

修 士 論 文 の 和 文 要 旨

研究科・専攻	大学院 情報理工 学研究科 情報・通信工学 専攻 博士前期課程		
氏 名	香村 友宏	学籍番号	1531039
論 文 題 目	錐線形計画に対する面削減法の実装の試み		
<p>要 旨</p> <p>近年、半正定値計画問題や二次錐計画問題などの錐線形計画がよく研究されている。錐線形計画においては双対ギャップがある場合、最適値は存在するが最適解が存在しない場合など、通常の線形計画にはありえない状況が起こる場合がある。もし錐線形計画に内点許容解がある場合には、双対ギャップがなく、双対問題には最適解が存在することが知られている。</p> <p>面削減法とは、内点許容解がない錐線形計画問題が与えられたときに、錐をある面に制限することにより、内点許容解を持つ問題に変換しようとするものである。しかし、面削減法は数値誤差に弱い手法である。これに対し、Lourenço, Muramatsu, Tsuchiya は削減方向の計算時に、主問題と双対問題の両方に内点許容解が存在するような新しい定式化(LMT)を用いることを提案している。LMT は、従来の計算法よりも数値的に安定して解けることが期待される。</p> <p>本論文の目的は、LMT により削減方向を計算することが、どの程度有用なのかを数値計算により確認すること、また、面削減法の実装において数値誤差に対する弱さを克服できないのか検討することである。</p> <p>結果として、主問題が強実行可能かつ双対問題が弱実行可能な半正定値計画問題について、削減方向の計算に従来の定式化と LMT のどちらを用いても概ね正しい削減方向が計算できることや、面削減法の方がその他の方法よりも安定して解くことができることなどが判明した。</p> <p>また、弱実行不可能な半正定値計画問題について、既存のソルバをそのまま用いるよりも面削減法を用いた方が適していることや、削減方向の計算には LMT の方が適していることなどが判明した。</p> <p>加えて、実行可能だが双対ギャップが存在する半正定値計画問題について、面削減法は解くのに適しているとは断言できないことや、削減方向の計算方法は、解くことができる問題数には関係が無いことなどが判明した。</p>			

平成 28 年度 電気通信大学 情報・通信工学専攻 修士論文

錐線形計画に対する面削減法の実装の試み

指導教員 村松正和 教授

平成 29 年 3 月 12 日

電気通信大学大学院 情報理工学研究科 情報・通信工学専攻

1531039

香村友宏

目次

1	はじめに	4
1.1	背景および研究内容	4
1.2	構成	5
2	基礎知識	6
2.1	錐線形計画	6
2.1.1	線形計画問題	7
2.1.2	2次錐計画問題	7
2.1.3	半正定値計画問題	8
2.1.4	自己双対形	9
2.2	錐線形計画問題の実行可能性について	13
2.3	面削減法	15
2.3.1	面削減法	15
2.3.2	半正定値計画問題に対する面削減法の適用	15
2.3.3	削減方向の計算	16
2.3.4	小さな問題の生成	18
2.3.5	錐拡大法	20
3	実験1：弱実行可能な場合	22
3.1	実験内容	22
3.1.1	実験の手順	22
3.1.2	問題の生成方法	23
3.1.3	実験詳細	24
3.1.4	実験環境	24
3.2	結果	24
3.2.1	削減方向の計算	24
3.2.2	$N = 10$ の場合	25
3.2.3	$N = 20$ の場合	27
3.2.4	ソルバの終了条件の変更	29
3.2.5	ソルバをそのまま用いる場合	30
4	実験2：弱実行不可能な場合	32
4.1	実験内容	32
4.1.1	文献 [5] の実験内容	32
4.2	結果	33
4.2.1	削減方向の計算	33
4.2.2	面削減法および錐拡大法の適用	34
4.2.3	文献 [5] との比較	35

4.2.4	ソルバの終了条件の変更	36
5	実験 3 : 双対ギャップが存在する場合	37
5.1	実験内容	37
5.1.1	問題の生成方法	37
5.1.2	生成した問題	38
5.2	結果	38
5.2.1	面削減法および錐拡大法の適用	38
5.2.2	ソルバをそのまま用いる場合	39
5.2.3	双対ギャップ	39
6	結論	41

1 はじめに

1.1 背景および研究内容

近年，半正定値計画問題や二次錐計画問題などの錐線形計画がよく研究されている．例えば，田村・村松は文献 [1] において

効率的解法の存在と適用範囲の広さの点で，錐線形計画は線形計画に通じるものがあり，「21 世紀の線形計画」とよばれることがある

と述べており，同文献内では半正定値計画問題に対する主双対内点法や半正定値計画問題の組合せ最適化への応用例が紹介されている．

しかし，錐線形計画においては双対ギャップがある場合，最適値は存在するが最適解が存在しない場合など，通常の線形計画にはありえない状況が起こる場合がある．

もし錐線形計画に内点許容解がある場合には，文献 [1] にて

主問題 (4.2) に内点許容解が存在し，かつ，双対問題 (4.5) に許容解が存在すれば，最適値は一致し，双対問題には最適解が存在する．

と述べられている通り，双対ギャップがなく，双対問題には最適解が存在することが知られている．しかしながら，現実には内転許容解が無い問題を解かなければならないことがしばしばある．

面削減法とは，内点許容解がない錐線形計画問題が与えられたときに，錐をある面に制限することにより，内点許容解を持つ問題に変換しようとするものである．例えば，自己双対形の問題の解は，ある 2 つの変数の値によって 2 通りに場合分けできることは一般的である．それらの変数が実際に取りうる値のパターンとしては 3 通りが考えられるが，残りの 1 通りは実行可能なのか不可能なのかがよく分からないものとして考えられていた．そういった場合に面削減法を用いることで，実行可能か否かが不明な最適化問題の実行可能性を判定できるようになる．

面削減法は Wolkovitz らにより，初めて凸計画問題に対して提案された．続いて，Waki and Muramatsu により，錐線形計画問題に対する面削減法が提案された．

面削減法は，制限する面を精度よく指定しないと，初期の問題の実行可能解が失われたり，逆に実行可能解がないのに実行可能になってしまったり，ということが起こりうる．Permenter ら [2] は面削減法の数値計算を行っているが，うまく計算できない場合が多数あることが報告されている．面削減法は数値誤差に弱い手法であると言える．

これに対し，Lourenço, Muramatsu, Tsuchiya(LMT)[4] は削減方向の計算時に，主問題と双対問題の両方に内点許容解が存在するような新しい定式化を用いることを提案している．前述の通り，主問題と双対問題の両方に内点許容解が存在する場合には，主問題と双対問題の最適値は一致し，両方に最適解が存在する．また，主問題と双対問題の双方に内点許容解が存在することは，主双対内点法が収束するための条件でもある．したがって，この新しい定式化は，従来の計算法よりも数値的に安定して解けることが期待される．

本論文の目的は、LMT らの新しい定式化により削減方向を計算することが、どの程度有用なのかを数値計算により確認すること、また、面削減法の実装において数値誤差に対する弱さを克服できないのか検討することである。

本論文では一貫して錐線形計画問題として半正定値計画問題を用いる。半正定値計画問題は錐線形計画問題の中でも、特に重要な位置を占めているからである。

1.2 構成

本稿の構成は、以下の通りである。第2章では、本稿を読むうえで必要になる知識について述べる。第3章では、弱実行可能な半正定値計画問題に対して行った実験について述べる。第4章では、弱実行不可能な半正定値計画問題に対して行った実験について述べる。第5章では、実行可能だが双対ギャップが存在する半正定値計画問題に対して行った実験について述べる。そして第6章では、本研究の結論を述べる。

2 基礎知識

2.1 錐線形計画

定義 1 集合 $\mathcal{K} \subseteq \mathbb{R}^n$ の任意の要素 $\mathbf{x} \in \mathcal{K}$ が, 任意の $\lambda > 0$ に対して $\lambda \mathbf{x} \in \mathcal{K}$ を満たすとき, \mathcal{K} を錐と呼ぶ. また, 凸集合である錐を凸錐, 閉集合かつ凸集合であるような錐を閉凸錐と呼ぶ.

今, 集合 $\mathcal{K} \subseteq \mathbb{R}^n$ を非空な閉凸錐, \mathbf{x} を決定変数として, 以下の最適化問題を考える:

$$\begin{cases} \min \langle \mathbf{c}, \mathbf{x} \rangle \\ \text{s.t. } \langle \mathbf{a}_i, \mathbf{x} \rangle = b_i \ (i = 1, \dots, m) \\ \mathbf{x} \in \mathcal{K}. \end{cases} \quad (1)$$

ただし, $\langle \cdot, \cdot \rangle$ は内積を表す. (1) 式において, $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{a}_i \in \mathbb{R}^n$ ($i = 1, \dots, m$), $b_i \in \mathbb{R}$ ($i = 1, \dots, m$) である. (1) 式のように, 目的関数が線形であり, 条件が線形制約および錐制約のみで構成されるような最適化問題を錐線形計画問題という. (1) 式において, 内積が $\mathbf{a}_i^T \mathbf{x}$ ($i = 1, \dots, m$) であるとき,

$$A = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_m^T \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

とおけば, (1) 式は以下のように変形できる:

$$\begin{cases} \min \mathbf{c}^T \mathbf{x} \\ \text{s.t. } A\mathbf{x} = \mathbf{b} \\ \mathbf{x} \in \mathcal{K}. \end{cases} \quad (2)$$

(2) 式の双対問題を考えると, スラック変数 \mathbf{s} および決定変数 \mathbf{y} , 双対錐 \mathcal{K}^* を用いて以下のように記述することができる:

$$\begin{cases} \max \mathbf{b}^T \mathbf{y} \\ \text{s.t. } \mathbf{s} = \mathbf{c} - A^T \mathbf{y} \\ \mathbf{s} \in \mathcal{K}^*, \mathbf{y} \in \mathbb{R}^m. \end{cases} \quad (3)$$

ここで, $\mathcal{K}^* := \{\mathbf{s} \in \mathbb{R}^n \mid \mathbf{s}^T \mathbf{x} \geq 0, \forall \mathbf{x} \in \mathcal{K}\}$ である.

(1) 式において, \mathcal{K} の内点であり, かつ許容解であるような点を内点許容解と呼ぶ. また, (2) 式において, \mathcal{K}^* の内点であり, かつ許容解であるような点を内点許容解と呼ぶ. このとき, 以下の定理が成り立つ:

定理 1 主問題と双対問題の両方に内点許容解が存在する場合、主問題と双対問題の最適値が一致し、最適解が存在する。

証明は文献 [1] を参照。

以上のような錐線形計画問題は、問題の形によって通常の線形計画問題として扱うことができる場合がある。また、特殊な場合として、二次錐計画問題、半正定値計画問題と呼ばれる問題に分類できることがある。以下、この3つの場合について説明する。

2.1.1 線形計画問題

\mathbb{R}^n における第1象限を考える。今、(2)式における \mathcal{K} として第1象限をとる、すなわち $\mathcal{K} := \{\mathbf{x} \in \mathbb{R}^n | x_i \geq 0 (i = 1, \dots, n)\}$ であるとする。このとき、最適化問題は次のように書くことができる：

$$\begin{cases} \min \mathbf{c}^T \mathbf{x} \\ \text{s.t. } A\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0}. \end{cases} \quad (4)$$

\mathcal{K} は閉凸錐であるから、(4)式は錐線形計画問題である。また、(4)式の形は等式標準形となっている。したがって、錐 \mathcal{K} として第1象限を取った場合、(2)式は線形計画問題として扱うことができる。

(4)式の双対問題を考えると、以下のようになる：

$$\begin{cases} \max \mathbf{b}^T \mathbf{y} \\ \text{s.t. } A^T \mathbf{y} \geq \mathbf{c}. \end{cases} \quad (5)$$

2.1.2 2次錐計画問題

集合 $\mathcal{H}(N)$ を、以下のように定義する (N は自然数)：

$$\mathcal{H}(N) \stackrel{\text{def}}{=} \left\{ \mathbf{x} \in \mathbb{R}^N \mid x_1 \geq \sqrt{\sum_{i=2}^N x_i^2} \right\}. \quad (6)$$

このとき、 $\mathcal{H}(N)$ は閉凸錐である。このような $\mathcal{H}(N)$ を N 次元空間の2次錐と呼ぶ。今、 $N = \sum_{i=1}^p N_i$ を満たすような自然数 $N_i (i = 1, \dots, p)$ を考えると、(6)式の N 次元ベクトル \mathbf{x} は以下のように分解できる：

$$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p), \quad \mathbf{x}_i \in \mathbb{R}^{N_i} (i = 1, \dots, p). \quad (7)$$

このとき、集合 \mathcal{H} を以下のように定義する：

$$\mathcal{H} \stackrel{\text{def}}{=} \{ \mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p) \mid \mathbf{x}_i \in \mathcal{H}(N_i) (i = 1, \dots, p) \}. \quad (8)$$

また、(2) 式において錐 \mathcal{K} を \mathcal{H} に置き換えた式を考えると、以下のようになる：

$$\begin{cases} \min \mathbf{c}^T \mathbf{x} \\ \text{s.t. } A\mathbf{x} = \mathbf{b} \\ \mathbf{x} \in \mathcal{H}. \end{cases} \quad (9)$$

\mathcal{H} は閉凸錐であるから、(9) 式は錐線形計画問題である。このように、錐 \mathcal{K} として \mathcal{H} を用いた錐線形計画問題を 2 次錐計画問題という。

2.1.3 半正定値計画問題

$n \times n$ 実正方行列 X および任意のベクトル $\mathbf{y} \in \mathbb{R}^n$ に対して、

$$\mathbf{y}^T X \mathbf{y} \geq 0$$

が成り立つとき、 X を半正定値行列という。

今、 S^n を $n \times n$ 実対称行列の集合、 S_+^n を $n \times n$ 半正定値行列の集合とする。ここで、

$$\forall X, Y \in S^n, X \cdot Y \stackrel{\text{def}}{=} \text{tr}XY = \sum_{j=1}^n \sum_{i=1}^n X_{ij}Y_{ij} \quad (10)$$

と定義すると、 $C \in S^n$, $A_i \in S^n (i = 1, \dots, m)$, $b_i \in \mathbb{R} (i = 1, \dots, m)$ を用いて以下のような最適化問題を構築できる：

$$\begin{cases} \min C \cdot X \\ \text{s.t. } A_i \cdot X = b_i \ (i = 1, \dots, m) \\ X \in S_+^n. \end{cases} \quad (11)$$

ここで、(11) 式の大対問題は以下のように表される：

$$\begin{cases} \max \sum_{i=1}^m b_i y_i \\ \text{s.t. } S = C - \sum_{i=1}^m y_i A_i \\ S \in S_+^n. \end{cases} \quad (12)$$

(11) 式のように、決定変数が半正定値行列であるような錐線形計画問題を半正定値計画問題という。半正定値計画問題の効率的な解法として、主双対内点法¹と呼ばれる手法が存在する。本稿では錐線形計画のうち、特にこの半正定値計画について、後述の面削減法および錐拡大法の実装を試みている。

以上のような錐線形計画問題の解法の一つに、問題を自己双対形と呼ばれる形へと変換する方法が存在する。

¹詳細は文献 [1] を参照

2.1.4 自己双対形

錐線形計画において、行列 $A \in \mathbb{R}^{n \times n}$ 、ベクトル $\mathbf{c} \in \mathbb{R}^n$ 、決定変数 $\mathbf{x} \in \mathbb{R}^n$ 、閉凸錐 $\mathcal{K} \subseteq \mathbb{R}^n$ およびその双対錐 \mathcal{K}^* を用いて、次の形で表される最適化問題を、自己双対形という：

$$\begin{cases} \min \mathbf{c}^T \mathbf{x} \\ \text{s.t. } A\mathbf{x} + \mathbf{c} \in \mathcal{K}^* \\ \mathbf{x} \in \mathcal{K}. \end{cases} \quad (13)$$

ただし、 $(A = -A^T)$ である。任意の線形計画問題は自己双対形に変換できる (証明：[1])。任意の錐線形計画問題もまた自己双対形に変換できる。今、錐線形計画問題の主問題を (2) 式、双対問題を (3) 式として、以下の同次システムを考える：

$$\begin{cases} A\mathbf{x} - \mathbf{b}\tau = \mathbf{0} \\ -A^T \mathbf{y} - \mathbf{s} + \mathbf{c}\tau = \mathbf{0} \\ \mathbf{b}^T \mathbf{y} - \mathbf{c}^T \mathbf{x} - \kappa = 0 \\ (\mathbf{x}, \mathbf{s}, \mathbf{y}, \tau, \kappa) \in \mathcal{K} \times \mathcal{K}^* \times \mathbb{R}^m \times \mathbb{R}_+ \times \mathbb{R}_+. \end{cases} \quad (14)$$

\mathbb{R}_+ は非負の実数全体を表す。このとき、以下の定理が成り立つ。

定理 2 同次システム (14) 式は、自己双対形の錐線形計画問題である。

(証明) (14) 式より、以下の等式を考える。

$$\begin{aligned} & \begin{bmatrix} A & 0 & 0 & -\mathbf{b} & 0 \\ 0 & -I & -A^T & \mathbf{c} & 0 \\ -\mathbf{c}^T & 0 & \mathbf{b}^T & 0 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{s} \\ \mathbf{y} \\ \tau \\ \kappa \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ 0 \end{bmatrix} \quad (I \text{ は恒等行列}) \\ \Leftrightarrow & \begin{bmatrix} 0 & A & -\mathbf{b} & 0 & 0 \\ -A^T & 0 & \mathbf{c} & -I & 0 \\ \mathbf{b}^T & -\mathbf{c}^T & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \\ \tau \\ \mathbf{s} \\ \kappa \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ 0 \end{bmatrix} \\ \Leftrightarrow & \begin{bmatrix} 0 & A & -\mathbf{b} \\ -A^T & 0 & \mathbf{c} \\ \mathbf{b}^T & -\mathbf{c}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \\ \tau \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{s} \\ \kappa \end{bmatrix} \end{aligned}$$

上式は、(13) 式において $A = \begin{bmatrix} 0 & A & -\mathbf{b} \\ -A^T & 0 & \mathbf{c} \\ \mathbf{b}^T & -\mathbf{c}^T & 0 \end{bmatrix}$ 、 $\mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \\ \tau \end{bmatrix}$ 、 $\mathbf{c} = \mathbf{0}$ とおき、 \mathcal{K} を $\mathbb{R}^m \times \mathcal{K} \times \mathbb{R}_+$ 、 \mathcal{K}^* を $\{\mathbf{0}\} \times \mathcal{K}^* \times \mathbb{R}_+$ とおいた場合の形になっている。 ■

このとき、目的関数値は常に0であり、こういった場合には(14)式のように条件式のみを記述することがある。このような自己双対形の問題を、自己双対同次モデルという。

定理 3 同次システム(14)式の解 (x, y, s, τ, κ) に関して以下が成り立つ。

1. $\tau > 0 \wedge \kappa = 0$ のとき、 $\frac{x}{\tau}$, $\frac{1}{\tau}(s, y)$ がそれぞれ主問題(2)式および双対問題(3)式の最適解となる。
2. $\tau = 0 \wedge \kappa > 0$ のとき、 $b^T y > 0 \vee c^T x < 0$ が成り立つ。 $b^T y > 0$ のとき、主問題の許容解が存在しないため、主問題は実行不可能となる。また、 $c^T x < 0$ のとき、双対問題の許容解が存在しないため、双対問題は実行不可能となる。

(証明) まず、 τ および κ の取りうる値を示す。次に、1. および 2. が成り立つことを証明する。

同次システム(14)式において τ と κ の積を考えると、

$$\begin{aligned} \tau\kappa &= \tau(b^T y - c^T x) \\ &= \tau(\langle b, y \rangle - \langle c, x \rangle) \\ &= \langle b\tau, y \rangle - \langle c\tau, x \rangle \\ &= \langle Ax, y \rangle - \langle s + A^T y, x \rangle \\ &= -\langle s, x \rangle \\ &= -s^T x \end{aligned}$$

となる。ここで、 τ , κ は共に非負であり、 $s^T x$ もまた非負であるから、 $\tau\kappa = 0$ となる。よって、 τ と κ の取りうる値は

$$\begin{cases} \tau > 0 \wedge \kappa = 0 \\ \tau = 0 \wedge \kappa > 0 \\ \tau = \kappa = 0 \end{cases}$$

の3通りである。

1. $\tau > 0 \wedge \kappa = 0$

このとき、 $\frac{x}{\tau}$, $\frac{1}{\tau}(s, y)$ がそれぞれ主問題(2)式および双対問題(3)式の最適解となることを示す。実際、

$$\frac{x}{\tau} = \bar{x}$$

とおけば、 $x \in \mathcal{K}$ および $\tau > 0$ より、

$$\frac{x}{\tau} \in \mathcal{K}$$

である。よって、

$$\bar{\mathbf{x}} \in \mathcal{K}.$$

また、(14) 式より

$$A\mathbf{x} = \mathbf{b}\tau$$

であり、これと $\tau > 0$ より

$$A\bar{\mathbf{x}} = \mathbf{b}$$

である。したがって、 $\bar{\mathbf{x}}$ は主問題の許容解であり、その目的関数値は

$$\frac{\mathbf{c}^T \bar{\mathbf{x}}}{\tau}$$

となる。一方、

$$\frac{\mathbf{s}}{\tau} = \bar{\mathbf{s}}$$

とおけば、 $\langle \bar{\mathbf{s}}, \bar{\mathbf{x}} \rangle = \frac{1}{\tau^2} \langle \mathbf{s}, \mathbf{x} \rangle \geq 0$ より

$$\bar{\mathbf{s}} \in \mathcal{K}^*$$

である。また、

$$\frac{\mathbf{y}}{\tau} = \bar{\mathbf{y}}$$

とおけば、

$$\bar{\mathbf{y}} \in \mathbb{R}^m$$

である。(14) 式より

$$\mathbf{s} = \mathbf{c}\tau - A^T \mathbf{y}$$

であるので、

$$\bar{\mathbf{s}} = \mathbf{c} - A^T \bar{\mathbf{y}}$$

である。よって $\bar{\mathbf{s}}, \bar{\mathbf{y}}$ は双対問題の許容解であり、その目的関数値は

$$\frac{\mathbf{b}^T \bar{\mathbf{y}}}{\tau}$$

となる。今、 $\kappa = 0$ であるから、(14) 式より

$$\mathbf{c}^T \mathbf{x} = \mathbf{b}^T \mathbf{y}$$

である。即ち

$$\frac{\mathbf{c}^T \bar{\mathbf{x}}}{\tau} = \frac{\mathbf{b}^T \bar{\mathbf{y}}}{\tau}.$$

したがって、主問題と双対問題の目的関数値が等しいため、 $(\bar{\mathbf{x}}, \bar{\mathbf{s}}, \bar{\mathbf{y}})$ は主問題および双対問題の最適解となる。

2. $\tau = 0 \wedge \kappa > 0$

このとき,

$$\mathbf{b}^T \mathbf{y} > 0 \vee \mathbf{c}^T \mathbf{x} < 0$$

が成り立つことを示す. 実際,

$$\mathbf{b}^T \mathbf{y} \leq 0 \wedge \mathbf{c}^T \mathbf{x} \geq 0$$

であると仮定すると,

$$\mathbf{b}^T \mathbf{y} - \mathbf{c}^T \mathbf{x} \leq 0$$

であり, (14) 式および $\kappa > 0$ より

$$\mathbf{b}^T \mathbf{y} - \mathbf{c}^T \mathbf{x} > 0$$

であることと矛盾する.

$\mathbf{b}^T \mathbf{y} > 0$ のとき, 主問題に許容解 $\bar{\mathbf{x}}$ が存在すると仮定する. このとき, $\tau = 0$ より

$$A\bar{\mathbf{x}} = \mathbf{0}$$

であるが,

$$0 < \mathbf{b}^T \mathbf{y} = (A\bar{\mathbf{x}})^T \mathbf{y} = 0$$

となり矛盾が生じる. したがって, 主問題は実行不可能となる. また, $\mathbf{c}^T \mathbf{x} < 0$ のとき, 双対問題に許容解 $\bar{\mathbf{y}}$ が存在すると仮定すると, $\tau = 0$ より

$$\mathbf{s} + A^T \bar{\mathbf{y}} = \mathbf{0}$$

であるが,

$$0 > \mathbf{c}^T \mathbf{x} = (\mathbf{s} + A^T \bar{\mathbf{y}})^T \mathbf{x} = 0$$

となり矛盾が生じる. したがって, 双対問題は実行不可能となる. ■

注 1 定理 3 において, $\tau = \kappa = 0$ のとき, 主問題および双対問題は実行可能性を判定できない場合がほとんどである. しかし, 近年, $\tau = \kappa = 0$ の場合に後述の面削減法を用いて実行可能性を判定する手法が研究されている.

2.2 錐線形計画問題の実行可能性について

本稿では、後述の実験において最適化問題の実行可能性を判定している。その際、扱う問題を「実行可能」「実行不可能」の2種類だけではなく、更に詳しく分類しているため、以下にその定義を示す。

$n \times n$ 実正方行列 X および任意の非ゼロベクトル $\mathbf{y} \in \mathbb{R}^n$ に対して、

$$\mathbf{y}^T X \mathbf{y} > 0$$

が成り立つとき、 X を正定値行列という。

$n \times n$ 実行列 X に対して、そのノルムを

$$\|X\|_F \stackrel{\text{def}}{=} \sqrt{\text{tr} X^T X}$$

と定義する。また、 $n \times n$ 実行列の集合 \mathcal{A} , \mathcal{B} の距離、 $\text{dist}(\mathcal{A}, \mathcal{B})$ を以下のように定義する：

$$\text{dist}(\mathcal{A}, \mathcal{B}) \stackrel{\text{def}}{=} \inf \{ \|A - B\|_F \mid A \in \mathcal{A}, B \in \mathcal{B} \}.$$

今、主問題 (P) として (11) 式、その双対問題 (D) として (12) を考える。また、 S_{++}^n を $n \times n$ 正定値行列の集合とする。このとき、 (P) および (D) の実行可能性を、以下のように定義する。

定義 2 主問題 (P) について、

(P) が強実行可能 $\stackrel{\text{def}}{\Leftrightarrow} \exists \bar{X} \in S_{++}^n, A_i \cdot \bar{X} = b_i \ (i = 1, \dots, m)$

(P) が弱実行可能 $\stackrel{\text{def}}{\Leftrightarrow} (P)$ が実行可能 \wedge 強実行可能でない

(P) が強実行不可能 $\stackrel{\text{def}}{\Leftrightarrow} \text{dist}(S_+^n, \{X : A_i X = b_i \ (i = 1, \dots, m)\}) > 0$

(P) が弱実行不可能 $\stackrel{\text{def}}{\Leftrightarrow} (P)$ が実行不可能 \wedge 強実行不可能でない

定義 3 双対問題 (D) について、

(D) が強実行可能 $\stackrel{\text{def}}{\Leftrightarrow} \exists \bar{S} \in S_{++}^n, \bar{S} + \sum_{i=1}^m y_i A_i = C$

(D) が弱実行可能 $\stackrel{\text{def}}{\Leftrightarrow} (D)$ が実行可能 \wedge 強実行可能でない

(D) が強実行不可能 $\stackrel{\text{def}}{\Leftrightarrow} \text{dist}\left(S_+^n, \left\{S : S + \sum_{i=1}^m y_i A_i = C\right\}\right) > 0$

(D) が弱実行不可能 $\stackrel{\text{def}}{\Leftrightarrow} (D)$ が実行不可能 \wedge 強実行不可能でない

注 2 半正定値計画問題における弱実行不可能の定義は定義 2 および定義 3 の通りであるが、一般的な錐線形計画問題における弱実行不可能な場合の例として、以下の図 1 のような場合が考えられる。

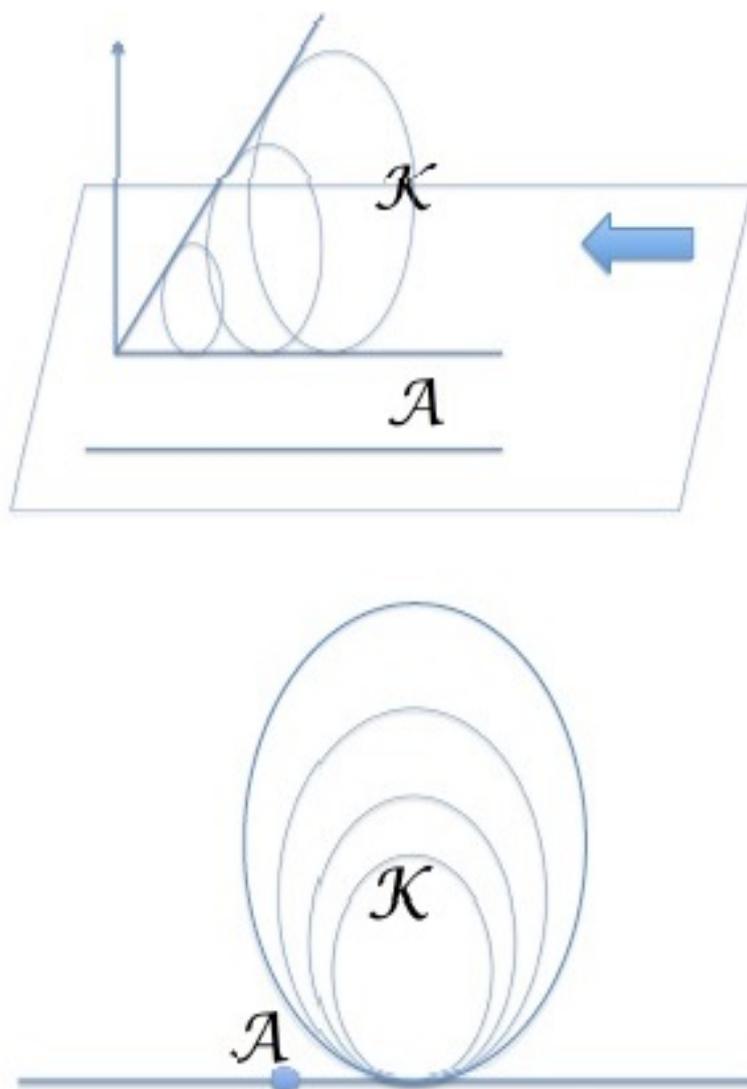


図 1: 弱実行不可能な場合の例

出典：文献 [12] p.6

図 1 において、 \mathcal{K} は錐である。また、 \mathcal{A} はアフィン空間を表す。このとき、許容領域の条件は $\mathcal{A} \cap \mathcal{K}$ である。

2.3 面削減法

2.3.1 面削減法

集合 $\mathcal{K} \subseteq \mathbb{R}^n$ を非空な閉凸錐とする。集合 $\mathcal{F} \subseteq \mathcal{K}$ が以下を満たすとき、 \mathcal{F} を \mathcal{K} の面であるという：

$$\forall \mathbf{x}, \mathbf{y} \in \mathcal{K}, \left(\frac{\mathbf{x} + \mathbf{y}}{2} \in \mathcal{F} \Rightarrow \mathbf{x}, \mathbf{y} \in \mathcal{F} \right). \quad (15)$$

ここで、 $\mathbf{s} \in \mathcal{K}^*$ について $\mathbf{s}^\perp := \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}^T \mathbf{s} = 0\}$ と定義する。 \mathbf{s}^\perp は超平面と言われる。今、錐線形計画問題 (2) の実行可能性が判明していないとする。ここで、(2) 式の実行可能領域を含むような \mathbf{s}^\perp を考える。このとき、 \mathcal{K} の任意の要素 \mathbf{x}, \mathbf{y} および $\mathcal{K} \cap \mathbf{s}^\perp \subset \mathcal{K}$ に対して

$$\begin{aligned} \frac{\mathbf{x} + \mathbf{y}}{2} \in \mathcal{K} \cap \mathbf{s}^\perp &\Rightarrow \left(\frac{\mathbf{x} + \mathbf{y}}{2} \right)^T \mathbf{s} = 0 \\ &\Leftrightarrow \frac{1}{2} (\mathbf{x} + \mathbf{y})^T \mathbf{s} = 0 \\ &\Leftrightarrow \frac{1}{2} (\mathbf{x}^T \mathbf{s} + \mathbf{y}^T \mathbf{s}) = 0 \\ &\Leftrightarrow \mathbf{x}^T \mathbf{s} + \mathbf{y}^T \mathbf{s} = 0 \\ &\Leftrightarrow \mathbf{x}^T \mathbf{s} = 0 \wedge \mathbf{y}^T \mathbf{s} = 0 \quad (\because \mathbf{x}^T \mathbf{s} \geq 0 \wedge \mathbf{y}^T \mathbf{s} \geq 0) \\ &\Leftrightarrow \mathbf{x}, \mathbf{y} \in \mathcal{K} \cap \mathbf{s}^\perp \end{aligned}$$

であるから、 $\mathcal{K} \cap \mathbf{s}^\perp$ は (15) 式を満たす。よって、 $\mathcal{K} \cap \mathbf{s}^\perp$ は \mathcal{K} の面である。ここで、 \mathcal{K} を面 $\mathcal{K} \cap \mathbf{s}^\perp$ に置き換えると、(2) 式は次の問題に書き換えることができる：

$$\begin{cases} \min \mathbf{c}^T \mathbf{x} \\ \text{s.t. } A\mathbf{x} = \mathbf{b} \\ \mathbf{x} \in \mathcal{K} \cap \mathbf{s}^\perp. \end{cases} \quad (16)$$

\mathbf{s}^\perp は条件式 $A\mathbf{x} = \mathbf{b}$ のすべての解を含んでいるため、(2) 式と (16) 式は同じ最適解を持つ。また、 $\mathcal{K} \cap \mathbf{s}^\perp \subset \mathcal{K}$ であるから、 $\mathcal{K} \cap \mathbf{s}^\perp$ は \mathcal{K} よりも小さくなっている。このような \mathbf{s} を削減方向と呼ぶ。次に (16) 式について新たに \mathbf{s}^\perp を考え、面 $\mathcal{K} \cap \mathbf{s}^\perp$ を新たな面に置き換えると、同じ最適解を保ったまま更に面を小さくすることができる。これを新しい \mathbf{s}^\perp が存在しなくなるまで繰り返すことで、最終的に構成した問題の実行可能性を判定することができる可能性がある。

このように、最適化問題の面を小さくすることで内点許容解を持つ問題に変換する手法を面削減法という。

2.3.2 半正定値計画問題に対する面削減法の適用

半正定値計画問題に対して、面削減法を適用することを考える。今、最適化問題として (11) 式および (12) 式が与えられているとする。このとき、削減方向 W を用いて (12) 式に

面削減法を適用することを考える．錐 S_+^n の双対錐は S_+^n となるため， S_+^n は自己双対錐である．また， $\{W\}^\perp$ がアフィン集合 $\{S \mid S = C - \sum_{i=1}^m y_i A_i\}$ を含むための条件は，

$$\begin{aligned} S \cdot W &= C \cdot W - \sum_{i=1}^m y_i A_i \cdot W \\ &= 0 \end{aligned}$$

より，

$$C \cdot W = 0 \wedge A_i \cdot W = 0 \quad (i = 1, \dots, m)$$

となることである．以上より， W は以下の条件式を満たす：

$$\begin{cases} C \cdot W = 0 \\ A_i \cdot W = 0 \quad (i = 1, \dots, m) \\ W \in S_+^n, W \neq O. \end{cases} \quad (17)$$

(17) 式を満たす W が求められたと仮定し，それを用いて (12) 式を置き換えると，以下の式が導ける：

$$\begin{cases} \max \sum_{i=1}^m b_i y_i \\ \text{s.t. } S = C - \sum_{i=1}^m y_i A_i \\ S \in S_+^n \cap \{W\}^\perp. \end{cases} \quad (18)$$

2.3.1 節より，(18) 式の実行可能領域は (12) 式と等しい．

以上より，半正定値問題に面削減法を適用することができた．これは面削減法の 1 反復である．理論的には，これを再帰的に繰り返すことで，最終的に内点許容解を得るか，あるいは実行不可能であることの証拠を得ることになる．しかし，面削減法を数値計算として再帰的に計算することは大変難しく，そのような文献は見当たらない．本研究でも，1 反復だけすることを目指す．

以下の節では，面削減法の 1 反復における各計算についてより詳しく述べる．

2.3.3 削減方向の計算

実際に (17) 式の条件を満たすような削減方向 W を求める場合，以下のような最適化問題を考えることは自然である：

$$\begin{cases} \min O \cdot W \\ \text{s.t. } C \cdot W = 0 \\ A_i \cdot W = 0 \quad (i = 1, \dots, m) \\ I \cdot W = 1 \\ W \in S_+^n. \end{cases} \quad (19)$$

ここで、 I は恒等行列である。(19)式の解は、それ自体が削減方向 W となっている。以下では、この最適化問題を解いて削減方向を求める手法を「naive定式化」と呼ぶことにする。

一方、文献[4]では削減方向を求める方法として、(19)式とは異なる以下のような定式化を用いている：

$$\left\{ \begin{array}{l} \min_{X,t,w} t \\ \text{s.t. } -C \cdot (X - tI) + t - w = 0 \\ I \cdot X + w = 1 \\ \begin{bmatrix} A_1 \cdot X \\ A_2 \cdot X \\ \vdots \\ A_m \cdot X \end{bmatrix} - t \begin{bmatrix} A_1 \cdot I \\ A_2 \cdot I \\ \vdots \\ A_m \cdot I \end{bmatrix} = 0 \\ (X, t, w) \in S_+^n \times S_+^1 \times S_+^1. \end{array} \right. \quad (20)$$

(20)式において、求める削減方向は X である。以下では、この最適化問題を解いて削減方向を求める手法を「LMT定式化」と呼ぶことにする。(20)式の双対問題は、以下のようになる：

$$\left\{ \begin{array}{l} \max_{y_1, y_2, y_3} y_2 \\ \text{s.t. } y_1 C - y_2 I - \sum_{i=1}^m y_3^i A_i \in S_+^n \\ 1 - y_1 (1 + C \cdot I) + I \cdot \left(\sum_{i=1}^m y_3^i A_i \right) \geq 0 \\ y_1 - y_2 \geq 0. \end{array} \right. \quad (21)$$

ここで、

$$t = \frac{1}{I \cdot I + 1}, w = \frac{1}{I \cdot I + 1}, X = \frac{I}{I \cdot I + 1}$$

とおけば、 (t, w, X) は(20)式の内点許容解である。実際、

$$\begin{aligned} -C \cdot (X - tI) + t - w &= -C \cdot \left(\frac{I}{I \cdot I + 1} - \frac{I}{I \cdot I + 1} \right) \\ &= -C \cdot 0 \\ &= 0, \end{aligned}$$

$$\begin{aligned} I \cdot X + w &= \frac{I \cdot I}{I \cdot I + 1} + \frac{1}{I \cdot I + 1} \\ &= \frac{I \cdot I + 1}{I \cdot I + 1} \\ &= 1, \end{aligned}$$

$$A_i \cdot \left(\frac{I}{I \cdot I + 1} \right) - \frac{1}{I \cdot I + 1} A_i \cdot I = \frac{A_i \cdot I}{I \cdot I + 1} - \frac{A_i \cdot I}{I \cdot I + 1} \quad (i = 1, \dots, m)$$

$$= 0$$

となり, (20) 式の条件を満たす. また, $(y_1, y_2, \mathbf{y}_3) = (0, -1, \mathbf{0})$ とおけば, (y_1, y_2, \mathbf{y}_3) は (21) 式の内点許容解である. 実際,

$$y_1 C - y_2 I - \sum_{i=1}^m y_3^i A_i = 0C - (-1)I - \sum_{i=1}^m 0A_i$$

$$= I \in S_n^+,$$

$$1 - y_1(1 + C \cdot I) + I \cdot \left(\sum_{i=1}^m y_3^i A_i \right) = 1 - 0(1 + C \cdot I) + I \cdot \left(\sum_{i=1}^m 0A_i \right)$$

$$= 1 \geq 0,$$

$$y_1 - y_2 = 0 - (-1)$$

$$= 1 \geq 0$$

となり, (21) 式の条件を満たす. 以上より, (20) 式と (21) 式には両方内点許容解が存在する. このとき, 定理 1 より, (20) 式と (21) 式の最適値は一致し, 両方に最適解が存在する. また, 主問題と双対問題の双方に内点許容解が存在することは, 主双対内点法が収束するための条件でもある. 一方, naive 定式化では, 内点許容解があることは保障されていない. したがって, naive 定式化よりも数値的に安定して解くことができることが期待される.

2.3.4 小さな問題の生成

削減方向が得られた場合に面削減法を適用して, 新たな最適化問題を得る方法について説明する. 例として, (11) 式から構成した (19) 式より削減方向 W が得られたとする. その際, (12) 式に面削減法を適用して, 新たな最適化問題を得る.

まず, W を固有値分解すると, 以下のように表せる:

$$W = Q \Lambda Q^T. \quad (22)$$

(22) 式において, Λ は対角要素に固有値を持つ行列であり, Q は直交行列である. ここで, Λ および Q をブロック行列を用いて表現すると, 以下のように書ける:

$$W = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} \Lambda_1 & O \\ O & \Lambda_2 \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}. \quad (23)$$

(23) 式において, Λ_1 はその値がほとんど 0 である固有値を対角要素に持ち, Λ_2 はそれ以外の固有値を対角要素に持つとする. また, Q_1, Q_2 はそれぞれ Λ_1, Λ_2 に対応する固有

ベクトルを列として持つ。\$W\$ を (23) 式を用いて変形する際、\$\Lambda_1\$ と \$\Lambda_2\$ の境界 (\$W\$ のランク) は以下のように決定する。

ランク決定のルール

今、\$n \times n\$ 行列 \$W\$ を固有値分解して行列 \$\Lambda\$ を得られたとする。また、\$\mathbf{p}\$ を \$\Lambda\$ の対角要素を昇順に並べた \$n\$ 次ベクトル、\$a\$ を \$W\$ の最大の固有値とする。このとき、求めたいランク \$r\$ は、\$p_i\$ (\$i = 1, \dots, n-1\$) と \$\frac{a}{n-i}\$ (\$i = 1, \dots, n-1\$) を \$i = 1\$ から昇順に比較していき、初めて

$$p(i) > \frac{a}{n-i} \quad (24)$$

を満たした時の \$i\$ であるとする。(24) 式を満たすような \$i\$ が存在しない場合、\$r = n-1\$ とする。

以降、このルールを AO 法と呼ぶ。\$\Lambda_1\$ の要素が 0 であると考え、(23) 式は次のように書ける：

$$W = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} O & O \\ O & \Lambda_2 \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} = Q_2 \Lambda_2 Q_2^T. \quad (25)$$

(25) 式および (19) 式より、以下が導ける：

$$\begin{aligned} W \cdot S = 0 &\Leftrightarrow Q_2 \Lambda_2 Q_2^T \cdot S = 0 \\ &\Leftrightarrow \Lambda_2 \cdot Q_2^T S Q_2 = 0 \quad (\because X \cdot Y = \text{tr}XY). \end{aligned}$$

ここで、\$C = Q_2^T S Q_2\$ とおくと、\$C\$ は半正定値である。\$\lambda_i (\in \Lambda_2) > 0\$、\$c_{ii} (\in C) \ge 0\$ より、

$$\Lambda_2 \cdot C = \sum_{i=1}^r \lambda_i c_{ii} \geq 0.$$

これが 0 となるのは、すべての \$i\$ について \$c_{ii} = 0\$ のときのみである。対角要素がすべて 0 となる半正定値行列は \$O\$ のみであるから、\$C = O\$。したがって、

$$Q_2^T S Q_2 = O \quad (26)$$

である。また、(12) 式における条件式より、

$$\begin{aligned} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} S \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} &= \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} C \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} - \sum_{i=1}^m y_i \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} A_i \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \\ \Leftrightarrow \begin{bmatrix} Q_1^T S Q_1 & Q_1^T S Q_2 \\ Q_2^T S Q_1 & Q_2^T S Q_2 \end{bmatrix} &= \begin{bmatrix} Q_1^T C Q_1 & Q_1^T C Q_2 \\ Q_2^T C Q_1 & Q_2^T C Q_2 \end{bmatrix} - \sum_{i=1}^m y_i \begin{bmatrix} Q_1^T A_i Q_1 & Q_1^T A_i Q_2 \\ Q_2^T A_i Q_1 & Q_2^T A_i Q_2 \end{bmatrix} \end{aligned}$$

となる。これと (26) 式より、以下の 2 式が導ける：

$$\begin{aligned} Q_1^T S Q_1 &= Q_1^T C Q_1 - \sum_{i=1}^m y_i Q_1^T A_i Q_1 \\ \sum_{i=1}^m y_i Q_2^T A_i &= Q_2^T C \quad (\because Q_2 \text{ はフルランク行列}). \end{aligned} \quad (27)$$

(27) 式を用いて (12) 式を変換すると、以下ようになる：

$$\left\{ \begin{array}{l} \max \sum_{i=1}^m b_i y_i \\ \text{s.t. } \tilde{S} = \tilde{C} - \sum_{i=1}^m y_i \tilde{A}_i \\ \tilde{S} \in S_+^n \\ \sum_{i=1}^m y_i Q_2^T A_i = Q_2^T C. \end{array} \right. \quad (28)$$

(28) 式において、 $\tilde{S} = Q_1^T S Q_1$ 、 $\tilde{C} = Q_1^T C Q_1$ 、 $\tilde{A}_i = Q_1^T A_i Q_1$ である。

以上より、(12) 式に面削減法を適用して、より小さな最適化問題 (28) 式を得ることができた。

2.3.5 錐拡大法

今、(17) 式より削減方向 W が求められたとする。このとき、 W およびパラメータ α を用いて (11) 式を以下の形に置き換える：

$$\left\{ \begin{array}{l} \min C \cdot (X + \alpha W) \\ \text{s.t. } A_i \cdot (X + \alpha W) = b_i \quad (i = 1, \dots, m) \\ (X + \alpha W) \in S_+^n, \alpha \in \mathbb{R}. \end{array} \right. \quad (29)$$

W は (17) 式より求めたものであるので、 $C \cdot W = 0$ 、 $A_i \cdot W = 0$ が成り立つ。したがって、(29) 式は次の形に変形できる：

$$\left\{ \begin{array}{l} \min C \cdot X \\ \text{s.t. } A_i \cdot X = b_i \quad (i = 1, \dots, m) \\ \tilde{X} = X + \alpha W \\ \tilde{X} \in S_+^n, \alpha \in \mathbb{R}. \end{array} \right. \quad (30)$$

(30) 式のように、元の問題よりも錐の範囲を拡大して解を求める手法を、錐拡大法と呼ぶ。(30) 式の双対問題は、以下ようになる：

$$\left\{ \begin{array}{l} \max \sum_{i=1}^m b_i y_i \\ \text{s.t. } S = C - \sum_{i=1}^m y_i A_i \\ S \in S_+^n \cap \{W\}^\perp. \end{array} \right. \quad (31)$$

(31) 式は, (18) 式と等しい. よって, 錐拡大法を適用した問題を解くことで, 面削減法を 1 反復適用した問題の解を得られることが予想される. 錐拡大法は, 半正定値計画問題の意味では内点許容解を回復しない. (31) 式には, 正定値である許容解は存在しないのである. しかし, 双対問題 (30) 式は実行可能領域が広がっており, 内点許容解が存在する可能性がある. 一般に, 双対ギャップのある問題では, 主問題も双対問題も弱実行可能であることが知られている. このような場合に, 双対問題の内点許容解を回復する手法は有効かもしれない.

(30) 式および (31) 式では求めた削減方向 W をそのまま用いているが, 前節で示している通り, W は (25) 式のように変形できる. 本稿の実験では, 求めた W をそのまま用いる場合と (25) 式を用いて変形する場合の 2 通りについて, それぞれ錐拡大法を適用して結果を比較している.

3 実験1：弱実行可能な場合

3.1 実験内容

本実験の目的は、以下の3点である。

- 弱実行可能な半正定値計画問題に対して面削減法および錐拡大法を適用し、問題の実行可能性を正確に判定できるか否かを調べる。
- 弱実行可能な半正定値計画問題の削減方向を求める際、(19)式と(20)式のどちらが適しているかを調べる。
- 弱実行可能な半正定値計画問題に対して、面削減法および錐拡大法のどちらの手法が適しているかを調べる。

そこで、(11)式が強実行可能、(12)式が弱実行可能であるような半正定値計画問題をランダムに生成し、面削減法および錐拡大法を用いて実行可能性を判定し、その結果を比較、考察した。

3.1.1 実験の手順

1つの半正定値計画問題を解く場合、まず始めに与えられた問題を naive 定式化または LMT 定式化に変形し、ソルバを用いて削減方向を求める。2.3.3節および2.3.4節より、半正定値計画問題に面削減法および錐拡大法を適用する際には、共通する削減方向を用いてよい。よって、求めた削減方向を用いて面削減法または錐拡大法を適用し、問題を解く。

面削減法を用いて半正定値計画問題を解く手順を、以下に示す。

1. 求めた削減方向を用いて、与えられた問題の双対問題に面削減法を適用し、(28)式に変形する。
2. (28)式をソルバを用いて解く。

本実験では、手順1で(28)式に変形する際の反復は1回のみ行った。

錐拡大法を用いて半正定値計画問題を解く手順を、以下に示す。

1. 求めた削減方向を用いて、与えられた問題の双対問題に錐拡大法を適用し(30)式に変形する。
2. (30)式をソルバを用いて解く。

問題に錐拡大法を適用する場合、求めた削減方向をそのまま用いる場合と、(25)式を用いて変形してから用いる場合の2通りが考えられる。本稿では、以降、求めた削減方向をそのまま用いる場合を「錐拡大法(1)」、(25)式を用いて変形する場合を「錐拡大法(2)」と表記する。

3.1.2 問題の生成方法

以下の方法では、主問題が強実行可能、双対問題が弱実行可能であるような半正定値問題をランダムに生成する。生成する半正定値問題の主問題および双対問題は、それぞれ (11) 式, (12) 式であるとする。また、行列のサイズは N , 等式の数 m , 実行可能な行列のランクは $r < N$ であるとし、乱数は区間 $(0, 1)$ で一様分布したものをを用いた。

まず、以下の方法により $\tilde{A}_i (i = 1, \dots, m)$, $b_i (i = 1, \dots, m)$, \tilde{C} を生成する。

Algorithm 1 Calculate $\tilde{A}_i (i = 1, \dots, m)$, $b_i (i = 1, \dots, m)$, \tilde{C}

$R \leftarrow r \times r$ 乱数行列

$\tilde{X} = R^T R$

$\tilde{S} = R^T R$

$\tilde{C} = \tilde{S}$

$\mathbf{y} \leftarrow r$ 次乱数ベクトル

for $i = 1$ to m **do**

$\tilde{A}_i = (R + R^T) / 2$

$b_i = \tilde{A}_i \cdot \tilde{X}$

$\tilde{C} = \tilde{C} + y_i \tilde{A}_i$

end for

ここで、求めた $\tilde{A}_i (i = 1, \dots, m)$, $b_i (i = 1, \dots, m)$, \tilde{C} を用いて、以下のような最適化問題 (\tilde{P}) および (\tilde{D}) を考える：

$$(\tilde{P}) \begin{cases} \min \tilde{C} \cdot X \\ \text{s.t. } \tilde{A}_i \cdot X = b_i (i = 1, \dots, m) \\ X \in S_+^r \end{cases} \quad (32)$$

$$(\tilde{D}) \begin{cases} \max \sum_{i=1}^m b_i y_i \\ \text{s.t. } \tilde{S} = \tilde{C} - \sum_{i=1}^m y_i \tilde{A}_i \\ \tilde{S} \in S_+^n. \end{cases} \quad (33)$$

このとき、(32) 式および (33) 式は共に強実行可能な半正定値計画問題となっている。

次に、以下のように $A_i (i = 1, \dots, m)$, C を生成する：

Algorithm 2 Calculate $A_i (i = 1, \dots, m)$, C

 $Q \leftarrow N \times N$ 直交行列 $O_1 \leftarrow (N - r) \times (N - r)$ 零行列 $O_2 \leftarrow (N - r) \times r$ 零行列 $O_3 \leftarrow r \times (N - r)$ 零行列**for** $i = 1$ to m **do**

$$A_i = Q^T \begin{bmatrix} O_1 & O_2 \\ O_3 & \tilde{A}_i \end{bmatrix} Q$$

end for

$$C = Q^T \begin{bmatrix} O_1 & O_2 \\ O_3 & \tilde{C} \end{bmatrix} Q$$

以上で生成した $A_i (i = 1, \dots, m)$, $b_i (i = 1, \dots, m)$, C を用いて半正定値問題を生成すると、主問題が強実行可能、双対問題が弱実行可能な問題が生成できる。

3.1.3 実験詳細

本実験では、上記の生成方法において $N = 10$ かつ $r = 3$ の場合, $r = 6$ の場合, $r = 9$ の場合の 3 通り, $N = 20$ かつ $r = 3$ の場合, $r = 6$ の場合, $r = 9$ の場合の 3 通り, 計 6 通りの場合についてランダムに 100 問ずつ半正定値問題を生成した. そして, naive と LMT, 2 通りの方法で削減方向を計算し, 面削減法と錐拡大法 (1)(2) を用いて生成した問題の実行可能性を判定した.

3.1.4 実験環境

本稿の実験は, すべて以下の環境にて行った.

OS	Mac OS X 10.10.5
CPU	Intel Core i5 2.6GHz
メモリ	8GB
ソルバ	SDPT3-4.0 on MATLAB 8.5.0.197613(R2015a)
OPTION	version=NT, predcorr=1

3.2 結果

3.2.1 削減方向の計算

6 通りの問題, 計 600 問について, naive と LMT をそれぞれ用いて削減方向を計算した結果を, 以下の表 1 に示す. 表 1 内の数字は, 削減方向の計算時に許容解を得ることができた問題の数を表している. 表 1 が示す通り, naive と LMT のどちらを用いた場合も, 本実験で生成した全ての問題について, 削減方向の計算時に許容解を得ることができ

表 1: 削減方向の計算結果

	$N = 10$		$N = 20$	
	naive	LMT	naive	LMT
$r=3$	100	100	100	100
$r=6$	100	100	100	100
$r=9$	100	100	100	100

た. よって, 弱実行可能な問題について面削減法および錐拡大法を適用する場合, naive と LMT のどちらを用いる場合でも, 少なくとも削減方向は求めることができると考えられる.

3.2.2 $N = 10$ の場合

$r = 3$ の問題に対して naive を用いた結果, 面削減法を適用して実行可能と回答した問題は 99 問, 実行可能と回答しなかった問題 (その他と表記) は 1 問であった. 錐拡大法 (1) を適用して実行可能と回答した問題は 98 問, 実行可能と回答しなかった問題は 2 問であった. 錐拡大法 (2) を適用して実行可能と回答した問題は 96 問, 実行可能と回答しなかった問題は 4 問であった. また, LMT を用いた結果, 面削減法を適用して実行可能と回答した問題は 99 問, 実行可能と回答しなかった問題は 1 問であった. 錐拡大法 (1) を適用して実行可能と回答した問題は 99 問, 実行可能と回答しなかった問題は 1 問であった. 錐拡大法 (2) を適用して実行可能と回答した問題は 99 問, 実行可能と回答しなかった問題は 1 問であった. これらの結果を以下の表 2 に示す. 表 2 より, $N = 10$ かつ $r = 3$ のとき, 面

表 2: 実行結果 ($N = 10, r = 3$)

	naive		LMT	
	実行可能	その他	実行可能	その他
面削減法	99	1	99	1
錐拡大法 (1)	98	2	99	1
錐拡大法 (2)	96	4	99	1

削減法, 錐拡大法 (1), 錐拡大法 (2) のうちどの手法を適用した場合の結果にも大きな差が無く, ほとんどの問題を実行可能と判定できていることが分かる. また, naive と LMT を比較すると, 若干 LMT の方が実行可能と判定できた問題数が多いが, その差は僅かであり, LMT の方が優れているとは明言できないと考えられる. したがって, $N = 10$ かつ $r = 3$ のときは naive と LMT, および面削減法と錐拡大法 (1)(2), どの組み合わせを用いても大きな差は無く, 弱実行可能な半正定値計画問題の実行可能性を, 概ね正しく判定することができると考えられる.

$r = 6$ の問題に対して naive を用いた結果、面削減法を適用して実行可能と回答した問題は 100 問、実行可能と回答しなかった問題は 0 問であった。錐拡大法 (1) を適用して実行可能と回答した問題は 93 問、実行可能と回答しなかった問題は 7 問であった。錐拡大法 (2) を適用して実行可能と回答した問題は 100 問、実行可能と回答しなかった問題は 0 問であった。また、LMT を用いた結果、面削減法を適用して実行可能と回答した問題は 100 問、実行可能と回答しなかった問題は 0 問であった。錐拡大法 (1) を適用して実行可能と回答した問題は 17 問、実行可能と回答しなかった問題は 83 問であった。錐拡大法 (2) を適用して実行可能と回答した問題は 100 問、実行可能と回答しなかった問題は 0 問であった。これらの結果を以下の表 3 に示す。表 3 より、 $N = 10$ かつ $r = 6$ のとき、面

表 3: 実行結果 ($N = 10, r = 6$)

	naive		LMT	
	実行可能	その他	実行可能	その他
面削減法	100	0	100	0
錐拡大法 (1)	93	7	17	83
錐拡大法 (2)	100	0	100	0

削減法と錐拡大法 (2) を用いた場合には、全ての問題を実行可能と判定できていることが分かる。しかし、錐拡大法 (1) を用いた場合、削減方向の計算に LMT を用いると、naive を用いた場合と比較して実行可能性を正しく判定することができた問題数が大きく減少していることが分かる。よって、 $N = 10$ かつ $r = 6$ のとき、面削減法と錐拡大法 (2) は錐拡大法 (1) よりも実行可能な半正定値計画問題を解くのに適しており、錐拡大法 (1) を用いる場合には LMT よりも naive の方が弱実行可能な半正定値計画問題の実行可能性を判定するのに適していると考えられる。

$r = 9$ の問題に対して naive を用いた結果、面削減法を適用して実行可能と回答した問題は 94 問、実行可能と回答しなかった問題は 6 問であった。錐拡大法 (1) を適用して実行可能と回答した問題は 99 問、実行可能と回答しなかった問題は 1 問であった。錐拡大法 (2) を適用して実行可能と回答した問題は 100 問、実行可能と回答しなかった問題は 0 問であった。また、LMT を用いた結果、面削減法を適用して実行可能と回答した問題は 100 問、実行可能と回答しなかった問題は 0 問であった。錐拡大法 (1) を適用して実行可能と回答した問題は 45 問、実行可能と回答しなかった問題は 55 問であった。錐拡大法 (2) を適用して実行可能と回答した問題は 70 問、実行可能と回答しなかった問題は 30 問であった。これらの結果を以下の表 4 に示す。表 4 より、 $N = 10$ かつ $r = 9$ のとき、面削減法を用いた場合にはほとんどの問題を実行可能と判定できていることが分かる。しかし、錐拡大法 (1) および錐拡大法 (2) を用いた場合、削減方向の計算に LMT を用いると、naive を用いた場合と比較して実行可能性を正しく判定することができた問題数が大きく減少していることが分かる。よって、 $N = 10$ かつ $r = 6$ のとき、面削減法は錐拡大法 (1) と錐

表 4: 実行結果 ($N = 10, r = 9$)

	naive		LMT	
	実行可能	その他	実行可能	その他
面削減法	94	6	100	0
錐拡大法 (1)	99	1	45	55
錐拡大法 (2)	100	0	70	30

拡大法 (2) よりも弱実行可能な半正定値計画問題の実行可能性を正しく判定するのに適しており、面削減法を用いた場合、若干ではあるが LMT を用いて削減方向を求めた方が弱実行可能な半正定値計画問題の実行可能性を正しく判定することができる確率が高いと考えられる。しかし、錐拡大法 (1) および錐拡大法 (2) を用いる場合には LMT よりも naive の方が弱実行可能な半正定値計画問題の実行可能性を判定するのに適していると考えられる。

3.2.3 $N = 20$ の場合

$r = 3$ の問題に対して naive を用いた結果、面削減法を適用して実行可能と回答した問題は 92 問、実行可能と回答しなかった問題は 8 問であった。錐拡大法 (1) を適用して実行可能と回答した問題は 98 問、実行可能と回答しなかった問題は 2 問であった。錐拡大法 (2) を適用して実行可能と回答した問題は 100 問、実行可能と回答しなかった問題は 0 問であった。また、LMT を用いた結果、面削減法を適用して実行可能と回答した問題は 91 問、実行可能と回答しなかった問題は 9 問であった。錐拡大法 (1) を適用して実行可能と回答した問題は 99 問、実行可能と回答しなかった問題は 1 問であった。錐拡大法 (2) を適用して実行可能と回答した問題は 95 問、実行可能と回答しなかった問題は 5 問であった。これらの結果を以下の表 5 に示す。表 5 より、 $N = 20$ かつ $r = 3$ のとき、錐拡大法

表 5: 実行結果 ($N = 20, r = 3$)

	naive		LMT	
	実行可能	その他	実行可能	その他
面削減法	92	8	91	9
錐拡大法 (1)	98	2	99	1
錐拡大法 (2)	100	0	95	5

(1) および錐拡大法 (2) を適用した場合の結果には大きな差が無く、ほとんどの問題を実行可能と判定できていることが分かる。面削減法を用いた場合は naive と LMT のどちらを用いた場合も、錐拡大法 (1)(2) の双方より実行可能性を正しく判定することができた問題数が若干少ないが、それでも 9 割以上の問題の実行可能性を正しく判定することができ

ている。したがって、 $N = 20$ かつ $r = 3$ のときは naive と LMT、および面削減法と錐拡大法 (1)(2)、どの組み合わせを用いても大きな差は無く、弱実行可能な半正定値計画問題の実行可能性を概ね正しく判定することができると考えられる。

$r = 6$ の問題に対して naive を用いた結果、面削減法を適用して実行可能と回答した問題は 100 問、実行可能と回答しなかった問題は 0 問であった。錐拡大法 (1) を適用して実行可能と回答した問題は 99 問、実行可能と回答しなかった問題は 1 問であった。錐拡大法 (2) を適用して実行可能と回答した問題は 99 問、実行可能と回答しなかった問題は 1 問であった。また、LMT を用いた結果、面削減法を適用して実行可能と回答した問題は 100 問、実行可能と回答しなかった問題は 0 問であった。錐拡大法 (1) を適用して実行可能と回答した問題は 16 問、実行可能と回答しなかった問題は 84 問であった。錐拡大法 (2) を適用して実行可能と回答した問題は 99 問、実行可能と回答しなかった問題は 1 問であった。これらの結果を以下の表 6 に示す。表 3 より、 $N = 20$ かつ $r = 6$ のとき、面削減法を

表 6: 実行結果 ($N = 20, r = 6$)

	naive		LMT	
	実行可能	その他	実行可能	その他
面削減法	100	0	100	0
錐拡大法 (1)	99	1	16	84
錐拡大法 (2)	99	1	99	1

用いた場合は全ての問題を、錐拡大法 (2) を用いた場合には、ほとんど全ての問題を実行可能と判定できていることが分かる。しかし、錐拡大法 (1) を用いた場合、削減方向の計算に LMT を用いると、naive を用いた場合と比較して実行可能性を正しく判定することができた問題数が大きく減少していることが分かる。よって、 $N = 20$ かつ $r = 6$ のとき、面削減法と錐拡大法 (2) は錐拡大法 (1) よりも実行可能な半正定値計画問題の実行可能性を正しく判定するのに適しており、錐拡大法 (1) を用いる場合には LMT よりも naive の方が弱実行可能な半正定値計画問題の実行可能性を判定するのに適していると考えられる。

$r = 9$ の問題に対して naive を用いた結果、面削減法を適用して実行可能と回答した問題は 100 問、実行可能と回答しなかった問題は 0 問であった。錐拡大法 (1) を適用して実行可能と回答した問題は 100 問、実行可能と回答しなかった問題は 0 問であった。錐拡大法 (2) を適用して実行可能と回答した問題は 100 問、実行可能と回答しなかった問題は 0 問であった。また、LMT を用いた結果、面削減法を適用して実行可能と回答した問題は 100 問、実行可能と回答しなかった問題は 0 問であった。錐拡大法 (1) を適用して実行可能と回答した問題は 51 問、実行可能と回答しなかった問題は 49 問であった。錐拡大法 (2) を適用して実行可能と回答した問題は 99 問、実行可能と回答しなかった問題は 1 問であった。これらの結果を以下の表 7 に示す。表 3 より、 $N = 20$ かつ $r = 9$ のとき、面削減法を用いた場合は全ての問題を、錐拡大法 (2) を用いた場合にも、ほとんど全ての問題

表 7: 実行結果 ($N = 20, r = 9$)

	naive		LMT	
	実行可能	その他	実行可能	その他
面削減法	100	0	100	0
錐拡大法 (1)	100	0	51	49
錐拡大法 (2)	100	0	99	1

を実行可能と判定できていることが分かる。しかし、錐拡大法 (1) を用いた場合、削減方向の計算に LMT を用いると、naive を用いた場合と比較して実行可能性を正しく判定することができた問題数が大きく減少していることが分かる。よって、 $N = 20$ かつ $r = 9$ のとき、面削減法と錐拡大法 (2) は錐拡大法 (1) よりも実行可能な半正定値計画問題の実行可能性を正しく判定するのに適しており、錐拡大法 (1) を用いる場合には LMT よりも naive の方が弱実行可能な半正定値計画問題の実行可能性を判定するのに適していると考えられる。

3.2.4 ソルバの終了条件の変更

$N = 10$ かつ $r = 6$, $N = 10$ かつ $r = 9$, $N = 20$ かつ $r = 6$, $N = 20$ かつ $r = 9$ の 4 通りの場合において、弱実行可能な半正定値計画問題に対して LMT と錐拡大法 (1) を用いた場合に、「その他」と判定された問題が多いことについて考える。例えば、 $N = 10$ かつ $r = 6$ の問題で「その他」と判定された 83 問の内訳は、「lack of progress in infeas」というエラーで止まった問題が 41 問、「progress in duality gap has deteriorated」というエラーで止まった問題が 42 問であった。今回用いたソルバ、SDPT3 の内部では反復を繰り返して解を求めているが、双方ともに、何らかの数値の進捗が見られなくなって終了している。そこで、LMT を用いて削減方向を計算する際にソルバの終了条件を変更し、新しく得られた削減方向を用いて再度錐拡大法 (1) を適用する実験を行った。終了条件は、既定値である 10^{-8} から 10^{-10} , 10^{-12} の 2 通りに変化させた。その結果を以下の表 8~表 11 に示す。表 8 は $N = 10$ かつ $r = 6$ の場合の結果を、表 9 は $N = 10$ かつ $r = 9$ の場合の

表 8: 終了条件変更 ($N = 10, r = 6$)

終了条件	10^{-8}	10^{-10}	10^{-12}
許容解取得数 (削減方向計算時)	100	83	11
実行可能と判定した問題数	17	80	81

結果を、表 10 は $N = 20$ かつ $r = 6$ の場合の結果を、表 11 は $N = 20$ かつ $r = 9$ の場合の結果を、それぞれ表している。表 8~表 11 より、終了条件の変更を行った 4 通りの全ての場合において、終了条件が 10^{-10} のときには実行可能性を正しく判定することができた

表 9: 終了条件変更 ($N = 10, r = 9$)

終了条件	10^{-8}	10^{-10}	10^{-12}
許容解取得数 (削減方向計算時)	100	92	10
実行可能と判定した問題数	45	92	92

表 10: 終了条件変更 ($N = 20, r = 6$)

終了条件	10^{-8}	10^{-10}	10^{-12}
許容解取得数 (削減方向計算時)	100	98	23
実行可能と判定した問題数	16	92	95

表 11: 終了条件変更 ($N = 20, r = 9$)

終了条件	10^{-8}	10^{-10}	10^{-12}
許容解取得数 (削減方向計算時)	100	99	26
実行可能と判定した問題数	51	93	96

問題数が既定値のときに比べて増加していることが分かる。しかし、100 問近く実行可能性を正しく判定することができるようになる場合もあれば、80 問程度までしか実行可能性を正しく判定することができる問題数が増加しない場合もあり、終了条件を厳しくしたとしても面削減法と比較すると安定性に欠けると考えられる。また、条件を 10^{-12} に変更したときは、実行可能性を正しく判定することができた問題数は条件が 10^{-10} のときと比べて変化が無いが、あるいは僅かに増加しているのに対し、削減方向を計算する際に許容解を得ることができた問題数が大きく減少していることが分かる。以上より、以下の 2 点が考えられる。

- 弱実行可能な半正定値計画問題に LMT を用いて錐拡大法 (1) を適用する場合、削減方向を計算する際にソルバの終了条件を既定値よりも厳しくした方が、安定して問題の実行可能性を正しく判定することができる。
- 弱実行可能な半正定値計画問題に対して、たとえ削減方向の計算時にソルバが許容解を返さなくても、最終的に得られている解を用いて錐拡大法 (1) を適用してよい。

3.2.5 ソルバをそのまま用いる場合

$N = 10$ かつ $r = 3, 6, 9$ の場合、および $N = 20$ かつ $r = 3, 6, 9$ の場合の 6 通りについて、面削減法や錐拡大法を適用せず、ソルバをそのまま用いて実行可能性を判定した結果を以下の表 12 に示す。表 12 中の数字は、正確に実行可能性を判定することのできた問題数を表している。表 12 より、面削減法および錐拡大法を用いない場合でも、ほぼ全て

表 12: 実行結果 (sdpt3 据え置き)

	$r = 3$	$r = 6$	$r = 9$
$N = 10$	96	100	100
$N = 20$	95	99	100

の問題の実行可能性を正確に判定できていることが分かる。表 2~7 の結果と比較しても、各手法を用いた場合と同等かそれ以上に正確に実行可能性を判定できていることが分かる。よって、弱実行可能な問題に対してソルバとして SDPT3 を用いる場合、実行可能性の判定という点では、面削減法および錐拡大法は適用せず、ソルバをそのまま用いた方がよいと考えられる。

4 実験 2 : 弱実行不可能な場合

4.1 実験内容

本実験の目的は、以下の 3 点である。

- 弱実行不可能な半正定値計画問題に面削減法および錐拡大法を適用することが、既存の手法と比較して有効か否かを調べる。
- 弱実行不可能な半正定値計画問題の削減方向を求める際、(19) 式と (20) 式のどちらが適しているかを調べる。
- 弱実行不可能な半正定値計画問題に対して、面削減法および錐拡大法のどちらの手法が適しているかを調べる。

そこで、文献 [5] の実験に使用されたものと同じ弱実行不可能な半正定値計画問題を、面削減法および錐拡大法 (1)、錐拡大法 (2) を用いて実行可能性を判定する実験を行った。そして、その結果を文献 [5] の結果と比較した。文献 [5] では弱実行不可能な問題の他、弱実行不可能か強実行不可能か不明な問題についても同様の実験を行っているが、本稿では弱実行不可能な問題のみ扱うこととする。

削減方向の計算法および面削減法、錐拡大法 (1)、錐拡大法 (2) を用いて半正定値計画問題を解く方法は、実験 1 と同様のものを用いた。

4.1.1 文献 [5] の実験内容

文献 [5] では、以下の 2 項目：

- A_i ($i = 1, \dots, m$) が clean/messy
- $m = 10/20$

の順列、すなわち 4 通りの条件についてそれぞれ 100 問ずつ弱実行不可能な半正定値計画問題を生成し、複数のソルバを用いて実行可能性を判定している。そして、ソルバごとに 100 問中何問の実行可能性を正しく判定することができたのか、結果を比較している。ここで、「clean」とは文献 [5] におけるアルゴリズム 2 および 3 を用いて生成されたデータセット A_i ($i = 1, \dots, m$) を表す。また、「messy」とはデータセット「clean」を文献 [5] における Messing step を用いて変化させたデータセット A_i ($i = 1, \dots, m$) を表しており、一般に messy の方が難しい問題と言われている。以下の表 13 に、生成した問題の場合分けについてまとめた。

表 13: 場合分け

生成した問題	$A_i (i = 1, \dots, m)$	m
弱実行不可能	clean	10
		20
	messy	10
		20

4.2 結果

4.2.1 削減方向の計算

naive および LMT のそれぞれについて、表 13 の場合分け 1 通りにつき 100 問、計 400 問の削減方向を計算した結果を、以下の表 14、表 15 に示す。表 14 は、 $m = 10$ および

表 14: 削減方向の計算結果：naive

	$m = 10$		$m = 20$	
	clean	messy	clean	messy
許容解取得	0	0	0	0
実行不可能	79	98	77	97

表 15: 削減方向の計算結果：LMT

	$m = 10$		$m = 20$	
	clean	messy	clean	messy
許容解取得	100	100	100	100
実行不可能	0	0	0	0

$m = 20$ の問題について naive を適用した結果、許容解を得た問題の数および実行不可能と判定された問題の数を表している。naive を用いた場合はどの問題においても削減方向の計算時に許容解を得ることができず、ほとんどの問題でソルバに実行不可能と回答された。表 15 は、 $m = 10$ および $m = 20$ の問題について LMT を適用した結果、許容解を得た問題の数および実行不可能と判定された問題の数を表している。LMT を用いた場合は、全ての問題で許容解を得ることができた。

ソルバに実行不可能と回答された場合、返される削減方向はソルバが実行不可能と判断した時点での計算結果である。このとき、最適化問題の等式条件を満たした解は返されていない。よって、表 14 および表 15 より、naive と LMT を比較すると、naive より LMT の方が安定して削減方向を求められていると考えられる。

4.2.2 面削減法および錐拡大法の適用

得られた削減方向を用いて，与えられた問題を (28) 式および (30) 式に変形し，前述の 400 問に対して面削減法および錐拡大法 (1)，錐拡大法 (2) を適用して解いた結果を以下の表 16，17 に示す．表 16 および表 17 内の数字は，各手法を各問題に適用した結果，実行

表 16: 実行不可能と判定された問題の数 ($m = 10$)

	naive		LMT	
	clean	messy	clean	messy
面削減法	100	100	100	100
錐拡大法 (1)	84	60	100	97
錐拡大法 (2)	76	36	100	96

表 17: 実行不可能と判定された問題の数 ($m = 20$)

	naive		LMT	
	clean	messy	clean	messy
面削減法	32(68)	24(76)	100	100
錐拡大法 (1)	85	63	100	100
錐拡大法 (2)	59	22	100	99

不可能と判定された問題の数を表している．また，表 17 における () 内の数は，その条件においてソルバが実行可能だと判定した問題の数を表している．

表 16，17 より，naive と LMT の結果を比較すると， $m = 10$ のとき，面削減法を用いた場合は naive と LMT の結果に差はない．しかし，錐拡大法 (1) および錐拡大法 (2) を用いた場合は，どちらの場合においても実行不可能と判定された問題の数は明らかに LMT の方が多い．また， $m = 20$ のとき，面削減法，錐拡大法 (1)，錐拡大法 (2) のいずれを用いた場合においても，実行不可能と判定された問題の数は明らかに LMT の方が多い．加えて，naive を用いた場合，面削減法を適用すると実行可能だと判定してしまい，弱実行不可能な問題を正確に判定できていないことが分かる．以上より，弱実行不可能な問題に面削減法，錐拡大法 (1)(2) を適用する場合は，naive よりも LMT の方が優れていると考えられる．

表 16，17 より，面削減法と錐拡大法 (1)，錐拡大法 (2) を比較すると，naive を用いた場合， $m = 10$ のときは面削減法では clean および messy どちらの場合でも全ての問題の実行可能性を正しく判定することができているのに対し，錐拡大法 (1) および錐拡大法 (2) では clean と messy どちらの場合も全ての問題の実行可能性を正しく判定することができていない．よって， $m = 10$ のときに naive を用いる場合は，錐拡大法 (1)(2) よりも面削減法の方が優れていると考えられる．しかし， $m = 20$ のとき，面削減法を用いると半分

以上の問題で実行可能な問題だと判定してしまっており、問題を正確に判定できた数は錐拡大法(1)が最も多い。よって、 $m = 20$ のときに naive を用いる場合は、錐拡大法(1)が最も優れていると考えられる。LMT を用いた場合、 m の値に関わらずいずれの手法でもほとんど全ての問題の実行可能性を正しく判定することができているが、 $m = 10$ のときの錐拡大法(1)(2)、および $m = 20$ のときの錐拡大法(2)では、messy の場合に数問だけではあるが実行可能性を正しく判定できていない問題が存在する。よって、LMT を用いた場合は、 m の値に関わらず錐拡大法(1)(2)よりも面削減法の方が優れていると考えられる。

4.2.3 文献 [5] との比較

文献 [5] で用いられたソルバは SEDUMI, SDPT3, MOSEK, および SEDUMI に Permenter and Parrio[7] のアルゴリズムを用いたもの (以下, PP+SEDUMI と表記) の 4 種類である。それぞれのソルバによる実行結果を、以下の表 18 に示す。表 18 は、 $m = 10$

表 18: 文献 [5] の実行結果 (弱実行不可能な問題)

	$m = 10$		$m = 20$	
	clean	messy	clean	messy
SEDUMI	0	0	1	0
SDPT3	0	0	0	0
MOSEK	0	0	11	0
PP+SEDUMI	100	0	0	0

および $m = 20$ の場合に各ソルバが実行不可能と回答した問題の数を表している。表 18 より、文献 [5] で用いられているソルバでは、ほとんどの場合、生成した問題の実行可能性を正しく判定することができないことが分かる。特に、問題が弱実行不可能かつ messy な場合は、どのソルバでも 1 問も実行可能性を正しく判定することができていないことが分かる。また、文献 [5] では SDPT3 の仕様が明記されていなかったため、本実験の仕様で同様の実験を行った。しかし、結果は表 18 のものと変わらなかった。

一方、表 16 および 17 で示している通り、LMT を用いた面削減法および錐拡大法(1)(2)では、clean の場合、どの手法を用いた場合も、 m の値に関わらず、100 問中 100 問の実行可能性を正しく判定することができている。messy の場合、面削減法を用いた場合は m の値に関わらず 100 問全てを、錐拡大法(1)(2)を用いた場合も m の値に関わらず、少なくとも 96 問以上は実行可能性を正しく判定することができている。以上より、LMT を用いた場合は、文献 [5] で用いられている 4 つのソルバよりも、弱実行不可能な半正定値計画問題の実行可能性を判定する場合において優れていると考えられる。

4.2.4 ソルバの終了条件の変更

表 16 および表 17 より，削減方向の計算に naive を用いた場合は LMT を用いた場合に比べて，実行可能性を正しく判定することができた問題数が少ない．そこで，naive を用いて削減方向を計算する際にソルバの終了条件を変更し，新しく得られた削減方向を用いて再度実験を行った．行った実験は， $m = 10$ の場合の錐拡大法 (1) および (2)， $m = 20$ の場合の面削減法，錐拡大法 (1) および (2) の 5 通りである．終了条件は，既定値である 10^{-8} から 10^{-10} に変化させた．その結果を以下の表 19 および表 20 に示す．表 19 および

表 19: 終了条件の変更 (naive, $m = 10$)

	10^{-8}		10^{-10}	
	clean	messy	clean	messy
錐拡大法 (1)	84	60	84	60
錐拡大法 (2)	76	36	0	22

表 20: 終了条件の変更 (naive, $m = 20$)

	10^{-8}		10^{-10}	
	clean	messy	clean	messy
面削減法	32	24	32	24
錐拡大法 (1)	85	63	85	63
錐拡大法 (2)	59	22	59	22

表 20 内の数字は，各条件で実行可能性を正しく判定することができた問題数を表している．また，削減方向を計算する際，終了条件を変更しても許容解を得ることができた問題は 1 問も無かった．表 19 および表 20 が示す通り，再度実験を行った全ての場合において，終了条件を変更しても実行可能性を正しく判定することができた問題数は増加しなかった．特に， $m = 10$ のとき，錐拡大法 (2) を用いると，実行可能性を正しく判定することができた問題数は clean, messy 共に減少している．以上より，弱実行不可能な半正定値計画問題に対して naive を用いる場合，削減方向を計算する際に終了条件を厳しくしても，実行可能性を正しく判定することができる問題数は増加しないと考えられる．

5 実験3：双対ギャップが存在する場合

5.1 実験内容

本実験の目的は、以下の3点である。

- 実行可能だが双対ギャップが存在する半正定値計画問題に対して面削減法および錐拡大法を適用し、問題の実行可能性を正確に判定できるか否かを調べる。
- 実行可能だが双対ギャップが存在する半正定値計画問題の削減方向を求める際、(19)式と(20)式のどちらが適しているかを調べる。
- 実行可能だが双対ギャップが存在する半正定値計画問題に対して、面削減法および錐拡大法のどちらの手法が適しているかを調べる。

そこで、文献[6]のアルゴリズム1.0.3において述べられている、実行可能であるが双対ギャップが存在する半正定値計画問題を作成する手法を用いて半正定値計画問題を100問生成し、面削減法および錐拡大法(1)(2)を用いて解く実験を行った。

削減方向の計算法および面削減法、錐拡大法(1)、錐拡大法(2)を用いて半正定値計画問題を解く方法は、実験1と同様のものを用いた。

5.1.1 問題の生成方法

行列の次元が n 、データセット数が m 、双対ギャップが g であるような半正定値問題を生成する。このとき、生成する問題を(11)式であるとする。

まず、以下の2式を満たすように自然数 r_1, r_3, p を決定する：

$$\begin{aligned} r_1 + r_3 + 1 &= n \\ p &\leq r_3. \end{aligned}$$

決定した r_3 および p を用いて、 A_1, \dots, A_p を以下のように決定する：

$$A_i \ (i = 1, \dots, p) = \begin{bmatrix} O & O \\ O & S_+^{r_3} \end{bmatrix}.$$

次に、 A_{p+1}, \dots, A_m を以下のように決定する：

$$A_i \ (i = p + 1, \dots, m) = \begin{bmatrix} O & O & (A_i)_{13} \\ O & (A_i)_{22} & * \\ (A_i)_{13}^T & * & * \end{bmatrix}.$$

ここで、 $*$ は任意の要素を持つブロック行列であり、 $(A_i)_{13}$ は線形独立である。また、ある $i \in \{p + 1, \dots, m\}$ について $(A_i)_{22} \succ O$ である。最後に、 C および $b_i \ (i = 1, \dots, m)$ を

以下のように決定する：

$$C = \begin{bmatrix} O & O & O \\ O & \sqrt{g} & O \\ O & O & O \end{bmatrix}, \quad b_i = A_i \cdot C \quad (i = 1, \dots, m).$$

5.1.2 生成した問題

今回の実験で用いた問題は、上記の生成方法において

$$n = 10, m = 10, r_1 = 5, p = 4, g = 100$$

として生成した問題 100 問である。

5.2 結果

5.2.1 面削減法および錐拡大法の適用

面削減法および錐拡大法 (1)(2) を用いて、生成した問題 100 問を解いた結果を以下の表 21 および表 22 に示す。表 21 および表 22 は、面削減法、錐拡大法 (1)、錐拡大法 (2) を用

表 21: naive(双対ギャップあり)

	実行可能	実行不可能	その他
面削減法	15	0	85
錐拡大法 (1)	53	0	47
錐拡大法 (2)	57	0	43

表 22: LMT(双対ギャップあり)

	実行可能	実行不可能	その他
面削減法	4	2	94
錐拡大法 (1)	58	32	10
錐拡大法 (2)	34	1	65

いた場合に各問題を実行可能、実行不可能と判定した数、およびそのどちらとも判定しなかった問題の数(その他と表記)を表している。表 21 が示す通り、naive を用いた場合、錐拡大法 (1) および錐拡大法 (2) では約半分の問題の実行可能性を正しく判定することができるのに対し、面削減法では 15 問しか実行可能性を正しく判定することができていない。また、表 22 が示す通り、LMT を用いた場合、錐拡大法 (1) では約半分の問題の実

行可能性を正しく判定できているのに対し、錐拡大法 (2) では約 3 割、面削減法では 4 問しか実行可能性を正しく判定することができていない。以上より、双対ギャップの存在する実行可能な問題の実行可能性を判定する場合には、面拡大法よりも錐拡大法、特に錐拡大法 (1) の方が優れていると考えられる。しかし、錐拡大法 (1) を用いた場合であっても、LMT を用いた場合は実行不可能と判定した問題が 32 問存在していたり、そもそも 100 問中最大で 58 問しか正確な判定ができていないため、錐拡大法が双対ギャップが存在する実行可能な問題の実行可能性を判定するのに適しているとは断言できないと考えられる。

5.2.2 ソルバをそのまま用いる場合

生成した問題 100 問に対し、面削減法および錐拡大法を適用せずにソルバをそのまま用いて解く実験を行った。その結果、100 問全ての問題の実行可能性を正しく判定することができた。これより、実行可能だが双対ギャップが存在する半正定値計画問題の実行可能性を判定する場合は、面削減法および錐拡大法を適用せずにソルバをそのまま用いた方がよいと考えられる。

5.2.3 双対ギャップ

本実験で生成した問題は、100 問全てに双対ギャップが存在し、その値は 100 である。しかし、面削減法を用いると、主問題と双対問題の最適値が一致する。また、5.1.1 節より、今回生成した双対問題の目的関数値は 0 である。よって、双対問題の目的関数値 0 に合わせて、主問題の目的関数値も 0 になると考えられる。したがって、問題を解いた際に最終的に得られた主問題と双対問題の目的関数値の差、すなわち双対ギャップが 0 に近ければ、実行可能性を判定するという点とは別に、最適値を求めるという点においては、その問題に対してうまく各手法を適用できていると考えられる。そこで、生成した問題に対して面削減法、錐拡大法 (1) および (2) を適用して解いた場合に得られた目的関数値と双対ギャップを、naive と LMT の 2 通りについて調べた。結果を、以下の表 23~25 に示す。

表 23: 目的関数値の誤差が 10^{-8} 以下の問題数：主問題

	naive	LMT
面削減法	0	0
錐拡大法 (1)	57	36
錐拡大法 (2)	100	100

表 23 内の数字は、各手法において求めた主問題の目的関数値の誤差が 10^{-8} 以下であった問題数を表している。表 24 内の数字は、各手法において求めた双対問題の目的関数値の誤差が 10^{-8} 以下であった問題数を表している。表 25 内の数字は、各手法において双対ギャップの値が 10^{-8} 以下であった問題数を表している。

表 24: 目的関数値の誤差が 10^{-8} 以下の問題数：双対問題

	naive	LMT
面削減法	0	0
錐拡大法 (1)	57	55
錐拡大法 (2)	100	100

表 25: 双対ギャップが 10^{-8} 以下の問題数

	naive	LMT
面削減法	0	0
錐拡大法 (1)	57	35
錐拡大法 (2)	100	100

表 23～表 25 より，錐拡大法 (2) を用いた場合は主問題と双対問題，両方の目的関数値を全ての問題で正しく求められていることが分かる．他の 2 つの手法は程度の差はあれ錐拡大法 (2) よりも明らかに目的関数値を正しく求められている問題数が少なく，特に面削減法を用いた場合は，1 問も正しい値を求められていない．以上より，実行可能であるが双対ギャップが存在する半正定値計画問題の目的関数値を求める場合には，錐拡大法 (2) を用いた方がよいと考えられる．

6 結論

本稿では、面削減法と錐拡大法 (1), 錐拡大法 (2) の計 3 つの手法を用いて、主問題が強実行可能かつ双対問題が弱実行可能な半正定値計画問題, 弱実行不可能な半正定値計画問題, 実行可能だが双対ギャップが存在する半正定値計画問題を解いた. そして, 3 つの手法を適用して得られた結果を, それぞれの結果および他のソルバを適用して得られた結果と比較した. その結果として, 主問題が強実行可能かつ双対問題が弱実行可能な半正定値計画問題について, 以下の 3 点が判明した.

- naive 定式化と LMT 定式化のどちらを用いても, 概ね正しい削減方向が計算できている.
- 錐拡大法を適用する場合, 削減方向の計算には naive 定式化を用いた方が安定して問題を解くことができる.
- 面削減法の方が, 錐拡大法よりも安定して解くことができる.

また, 弱実行不可能な半正定値計画問題について, 以下の 3 点が判明した.

- 既存のソルバよりも面削減法および錐拡大法の方が適している.
- 削減方向の計算には, LMT 定式化の方が適している.
- 面削減法と錐拡大法のどちらがより適しているとは明言できない.

加えて, 実行可能だが双対ギャップが存在する半正定値計画問題について, 以下の 4 点が判明した.

- 面削減法よりも, 錐拡大法 (1) および錐拡大法 (2) の方が適している.
- しかし, 面削減法および錐拡大法のどちらを適用した場合も, 解くのに適しているとは断言できない.
- 削減方向の計算方法は, 解くことができる問題数には関係が無い.
- 目的関数値を求める場合, 錐拡大法 (2) が適している.

今後の課題としては, 実行可能だが双対ギャップが存在する半正定値計画問題の実行可能性を判定することができるような手法を考えること, より多くの多様な問題を解くことなどが挙げられる.

参考文献

- [1] 田村明久, 村松正和, 最適化法, 共立出版, 2002
- [2] Frank Permenter, Henrik A. Friberg, Erling D. Andersen, Solving conic optimization problems via self-dual embedding and facial reduction: a unified approach, 2016.8.17
- [3] 村松正和, 面の話, http://www.orsj.or.jp/archive2/or59-01/or59_1_5.pdf, 2016.12.22
- [4] Bruno F. Lourenço, Masakazu Muramatsu, Takashi Tsuchiya, Solving SDP Completely with an Interior Point Oracle, 2015.7
- [5] Minghui Liu, Gabor Pataki, Exact duals and short certificates of infeasibility and weak infeasibility in conic linear programming, 2015.6
- [6] Y.-L. Cheung, S. Schurr, H. Wolkowicz, Preprocessing and regularization for degenerate semidefinite programs, In Computational and Analytical Mathematics, pp 251-303, 2013
- [7] Frank Permenter and Pablo Parrio, Partial facial reduction: simplified, equivalent sdps via approximations of the psd cone, Technical report, <http://arxiv.org/abs/1408.4685>, 2014
- [8] Hans D. Mittelmann, Computing Errors, <http://plato.asu.edu/dimacs/node3.html>, 2002
- [9] Jos F. Sturm, SeDuMi—Optimization over symmetric cones, <http://sedumi.ie.lehigh.edu/>, 2017.1.27
- [10] Kim-Chuan Toh, Michael J. Todd, Reha H. Tutuncu, SDPT3 – a MATLAB software for semidefinite-quadratic-linear programming, <http://www.math.nus.edu.sg/mat-tohkc/sdpt3.html>, 2017.1.27
- [11] Gabor Pataki, A Simple Derivation of a Facial Reduction Algorithm and Extended Dual Systems, Dept. of Statistics and OR, 2000
- [12] Masakazu Muramatsu, Weak Status of Conic Programming, POC 2014 at Wuxi, 2014

謝辞

本研究を進めるにあたりご指導を頂きました村松正和教授をはじめ, 高橋里司助教, 村松・高橋研究室の皆様がこの場を借りて御礼申し上げます.