線形計画法*

半場 滋[†]

1 目的

本実験の目的は, いくつかのアルゴリズムを利用して実際に問題を解くことでシステム工学の重要な一分野である線形計画法に関する理解を深めることである。

2 原理

本節では線形計画法の原理について述べる。今後線形計画法を必要とする学生のために説明を詳しくしてあるが, 本節の内容が理解できなくても実験には差し障えない。**興味のない者はこの節をすべて読み飛ばして 14 ページの実験の節に進んでも構わない**。

2.1 線形計画問題

線形計画問題とは, 与えられた線形の等式および不等式制約のもとで, 線形目的関数を最大化あるいは最小化する問題である。まず例を挙げる。

例 1 パソコンショップの店主である A さんは, ある顧客から「PC/AT 互換機のシステムを組んで欲しい」という依頼を受けた。顧客の要望は, 「メモリ 100MB 以上, ディスク 5GB 以上, 予算はディスクとメモリを合わせて 10 万円以下, 予算の範囲でメモリもディスクも可能な限りたくさん載せたい」というものであった。

話を簡単にするため, メモリおよびディスクの大きさは任意の正の実数値を取るものとする。メモリの価格は 1MB あたり 100 円, ディスクの価格は 1GB あたり 2,500 円とする。また, 顧客の依頼したパソコンには, メモリは最大 800MB まで, ディスクは無限に増設できるものとする。A さんの店では, メモリは 1MB あたり 10 円, ディスクは 1GB あたり 200 円の利益があるものとする。A さんとしては, なるべく多くの利益をあげたい。さて, どのような意思決定をするのが良いだろうか?

メモリの大きさを x MB, ディスクの大きさを y GB としよう。すると, A さんの利益 (目的関数) は,

$$10x + 200y \tag{1}$$

となる。よって, A さんが解くべき問題は, 「$10x + 200y$ を最大化すること」となる。

メモリの大きさ x とディスクの大きさ y を勝手に設定できるなら, A さんはいくらでもお金を儲けられる。だが実際には制約条件がある。それは,

- メモリの大きさは正
- メモリの大きさは 100 MB 以上 (顧客の要望)
- メモリの大きさは 800 MB 以下 (計算機の仕様)
- ディスクの大きさは正
- ディスクの大きさは 5 GB 以上 (顧客の要望)
- メモリとディスクの合計金額は 10 万円以下

というものである。これらを書き直すと,

$$\begin{array}{rcl} x & \geq & 0 \\ x & \geq & 100 \\ x & \leq & 800 \\ y & \geq & 0 \\ y & \geq & 5 \\ 100x + 2500y & \leq & 100000 \end{array} \tag{2}$$

となる。

図 1 は, 制約条件を満たす領域の形と目的関数のようすをあらわしたものである。制約条件を満たす領域は図中の網のかかった台形の内部である。目的関数は図中に引かれた傾き -1 の一点鎖線上で一定値を取る。この直線が図中の右上に移動するほど目的関数の値は大きくなる。この図から, 台形の右上の頂点の部分で目的関数が最大値を取ることがわかる。

例 1 は, 結局

> 変数 (x, y), 目的関数 $10x + 200y$ および制約条件 (2) が与えられたとき, 与えられた制約条件のもとで目的関数を最大化するような x, y を求めよ

*この文書は琉球大学工学部電気電子工学科で開講されている 4 年次学生向けの専門実験 (電力工学実験, 電子工学実験, システム工学実験) 向けに執筆されたテキストに若干手を加えたものである。内容を改変しない限り自由に再配布して頂いて構わない。

[†]琉球大学工学部電気電子工学科

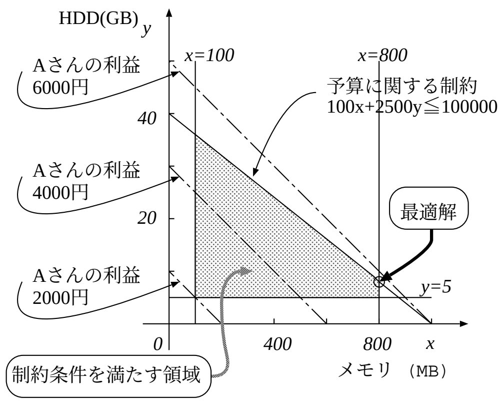


図 1: 制約条件を満たす領域と目的関数の挙動

という問題であった。これを一般化したのが線形計画問題である。

線形計画問題 (一般形) n 個の変数 $x_1, \ldots, x_n$ に対し, m 個の等式あるいは不等式の制約条件

$$\begin{array}{rcl} a_{11}x_1 + \ldots + a_{1n}x_n & \geq & b_1 \\ \cdots\cdots & & \\ a_{k1}x_1 + \ldots + a_{kn}x_n & = & b_k \\ \cdots\cdots & & \\ a_{l1}x_1 + \ldots + a_{ln}x_n & \leq & b_l \\ \cdots\cdots & & \end{array} \tag{3}$$

が与えられ, いくつかの変数に符号に関する制約条件 $x_{i_1} \geq 0, \ldots, x_{i_r} \geq 0$, $x_{j_1} \leq 0, \ldots, x_{j_s} \leq 0$ が課されているものとする。このとき, 制約条件 (3) を満たす範囲内で, 目的関数

$$c_1x_1 + \ldots + c_nx_n \tag{4}$$

を最小化 (あるいは最大化) せよ。

ここに, 最小化 (最大化) とは, 目的関数を最小 (最大) にする x_1 から x_n の値とその最小値 (最大値) を求めることを言う。

線形計画問題の例として, 工場における生産計画の問題などが挙げられる。実用的な線形計画問題では, 変数および制約条件の数は極めて大きくなることが多い。

線形計画問題の解法としてよく知られているものに, シンプレックス法と内点法がある。シンプレックス法は, 制約領域の隣接する頂点をたどりながら解を改善する方法である。一方, 内点法は, 制約条件を満たす領域の内部で目的関数が減少する方向に解を動かす方法である。

シンプレックス法は, 1947 年に G. B. Dantzig によって提案されて以降 40 年ほどの間, 最も効率的な線形計画問題の解法とされてきた。しかし, 1984 年に N. Karmarkar によって内点法が提案され, 多くの研究者の努力によって研究が進んだ結果, 今日では, 内点法は理論的な特性がシンプレックス法より良いばかりでなく, 大規模な問題を解くときには実用上もシンプレックス法より高速であることが知られている (ただし, 規模の小さい問題を解くときにはシンプレックス法の方が高速である)。

2.2 標準形と基準形

線形計画問題の一般形は, 異なる向きの不等号や制約条件が付いた変数と制約条件のない変数が混在しているために, 数学的に取り扱いにくい。そこで, ふつうは, 問題を標準的な形に変換してから解く。

標準的な形としてよく用いられるのは, 標準形と呼ばれるものと, 基準形と呼ばれるものである。以下でこれらについて説明する。

以下では, 特に断らない限り, $\boldsymbol{A}$ は m 行 n 列の実行列, $\boldsymbol{b}$ は m 次元のベクトル, $\boldsymbol{c}$, $\boldsymbol{x}$ は n 次元のベクトルとする。また, $\boldsymbol{A}$ が m 行 n 列の実行列であることを $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ という記法であらわし, $\boldsymbol{x}$ が n 次元の実ベクトルであるということを $\boldsymbol{x} \in \mathbb{R}^n$ という記法であらわす。さらに, 行列 $\boldsymbol{A}$ の第 i 番目の列ベクトルを $\boldsymbol{a}_i$ などと書き, ベクトル $\boldsymbol{b}$, $\boldsymbol{c}$ などの第 i 要素をそれぞれ b_i, c_i などと書く。また, $\boldsymbol{x}$ と $\boldsymbol{w}$ がともに n 次のベクトルであるとき, $\boldsymbol{x} \geq \boldsymbol{w}$ という記法によって, ベクトル $\boldsymbol{x}$ の各成分がベクトル $\boldsymbol{w}$ の対応する成分以上である, すなわち $x_i \geq w_i$ $(i = 1, \ldots, n)$ となっているということをあらわす。零ベクトルあるいは零行列をあらわすのには記号 $\mathbf{0}$ を使う。次元を明示する必要があるときには $\mathbf{0}_n$, $\mathbf{0}_{m\times n}$ などのように書く。

線形計画問題の標準形とは, 以下のような形式の問題である。

標準形: $\min\{z = \boldsymbol{c}^T\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \mathbf{0}\}$

この式の読み方は, 「制約条件 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, $\boldsymbol{x} \geq \mathbf{0}$ のもとで目的関数 $z = \boldsymbol{c}^T\boldsymbol{x}$ を最小化せよ」である。なお, min は minimize(最小化する) の略である。

一方, 線形計画問題の基準形とは, 次のような問題である。

基準形: $\min\{z = \boldsymbol{c}^T\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}, \boldsymbol{x} \geq \mathbf{0}\}$

続いて, 線形計画問題の一般形を標準形に変換する手順について説明する。このためには, 以下に述べるような一連の手順を実行すればよい。

1. 不等式制約 $a_{i,1}x_1 + \cdots + a_{i,n}x_n \geq b_i$: 新たな変数 $y_i \in \mathbb{R}$ を導入し, 制約条件を $a_{i,1}x_1 + \cdots + a_{i,n}x_n - y_i \geq b_i, y_i \geq 0$ と書き直す。

2. 不等式制約 $a_{i,1}x_1+\cdots+a_{i,n}x_n \le b_i$: 新たな変数 $y_i \in \mathbb{R}$ を導入し, 制約条件を $a_{i,1}x_1+\cdots+a_{i,n}x_n+y_i \ge b_i, y_i \ge 0$ と書き直す。

3. 符号制約なしの変数 x_i: 新たに 2 個の正の変数 x_{i-}, x_{i+} を導入し, $x_i = x_{i+} - x_{i-}$, $x_{i+} \ge 0$, $x_{i-} \ge 0$ と書き直す。また, $x_i = x_{i+} - x_{i-}$ を制約条件や目的関数の式に代入する。

4. 正でない変数 $x_i \le 0$: $x'_i = -x_i$ と変数を置き直す。また, $x_i = -x'_i$ を制約条件や目的関数の式に代入する。

5. $\boldsymbol{c}^T\boldsymbol{x}$ を最大化する問題 (最大化問題): $\boldsymbol{c}' = -\boldsymbol{c}$ と定義して, 新たな目的関数 $(\boldsymbol{c}')^T\boldsymbol{x}$ を最小化する。

上記の手続きのうち, 1, 2, 3 では変数の数がそれぞれ 1 個ずつ増えることを注意しておく。

上記と同様の手順によって, 線形計画問題の一般形を基準形に変換することもできる (制約条件や変数の数は増加する)。 なお, 上記とは別の方法を使うと, 等式制約条件や制約条件なしの変数を書き換える際に, 制約条件や変数の数を減らせることが知られている [6]。

2.3 シンプレックス法

本節ではシンプレックス法について簡単に説明する。

標準形で記述された線形計画問題:

$$\text{(STD)} \quad \min\left\{z = \boldsymbol{c}^T\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \ge \boldsymbol{0}\right\} \tag{5}$$

が与えられているものとする。先に注意したように, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, $\boldsymbol{c} \in \mathbb{R}^n$ である。また, $n \ge m$ で, 行列 $\boldsymbol{A}$ はフルランクであると仮定しておく。 さらに, (STD) の制約条件を満たす領域は空でないと仮定する。

シンプレックス法は, まず制約条件を満たす基底解と呼ばれる解[i] をひとつ見付け, 続いて最適解が得られるまで基底解を改善していく方法である。標準形の問題が一定の条件を満たすときは, シンプレックス法を用いると有限回の計算で最適解が得られることが保証される。

ではシンプレックス法の説明に入ろう。

行列 $\boldsymbol{A}$ の列ベクトル $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$ の中から適当な m 個の線形独立なベクトルを選び出し, これらを並べて

$$\boldsymbol{B} = [\boldsymbol{a}_{i_1}, \ldots, \boldsymbol{a}_{i_m}] \tag{6}$$

という行列を作る。また, 行列 $\boldsymbol{A}$ の列ベクトルから行列 $\boldsymbol{B}$ の列ベクトルを除いた残りを並べて,

$$\boldsymbol{N} = [\boldsymbol{a}_{j_1}, \ldots, \boldsymbol{a}_{j_{n-m}}] \tag{7}$$

という行列を作る。行列 $\boldsymbol{B}$, $\boldsymbol{N}$ はそれぞれ基底行列, 非基底行列と呼ばれる。さらに, 行列 $\boldsymbol{B}$ に対応する変数の組 $x_{i_1}, \ldots, x_{i_m}$ を集めて

$$\boldsymbol{x}_B = [x_{i_1}, \ldots, x_{i_m}]^T \tag{8}$$

なるベクトルを作る。$\boldsymbol{x}_B$ を基底変数ベクトルと呼ぶ。続いて, 行列 $\boldsymbol{N}$ に対応する変数の組 $x_{j_1}, \ldots, x_{j_{n-m}}$ を集めて

$$\boldsymbol{x}_N = [x_{j_1}, \ldots, x_{j_{n-m}}]^T \tag{9}$$

なるベクトルを作る。$\boldsymbol{x}_N$ を非基底変数ベクトルと呼ぶ。

ここで, 目的関数に対応するベクトル $\boldsymbol{c}$ を基底行列に対応した形に分割し,

$$\boldsymbol{c}_B = [c_{i_1}, \ldots, c_{i_m}], \quad \boldsymbol{c}_N = [c_{j_1}, \ldots, c_{j_{n-m}}] \tag{10}$$

とおくと, 目的関数 $z = \boldsymbol{c}^T\boldsymbol{x}$ は

$$z = \boldsymbol{c}_B^T\boldsymbol{x}_B + \boldsymbol{c}_N^T\boldsymbol{x}_N \tag{11}$$

と書き直される。

さて, 変数ベクトル $\boldsymbol{x}$ を

$$\boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b}, \quad \boldsymbol{x}_N = \boldsymbol{0} \tag{12}$$

となるように選んでみよう。等式制約条件 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ は

$$\boldsymbol{B}\boldsymbol{x}_B + \boldsymbol{N}\boldsymbol{x}_N = \boldsymbol{b} \tag{13}$$

と書き直せるから, (12) から決まる $\boldsymbol{x}$ は制約条件 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ を満たしている。(12) で定まる $\boldsymbol{x}$ を基底解と呼ぶ。ただし, この段階では, $\boldsymbol{x} \ge \boldsymbol{0}$ であることは保証されない。

ところで, $\boldsymbol{B}^{-1}\boldsymbol{b} \ge \boldsymbol{0}$ となる場合には, $\boldsymbol{x}_B \ge 0$, $\boldsymbol{x}_N \ge 0$ となり, 基底解 $\boldsymbol{x}$ は符号制約 $\boldsymbol{x} \ge \boldsymbol{0}$ を満たす。このとき基底解 $\boldsymbol{x}$ は問題 (STD) のひとつの解になる。$\boldsymbol{B}^{-1}\boldsymbol{b} \ge \boldsymbol{0}$ のとき, (12) により定まる $\boldsymbol{x}$ を実行可能基底解と呼び, 対応する行列 $\boldsymbol{B}$ を実行可能基底行列と呼ぶ。

以下では, まず実行可能基底行列がひとつ得られている場合に最適解を構成するためのアルゴリズムについて説明し, 続いて実行可能基底行列が未知の場合への拡張について述べる。

2.3.1 実行可能基底行列が得られている場合

実行可能基底行列 $\boldsymbol{B}$ がすでに得られ, 制約条件 $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ が (13) の形に書き直されているものとする。ここで,

$$\bar{\boldsymbol{b}} = \boldsymbol{B}^{-1}\boldsymbol{b}, \quad \bar{\boldsymbol{A}}_N = \boldsymbol{B}^{-1}\boldsymbol{N} \tag{14}$$

と定義し, (13) を

$$\boldsymbol{x}_B = \bar{\boldsymbol{b}} - \bar{\boldsymbol{A}}_N\boldsymbol{x}_N \tag{15}$$

[i] ここで言う「解」とは制約条件をすべて満たす変数ベクトルのことである。最適解 (制約条件をすべて満たし, かつ目的関数を最小化するもの) とは異なる。

のように書き直す。(15) を (11) に代入し,

$$z_B = \boldsymbol{c}_B^T \bar{\boldsymbol{b}}, \quad \bar{\boldsymbol{c}}_N^T = \boldsymbol{c}_N^T - \boldsymbol{c}_B^T \bar{\boldsymbol{A}}_N \tag{16}$$

とおくと,

$$z = z_B + \bar{\boldsymbol{c}}_N^T \boldsymbol{x}_N \tag{17}$$

なる表現が得られる。

ここで, 行列 $\bar{\boldsymbol{A}}_N$ の第 i 列ベクトルを $\bar{\boldsymbol{a}}_i$ と書くことにする。すなわち,

$$\bar{\boldsymbol{A}}_N = [\bar{\boldsymbol{a}}_1, \ldots, \bar{\boldsymbol{a}}_{n-m}] \tag{18}$$

と書く。(15) と (17) をまとめると, 問題 (SP) と等価な

$$\begin{aligned}(\mathrm{BF}) \quad \min\{z = z_B + \bar{\boldsymbol{c}}_N^T \boldsymbol{x}_N :\\ \boldsymbol{x}_B = \bar{\boldsymbol{b}} - \bar{\boldsymbol{A}}_N \boldsymbol{x}_N, \boldsymbol{x}_B \geq \boldsymbol{0}, \boldsymbol{x}_N \geq \boldsymbol{0}\}\end{aligned} \tag{19}$$

という問題が得られる。問題 (BF) を実行可能基底行列 $\boldsymbol{B}$ に対する基底形式表現と呼ぶ。

さて, 基底解 (12) は (BF) の制約条件を満たしているので, 最適とは限らない解である。そこで, 基底解 (12) を変更して目的関数 z をより小さくすることを考える。このために, (15) と (17) を見比べると, 以下のことがわかる。

- (16) のベクトル $\bar{\boldsymbol{c}}_N$ の要素の中から値が最小のものを探す。添字 s を持つ要素 $\bar{c}_{N,s}$ が最小値であったものとしよう。$\bar{c}_{N,s} \geq 0$ であれば, 非基底変数をどのように変更しても目的関数は減少しないから, 現在の z がすでに最小値である。逆に $\bar{c}_{N,s} < 0$ のときには, (17) で $\bar{c}_{N,s}$ に対応する非基底変数 x_{j_s} の値を零から増加させれば z は減少するから, 現在の解を改善できる可能性がある。

- 非基底変数 x_{j_s} は (19) の制約条件を崩さない範囲で選ばれなければならない。x_{j_s} の値を零から θ に変更したとき, $\boldsymbol{x}_B = \bar{\boldsymbol{b}} - \bar{\boldsymbol{a}}_s \theta$ となる。ベクトル $\bar{\boldsymbol{b}}$ の各成分を $\bar{b}_1, \ldots, \bar{b}_m$, ベクトル $\bar{\boldsymbol{a}}_s$ の各成分を $\bar{a}_{s;1}, \ldots, \bar{a}_{s;m}$ と書く。$\boldsymbol{x}_B \geq \boldsymbol{0}$ という制約から, $\bar{a}_{s,i}$ が正のときには, θ を $\bar{b}_i / \bar{a}_{s;i}$ 以下に取る必要がある。これに対し, $\bar{a}_{s,i} = 0$ のときは, θ をいくら大きく取っても制約条件はつねに満たされる。よって, 添字集合 $\{1, \ldots, m\}$ から $\bar{a}_{s;i} > 0$ となるものを抜き出し,

  $$\bar{\theta} = \min\{\bar{b}_i / \bar{a}_{s;i} : \bar{a}_{s;i} > 0, 1 \leq i \leq m\}$$

  とすると, この $\bar{\theta}$ が θ の取りうる上限となる。一方, $a_s \leq 0$ のときには, θ をいくらでも大きく取ることができるため, z の値をいくらでも小さくすることができる。

以上のような考え方からシンプレックス法のアルゴリズムが得られる[ii] 。

シンプレックス法

(初期化) 実行可能基底行列 $\boldsymbol{B}$ が与えられているものとする。$\boldsymbol{B}$ を使って基底形式表現 (19) を求める。

(ステップ 1) $\{\bar{c}_{N,1}, \ldots, \bar{c}_{N,n-m}\}$ の中で最小のものを $\bar{c}_{N,s}$ とする[iii] 。$\bar{c}_{N,s} \geq 0$ であれば

$$\boldsymbol{x}_B^* = \boldsymbol{B}^{-1} \boldsymbol{b}, \quad \boldsymbol{x}_N^* = \boldsymbol{0} \tag{20}$$

が最適解なので終了する。$\bar{c}_{N,s} < 0$ であればステップ 2 に進む。

(ステップ 2) 非基底行列 $\boldsymbol{N}$ の第 s 列 $\boldsymbol{a}_{j_s}$ を取り, 線形方程式

$$\boldsymbol{B} \bar{\boldsymbol{a}}_s = \boldsymbol{a}_{j_s} \tag{21}$$

を解いてベクトル $\bar{\boldsymbol{a}}_s$ を求める。$\bar{\boldsymbol{a}}_s \leq \boldsymbol{0}$ なら終了 (無限解が存在する)。そうでないときは, $\bar{\theta} = \min\{\bar{b}_i / \bar{a}_{s;i} : \bar{a}_{s;i} > 0, 1 \leq i \leq m\}$ とする。ある r に対して $\bar{\theta} = \bar{b}_r / \bar{a}_{s;r}$ となっているので, このような添字 r を選ぶ[iv] 。

(ステップ 3) 添字 r と s を用いて基底行列を

$$\boldsymbol{B} = [\boldsymbol{a}_{i_1}, \ldots, \boldsymbol{a}_{i_{r-1}}, \boldsymbol{a}_{i_r}, \boldsymbol{a}_{i_{r+1}}, \ldots, \boldsymbol{a}_{i_m}] \tag{22}$$

から

$$\boldsymbol{B}' = [\boldsymbol{a}_{i_1}, \ldots, \boldsymbol{a}_{i_{r-1}}, \boldsymbol{a}_{j_s}, \boldsymbol{a}_{i_{r+1}}, \ldots, \boldsymbol{a}_{i_m}] \tag{23}$$

に変更し, 新しい基底変数と非基底変数から対応する $\bar{\boldsymbol{c}}_N$ を求めてステップ 1 に戻る。

なお, (14) の第 1 式の $\bar{\boldsymbol{b}}$ を求めるときには, 逆行列は使わず, 線形方程式

$$\boldsymbol{B} \bar{\boldsymbol{b}} = \boldsymbol{b} \tag{24}$$

を解く。一般に, 逆行列の数値計算は計算量が多くかつ誤差の影響を受けやすいので, 数値解析の分野では逆行列の計算は可能な限り回避される。

[ii] ここで挙げたアルゴリズムは「改訂シンプレックス法」と呼ばれるものである。ところで, 「改訂」シンプレックス法が存在する以上, ただのシンプレックス法も存在する。シンプレックス法と改訂シンプレックスの違いは, シンプレックス法が基底形式表現そのものを更新するのに対し, 改訂シンプレックス法は基底行列に対応する添字を更新して対応する線形方程式を解く, という点にある。高速な線形方程式のソルバが利用可能である場合には改訂シンプレックス法の方が高速である。

[iii] 条件を満たすものが複数あるときには, どれを選んでもよい。

[iv] 条件を満たすものが複数あるときには, どれを選んでもよい。

各ステップでベクトル $\bar{\boldsymbol{c}}_N$ を求めるためには $\boldsymbol{c}_B^T\boldsymbol{B}^{-1}\boldsymbol{N}$ を計算する必要があるが, これを求めるためにはまず線形方程式 $\boldsymbol{B}^T\boldsymbol{c}_{N0} = \boldsymbol{c}_B$ を解いてベクトル $\boldsymbol{c}_{N0}$ を求め, 続いて $\boldsymbol{c}_{N0}$ に行列 $\boldsymbol{N}$ の転置を左から掛ける, という手順で計算をおこなう。

$$(\boldsymbol{c}_B^T\boldsymbol{B}^{-1}\boldsymbol{N})^T = \boldsymbol{N}^T(\boldsymbol{B}^T)^{-1}\boldsymbol{c}_B$$

だから, このようにしても正しい結果が得られることがわかる。$\bar{\boldsymbol{A}}_N = \boldsymbol{B}^{-1}\boldsymbol{N}$ のすべての成分を計算しても正しい結果は得られるが, これは計算量の観点からは無駄である。

ここで, シンプレックス法のステップ 2 で更新された基底解とステップ 3 の基底行列に対応する基底解が一致することを見よう。

記法の簡単のため, 基底行列 $\boldsymbol{B}$ の最後の列ベクトルと非基底行列 $\boldsymbol{N}$ の最後の列ベクトルが入れ換えられるという状況を考える。入れ換えられる列ベクトルを明示するために, $\boldsymbol{B}$ と $\boldsymbol{N}$ の最後の列ベクトルをそれぞれ $\boldsymbol{\beta}$, $\boldsymbol{\nu}$ とし,

$$\boldsymbol{B} = \left[\begin{array}{c|c} \boldsymbol{B}_0 & \boldsymbol{\beta} \end{array}\right], \quad \boldsymbol{N} = \left[\begin{array}{c|c} \boldsymbol{N}_0 & \boldsymbol{\nu} \end{array}\right] \tag{25}$$

と書く。また, $\boldsymbol{e}_{n-m}$ を第 $n-m$ 番目の単位ベクトルとする。

ステップ 2 では, $\boldsymbol{x}_B$ と $\boldsymbol{x}_N$ がそれぞれ $\boldsymbol{x}'_B$ と $\boldsymbol{x}'_N$ に変更されたものとする。すなわち,

$$\begin{aligned} \boldsymbol{x}_B &= \boldsymbol{B}^{-1}\boldsymbol{b}, \\ \boldsymbol{x}'_B &= \boldsymbol{B}^{-1}\boldsymbol{b} - \theta\boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{e}_{n-m}, \\ \boldsymbol{x}_N &= \boldsymbol{0}, \\ \boldsymbol{x}'_N &= \theta\boldsymbol{e}_{n-m} \end{aligned} \tag{26}$$

である。$\boldsymbol{B}$ の最後の列ベクトルと $\boldsymbol{N}$ の最後の列ベクトルが入れ換えられることを仮定したから, (26) のように $\boldsymbol{x}_B$ を変更すると, $\boldsymbol{x}_B$ の最後の要素は零になる。(25) を使うと, (26) の第 2 式は

$$\boldsymbol{x}'_B = \boldsymbol{B}^{-1}\boldsymbol{b} - \theta\boldsymbol{B}^{-1}\boldsymbol{\nu} \tag{27}$$

と書けることを注意しておく。

ここで, $\boldsymbol{x}_B$ と $\boldsymbol{x}'_B$ の第 m 要素とそれ以外を分け,

$$\boldsymbol{x}_B = \left[\begin{array}{c} \boldsymbol{\sigma}_0 \\ \hline \sigma_1 \end{array}\right], \quad \boldsymbol{x}'_B = \left[\begin{array}{c} \boldsymbol{\sigma}_2 \\ \hline 0 \end{array}\right] \tag{28}$$

とおく。 ただし, $\boldsymbol{\sigma}_0$ と $\boldsymbol{\sigma}_2$ は $m-1$ 次のベクトル, σ_1 はスカラーである。

証明すべきことは, 入れ換え後の基底行列

$$\left[\begin{array}{c|c} \boldsymbol{B}_0 & \boldsymbol{\nu} \end{array}\right] \tag{29}$$

に対応する解

$$\boldsymbol{x}''_B = \left[\begin{array}{c|c} \boldsymbol{B}_0 & \boldsymbol{\nu} \end{array}\right]^{-1}\boldsymbol{b}, \quad \boldsymbol{x}''_N = \boldsymbol{0}, \tag{30}$$

が添字の並べ換えによって $\boldsymbol{x}'_B$, $\boldsymbol{x}'_N$ と一致すること, すなわち

$$\boldsymbol{x}''_B = \left[\begin{array}{c} \boldsymbol{\sigma}_2 \\ \hline \theta \end{array}\right] \tag{31}$$

となることである。

これを示すための準備として, まず行列 (29) が正則であることを背理法によって示しておく。 行列 (29) が正則でないと仮定しよう。 行列 $\boldsymbol{B}$ は実行可能基底行列であって正則だから, この仮定はベクトル $\boldsymbol{\nu}$ が行列 $\boldsymbol{B}_0$ の列ベクトルの線形結合となることを意味する。さて, (27) より, $\boldsymbol{B}\boldsymbol{x}'_B = \boldsymbol{b} - \theta\boldsymbol{\nu}$ であるが, (28) の第 2 式を用いると, これは

$$\boldsymbol{B}_0\boldsymbol{\sigma}_2 = \boldsymbol{b} - \theta\boldsymbol{\nu} \tag{32}$$

と書き直せる。ところで, $\boldsymbol{\nu}$ は行列 $\boldsymbol{B}_0$ の列ベクトルの線形結合だったから, あるベクトル $\boldsymbol{\sigma}_3$ に対し, $\boldsymbol{\nu} = \boldsymbol{B}_0\boldsymbol{\sigma}_3$ となる。これを (32) に代入すると,

$$\boldsymbol{b} = \boldsymbol{B}_0(\boldsymbol{\sigma}_2 + \boldsymbol{\sigma}_3) \tag{33}$$

となる。 一方, (26) の第 1 式と (28) の第 1 式より, $\boldsymbol{B}_0\boldsymbol{\sigma}_0 + \boldsymbol{\beta}\sigma_1 = \boldsymbol{b}$ であり, これと (33) を合わせると

$$\boldsymbol{\beta}\sigma_1 = \boldsymbol{B}_0(\boldsymbol{\sigma}_2 + \boldsymbol{\sigma}_3 - \boldsymbol{\sigma}_0) \tag{34}$$

が得られる。 ゆえに, ベクトル $\boldsymbol{\beta}$ は行列 $\boldsymbol{B}_0$ の列ベクトルの線形結合である。一方, 行列 $\boldsymbol{B}$ は正則である。 これは矛盾である。以上によって行列 (29) が正則であることが示された。

続いて, (31) を示そう。 このために,

$$\boldsymbol{x}''_B = \left[\begin{array}{c} \boldsymbol{\sigma}_4 \\ \hline \sigma_5 \end{array}\right] \tag{35}$$

とおく。 このとき, (30) より, $\boldsymbol{B}_0\boldsymbol{\sigma}_4 + \boldsymbol{\nu}\sigma_5 = \boldsymbol{b}$ であり, これを (32) に代入して整理すると,

$$\boldsymbol{B}_0\left(\boldsymbol{\sigma}_4 - \boldsymbol{\sigma}_2\right) = \boldsymbol{\nu}\left(\theta - \sigma_5\right) \tag{36}$$

が得られる。 一方, ベクトル $\boldsymbol{\nu}$ が行列 $\boldsymbol{B}_0$ の各列ベクトルの線形結合で表されないことはすでに示されているから, (36) が成り立つということは, $\boldsymbol{\sigma}_4 = \boldsymbol{\sigma}_2$, $\sigma_5 = \theta$ を意味する。 以上によってシンプレックス法のステップ 2 で更新された解がステップ 3 の基底解と一致することが示された。

2.3.2　もとの問題の最適解の抽出

シンプレックス法終了時に得られている実行可能基底行列を $\boldsymbol{B}^*$ とすると, 線形方程式

$$\boldsymbol{B}^*\boldsymbol{x}_B^* = \boldsymbol{b} \tag{37}$$

を解くことにより最適解の一部 $\boldsymbol{x}_B^*$ が得られる。残りの変数 $\boldsymbol{x}_N^*$ は零である。$\boldsymbol{x}_B^*$ と $\boldsymbol{x}_N^*$ からもとの問題の最適解 $\boldsymbol{x}^*$ を構成するためには, どの変数が基底変数に入っていたかを確認する必要がある。

たとえば, 基底変数が x_1, x_3, x_5 であり, $\boldsymbol{x}_B^* = [10, 20, 30]$ となっている場合には, もとの問題の最適解は

$$[10, 0, 20, 0, 30, 0, ...]$$

である (非基底変数はすべて零である)。

例 2 シンプレックス法の挙動を調べるため, 以下のような 2 変数の簡単な線形計画問題を考える。

$$\min\{x_1 + x_2 : 1 \leq x_1 \leq 2, 1 \leq x_2 \leq 2, x_1 \geq 0, x_2 \geq 0\} \tag{38}$$

問題 (38) の制約条件を満たす領域を図 2 に示す。

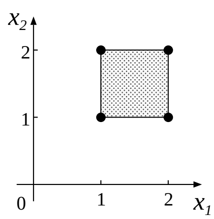


図 2: 問題 (38) の制約条件を満たす領域

目的関数は x_1+x_2 であるが, これは傾き -1 の直線上で一定値となる。そして, この直線が左下にずれるほど目的関数の値は小さくなる。よって, この問題の最適解は $x_1 = 1, x_2 = 1$ であることがわかる。

では, この問題を標準形に書き直そう。不等式 $1 \leq x_1$ は, 新しい変数 x_3 を導入して,

$$x_1 - x_3 = 1, \quad x_3 \geq 0$$

と書ける。不等式 $x_1 \leq 2$ は, 新しい変数 x_4 を導入することで,

$$x_1 + x_4 = 2, \quad x_4 \geq 0$$

のようになる。x_2 に関する不等式も同様にして

$$x_2 - x_5 = 1, x_5 \geq 0, x_2 + x_6 = 2, x_6 \geq 0$$

という等式に変わる。さて,

$$\boldsymbol{x} = [x_1, x_2, x_3, x_4, x_5, x_6]^T$$

とおくと, この問題は

$$\begin{aligned}
&\min\{\boldsymbol{c}^T\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\}, \\
&\boldsymbol{A} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \end{bmatrix}, \\
&\boldsymbol{c} = \begin{bmatrix} 1, 1, 0, 0, 0, 0 \end{bmatrix}^T
\end{aligned} \tag{39}$$

という線形計画問題に書き直される。これは標準形である。

次に, (39) の実行可能基底行列と実行可能基底解を求めてみよう。基底変数を決めるときには 1 から 6 までの添字の中から 4 個の添字を選ぶことになるので, 可能な組み合わせの数は $\begin{pmatrix} 6 \\ 4 \end{pmatrix} = 15$ 通りである。一方, 実行可能基底解は図 2 の制約条件を満たす領域 (正方形) の頂点であり, 頂点の数は 4 個しかないから, 上記の 15 種類の添字の選び方のうち実行可能なものは 4 個しかないと推察される。この推察は正しく, 実際に計算してみると, 特定の 4 通りの添字の選び方に対してのみ実行可能基底解が得られ, それ以外の選び方では, $\boldsymbol{B}$ が正則でなくなるか, $\boldsymbol{B}^{-1}\boldsymbol{b}$ の成分の中に負のものがあらわれるということがわかる。表 1 に実行可能な基底変数ベクトルと, 実行可能基底解に対応する x_1 と x_2 の値を示す。表 1 の右の欄の値から, これらが図 2 の 4 個の頂点に対応していることがわかる。

表 1: 実行可能な基底変数ベクトルと実行可能基底解

$\boldsymbol{x}_B$	(x_1, x_2) の値
$[x_1, x_2, x_3, x_5]^T$	(2, 2)
$[x_1, x_2, x_3, x_6]^T$	(2, 1)
$[x_1, x_2, x_4, x_5]^T$	(1, 2)
$[x_1, x_2, x_5, x_6]^T$	(1, 1)

続いて, 基底変数ベクトルを $\boldsymbol{x}_B = [x_1, x_2, x_3, x_5]^T$ とした場合のシンプレックス法の動作を見よう。シンプレックス法は制約条件を満たす領域の頂点をたどって最適解を探す方法なのだから, 点 $(x_1, x_2) = (2, 2)$ から出発した場合, 2 回の繰り返し計算で最適点 $(x_1, x_2) = (1, 1)$ に到達できるはずである。

1 回目 $\boldsymbol{x}_B = [x_1, x_2, x_3, x_5]^T$, $\boldsymbol{x}_N = [x_4, x_6]^T$ から出発する (点 $(x_1, x_2) = (2, 2)$ から始めることに対応)。これに対応する基底行列は $\boldsymbol{B} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \boldsymbol{a}_5]$ であり, 非基底行列は $\boldsymbol{N} = [\boldsymbol{a}_4, \boldsymbol{a}_6]$ である。また, $\boldsymbol{c}_B = [1, 1, 0, 0]^T$,

$\boldsymbol{c}_N = [0, 0]^T$ となる。ここから, $z_B = 4$,

$$\bar{\boldsymbol{b}} = \begin{bmatrix} 2 \\ 2 \\ 1 \\ 1 \end{bmatrix}, \bar{\boldsymbol{A}}_N = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \bar{\boldsymbol{c}}_N = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

が得られる。

(ステップ 1-1) ベクトル $\boldsymbol{c}_N$ の第 1 要素, 第 2 要素はともに -1 で負なので, このうち番号が若い $s = 1$ を選択して次に進む。

(ステップ 1-2) $\bar{\boldsymbol{a}}_1 = [1, 0, 1, 0]^T$ で, このうち第 1 成分と第 3 成分が正である。$\bar{b}_1/\bar{a}_{1;1} = 2$ $\bar{b}_3/\bar{a}_{1;3} = 1$ であり, $\bar{b}_3/\bar{a}_{1;3}$ の方が小さいので, 添字 $r = 3$ を選択して次に進む。

(ステップ 1-3) 基底行列は

$$\boldsymbol{B} = [\boldsymbol{a}_{i_1}, \boldsymbol{a}_{i_2}, \overset{\Downarrow}{\boldsymbol{a}}_{i_3}, \boldsymbol{a}_{i_4}]$$

から

$$\boldsymbol{B}' = [\boldsymbol{a}_{i_1}, \boldsymbol{a}_{i_2}, \overset{\Downarrow}{\boldsymbol{a}}_{j_1}, \boldsymbol{a}_{i_4}]$$

に変わる。ここで $\boldsymbol{x}_B = [x_1, x_2, x_3, x_5]^T$, $\boldsymbol{x}_B = [x_4, x_6]^T$ であったことを思い出すと, $i_1 = 1$, $i_2 = 2$, $i_3 = 3$, $i_4 = 5$, $j_1 = 4$, $j_2 = 6$ であることがわかる。ゆえに, 新しい基底行列は

$$\boldsymbol{B}' = [\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_4, \boldsymbol{a}_5]$$

となる。対応する基底変数ベクトルは $\boldsymbol{x}_B = [x_1, x_2, x_4, x_5]^T$ である。

2 回目 1 回目の終了時の状態を継承して, $\boldsymbol{x}_B = [x_1, x_2, x_4, x_5]^T$, $\boldsymbol{x}_N = [x_3, x_6]^T$ から出発する。表 1 から, これが点 $(x_1, x_2) = (1, 2)$ に対応することがわかる。先と同様の計算により, $z_B = 3$,

$$\bar{\boldsymbol{b}} = \begin{bmatrix} 1 \\ 2 \\ 1 \\ 1 \end{bmatrix}, \bar{\boldsymbol{A}}_N = \begin{bmatrix} -1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \bar{\boldsymbol{c}}_N = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

が得られる。

(ステップ 2-1) ベクトル $\boldsymbol{c}_N$ の第 2 要素のみ負なので, $s = 2$ を選択して次に進む。

(ステップ 2-2) $\bar{\boldsymbol{a}}_1 = [0, 1, 0, 1]^T$ で, このうち第 2 成分と第 4 成分が正である。$\bar{b}_2/\bar{a}_{2;2} = 2$, $\bar{b}_4/\bar{a}_{2;2} = 1$ であり, $\bar{b}_4/\bar{a}_{2;4}$ の方が小さいので, 添字 $r = 4$ を選択して次に進む。

(ステップ 2-3) 基底行列は

$$\boldsymbol{B} = [\boldsymbol{a}_{i_1}, \boldsymbol{a}_{i_2}, \boldsymbol{a}_{i_3}, \overset{\Downarrow}{\boldsymbol{a}}_{i_4}]$$

から

$$\boldsymbol{B}' = [\boldsymbol{a}_{i_1}, \boldsymbol{a}_{i_2}, \boldsymbol{a}_{i_3}, \overset{\Downarrow}{\boldsymbol{a}}_{j_2}]$$

に変わる。ここで $\boldsymbol{x}_B = [x_1, x_2, x_4, x_5]^T$, $\boldsymbol{x}_B = [x_3, x_6]^T$ であったことを思い出すと, $i_1 = 1$, $i_2 = 2$, $i_3 = 4$, $i_4 = 5$, $j_1 = 3$, $j_2 = 6$ であることがわかる。ゆえに, 新しい基底行列は

$$\boldsymbol{B}' = [\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_4, \boldsymbol{a}_6]$$

となる。対応する基底変数ベクトルは $\boldsymbol{x}_B = [x_1, x_2, x_4, x_6]^T$ である。

3 回目 2 回目の終了時の状態を継承して, $\boldsymbol{x}_B = [x_1, x_2, x_4, x_6]^T$, $\boldsymbol{x}_N = [x_3, x_5]^T$ から出発する。表 1 から, これが点 $(x_1, x_2) = (1, 1)$ に対応することがわかる。この点で目的関数は最小となるから, アルゴリズムはこの段階で終了するはずである。これを確認しよう。先と同様の計算により, $z_B = 2$,

$$\bar{\boldsymbol{b}} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \bar{\boldsymbol{A}}_N = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \bar{\boldsymbol{c}}_N = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{40}$$

が得られる。

(ステップ 3-1) ベクトル $\boldsymbol{c}_N$ のすべての要素は零である。よって, すでに最適解が得られているので, 計算を終了する。

もとの問題の解の復元 計算終了時の基底変数ベクトルが $\boldsymbol{x}_B = [x_1, x_2, x_4, x_6]^T$ であったことと最適解が $\boldsymbol{x}_B = \bar{\boldsymbol{b}}$, $\boldsymbol{x}_N = \boldsymbol{0}$ とあらわされることを思い出すと, (40) から, $x_1 = 1$, $x_2 = 1$, $x_3 = 0$, $x_4 = 1$, $x_5 = 0$, $x_6 = 1$ が最適解となる。よって, $x_1 = 1$, $x_2 = 1$ が目的関数を最小とする点であることがわかる。

以上の説明から, シンプレックス法のアルゴリズムが, 点 $(x_1, x_2) = (2, 2)$ から出発し, $(x_1, x_2) = (1, 2)$ という頂点を経由して最適点 $(x_1, x_2) = (1, 1)$ に到達していることがわかる。

2.3.3 ビッグ M 法

第 2.3.1 節で述べたアルゴリズムを利用して与えられた線形計画問題を解くためには実行可能基底行列 $\boldsymbol{B}$ が必要であった。ところで, 問題の規模が大きいときには実行可能基底行列をひとつ求めること自体も容易でないことが多い。

そのような状況に対処するためには, 第 2.3.1 節のアルゴリズムを実行可能基底行列が未知の場合にも適用できるように拡張しておく必要がある。

このような拡張としてよく知られているものに, 2 段階法とビッグ M 法がある。本節では, これらのうちビッグ M 法について述べる。

(STD) にビッグ M 法が適用可能であるためには (STD) の制約条件の右辺にあらわれるベクトル $\boldsymbol{b}$ の各成分が非負でなければならない。そこで, $\boldsymbol{b}$ の第 i 成分 b_i が負で, 対応する行列 $\boldsymbol{A}$ の行ベクトルが $\boldsymbol{\alpha}_i^T$ であるときには, b_i を $-b_i$ で置き換え, $\boldsymbol{\alpha}_i^T$ を $-\boldsymbol{\alpha}_i^T$ で置き換える。これにより, もとの制約条件と等価で $\boldsymbol{b} \geq \boldsymbol{0}$ を満たす制約条件が得られる。

次に, 人為変数と呼ばれる新たな変数

$$\boldsymbol{\xi} = [\xi_1, \ldots, \xi_m]^T \tag{41}$$

を導入し,

$$\boldsymbol{c}_2 = [\boldsymbol{c}^T, M \cdot \boldsymbol{1}_m^T]^T, \quad \boldsymbol{A}_2 = [\boldsymbol{A}, \boldsymbol{I}_m] \tag{42}$$

と定義して (ただし $\boldsymbol{1}_m$ は m 次の要素がすべて 1 のベクトル, M は十分大きい正の定数, $\boldsymbol{I}_m$ は m 次の単位行列), (5) から次のような問題を作る。

$$\min \left\{ \boldsymbol{c}_2^T \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\xi} \end{bmatrix} : \boldsymbol{A}_2 \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\xi} \end{bmatrix} = \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}, \boldsymbol{\xi} \geq \boldsymbol{0} \right\} \tag{43}$$

もとの問題は $\boldsymbol{b} \geq \boldsymbol{0}$ となるように書き換えられていたから, $\boldsymbol{x} = \boldsymbol{0}, \boldsymbol{\xi} = \boldsymbol{b}$ は (43) の制約条件を満たす。よって, これは実行可能基底解である。ゆえに, 問題 (43) において基底変数ベクトルとして $\boldsymbol{\xi}$ を取ると, 実行可能基底行列が得られるから, ここから出発してシンプレックス法を実行し, 最適解を求めることが可能である。

シンプレックス法の終了条件が満たされ, 繰り返し計算が終了したものとしよう。このとき問題となるのは, もとの問題の最適解をどのようにして復元するか, ということである。

パラメータ M が十分大きく, かつ計算終了時に有限の最小値が得られた場合を考える。この最小値に対応する $\boldsymbol{x}$ と $\boldsymbol{\xi}$ をそれぞれ $\boldsymbol{x}^*$ と $\boldsymbol{\xi}^*$ とする。このとき, $\boldsymbol{\xi}^* = \boldsymbol{0}$ であれば $\boldsymbol{x} = \boldsymbol{x}^*$ が問題 (5) の最適解であることが言え, $\boldsymbol{\xi}^* \neq \boldsymbol{0}$ であれば問題 (5) は解を持たないということが言える。ただし, パラメータ M をどの程度取れば十分であるかについての基準はない。よって, 計算終了時に $\boldsymbol{\xi}^* \neq \boldsymbol{0}$ となったときには, M を大きく取り直してシンプレックス法を再度実行する必要がある。

なお, 計算終了時の結果からもとの問題の最適解 $\boldsymbol{x}^*$ を構成する手順は第 2.3.2 節と同一である。

2.3.4 ビッグ M 法による最適化の結果ともとの問題の解の関係

以下でビッグ M 法による最適化の結果と問題 (5) の解の関係に関する事実を証明つきで述べる。ただし, M は十分大きいと仮定する。

2.3.4.1 ビッグ M 法により有限の最適解が求められたとき

ビッグ M 法で有限の最適解 $[\boldsymbol{x}^{*T}, \boldsymbol{\xi}^{*T}]^T$ が得られているものとする。 このとき, $\boldsymbol{\xi}^{*T}$ が零ベクトルなら, $\boldsymbol{x}^*$ は問題 (5) の最適解である。 一方, $\boldsymbol{\xi}^*$ が零ベクトルでないときには, 問題 (5) の制約条件を満たす領域は存在しない。

(証明) まず最初に, $\boldsymbol{\xi}^* = \boldsymbol{0}$ の場合を考える。制約条件を満たす任意の点 $\boldsymbol{x}$ に対し, $[\boldsymbol{x}^T, \boldsymbol{0}^T]^T$ は (43) の制約条件を満たす。一方, $[\boldsymbol{x}^{*T}, \boldsymbol{0}^T]^T$ は (43) の最小解だから,

$$\boldsymbol{c}_2^T \begin{bmatrix} \boldsymbol{x}^* \\ \boldsymbol{0} \end{bmatrix} \leq \boldsymbol{c}_2^T \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{0} \end{bmatrix} \tag{44}$$

である。 すなわち, $\boldsymbol{x}^*$ は (5) の最小解である。

次に, $\boldsymbol{\xi}^* \neq \boldsymbol{0}$ の場合を考える。このとき, (5) の制約条件を満たす点は存在しないことを背理法によって示す。まず最初に, $[\boldsymbol{x}^{*T}, \boldsymbol{\xi}^{*T}]^T$ の候補は実行可能基底解全体の集合であって, これは実行可能基底行列の集合から決まる有限集合ことに注意する。この集合は (43) の制約条件のみから決定され, M には依存しない。

さて, 制約条件を満たす点 $\boldsymbol{x}$ が存在すれば, $[\boldsymbol{x}^T, \boldsymbol{0}^T]^T$ は (43) の制約条件を満たすから,

$$\boldsymbol{c}_2^T \begin{bmatrix} \boldsymbol{x}^* \\ \boldsymbol{\xi}^* \end{bmatrix} \leq \boldsymbol{c}_2^T \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{0} \end{bmatrix} \tag{45}$$

でなければならない。 一方, (45) の左辺は M を大きくすれば $+\infty$ に発散する ($\boldsymbol{\xi}^*$ は有限で零でないことに注意)。これは矛盾である。 したがって, 制約条件を満たす点 $\boldsymbol{x}$ が存在しないことがいえる。

2.3.4.2 ビッグ M 法の問題が有限の最適解を持たないとき

問題 (43) が有限の最適解を持たないときには, 問題 (5) も有限の最適解を持たない。

(証明) ビッグ M 法が有限の最適解を持たないと判定された段階における基底変数ベクトルを $\boldsymbol{x}_B$, 非基底変数ベクトルを $\boldsymbol{x}_N$ とする。非基底変数ベクトルの第 i 成分を動かすことで目的関数が任意に小さくできるものとする。すなわち, $\boldsymbol{e}_i$ を第 i 単位ベクトルとしたとき, $\boldsymbol{x}_B$ と $\boldsymbol{x}_N$ をそれぞれ

$$\begin{array}{lccr} \boldsymbol{x}_B: & \boldsymbol{B}^{-1}\boldsymbol{b} & \rightarrow & \boldsymbol{B}^{-1}\boldsymbol{b} - \theta \boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{e}_i \\ \boldsymbol{x}_N: & \boldsymbol{0} & \rightarrow & \theta \boldsymbol{e}_i \end{array} \tag{46}$$

とすると目的関数が減少し, かつ θ は任意の正の値を取りうる。このとき, 解は $\Delta\boldsymbol{x}_B = -\boldsymbol{B}^{-1}\boldsymbol{N}\boldsymbol{e}_i$, $\Delta\boldsymbol{x}_N = \boldsymbol{e}_i$ なる方向に動く。 また, θ が正の任意の値を取っても制約条件 $\boldsymbol{x} \geq 0$ はつねに満たされることから, $\Delta\boldsymbol{x}_B \geq \boldsymbol{0}$ である。また, $\Delta\boldsymbol{x}_B$, $\Delta\boldsymbol{x}_N$ は M に依存しない。 さらに,

$$\boldsymbol{B}\Delta\boldsymbol{x}_B + \boldsymbol{N}\Delta\boldsymbol{x}_N = \boldsymbol{0} \tag{47}$$

である。

さて, $\Delta\boldsymbol{x}_B$, $\Delta\boldsymbol{x}_N$ を並べ換えてもとの変数 $\boldsymbol{x}$ と人為変数 $\boldsymbol{\xi}$ に対応させると, 解が動く方向のベクトル $\Delta\boldsymbol{x}$ と $\Delta\boldsymbol{\xi}$ が得られる。このとき, (47) より,

$$\boldsymbol{A}_2\begin{bmatrix}\Delta\boldsymbol{x}\\ \Delta\boldsymbol{\xi}\end{bmatrix} = \begin{bmatrix}\boldsymbol{A} & \boldsymbol{I}_m\end{bmatrix}\begin{bmatrix}\Delta\boldsymbol{x}\\ \Delta\boldsymbol{\xi}\end{bmatrix} = \boldsymbol{0} \tag{48}$$

である。

先の議論により, $\Delta\boldsymbol{x} \geq \boldsymbol{0}$, $\Delta\boldsymbol{\xi} \geq \boldsymbol{0}$ であるが, 実は, 目的関数が任意に小さい値を取りうることから, $\Delta\boldsymbol{\xi} = \boldsymbol{0}$ となることがいえる。 これはなぜかというと, 問題 (43) の解を

$$\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{\xi}\end{bmatrix} + \theta\begin{bmatrix}\Delta\boldsymbol{x}\\ \Delta\boldsymbol{\xi}\end{bmatrix} \tag{49}$$

に変更すると目的関数は $\theta(\boldsymbol{c}^T\Delta\boldsymbol{x} + M\boldsymbol{1}^T\Delta\boldsymbol{\xi})$ だけ変動し, この変動が負であることから $\boldsymbol{c}^T\Delta\boldsymbol{x} + M\boldsymbol{1}^T\Delta\boldsymbol{\xi} < 0$ でなければならないのだが, M が十分大きく $\Delta\boldsymbol{\xi} \neq \boldsymbol{0}$ のときにはこれは負の値を取りえないからである。ゆえに $\Delta\boldsymbol{\xi} = 0$ である。 ここから, $\boldsymbol{c}^T\Delta x < 0$ が導かれる。

ここで, 問題 (5) が有限の最適解を持つと仮定して矛盾を導こう。$\boldsymbol{x}^*$ が問題 (5) の有限の最適解であるとする。さて, $\Delta\boldsymbol{x} \geq \boldsymbol{0}$ であり, 一方 (48) より, $\boldsymbol{A}\Delta\boldsymbol{x} = \boldsymbol{0}$ となるから, $\Delta\boldsymbol{x}$ の方向に解を動かしても問題 (5) の制約条件が脅かされることはない。 よって, 正のパラメータ θ に対し, $\boldsymbol{x}^* + \theta\Delta\boldsymbol{x}$ も解である。一方, $\boldsymbol{c}^T\Delta\boldsymbol{x} < 0$ より, $\boldsymbol{x}^* + \theta\Delta\boldsymbol{x}$ は最適解 $\boldsymbol{x}^*$ より目的関数値を小さくする解である。 これは $\boldsymbol{x}^*$ が最適であることに矛盾する。以上により, 問題 (5) が有限の最適解を持たないことが示された。

2.4 内点法

本節では内点法について述べる。

内点法は大きく分けると主内点法と主-双対法の 2 種類に分類されるのだが, 本節では主-双対法と呼ばれるアルゴリズムのうち比較的簡単なもの 2 種類を文献 [6, 7] にしたがって紹介する。

まず, いくつか記号の定義をしておく。ベクトルのノルムは, $\|\boldsymbol{x}\| = \sqrt{\sum_{i=1}^n x_i^2}$ によって定義される。ベクトル同士の Hadamard 積を, 以下によって定義する。

$$\boldsymbol{x}\boldsymbol{s} = [x_1s_1, \ldots, x_ns_n]^T \tag{50}$$

また, 記法の簡単のため, ベクトル $\boldsymbol{x}$ に対し,

$$1/\boldsymbol{x} = [1/x_1, \ldots, 1/x_n]^T \tag{51}$$

と定義する。

2.4.1 内点法のアルゴリズム

基準形で記述された最適化問題

$$\text{(P)} \quad \min\left\{z = \boldsymbol{c}^T\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}\right\} \tag{52}$$

を考える。(52) を主問題と呼ぶ。(52) に対し, 双対問題と呼ばれる問題が,

$$\text{(D)} \quad \max\left\{w = \boldsymbol{b}^T\boldsymbol{y} : \boldsymbol{A}^T\boldsymbol{y} \leq \boldsymbol{c}, \boldsymbol{y} \geq \boldsymbol{0}\right\} \tag{53}$$

で定義される。ここに, max は maximize の略であり, 目的関数を最大化する問題をあらわす。なお, $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^m$ である。

主-双対法は, 簡単に言うと, (P) と (D) をまとめて解く方法である。以下では, まず繰り返し計算のための初期値を定め, 続いて (P) と (D) を同時に解くために新たな問題 (SP) を定義し, さいごに繰り返し計算のアルゴリズムを述べる。

まず, $\boldsymbol{x}$ と $\boldsymbol{y}$ の適当な初期値 $\boldsymbol{x}^0 > \boldsymbol{0}$, $\boldsymbol{y}^0 > \boldsymbol{0}$ を決める。これは正であればどのように取っても構わない。次に, $\boldsymbol{p}^0 = 1/\boldsymbol{x}^0$, $\boldsymbol{t}^0 = 1/\boldsymbol{y}^0$ とおく。パラメータ $\bar{\boldsymbol{b}}$, $\bar{\boldsymbol{c}}$, β を

$$\begin{aligned}\bar{\boldsymbol{b}} &= \boldsymbol{t}^0 + \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^0\\ \bar{\boldsymbol{c}} &= \boldsymbol{p}^0 - \boldsymbol{c} + \boldsymbol{A}^T\boldsymbol{y}^0\\ \beta &= 1 - \boldsymbol{b}^T\boldsymbol{y}^0 + \boldsymbol{c}^T\boldsymbol{x}^0\end{aligned} \tag{54}$$

と定義する。続いて, 行列 $\boldsymbol{M}$ およびベクトル $\boldsymbol{q}$ を

$$\boldsymbol{M} = \begin{bmatrix}\boldsymbol{0}_{m\times m} & \boldsymbol{A} & -\boldsymbol{b} & \bar{\boldsymbol{b}}\\ -\boldsymbol{A}^T & \boldsymbol{0}_{n\times n} & \boldsymbol{c} & \bar{\boldsymbol{c}}\\ \boldsymbol{b}^T & -\boldsymbol{c}^T & 0 & \beta\\ -\bar{\boldsymbol{b}}^T & -\bar{\boldsymbol{c}}^T & -\beta & 0\end{bmatrix}, \quad \boldsymbol{q} = \begin{bmatrix}\boldsymbol{0}_m\\ \boldsymbol{0}_n\\ 0\\ n+m+2\end{bmatrix} \tag{55}$$

と定義し, 次の問題を考える。

$$\text{(SP)} \quad \min\left\{\boldsymbol{q}^T\boldsymbol{\xi} : \boldsymbol{M}\boldsymbol{\xi} + \boldsymbol{q} \geq \boldsymbol{0}, \boldsymbol{\xi} \geq \boldsymbol{0}\right\} \tag{56}$$

ここに, $\boldsymbol{\xi} = [\boldsymbol{y}^T, \boldsymbol{x}^T, \kappa, \theta]^T$ は $n+m+2$ 次元のベクトルである。問題 (SP) のように, 制約条件に対応する行列 $\boldsymbol{M}$ が歪対称行列[v] であり, 非負のベクトル $\boldsymbol{q}$ が制約条件と目的関数の双方にあらわれる問題を, 歪対称問題と呼ぶ。

[v] $\boldsymbol{M}$ が正方行列で, となるとき, $\boldsymbol{M}$ を歪対称行列という。

問題 (SP) の初期値を

$$\boldsymbol{\xi}^0 = \left[(\boldsymbol{y}^0)^T, (\boldsymbol{x}^0)^T 1, 1\right]^T \tag{57}$$

とする。

与えられた $\boldsymbol{\xi}$ に対し, (SP) の制約条件であらわれた式 $\boldsymbol{M\xi}+\boldsymbol{q}$ が取る値を $\boldsymbol{s}(\boldsymbol{\xi})$ とおく。すなわち,

$$\boldsymbol{s}(\boldsymbol{\xi}) = \boldsymbol{M\xi}+\boldsymbol{q} \tag{58}$$

と定義する。以下では, 記法の簡単のために, しばしば, $\boldsymbol{s}(\boldsymbol{\xi})$ を $\boldsymbol{s}$ と略記する。また, $n+m+2$ 次のベクトル $\boldsymbol{\xi}$, $\boldsymbol{s}$ に対し, 行列 $\boldsymbol{\Xi}$ と $\boldsymbol{S}$ を

$$\boldsymbol{\Xi} = \mathrm{diag}[\xi_1, \ldots, \xi_{m+n+2}], \quad \boldsymbol{S} = \mathrm{diag}[s_1, \ldots, s_{m+n+2}] \tag{59}$$

と定義する。ただし, $\mathrm{diag}[x, \ldots, x_k]$ は対角要素が $x_1, \ldots, x_k$ でそれ以外の要素が零の正方行列をあらわす。

以上の準備のもとで, 2 種類の内点法のアルゴリズムについて述べる。

まず, ショートステップパス追跡法と呼ばれるアルゴリズムについて述べる。これは, 次のようなアルゴリズムである。

内点法 (ショートステップパス追跡法)

(初期化) 精度パラメータ ε を適当に決める。また, $\boldsymbol{\xi} := \boldsymbol{\xi}^0$, $\boldsymbol{s} := \boldsymbol{s}(\boldsymbol{\xi})$ とする。さらに,

$$\sigma = 1 - 0.4/\sqrt{n+m+2} \tag{60}$$

とおく。

(ループ) 以下のプログラムを実行する。

$$\begin{array}{l} \textbf{while } \boldsymbol{\xi}^T \boldsymbol{s} \geq \varepsilon \textbf{ do} \\ \qquad \mu := \boldsymbol{s}^T \boldsymbol{\xi}/(n+m+2) \\ \qquad \boldsymbol{\xi} := \boldsymbol{\xi} + \boldsymbol{\Delta\xi} \\ \qquad \boldsymbol{s} := \boldsymbol{s}(\boldsymbol{\xi}) \\ \textbf{end} \end{array}$$

ここに, $\boldsymbol{\Delta\xi}$ は線形方程式

$$(\boldsymbol{S} + \boldsymbol{\Xi M})\boldsymbol{\Delta\xi} = -\boldsymbol{s\xi} + \sigma\mu\boldsymbol{1} \tag{61}$$

の解である。

問題 (SP) を初期値 $\boldsymbol{\xi}^0$ のもとで解き, $\boldsymbol{q}^T\boldsymbol{\xi}^* = 0$ なる最適解 $\boldsymbol{\xi}^* = [\boldsymbol{y}^*, \boldsymbol{x}^*, \kappa^*, \theta^*]^T$ が得られたとしよう。このとき $\kappa^* \neq 0$ であれば, もとの問題の最適解は

$$\bar{\boldsymbol{x}} = \frac{\boldsymbol{x}^*}{\kappa^*}, \quad \bar{\boldsymbol{y}} = \frac{\boldsymbol{y}^*}{\kappa^*} \tag{62}$$

となる。$\kappa^* = 0$ のときには, 最適解は存在しない。すなわち, 制約条件を満たす領域が存在しないか, 目的関数の値は有限にならない。これについては, 第 2.5.1 節においてもう少し詳しく説明する。

ショートステップパス追跡法には, 理論的な特性は良いが, 実際の問題に適用すると実行が極めて遅いという特徴がある。

理論的な特性はショートステップパス追跡法に劣るが実用上はより高速なアルゴリズムに, ロングステップパス追跡法がある。

ロングステップパス追跡法について述べる前に, 記号の定義をしておく。(61) を解いて得られる $\boldsymbol{\xi}$ とパラメータ α (ただし $0 \leq \alpha \leq 1$) に対し,

$$\boldsymbol{\xi}_\alpha = \boldsymbol{\xi} + \alpha\boldsymbol{\Delta\xi}, \quad \boldsymbol{s}_\alpha = \boldsymbol{s}(\boldsymbol{\xi}_\alpha) \tag{63}$$

と定義する。

ロングステップ追跡法のショートステップパス追跡法との相異は, (61) を解いて得られる $\Delta\boldsymbol{\xi}$ の方向に解 $\boldsymbol{\xi}$ を可能な限り大きく動かすことである。 具体的には, $\mu = \boldsymbol{s}^T\boldsymbol{\xi}/(n+m+2)$ とあらかじめ定められた定数 γ に対し, 直線探索と呼ばれる方法を用いて

$$\boldsymbol{\xi}_\alpha \boldsymbol{s}_\alpha \geq \gamma\mu\boldsymbol{1} \tag{64}$$

を満たす範囲で最大の α を求め (ただし $0 \leq \alpha \leq 1$), $\boldsymbol{\xi} := \boldsymbol{\xi} + \alpha\boldsymbol{\Delta\xi}$ により解を更新する。

内点法 (ロングステップパス追跡法)

(初期化) 精度パラメータ ε を適当に決める。定数 γ, $\sigma_{\min}$, $\sigma_{\max}$ を $0 < \gamma < 1$, $0 < \sigma_{\min} < \sigma_{\max} < 1$ となるように定める。

(ループ) 以下のプログラムを実行する。

$$\begin{array}{l} \textbf{while } \boldsymbol{\xi}^T \boldsymbol{s} \geq \varepsilon \textbf{ do} \\ \qquad \mu := \boldsymbol{s}^T \boldsymbol{\xi}/(n+m+2) \\ \qquad \sigma \in [\sigma_{\min}, \sigma_{\max}] \text{ を定める} \\ \qquad \text{線形方程式 (61) を解いて } \Delta\boldsymbol{\xi} \text{ を求める} \\ \qquad \text{(64) を満たす最大の } \alpha (0 \leq \alpha \leq 1) \text{ を求める} \\ \qquad \boldsymbol{\xi} := \boldsymbol{\xi} + \alpha\boldsymbol{\Delta\xi} \\ \qquad \boldsymbol{s} := \boldsymbol{s}(\boldsymbol{\xi}) \\ \textbf{end} \end{array}$$

直線探索により α を決めるときには, たとえば 2 分探索を用いればよい。

ロングステップパス追跡法では, パラメータ σ の値はステップごとに変えてよい。σ の決め方としては, たとえばステップごとに乱数を用いてランダムに σ を決める, といった方法が考えられる。

実は, σ を 0 に近い値にすれば $\boldsymbol{\Delta\xi}$ は目的関数をより減少させる方向 (アフィンスケーリング方向) を向き, σ を 1 に近い値にすれば $\boldsymbol{\Delta\xi}$ は制約条件を満たす領域からはずれないような方向 (中心化方向) を向く (第 2.5.2.3 節参照)。

内点法を用いた線形計画問題の求解パッケージでは, 以上で述べた 2 種類のアルゴリズムではなく, Mehrotra の予測子・修正子法というアルゴリズムが採用されていることが多い。Mehrotra の予測子・修正子法は高速ではあるが若干見通しが悪いので, ここでは述べない。興味のある読者は文献 [3, 7] を参照してほしい。

2.5 内点法の詳細

以下では, まず第 2.5.1 節において, 本節で利用された行列 $\boldsymbol{M}$ などの意味を説明する。続いて, 第 2.5.2 節では, 内点法の一般的なアルゴリズムについて説明する。

2.5.1 線形計画問題 (SP) の意味

本節では, 第 2.4 節の行列 $\boldsymbol{M}$ などの意味についていくつかの証明を省いた上で述べる。この節は実験そのものには関係ないので, 興味のない読者は読み飛ばしてよい。

主問題 (P) とその双対問題 (D) が与えられているものとする。主問題の制約条件 ($\boldsymbol{Ax} \geq \boldsymbol{b}, \boldsymbol{x} \geq \boldsymbol{0}$) を満たす点の集合を $\mathcal{P}$ と書き, 双対問題の制約条件 ($\boldsymbol{A}^T\boldsymbol{y} \leq \boldsymbol{c}, \boldsymbol{y} \geq \boldsymbol{0}$) を満たす点の集合を $\mathcal{D}$ と書く。$\mathcal{P}$ や $\mathcal{D}$ が空集合となることもある。このような場合には, 主問題や双対問題には解が存在しない。

さて, 適当に取られた $\boldsymbol{x} \in \mathcal{P}$ と $\boldsymbol{y} \in \mathcal{D}$ に対し, $\boldsymbol{b}^T\boldsymbol{y} \leq (\boldsymbol{x}^T\boldsymbol{A}^T)\boldsymbol{y} = \boldsymbol{x}^T(\boldsymbol{A}^T\boldsymbol{y}) \leq \boldsymbol{x}^T\boldsymbol{c}$ より, $\boldsymbol{b}^T\boldsymbol{y} \leq \boldsymbol{c}^T\boldsymbol{x}$ という不等式が成り立つことがわかる。すなわち, 双対問題の目的関数値は必ず主問題の目的関数値以下となる。このことから, ある $\boldsymbol{x}^* \in \mathcal{P}$ と $\boldsymbol{y}^* \in \mathcal{D}$ が等式 $\boldsymbol{b}^T\boldsymbol{y}^* = \boldsymbol{c}^T\boldsymbol{x}^*$ を満たせば, $\boldsymbol{x}^*$ は主問題の最適解であり, $\boldsymbol{y}^*$ は双対問題の最適解であるということがわかる。実は, この逆も成り立つ。すなわち, 主問題と双対問題がそれぞれ有限の最適解 $\boldsymbol{x}^*$ と $\boldsymbol{y}^*$ を持つとき, 等式 $\boldsymbol{b}^T\boldsymbol{y}^* = \boldsymbol{c}^T\boldsymbol{x}^*$ が成り立つということが言える。後半の結果を双対定理という。双対定理の証明はやや繁雑なのでここでは述べない。

さて, 最適解のための等式 $\boldsymbol{b}^T\boldsymbol{y} = \boldsymbol{c}^T\boldsymbol{x}$ に不等号を追加した不等式

$$\boldsymbol{b}^T\boldsymbol{y} \geq \boldsymbol{c}^T\boldsymbol{x} \tag{65}$$

を考えよう。不等式 $\boldsymbol{b}^T\boldsymbol{y} \leq \boldsymbol{c}^T\boldsymbol{x}$ はつねに成り立つのだから, これは実質的にもとの等式と変わらない。(65) の右辺を左辺に移項し, 主問題の制約条件, 双対問題の制約条件の両辺に -1 を乗じたものと合わせて整理すると,

$$\begin{aligned} \boldsymbol{Ax} - \boldsymbol{b} &\geq \boldsymbol{0} \\ -\boldsymbol{A}^T\boldsymbol{y} + \boldsymbol{c} &\geq \boldsymbol{0} \\ \boldsymbol{b}^T\boldsymbol{y} - \boldsymbol{c}^T\boldsymbol{x} &\geq 0 \end{aligned} \tag{66}$$

という不等式が得られる。これを行列の形でまとめると,

$$\begin{bmatrix} \boldsymbol{0} & \boldsymbol{A} & -\boldsymbol{b} \\ -\boldsymbol{A}^T & \boldsymbol{0} & \boldsymbol{c} \\ \boldsymbol{b}^T & -\boldsymbol{c}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{x} \\ 1 \end{bmatrix} \geq \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ 0 \end{bmatrix}, \boldsymbol{x} \geq \boldsymbol{0}, \boldsymbol{y} \geq \boldsymbol{0} \tag{67}$$

という形になる。この段階ではまだ (67) が解を持つかどうかは明らかでないことに注意しよう。

ここで, (67) の左辺のベクトルの最下段にあらわれた数 1 を変数 κ で置き変えた不等式

$$\begin{bmatrix} \boldsymbol{0} & \boldsymbol{A} & -\boldsymbol{b} \\ -\boldsymbol{A}^T & \boldsymbol{0} & \boldsymbol{c} \\ \boldsymbol{b}^T & -\boldsymbol{c}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{x} \\ \kappa \end{bmatrix} \geq \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{x} \\ \kappa \end{bmatrix} \geq \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ 0 \end{bmatrix} \tag{68}$$

を考える。(68) を満たす $(\boldsymbol{y}^T, \boldsymbol{x}^T, \kappa)$ が求められていて, かつ $\kappa > 0$ であれば, $(\boldsymbol{y}^T/\kappa, \boldsymbol{x}^T/\kappa, 1)$ は (67) を満たす。すなわち, 新しい変数 κ を導入することはもとの問題 (67) の解を求めることの障害とはならない。では, 新しい変数 κ を導入することには何か利点はあるのだろうか。まず (68) が自明な解 $\boldsymbol{x} = \boldsymbol{0}$, $\boldsymbol{y} = \boldsymbol{0}$, $\kappa = 0$ を持つことに注意する。ところで, 内点法を用いて問題 (68) の解を求めると, 問題 (67) に解があるときには κ は零以外の値に収束し, 問題 (67) に解がないときには κ は零に収束することが示せる (証明は略す)。すなわち, κ を (67) の解の存在判定に使うことができる。

さて, (68) の左端の行列は先に内点法のアルゴリズムで使った (55) と似てはいるが, 同一ではない。(55) の行列 $\boldsymbol{M}$ の右端および下段の要素は (68) にはないものである。これはどのような経緯で出て来たのだろうか?

実は, 線形計画問題を内点法で解くためには初期値として制約領域の内点 (制約条件から等号を取り除いた不等式を満たす点) が必要になるのだが, 内点を 1 個定めることはそれほど容易でない。そのため, 代替策として, 新たに変数を 1 個追加して, (68) から内点が既知の拡大された問題に変換する。(55) は (68) にそのような手法を適用した結果なのである。以下では, この方法について見てゆこう。

初期値 $\boldsymbol{x}^0 > \mathbf{0}$, $\boldsymbol{y}^0 > \mathbf{0}$ を適当に決め (各成分が 0 でなければ何でもよい), $\boldsymbol{p}^0 = 1/\boldsymbol{x}^0$, $\boldsymbol{t}^0 = 1/\boldsymbol{y}^0$ とおく。パラメータ $\bar{\boldsymbol{b}}$, $\bar{\boldsymbol{c}}$, β を (54) によって定義する。また, 行列 $\boldsymbol{M}$ およびベクトル $\boldsymbol{q}$ を (55) によって決め, (56) によって定められた問題 (SP) を考える。すると, 初期値 (57) が問題 (56) の内点となるのである。これを見ておこう。$\boldsymbol{\xi}^0 > \mathbf{0}$ は定義より明らかだから, $\boldsymbol{M\xi}^0 + \boldsymbol{q} > \mathbf{0}$ となることを見ればよい。これは, (54) を使えば,

$$\begin{aligned}\boldsymbol{M\xi}^0 + \boldsymbol{q} &= \begin{bmatrix} \boldsymbol{A}x^0 - \boldsymbol{b} + \bar{\boldsymbol{b}} \\ -\boldsymbol{A}^T\boldsymbol{y}^0 + \boldsymbol{c} + \bar{\boldsymbol{c}} \\ \boldsymbol{b}^T\boldsymbol{y}^0 - \boldsymbol{c}^T\boldsymbol{x}^0 + \beta \\ -\bar{\boldsymbol{b}}^T\boldsymbol{y}^0 - \bar{\boldsymbol{c}}^T\boldsymbol{x}^0 - \beta + (n+m+2) \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{t}^0 \\ \boldsymbol{p}^0 \\ 1 \\ 1 \end{bmatrix}\end{aligned} \tag{69}$$

となることからわかる。

2.5.2 内点法のアルゴリズムの一般形

本節では, 内点法のアルゴリズムの一般形について簡単に述べる。

第 2.5.1 節の結果から, 主問題と双対問題を解くことは, 問題

$$\text{(SP)} \quad \min\left\{\boldsymbol{q}^T\boldsymbol{\xi} : \boldsymbol{M\xi} + \boldsymbol{q} \geq \mathbf{0}, \boldsymbol{\xi} \geq \mathbf{0}\right\}$$

を解くことに帰着されることがわかっている。ここに, $\boldsymbol{M}$ は歪対称行列であり, $\boldsymbol{q}$ は各成分が非負のベクトルである。そこで, 本節では (SP) を出発点とする。

2.5.2.1 強相補性条件

行列 $\boldsymbol{M}$ が歪対称であることから, 目的関数値は $\boldsymbol{q}^T\boldsymbol{\xi} = \boldsymbol{s}^T\boldsymbol{\xi}$ を満たす[vi]。

さて, 問題 (SP) は自明な最適解 $\boldsymbol{\xi} = \mathbf{0}$ を持つ。しかし, 第 2.5.1 で説明したように, 我々は $\boldsymbol{\xi}$ のある成分が収束する値が 0 であるか否かに応じて主問題 (P) と双対問題 (D) の解の存在性の判定をするので, 自明な解は我々の望むものではない。ところで, $\boldsymbol{\xi}$ が最適であり ($\boldsymbol{s}^T\boldsymbol{x} = 0$), かつ $\boldsymbol{\xi}$ の要素の中の零でないものがあったとしよう。$\boldsymbol{\xi} \geq \mathbf{0}$, $\boldsymbol{s} \geq \mathbf{0}$ であることを思いだすと, $\boldsymbol{\xi}$ と $\boldsymbol{s}$ の成分は,

- $\xi_i = 0$ なら $s_i > 0$,
- $\xi_i > 0$ なら $s_i = 0$

という条件を満たすことがわかる。この条件を強相補性条件という。強相補性条件は, より簡単に,

$$\boldsymbol{s} + \boldsymbol{\xi} > \mathbf{0} \tag{70}$$

とも書ける。以上の議論により, 我々は問題 (SP) の強相補性条件を満たす最適解を (もしあれば) 見付けなければならないことがわかる。

2.5.2.2 中心パス

ここで, 中心パスという概念を導入しておく。$\mathbf{1}$ をすべての要素が 1 のベクトルとし, μ を正の定数として,

$$\boldsymbol{s\xi} = \mu\mathbf{1} \tag{71}$$

なる等式を考える。ここでは証明を省くが, (SP) の制約条件を満たす領域が内点を持つ (開集合を含む) とき, (71) と (SP) の制約条件を連立させた非線形連立方程式

$$\boldsymbol{s} = \boldsymbol{M\xi} + \boldsymbol{q}, \quad \boldsymbol{s\xi} = \mu\mathbf{1} \tag{72}$$

を $\boldsymbol{\xi}$ について解くと, 与えられた μ に対して制約領域の内点にある $\boldsymbol{\xi}(\mu)$ が一意的に定まることが言える。さらに, 次の事実が成り立つ。

- $\boldsymbol{\xi}(\mu)$ は μ の連続関数である。
- $\mu \to 0$ とすると, $\boldsymbol{\xi}(\mu)$ は強相補性条件を満たす点に収束する。

これらの事実から, 軌道 $\boldsymbol{\xi}(\mu)$(これを中心パスと呼ぶ) を追跡することにより強相補性条件を満たす解が求められることがわかる。中心パスは, 制約条件を満たす領域の内部を通って最適解に到達する, 幾何学的に性質の良い軌道である。

2.5.2.3 アフィンスケーリング方向と中心化方向

以上の議論を踏まえた上で, 2 種類の異なる探索ベクトルの決め方について述べよう。

まず最初に, (SP) を変数 $\boldsymbol{x}$ と $\boldsymbol{s}$ に関する非線形関数

$$\boldsymbol{F}(\boldsymbol{\xi}, \boldsymbol{s}) = \begin{bmatrix} \boldsymbol{M\xi} - \boldsymbol{s} + \boldsymbol{q} \\ \boldsymbol{s\xi} \end{bmatrix} \tag{73}$$

の零点を求める問題とみなす。点 $\boldsymbol{\xi}$ と $\boldsymbol{s} = \boldsymbol{M\xi} + \boldsymbol{q}$ が与えられているとき, $\boldsymbol{\xi}' = \boldsymbol{\xi} + \boldsymbol{\Delta\xi}_a$ および $\boldsymbol{s}' = \boldsymbol{s} + \boldsymbol{\Delta s}_a$ として $\boldsymbol{F}(\boldsymbol{\xi}', \boldsymbol{s}') = \mathbf{0}$ を $\boldsymbol{\Delta\xi}_a$ と $\boldsymbol{\Delta s}_a$ について Tailor 展開によって近似的に解くと, $\boldsymbol{\Delta\xi}_a$ と $\boldsymbol{\Delta s}_a$ は方程式

$$\begin{bmatrix} \boldsymbol{M} & -\boldsymbol{I} \\ \boldsymbol{S} & \boldsymbol{\Xi} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta\xi}_a \\ \boldsymbol{\Delta s}_a \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ -\boldsymbol{s\xi} \end{bmatrix} \tag{74}$$

[vi] 先と同様に, $\boldsymbol{s} = \boldsymbol{M\xi} + \boldsymbol{q}$ とする

の解となる。なお, 適当な正の定数αを取り, $\boldsymbol{\xi}' = \boldsymbol{\xi} + \alpha\boldsymbol{\Delta\xi}_a$ および $\boldsymbol{s}' = \boldsymbol{s} + \alpha\boldsymbol{\Delta s}_a$ とすると,

$$\boldsymbol{M}(\boldsymbol{\xi}') + \boldsymbol{q} = \boldsymbol{s} + \alpha\boldsymbol{M}\boldsymbol{\Delta\xi}_a = \boldsymbol{s}'$$

となる。よって, 任意のαに対し, $\boldsymbol{s}'$と$\boldsymbol{\xi}'$は等式$\boldsymbol{s}' = \boldsymbol{M}\boldsymbol{\xi}' + \boldsymbol{q}$を満たす。(74) を解くことによって得られる探索ベクトル $(\boldsymbol{\Delta\xi}_a^T, \boldsymbol{\Delta s}_a^T)^T$ をアフィンスケーリング方向と呼ぶ。

アフィンスケーリング方向は目的関数を減少させるには都合が良いが, 制約条件を見たす領域の境界に解を誘導する可能性があるという点では不都合である。ところで, 解を内点に誘導するには, 中心パス上の点を利用するのが良い。そこで, パラメータμを適当に決め, それ対応する中心パス上の点を求める問題を考える。パラメータμの選び方にはいろいろなものが考えられるが, たとえば目的関数値を現在の値の $1/\dim\boldsymbol{\xi}$ にすることを目指す, すなわち $\mu = (1/\dim\boldsymbol{\xi})\boldsymbol{s}^T\boldsymbol{\xi}$ とするという方法が考えられる。この問題も, 先と同様に, 関数

$$\boldsymbol{G}(\boldsymbol{\xi}, \boldsymbol{s}) = \begin{bmatrix} \boldsymbol{M}\boldsymbol{\xi} - \boldsymbol{s} + \boldsymbol{q} \\ \boldsymbol{s}\boldsymbol{\xi} - \mu\boldsymbol{1} \end{bmatrix} \tag{75}$$

の零点を求める問題に帰着される。$\boldsymbol{\xi}' = \boldsymbol{\xi} + \boldsymbol{\Delta\xi}_c$, $\boldsymbol{s}' = \boldsymbol{s} + \boldsymbol{\Delta s}_c$ とし, (75) を Tailor 展開によって近似的に解くと, $\boldsymbol{\Delta\xi}_c$ と $\boldsymbol{\Delta s}_c$ は方程式

$$\begin{bmatrix} \boldsymbol{M} & -\boldsymbol{I} \\ \boldsymbol{S} & \boldsymbol{\Xi} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta\xi}_c \\ \boldsymbol{\Delta s}_c \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{s}\boldsymbol{\xi} + \mu\boldsymbol{1} \end{bmatrix} \tag{76}$$

の解となる。(76) を解くことによって得られる探索ベクトル $(\boldsymbol{\Delta\xi}_c^T, \boldsymbol{\Delta s}_c^T)^T$ を中心化方向と呼ぶ。

これらの探索ベクトルを用いた解法は, Newton 法による非線形方程式の求解の一種となる。

2.5.2.4 アルゴリズムの一般形 アフィンスケーリング方向は目的関数を減少させる方向, 中心化方向は中心パスに向かう方向であった。Newton 法型の内点法のアルゴリズムでは, アフィンスケーリング方向と中心化方向の線形結合を使って解を更新してゆく。

σを 0 以上 1 以下の定数とし, 探索ベクトルを

$$\begin{bmatrix} \boldsymbol{\Delta\xi} \\ \boldsymbol{\Delta s} \end{bmatrix} = (1-\sigma) \begin{bmatrix} \boldsymbol{\Delta\xi}_a \\ \boldsymbol{\Delta s}_a \end{bmatrix} + \sigma \begin{bmatrix} \boldsymbol{\Delta\xi}_c \\ \boldsymbol{\Delta s}_c \end{bmatrix} \tag{77}$$

とすると, 探索ベクトル $(\boldsymbol{\Delta\xi}^T, \boldsymbol{\Delta s}^T)^T$ は

$$\begin{bmatrix} \boldsymbol{M} & -\boldsymbol{I} \\ \boldsymbol{S} & \boldsymbol{\Xi} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta\xi} \\ \boldsymbol{\Delta s} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{s}\boldsymbol{\xi} + \sigma\mu\boldsymbol{1} \end{bmatrix} \tag{78}$$

の解となる。

なお, (78) は $\boldsymbol{\Delta\xi}$ と $\boldsymbol{\Delta s}$ に関する線形方程式であるが, (78) の上半分は $\boldsymbol{\Delta s}$ について解けて $\boldsymbol{\Delta s} = \boldsymbol{M}\boldsymbol{\Delta\xi}$ となるので, 実際にはこれを (78) の下半分に代入した

$$\Big(\boldsymbol{S} + \boldsymbol{\Xi}\boldsymbol{M}\Big)\boldsymbol{\Delta\xi} = -\boldsymbol{s}\boldsymbol{\xi} + \sigma\mu\boldsymbol{1} \tag{79}$$

を $\boldsymbol{\Delta\xi}$ について解けばよい。

(78) の解 $(\boldsymbol{\Delta\xi}^T, \boldsymbol{\Delta s}^T)^T$ を探索ベクトルとして利用する内点法のアルゴリズムの一般形は, 以下に述べるようなものである。

内点法 (一般形)

(初期化) 精度パラメータεを適当に決める。また, $\boldsymbol{\xi} := \boldsymbol{\xi}^0$, $\boldsymbol{s} := \boldsymbol{s}(\boldsymbol{\xi})$ とする。

(ループ) 以下のプログラムを実行する。

while $\boldsymbol{\xi}^T\boldsymbol{s} \geq \varepsilon$ **do**
　σを決める
　$\mu := \boldsymbol{s}^T\boldsymbol{\xi}/(n)$ とする
　(79) を解いて $\boldsymbol{\Delta\xi}$, $\boldsymbol{\Delta s}$ を決める
　αを決める
　$\boldsymbol{\xi} := \boldsymbol{\xi} + \alpha\boldsymbol{\Delta\xi}$
　$\boldsymbol{s} := \boldsymbol{s} + \alpha\boldsymbol{\Delta s}$
end

第 2.4.1 節で述べたショートステップパス追跡法は, 上記のアルゴリズムにおいて $\alpha = 1$, $\sigma = 1 - 0.4/\sqrt{\dim\boldsymbol{\xi}}$ と固定した特別な場合である。大規模な問題では, このようにσを決めると, 探索ベクトルが中心化方向とほぼ一致してしまうため, ショートステップパス追跡法は極めて低速になる。

ループ内の各ステップでパラメータαやσの値を切り換えると, より効率の良いアルゴリズムが構成しうることが知られている。興味のある読者はたとえば文献 [3, 6, 7] を参照してほしい。

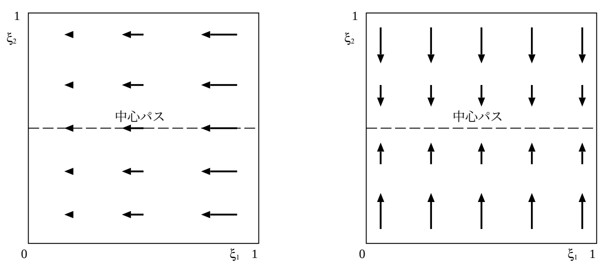
例 3 2 変数の歪対称問題

$$\begin{aligned} &\min \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}^T \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} : \right. \\ &\left. \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\} \end{aligned} \tag{80}$$

を考える。

(80) の制約条件は, $0 \leq \xi_1$, $0 \leq \xi_2 \leq 1$ と書き直される。目的関数は ξ_1 なので, $\xi_1 = 0$ を満たす任意の点で目的関数は最小となる。

この問題において, 各点でアフィンスケーリング方向と中心化方向がどのようになるかを見よう。この問題における中心パスは $\xi_2 = 1/2$ という直線であり, 強相補解は $(\xi_1, \xi_2) = (0, 1/2)$ という点である。図 3 に各点におけるアフィンスケーリング方向と中心化方向を示す。この例ではアフィンスケーリング方向は中心パスと平行になる。それゆえ, アフィンスケーリング方向のみを用いて解を更新すると強相補解に到達できない。また, $\mu = (1/2)\boldsymbol{s}^T\boldsymbol{\xi}$ の場合の中心化方向は中心パスと垂直となり, これを用いて解を更新すると目的関数は減少しない。



(a) アフィンスケーリング方向　(b) 中心化方向

図 3: 各点における探索ベクトルの方向

3　実験

3.1　実験課題

シンプレックス法 (ビッグ M 法), 内点法 (ショートステップパス追跡法) および内点法 (ロングステップパス追跡法) を用いて与えられた線形計画問題を解け。また, 解くべき問題の大きさと計算時間の関係を調べよ。

3.2　実験方法

ディレクトリ`~hanba/info-eng/`に, ビッグ M 法, ショートステップパス追跡法 およびロングステップパス追跡法 の完成したプログラムが置かれている。ファイル名はそれぞれ `bigM.sce`, `spf.sce`, `lpf.sce` である。これらは n 次元立方体 $1 \leq x_i \leq 2$ $(1 \leq i \leq n)$ の内部 (境界を含む) で目的関数 $\mathbf{1}^T\boldsymbol{x}$ を解くためのプログラムである。 最適解は $x_1 = 1, \ldots, x_n = 1$ である。いずれも, プログラムの 2 行目の `n=1;` という部分で次元が指定されている (最初は値が 1 に設定されている)。計算が終了すると, 変数 `time` に計算にかかった時間が格納される。この値は計算終了後に Scilab のウィンドウ内で

$$\texttt{time} \tag{81}$$

と入力することで参照できる。問題の解は `sol` という変数に格納される。計算が正常に終了すれば, これらは全要素がすべて 1 のベクトルになる (若干の数値誤差を含む可能性はあるが)。 値を確認するには, Scilab のウィンドウ内で

$$\texttt{sol} \tag{82}$$

と入力する。なお, シンプレックス法では, 人為変数の値を含んだ解が `asol` という変数に格納される。

これらを利用して, ショートステップパス追跡法 およびロングステップパス追跡法のそれぞれについて, n の値を 1 から 20 まで増加させて計算時間を測る。ただし, n の刻みは 1 ではなく, 2 ～ 3 程度に取って構わない。

[注意事項]

- レポート提出は電子メールのみで受け付ける。提出物は,
  - 作成した LaTeX のソースコード
  - レポート本文の PostScript ファイル

  である。実験用プログラム (`bigM.sce` など) が置かれているディレクトリにレポート作製のためのサンプル `sample.tex` があるので, これを利用せよ。

- 計算に要する時間は, 同一のプログラムを利用していても, 計算機の状態によって微妙に変わる。だから, 他人の実験結果をレポートに写した場合, 比較すればすぐにそれとわかる。複数のレポートに同一の実験結果が記載されていた場合, 該当するレポートはすべて零点とするので注意せよ。

- 各学生が提出したレポートは, 過去に他のグループなどの学生が提出したレポートと電子的に比較される。他の学生のレポートとの類似性が著しいレポートは他人のレポートがまる写ししたものとみなされる。この場合, レポートの得点は 0 点となる。

- サンプルファイルに実験の原理や方法に関する簡単な記述がある。そのままレポートに利用してよい。 また, より詳しい説明をレポートに記載する必要はない。 実験指導書を長々と書き写しても評価の対象とはならないので注意すること。

4 データ解析

- ビッグ M 法, ショートステップパス追跡法 およびロングステップパス追跡法のそれぞれについて, 横軸に n の大きさ, 縦軸に計算時間を取ったグラフを作成し, これらの挙動を比較せよ。 レポートには, 必ず, n の大きさと計算時間が記載された表と対応するグラフの両方を掲載すること。グラフ作成には Scilab や Gnuplot などを利用せよ。たとえば $n = 1, 2, 3, 4$ のそれぞれについて計算時間が 0.0078, 0.016, 0.023, 0.047 となった場合, Scilab でこのグラフを描画するには, たとえば

```
n=[1,2,3,4];
t=[0.0078,0.016,0.023,0.047];
plot(n,t,"n","time");
```

  のようにする (この例では, 横軸に `n`, 縦軸に `time` というラベルを入れている)。

- ビッグ M 法において M を小さくすると正解が得られなくなることを確認せよ。 問題の次元 `n` は 1 とする。このためには, ファイル `bigM.sce` の 3 行目で `M=1e5;` のように M が定義されているので, これを小さい正の値に変更して計算結果を確認すればよい。正解が得られなくなったときには, 計算終了時に `sol` に格納された値が正解 (全要素が 1) と違うものになるはずである。レポートに必す正解が得られなくなったときの M の値と対応する解を記載すること。

参考文献

[1] Mokhtar S. Bazaraa, John J. Jarvis, Hanif D. Sherali: Linear Programming and Network Flows, 2nd ed., John Wiley & Sons (1990)

[2] S. -C. Fang, S. Puthenpura: Linear Optimization and Extensions: Theory and Algorithms, Prentice Hall (1993)

[3] 小島 政和, 土谷 隆, 水野 眞治, 矢部 弘: 「内点法」, 朝倉書店 (2001)

[4] 今野 浩: 「線形計画法」, 日科技連 (1987)

[5] 藤田 宏, 今野 浩, 田邊 國士: 「最適化法」, 岩波書店 (1994)

[6] C. Roos, T. Terlaky and J. -Ph. Vial: Theory and Algorithms for Linear Optimization, Wiley (1997)

[7] S. J. Wright: Primal-Dual Interior-Point Mehtods, SIAM (1997)