2 乗スラック変数法を用いて再定式化した問題を (NLP2) とする.

$$\begin{aligned}
\underset{X}{\text{minimize}} \quad & \langle (X\circ X)-H, (X\circ X)-H\rangle && \text{(NLP2)}\\
\text{subject to} \quad & (X\circ X)_{ii}=1, \qquad i\in\{1,\dots,m\}\\
& X\in S^m
\end{aligned}$$

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 を生成し実験を行った．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．問題 (NSDP2) と問題 (NLP2) を比較する際には H は同じものを使用する.

3.2.1　実験結果と分析

表 2，**表 3**，**表 4**，**表 5**，**表 6**，**表 7** 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）をアルゴリズムごと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（問題例）は除いている．比較実験において，$m=30,50$ 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 1,500 回にする.

表 8 は各 m に対して，各アルゴリズムが最適化に失敗した場合のその理由をまとめたものである．//// はすべてのデータが最適化に成功していることを表し，IE は内部エラー，OVR はアルゴリズムの反復回数上限の超過，E はエラーを表している．内部エラーは Numerical Optimizer のエラーのことである．エラーはアルゴリズムでの反復中に数値的問題が発生したことやアルゴリズムの失敗条件を満たしたことを示す．例をあげると，内点法で，双対ギャップが十分小さくならないときに，「エラー」として表す.

表 2　求解速度と安定性の比較表（$m = 5$，反復回数 150 回）
Table 2　Comparison of running times and success rates ($m = 5$, max iterations = 150).

	Algorithm	Max (s)	Min (s)	Mean (s)	Success (%)
(NSDP2)	lmsdp	0.018	0.006	0.00736	100
	qnsdp	0.032	0.013	0.01831	99
	trsdp	0.022	0.008	0.00993	100
(NLP2)	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 3　求解速度と安定性の比較表（$m = 10$，反復回数 150 回）
Table 3　Comparison of running times and success rates ($m = 10$, max iterations = 150).

	Algorithm	Max (s)	Min (s)	Mean (s)	Success (%)
(NSDP2)	lmsdp	0.029	0.009	0.01104	100
	qnsdp	0.203	0.031	0.06118	85
	trsdp	0.036	0.012	0.01462	100
(NLP2)	bfgs	0.153	0.095	0.10295	99
	lsqp	0.121	0.076	0.08437	100
	slpsqp	1.155	0.073	0.13989	99
	tipm	0.288	0.049	0.11323	100
	tsqp	0.316	0.059	0.10028	100

表 4　求解速度と安定性の比較表（$m = 15$，反復回数 150 回）
Table 4　Comparison of running times and success rates ($m = 15$, max iterations = 150).

	Algorithm	Max (s)	Min (s)	Mean (s)	Success (%)
(NSDP2)	lmsdp	0.039	0.02	0.02276	100
	qnsdp	0.302	0.061	0.13977	91
	trsdp	0.052	0.028	0.03233	100
(NLP2)	bfgs	0.633	0.558	0.59621	75
	lsqp	0.565	0.472	0.51561	100
	slpsqp	1.625	0.367	0.51552	100
	tipm	2.444	0.273	0.83137	99
	tsqp	2.756	0.315	0.68658	97

表 5　求解速度と安定性の比較表（$m = 20$，反復回数 150 回）
Table 5　Comparison of running times and success rates ($m = 20$, max iterations = 150).

	Algorithm	Max (s)	Min (s)	Mean (s)	Success (%)
(NSDP2)	lmsdp	0.098	0.067	0.07438	100
	qnsdp	0.656	0.202	0.32409	91
	trsdp	0.106	0.085	0.09655	100
(NLP2)	bfgs	—*1	—*1	—*1	0
	lsqp	—*1	—*1	—*1	0
	slpsqp	4.936	1.484	2.75865	99
	tipm	8.473	0.805	2.98175	100
	tsqp	12.386	1.323	2.25866	97

*1 すべてのデータが最適化に失敗している．

図 1，**図 2**，**図 3**，**図 4**，**図 5**，**図 6** のグラフは 2 つの問題で，最も安定性が高い（最適化成功率が一番高い）アルゴリズムどうしを比較しているグラフである．最適化成功率が同値の場合は計算時間の平均値が短い（良い）法を比較している．すべての H に対しての求解速度を比較して

表 6　求解速度と安定性の比較表（$m = 30$，反復回数 1,500 回）
Table 6　Comparison of running times and success rates ($m = 30$, max iterations = 1,500).

	Algorithm	Max (s)	Min (s)	Mean (s)	Success (%)
(NSDP2)	lmsdp	0.627	0.559	0.59885	100
	qnsdp	12.853	1.209	1.89242	100
	trsdp	0.733	0.664	0.70654	100
(NLP2)	bfgs	—*1	—*1	—*1	0
	lsqp	13.73	11.032	12.06689	100
	slpsqp	87.494	23.214	32.42451	100
	tipm	42.106	10.046	22.18124	100
	tsqp	1168.427	9.905	55.51244	97

*1 すべてのデータが最適化に失敗している．

表 7　求解速度と安定性の比較表（$m = 50$，反復回数 1,500 回）
Table 7　Comparison of running times and success rates ($m = 50$, max iterations = 1,500).

	Algorithm	Max (s)	Min (s)	Mean (s)	Success (%)
(NSDP2)	lmsdp	11.495	10.416	10.71274	100
	qnsdp	49.704	20.532	28.15894	100
	trsdp	12.706	11.48	12.00278	100
(NLP2)	bfgs	—*1	—*1	—*1	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

*1 すべてのデータが最適化に失敗している．

表 8　アルゴリズムの最適化に失敗した原因一覧（反復回数 150 回，1,500 回）
Table 8　Overview of causes of failure (max iterations = 50, 1,500).

	Algorithm	5	10	15	20	30	50
(NSDP2)	lmsdp	////*1	////*1	////*1	////*1	////*1	////*1
	qnsdp	E*4	E*4	E*4	E*4	////	////
	trsdp	////	////	////	////	////	////
(NLP2)	bfgs	////	OVR*3	OVR*3	OVR*3	OVR*3	OVR*3
	lsqp	OVR*3	////	////	OVR*3	////	////
	slpsqp	IE*2	IE*2	////	IE*2	////	////
	tipm	////	////	OVR*3	////	////	////
	tsqp	////	////	OVR*3	OVR*3	OVR*3	OVR*3

*1 すべてのデータが最適化に成功している．
*2 内部エラー．
*3 アルゴリズムの反復上限の回数を超えた．
*4 エラーが発生し，計算を終了した．

いるグラフである．縦軸が求解速度（秒），横軸が NLP が短い時間で解けた方から何番目の H かを表している．問題 (NLP2) 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 2〜表 4（$m = 5, 10, 15$）では bfgs と lsqp はある程度最適化に成功しているが，表 5（$m = 20$）になると，どちらも 0%になっている．さらに規模の大きい問題を解く際にアルゴリズムの反復回数の上限を上げると，lsqp は最適化成功率が 100%になるのに対し，bfgs は 0%のままであった．

図 1（$m = 5$）で問題 (NSDP2) よりも問題 (NLP2) の方が求解速度が速いデータが 4 つあり，同程度の速度がいくつか

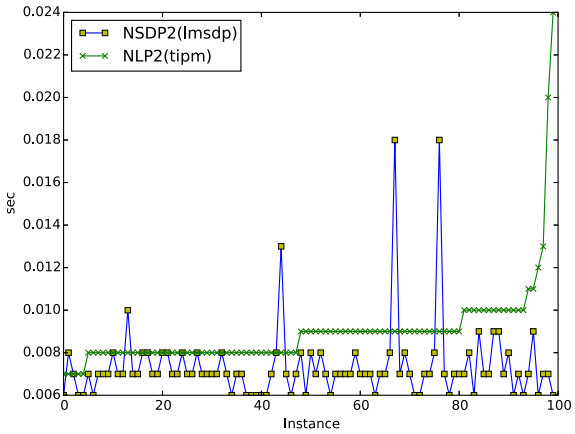

図 1　lmsdp と tipm の求解速度の比較グラフ（$m = 5$），反復回数 150 回

Fig. 1　Comparison between the running times of lmsdp and tipm ($m = 5$, max iterations = 150).

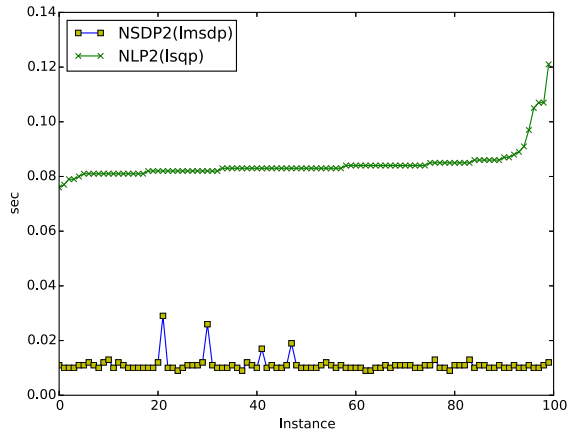

図 2　lmsdp と lsqp の求解速度の比較グラフ（$m = 10$），反復回数 150 回

Fig. 2　Comparison between the running times of lmsdp and lsqp ($m = 10$, max iterations = 150).

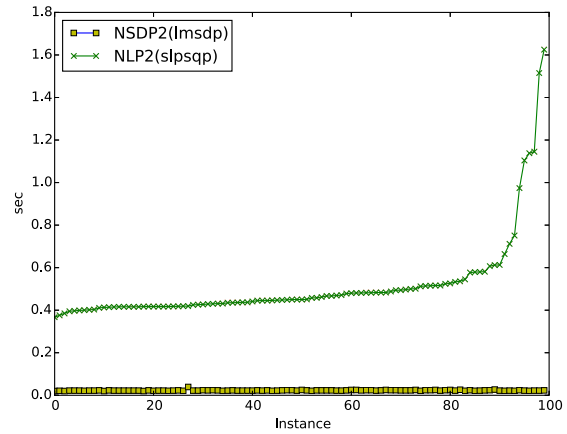

図 3　lmsdp と slpsqp の求解速度の比較グラフ（$m = 15$），反復回数 150 回

Fig. 3　Comparison between the running times of lmsdp and slpsqp ($m = 15$, max iterations = 150).

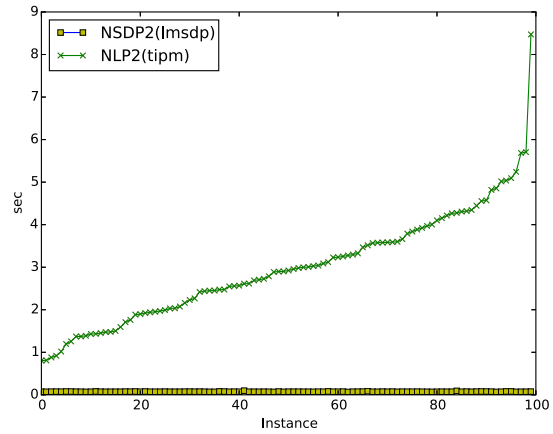

図 4　lmsdp と tipm の求解速度の比較グラフ（$m = 20$），反復回数 150 回

Fig. 4　Comparison between the running times of lmsdp and tipm ($m = 20$, max iterations = 150).

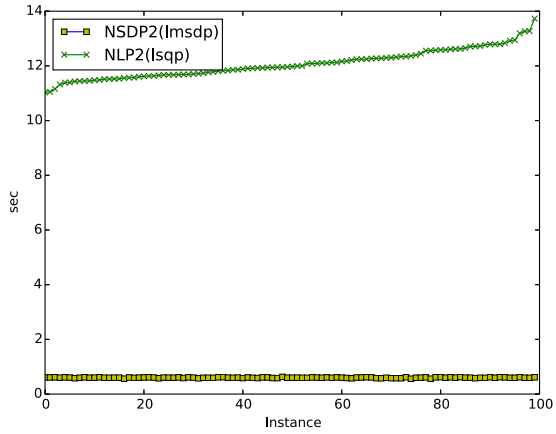

図 5　lmsdp と lsqp の求解速度の比較グラフ（$m = 30$，反復回数 1,500 回）

Fig. 5　Comparison between the running times of lmsdp and lsqp ($m = 30$, max iterations = 1,500).

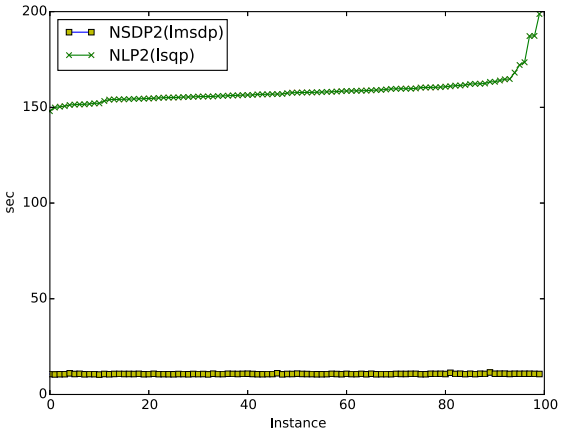

図 6　lmsdp と lsqp の求解速度の比較グラフ（$m = 50$，反復回数 1,500 回）

Fig. 6　Comparison between the running times of lmsdp and lsqp ($m = 50$, max iterations = 1,500).

あった．しかし, 図 1 の縦軸は非常に狭い幅なので, 速度差は誤差程度の差しかない．図 2〜図 6 ($m = 10, 15, 20, 30, 50$) ではすべてのデータで問題 (NSDP2) の求解速度が上だった．図 2〜図 6 ($m = 10, 15, 20, 30, 50$) では問題 (NSDP2) の求解速度においてはばらつきが少ない，問題 (NLP2) の求解速度はばらつきが多い．これは問題 (NLP2) の変数の数が問題 (NSDP2) 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．

3.3 実験 3

3.3.1 問題詳細

以下に問題 (NSDP3) を示す．

$$\begin{aligned} &\underset{X,z}{\text{minimize}} && \langle zX - H, zX - H\rangle && \text{(NSDP3)}\\ &\text{subject to} && zX_{ii} = 1, && i \in \{1, \ldots, m\}\\ &&& I_m \preceq X \preceq kI_m \end{aligned}$$

ここで，k は 1 より大きい正の数である．$X \succeq kI_m$ は $X - kI_m \in S^m_+$ を表す．行列 $H \in S^m$ 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．

問題 (NSDP3) を 2 乗スラック変数法を用いて再定式化した問題を (NLP3) とする．

$$\begin{aligned} &\underset{X,z,Y_1,Y_2}{\text{minimize}} && \langle zX - H, zX - H\rangle && \text{(NLP3)}\\ &\text{subject to} && zX_{ii} = 1, && i \in \{1, \ldots, m\}\\ &&& kI_m - X = Y_1 \circ Y_1\\ &&& X - I_m = Y_2 \circ Y_2 \end{aligned}$$

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．問題 (NSDP3) と問題 (NLP3) では H は同値のものを使用する．

3.3.2 実験結果と分析

表 9 はアルゴリズムの反復回数の上限を 150 回，**表 10**，**表 11**，**表 12**，**表 13** はアルゴリズムの反復回数の上限を 500 回にした際の求解速度の最大値 (Max)，最小値 (Min)，平均値 (Mean)，最適化成功率 (Success) をアルゴリズムごと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．

表 14，**表 15** は各 m に対して，アルゴリズムごとの最適化に失敗した理由をまとめたものである．表 8 と同じ表記を用いる．

表 9 ($m = 20$) では問題 (NLP3) の最適化成功率が低い．表 14 から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 回に設定して再実験を行った．

表 9　求解速度と安定性の比較表（$m = 20$，反復回数 150 回）
Table 9　Comparison of running times and success rates ($m = 20$, max iterations = 150).

	Algorithm	Max (s)	Min (s)	Mean (s)	Success (%)
(NSDP3)	lmsdp	1.063	0.786	0.90714	100
	qnsdp	3.449	1.403	1.82529	80
	trsdp	0.259	0.214	0.23033	100
(NLP3)	bfgs	43.472	33.423	36.39496	79
	lsqp	80.166	29.243	39.58897	79
	slpsqp	190.794	29.432	78.45750	22
	tipm	9.096	4.814	6.55762	55
	tsqp	67.627	17.998	31.07482	57

表 10　求解速度と安定性の比較表（$m = 5$，反復回数 500 回）
Table 10　Comparison of running times and success rates ($m = 5$, max iterations = 500).

	Algorithm	Max (s)	Min (s)	Mean (s)	Success (%)
(NSDP3)	lmsdp	0.037	0.011	0.01442	100
	qnsdp	0.081	0.021	0.03731	99
	trsdp	0.027	0.013	0.01575	100
(NLP3)	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 11　求解速度と安定性の比較表（$m = 10$，反復回数 500 回）
Table 11　Comparison of running times and success rates ($m = 10$, max iterations = 500).

	Algorithm	Max(s)	Min(s)	Mean(s)	Success (%)
(NSDP3)	lmsdp	0.162	0.031	0.04276	100
	qnsdp	0.463	0.071	0.14467	99
	trsdp	0.056	0.028	0.03091	100
(NLP3)	bfgs	1.536	0.437	0.54758	100
	lsqp	1.467	0.357	0.51828	99
	slpsqp	10.691	0.441	1.97075	32
	tipm	2.126	0.171	0.53929	92
	tsqp	3.523	0.492	0.86525	93

表 12　求解速度と安定性の比較表（$m = 15$，反復回数 500 回）
Table 12　Comparison of running times and success rates ($m = 15$, max iterations = 500).

	Algorithm	Max (s)	Min (s)	Mean (s)	Success (%)
(NSDP3)	lmsdp	0.29	0.157	0.2024	100
	qnsdp	2.02	0.319	0.65745	100
	trsdp	0.09	0.069	0.07956	100
(NLP3)	bfgs	21.50	4.956	6.03665	96
	lsqp	23.18	4.269	6.67216	94
	slpsqp	24.49	4.511	11.15871	28
	tipm	5.90	0.929	2.28178	89
	tsqp	32.44	2.935	5.09243	90

表 10，表 11 ($m = 5, 10$) では qnsdp が最適化に失敗するデータがあるのに対して，より規模の大きい問題の表 12，表 13 ($m = 15, 20$) ではすべ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，

表 13　求解速度と安定性の比較表（$m = 20$，反復回数 500 回）
Table 13　Comparison of running times and success rates ($m = 20$, max iterations = 500).

	Algorithm	Max (s)	Min (s)	Mean (s)	Success (%)
(NSDP3)	lmsdp	1.063	0.79	0.91076	100
	qnsdp	11.628	1.39	2.74977	100
	trsdp	0.252	0.21	0.22988	100
(NLP3)	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 14　アルゴリズムの最適化に失敗した原因一覧（反復回数 150 回）
Table 14　Overview of the causes of failure (max iterations = 150).

	Algorithm	5	10	15	20
(NSDP3)	lmsdp	////*1	////*1	////*1	////*1
	qnsdp	E*4	E*4	E*4	E*4
	trsdp	////*1	////*1	////*1	////*1
(NLP3)	bfgs	OVR*3	OVR*3	OVR*3	OVR*3
	lsqp	OVR*3	E*4	OVR*3&E*4	OVR*3&E*4
	slpsqp	IE&OVR*3	IE&OVR*3	IE&OVR*3	IE&OVR*3
	tipm	OVR*3	OVR*3	OVR*3	OVR*3
	tsqp	E*4	OVR*3&E*4	OVR*3&E	OVR*3&E

*1 すべてのデータが最適化に成功している．
*2 内部エラー．
*3 アルゴリズムの反復上限の回数を超えた．
*4 エラーが発生し，計算を終了した．

表 15　アルゴリズムの最適化に失敗した原因一覧（反復回数 500 回）
Table 15　Overview of the causes of failure (max iterations = 500).

	Algorithm	5	10	15	20
(NSDP3)	lmsdp	////*1	////*1	////*1	////*1
	qnsdp	E*4	E*4	////*1	////*1
	trsdp	////*1	////*1	////*1	////*1
(NLP3)	bfgs	////*1	////*1	OVR*3	OVR*3
	lsqp	OVR*3	E *4	OVR*3&E*4	OVR*3&E*4
	slpsqp	IE*2&OVR*3	IE*2&OVR*3	IE*2&OVR*3	IE*2&OVR*3
	tipm	////*1	OVR*3	OVR*3	OVR*3
	tsqp	E*4	OVR*3&E*4	OVR*3&E*4	OVR*3&E*4

*1 すべてのデータが最適化に成功している．
*2 内部エラー．
*3 アルゴリズムの反復上限の回数を超えた．
*4 エラーが発生し，計算を終了した．

qnsdp は上がっている．

図 8，**図 9**，**図 10**（$m = 10, 15, 20$）では求解速度が最も遅い問題では最適化に失敗している．すべての図において問題 (NLP3) のグラフの形は非常に似ている．三角形のマーカーは最適化に失敗していることを表している．

3.4　分析

3 つの実験を通して NSDP を NLP の形にして解くことができた．

実験 2 と実験 3 の各アルゴリズムの最適化成功率を m の値ごとにまとめたものが**表 16** である．この表から実験 2 ではすべての m に対して，NSDP の最適化成功率が

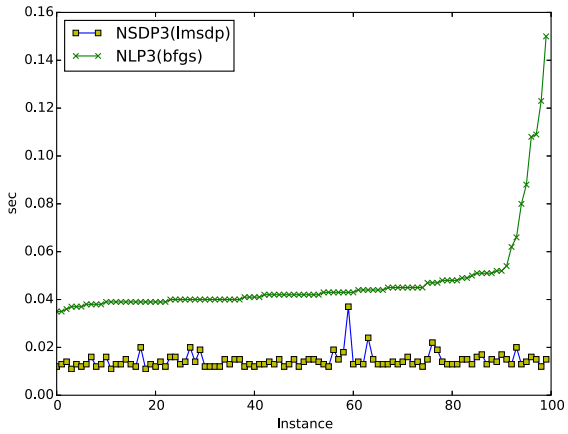

図 7　lmsdp と bfgs の求解速度の比較グラフ（$m = 5$，反復回数 500 回）
Fig. 7　Comparison between the running times of lmsdp and bfgs ($m = 5$, max iterations = 500).

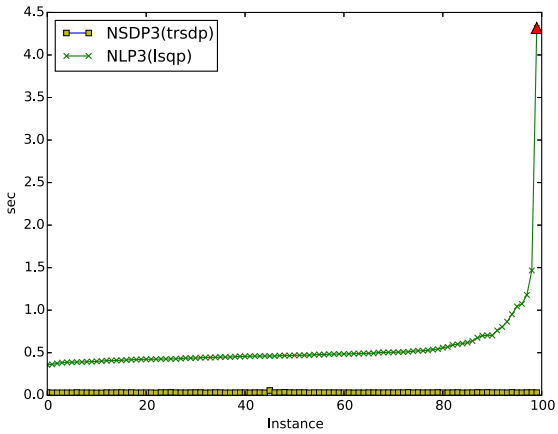

図 8　trsdp と lsqp の求解速度の比較グラフ（$m = 10$，反復回数 500 回）
Fig. 8　Comparison between the running times of trsdp and lsqp ($m = 10$, max iterations = 500).

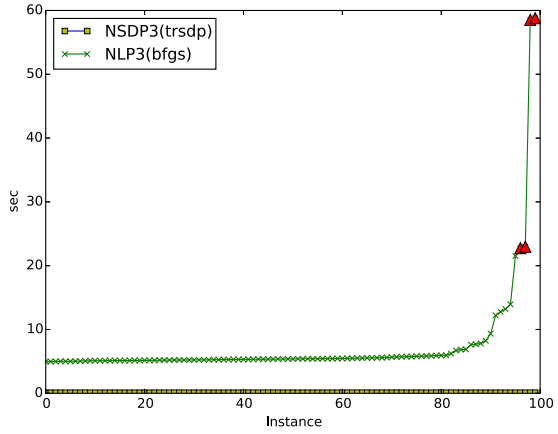

図 9　trsdp と bfgs の求解速度の比較グラフ（$m = 15$，反復回数 500 回）
Fig. 9　Comparison between the running times of trsdp and bfgs ($m = 15$, max iterations = 500).

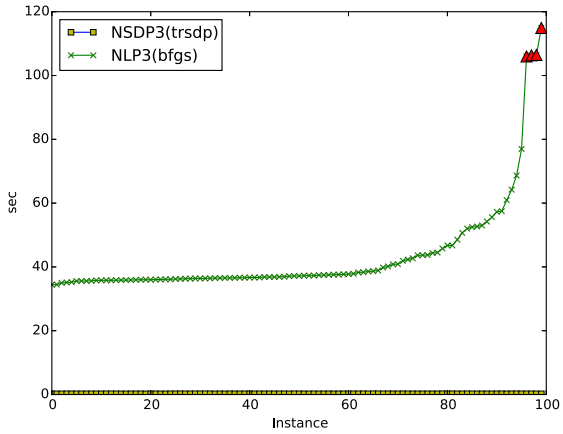

図 10　trsdp と bfgs の求解速度の比較グラフ（$m = 20$，反復回数 500 回）

Fig. 10　Comparison between the running times of trsdp and bfgs ($m = 20$, max iterations = 500).

表 16　各アルゴリズムの最適化成功率

Table 16　Overview of the success rates for each algorithm.

	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

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できるかもしれない．

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の非線形計画問題を解くことができるが，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．

そして Numerical Optimizer の非線形半正定値ソルバーは非常に高い安定性と求解速度を有していることも分かった．

本論文の数値実験結果と先行研究 [6] から，非線形半正定値計画問題を 2 乗スラック変数法で再定式化して非線形計画問題として解くことは，求解の可能性を広げるものの，求解時間を増加させる可能性があることが分かった．さらに，2 乗スラック変数法では誤差の影響は考慮する必要があると思われる．

今後，非線形半正定値計画問題の研究が進み，様々なアルゴリズムを汎用数理計画ソフトに実装することで直接的に非線形半正定値計画問題を解くことができるようになれば，さらに多くの現実問題が数理計画問題として解けるようになっていくと考えられる．

謝辞　本論文に貴重なご助言をいただいた査読委員の先生方に心より感謝いたします．本研究の一部は，JSPS 科研費 26350435 の助成を受けたものです．

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] Fujisawa, K., Nakata, K., Yamashita, M. and Fukuda, M.: SDPA project: Solving large-scale semidefinite programs, *Journal of the Operations Research Society of Japan*, Vol.50, No.4, pp.278–298 (2007).
[5] 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).
[6] Lourenço, B.F., Fukuda, E.H. and Fukushima, M.: Optimality conditions for nonlinear semidefinite programming via squared slack variables, *Mathematical Programming*, pp.1–24 (online), DOI: 10.1007/s10107-016-1014-4 (2017).
[7] Luenberger, D.: *Linear and Nonlinear Programming*, Addison-Wesley (1973).
[8] Sturm, J.F.: Using SeDuMi 1.02, A Matlab toolbox for optimization over symmetric cones, *Optimization Methods and Software*, Vol.11, No.1-4, pp.625–653 (1999).
[9] Todd, M.J.: Semidefinite optimization, *Acta Numerica*, Vol.10, pp.1–41 (2001).
[10] Toh, K.-C., Todd, M.J. and Tütüncü, R.: On the Implementation and Usage of SDPT3 - A Matlab Software Package for Semidefinite-Quadratic-Linear Programming, Version 4.0, *Handbook on Semidefinite,*

Conic and Polynomial Optimization, Anjos, M.F. and Lasserre, J.B. (Eds.), Springer US, pp.715–754 (2012).
[11] Vandenberghe, L. and Boyd, S.: Semidefinite Programming, *SIAM Review*, Vol.38, No.1, pp.49–95 (1996).
[12] 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).
[13] 株式会社 NTT データ数理システム：14.2 アルゴリズム一覧，入手先 ⟨https://www.msi.co.jp/nuopt/docs/v18/manual/html/14-02-00.html⟩（参照 2016-07-20）.
[14] 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, 入手先 ⟨https://www.msi.co.jp/nuopt/⟩（参照 2016-07-20）.
[15] エレン福田秀美，福島雅夫：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).

加藤 拓海

2017 年成蹊大学理工学部情報科学科卒業．非線形半正定値計画問題に興味を持ち，本研究を進めた．

ブルノ フィゲラ ロウレンソ

2011 年ブラジリア大学計算機科学部卒業．2012 年ブラジリア大学数学科修士課程修了．2016 年東京工業大学大学院数理計算科学専攻博士課程修了．2016 年より成蹊大学助教．数理最適化研究に従事．日本 OR 学会正会員．

池上 敦子　（正会員）

立教大学理学部数学科卒業．成蹊大学大学助手，講師，准教授を経て，2009 年同大教授．現実問題のモデル化とアルゴリズム構築に興味を持ち，組合せ最適化研究に従事．博士（工学）．1997 年日本 OR 学会事例研究奨励賞，2003 年日本人間工学会研究奨励賞，2004 年日本 OR 学会事例研究賞，2008 年スケジューリング学会技術賞各受賞．日本 OR 学会フェロー．