

# 基底列挙による線形計画問題の表計算

日大生産工(学部) ○倉田 康平  
日大生産工 篠原 正明

## 1. 始めに

線形計画法は非常に応用のきく手法であり、数多いオペレーションズ・リサーチの手法のなかでも、最も普及しており、実務的に最も効果がある手法の一つである。

本研究では、従来の方法と異なり表計算ソフト(Microsoft Excel)を用いて線形計画法の解法を求める。

## 2. 線形計画法

線形計画法(LP)とは、制約式と呼ばれるいくつかの1次不等式及び1次等式を満たす変数の値の中で、目的関数と呼ばれる1次関数を最大化または最小化する値を求める方法である。

線形計画問題は、以下の様en書き、与えられた定係数 $a_{mn}$ と $b_m, c_n$ ,及び(1)式の様な不等式制約の下、(3)式に示す目的式の最大値、及びそれを実現する変数 $x_n$ を求める問題が典型的な線形計画問題である。

$$\left. \begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &\leq b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &\leq b_m \end{aligned} \right\} (1)$$

$$x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0 \quad (2)$$

$$x_0 = c_1x_1 + c_2x_2 + \cdots + c_nx_n \quad (3)$$

また、(1)式の左辺の係数 $a_{mn}$ を要素とする行列を係数行列(以下 $A$ )、右辺の係数 $b_m$ を要素とする行列を右辺ベクトル(以下 $b$ )、(3)式の係

数 $c_j$ を要素とする行列を利益係数ベクトル(以下 $c$ )と表す。

## 3. 基底列挙法

基底列挙法は総当り法と酷似した解法であり、以下の様に表す事が出来る。

$$\begin{aligned} z = c^T x \rightarrow \max \quad w = b^T y \rightarrow \min \\ Ax \leq b \quad \Leftrightarrow A^T y \geq c \quad (4) \\ x \geq 0 \quad y \geq 0 \end{aligned}$$

まず不等式である制約式にスラック変数を導入して等式にした後、行列とする。以降、この行列を $A$ と置き、行列の大きさを以降 $m$ 行 $n$ 列と表現する。

次に、 $A$ より $m$ ヶ列を選択する。この時選択した列ベクトルを基底と呼び、基底で構成される $m$ 行 $m$ 列の正方行列を基底変数ベクトル( $B$ )とする。同様に選択した列と同列であった $c'$ を1行 $m$ 列の行列として $c_B$ 、他の選択され

なかった $c'$ の行列を $c_R$ とする。

$$\left. \begin{aligned} \text{そして非基底変数を } 0 \text{ とした上で、(5)式の} \\ B^{-1}b \geq 0 \\ c_B B^{-1}N - c_N \geq 0 \end{aligned} \right\} (5)$$

二つの式を満足できれば最適解が得られる。

基底列挙法では $A$ より選択できる全ての組合せ( ${}_n C_m$ 、ただし要素は重複しないものとする)で計算を行い、最適解を探索する。

## 4. アプリケーションについて

今回の研究対象である基底列挙法では目的

## Spread-sheet Calculation of Linear Programming Problem via Basis Enumeration

Kouhei KURATA † and Masaaki SHINOHARA

関数の値を比べ最適解を求める為。計算を行うアプリケーションを **Microsoft Excel** で作成した。詳細を以下の項目にて示す。

#### 4.1 構成

今回作成したアプリケーションは **Microsoft Excel** に内蔵されている関数、及び **Visual Basic for Application** を利用して作成した。アプリケーションは「条件入力シート」と「計算シート」の2つのシートからなり、それぞれに関数やマクロを用いて計算を行っている。

#### 4.2 条件の入力

アプリケーションを起動して「マクロを有効にする」ボタンをクリックする。すると条件入力シートが表示される。「式 入力」ボタンをクリックし図 1 に示したフォームにキーボードを用いて目的関数と制約式を入力する。問題に合わせて各項目に入力する必要があるが、中段の項目、[変数]、[式数]は必ずしも入力する必要はない。



図 1 式入力フォーム

入力した後「設定 入力」ボタンをクリックすると、図 2 の様に対応したセルに内容が入力される。

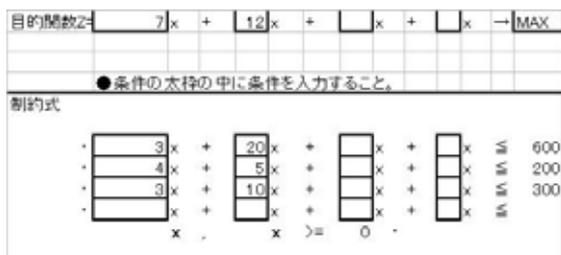


図 2

#### 4.3 計算シートについて

図 3 に計算シートの全体図を示す。

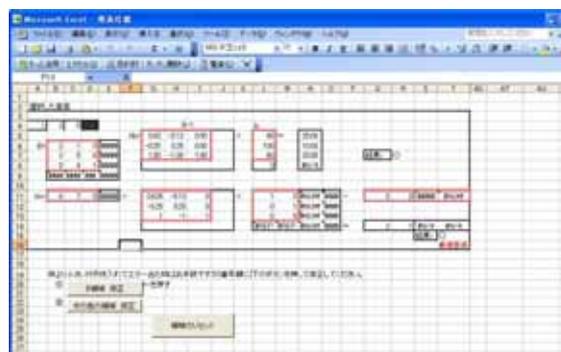


図 3

それぞれ太枠で囲まれた部分が (5)式で用いられている各行列部分を示しており、向かって上半分が(5)の上の式、下半分が下の式の具体的な数値である。

左上に表示されている 1 行 m 列の行列が現在選択している基底を指しており、この値を変更すると、行列 B の内容が連動して変更され、その基底での計算結果を見る事ができる。各行列部分には Excel の機能に内蔵されている「名前」機能を用いて定義しており、後の修正に関与する。

#### 4.4 修正について

このアプリケーションは、 $m$  行  $m$  列の  $A$ 、いわゆる正方行列が入力される事を想定して作成した。その為、それ以外の行列式が入力された場合、計算シートにジャンプしてもエラーが発生し正しい計算結果を見る事ができない。

その場合は、図 3 左下の各修正ボタンを順にクリックする事でマクロが作動し、計算式を修正する。その後に正しい計算結果を導き出す事が出来る。

#### 4.5 計算結果について

計算を正しく終えた場合、図 3 の右側に示す通り、「結果:」の横のセルを見る事で結果が判断できる。計算結果が(5)式の条件を満たす場合は○、そうでない場合は×を表示する。両方の計算結果が条件を満たす場合、右下端のセルに「最適基底」と表示される。

## 5. 実際の計算

以上の基底列挙法を用いて具体的数値を含んだモデルで最適解を求める。

### 5.1 例題

例題として以下の条件を設定した。

$$\left. \begin{array}{l} \text{目的} \quad 6x_1 + 7x_2 \rightarrow \max \\ \text{制約式} \quad 2x_1 + x_2 \leq 60 \\ \quad \quad \quad 2x_1 + 5x_2 \leq 100 \quad (6) \\ \quad \quad \quad 4x_2 \leq 60 \\ \quad \quad \quad x_1, x_2 \geq 0 \end{array} \right\}$$

この条件に基づきアプリケーションを用いて算出する。

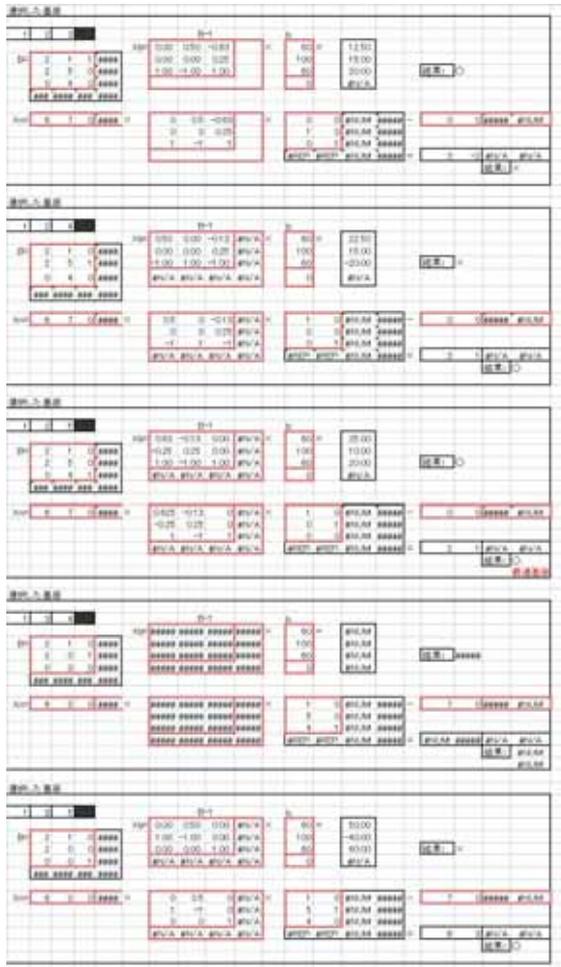


図4-1 例題1結果

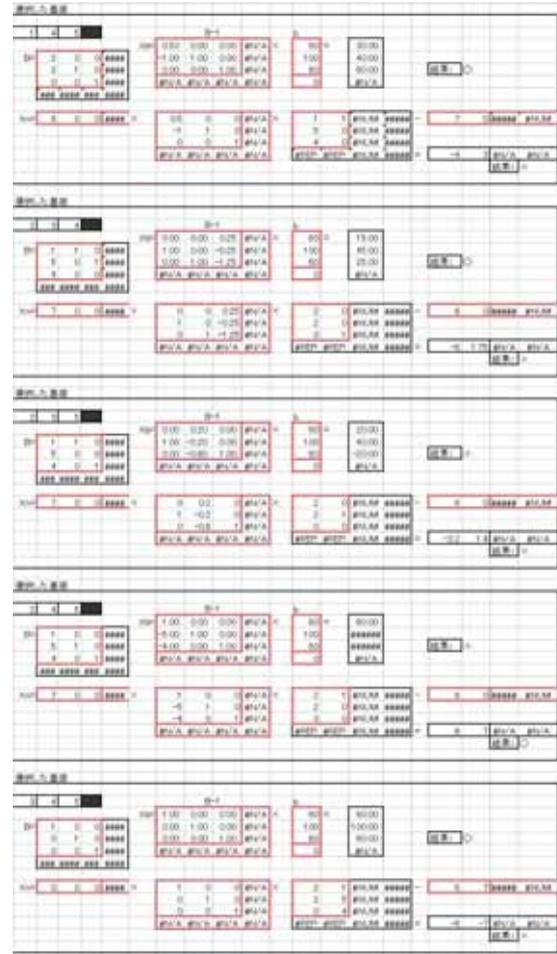


図4-2 例題1結果

### 5.2 結果

計算の結果、図4の示す通り、最適解は基底列(1, 2, 5)の組合せである事が判明した。

## 6. 既存アルゴリズムとの関連

線形計画法の代表的なアルゴリズムには、単体法と Karmarkar アルゴリズムに代表される内点法が存在する。本論文の基底列挙法を凸多面体の辺に沿って目的関数が増加するような経路を選定するアルゴリズムに変更したのが単体法である。従って提案する基底列挙法は教育効果と言う点で有意義であり、さらに強相補性、主退化、双対退化などと基底概念との関連を研究する場合の研究手法としても有意義と考えられる。

## 7. 終わりに

計算結果より、Microsoft Excel を用いて最適解を導き出せる事が判明した。しかし今回計算の為に構築したアプリケーションは要素の上限が低く、制約式や目的関数の係数等が一つでも増えると動作不能となってしまう。今後は多種多様な条件や退化、不能、無限などの特殊な状況にも対応できるよう機能を拡張していきたいと考えている。

## 8. 参考文献

[1]平本 巖/長谷 彰 共著：線形計画法，培風館，昭和 50 年 9 月 30 日

[2]一森 哲男 著：線形計画法-最適化の手法-，共立出版，1994 年 8 月 25 日

[3]森口 繁一 著：線形計画法入門，日科技連，1976 年 1 月 9 日 改装第 4 刷

[4]Windous プログラミング愛好会 著：ここが知りたかった！Excel VBA 500 の技，技術評論社，2005 年 7 月 15 日

[5]牧野都治 牧野京子著：パソコンによる OR，朝倉書店，1985 年 2 月 20 日

### 参考資料

○スラック変数について

スラック変数は制約式を行列に変換する際に不可欠であるが、導入時、使用者の付加を軽減する為や様々な制約式に対応するため、以下のマクロを作成した。

```
Sub 行列式作成()  
Dim 行数 As Integer  
Dim 列数 As Integer  
Dim コピー元セル As Range  
Dim 貼付先セル As Range  
Dim 条件目標 As Range  
Dim スラック貼付基準セル As Range
```

行数 = Range("AW2").Value

列数 = Range("AW3").Value

```
Set コピー元セル = Range("AU17")  
Set 貼付先セル = Range("AU23")  
Set 条件目標 = Range("N3")  
Set スラック貼付基準セル = Cells(23, 47 +  
列数)
```

```
Range("AU23:BB26").ClearContents
```

```
コピー元セル.Select  
Selection.Offset(行数 - 1, 列数 - 1).Select
```

```
Row_lastcell = ActiveCell.Row  
column_lastcell = ActiveCell.Column  
Range(コピー元セル, Cells(Row_lastcell,  
column_lastcell)).Copy
```

```
貼付先セル.PasteSpecial Paste:=xlFormulas  
Application.CutCopyMode = False
```

```
スラック貼付基準セル.Select
```

```
For i = 1 To 行数  
Select Case 条件目標  
Case "MAX"  
Selection.Value = 1  
Case "MIN"  
Selection.Value = -1  
Case Else  
Exit Sub  
End Select
```

```
Selection.Offset(1, 1).Select  
Next
```

```
Range("Q18").Select
```

```
End Sub
```