
2021年 5月

* 「線形計画法」の紹介 * [著者] 黒田 英夫

<プログラム開発環境>

OS : Windows 10 . . . Windows 7、8でも使用可

言語 : Visual Basic 2017 (Visual Studio 2017 Express)

<本書の配布について>

他者への配布は自由です。内容を引用する場合は出典元を明示してください。

汎用の線形計画法ソフトを提供します。線形計画法は、生産計画や経済計画などにおいて、種々の制約条件のもとに、目的の指標（目的関数）を最高にする問題を解く計算手法で、それらの制約条件や目的関数が線形のことを言います。本ソフトではシンプレックス法を用いて計算しています。

<目次>

	ページ
第1章 計算理論 -----	2
第2章 プログラムの機能・構成 -----	8
第3章 「データ」処理プログラム -----	10
第4章 「計算」プログラム -----	15
第5章 等式制約有りの計算事例 -----	18
[参考文献] -----	20

第1章 計算理論

1.1 線形計画の問題定義

一般的な線形計画を次のように定義します。

〔変数〕 x_1, x_2, \dots, x_n (n 個)

〔不等式制約〕 任意の個数 m 、不等号の向きはいずれでも可

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq (\text{または} \geq) d_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq (\text{または} \geq) d_2$$

$$\dots \dots \dots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq (\text{または} \geq) d_m$$

以上 (1-1)

〔等式制約〕 あってもなくても可、任意の個数 k

$$b_{11}x_1 + b_{12}x_2 + \dots + b_{1n}x_n = e_1$$

$$b_{21}x_1 + b_{22}x_2 + \dots + b_{2n}x_n = e_2$$

$$\dots \dots \dots$$

$$b_{k1}x_1 + b_{k2}x_2 + \dots + b_{kn}x_n = e_k$$

以上 (1-2)

注) 当然のことながら、等式制約の個数 k は変数の個数 n より少なくなければなりません。

〔非負制約〕 あってもなくても可、全変数一括

$$x_1, x_2, \dots, x_n \geq 0 \quad (1-3)$$

注) 通常の線形計画では、現実の事象を扱うので、変数は非負としています。

しかし、経済事象では負の数値もありますので、本ソフトでは非負制約の有無を選択できるようにしています。

〔目的関数〕 最大化または最小化

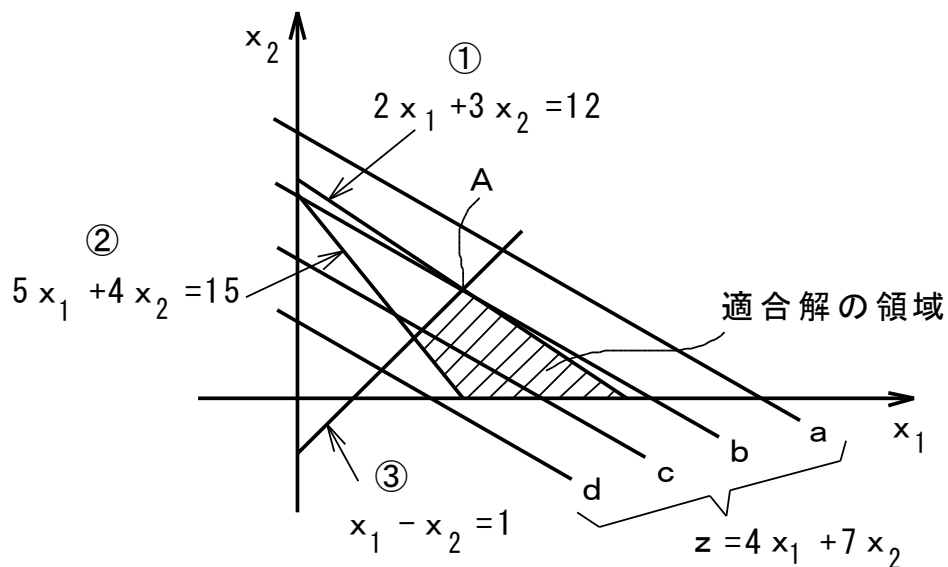
$$z = c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (1-4)$$

注) 係数 $c_1 \sim c_n$ の中には0があっても可。

1.2 問題事例

具体的な問題例を図1-1に示します。本例は、後述のソフト使用例でも取り上げるものです。図1-1では、制約式の①～③が不等式制約で、④が変数非負の制約です。グラフに示すように、これらの制約式を満足する領域は、斜線を施した領域です。そして、目的関数の式のグラフは、その関数値 z によって $a \sim d$ のように平行移動します。ここで、適合解の領域で、かつ z を最大にするのは、 b の直線となる場合であることがわかります。その b の直線を求めて、そのときに適合領域に接する点 A の座標 x_1, x_2 の値、および目的関数 z の値を算出することが、本線形計画を解くことになります。

なお、等式制約がある事例については第5章で説明します。



$$\begin{aligned}
 \text{[制約式]} \quad & 2x_1 + 3x_2 \leq 12 & \text{①} \\
 & 5x_1 + 4x_2 \geq 15 & \text{②} \\
 & x_1 - x_2 \geq 1 & \text{③} \\
 & x_1 \geq 0, x_2 \geq 0 & \text{④}
 \end{aligned}$$

$$\text{[目的関数]} \quad z = 4x_1 + 7x_2$$

図1-1 線形計画の問題事例

1.3 シンプレックス法の準備処理

本ソフトでは、最も一般的な解法のシンプレックス法^{B3)}を用いています。本解法は、変数および各制約式の右辺値が非負であり、かつ目的関数を最大化することを前提にしています。そのため、1.1節で示した一般的な問題に対して、プログラム内で次の準備処理を行ない、上記の前提に沿う問題に変形しています。

[一般的な問題の変換方法（プログラム内で自動処理）]

①制約式の右辺値（定数項）が負数の場合

制約式の両辺の各係数（含定数項）の符号を反転し、不等式においては不等号の向きを反転します。

②変数の非負制約のない場合

各変数 x_i が負数にも対応できるように次式のように2変数 x_{i1} 、 x_{i2} で置換します。

$$x_i = x_{i1} - x_{i2} \tag{1-5}$$

ただし、 x_{i1} 、 $x_{i2} \geq 0$

そして、解析結果が得られたとき、(1-5)式により x_{i1} 、 x_{i2} から元の変数 x_i の

値を算出します。

③最小化問題の場合

目的関数の各係数の符号を反転させることにより、最大化問題に変換します。
そして、解析結果が得られたとき、目的関数の値を反転させて解を得ます。

各制約式の右辺値が非負であり、かつ各変数が非負の形で表わされているとして、各制約式に対して、新たな変数（非負）を追加する事前処理を説明します。

[各制約式の前処理（プログラム内で自動処理）]

①「 \leq 」不等式の場合

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq d_i \quad (1-6)$$

スラック変数 x_{is} を導入して次式のように等式化します。スラック変数は、不等式を等式で表わすための非負の変数です。

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n + x_{is} = d_i \quad (1-7)$$

本式で初期の基底解を $x_{is} = d_i$ で与えます。

②「 \geq 」不等式の場合

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \geq d_i \quad (1-8)$$

スラック変数 x_{is} と人為変数 x_{ia} を導入して次式のように等式化します。スラック変数は、不等式を等式で表わすための非負の変数です。また、人為変数は初期解を与えるための非負の変数です。

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n - x_{is} + x_{ia} = d_i \quad (1-9)$$

本式で初期の基底解を $x_{ia} = d_i$ で与えます。

③等式の場合

$$b_{i1}x_1 + b_{i2}x_2 + \cdots + b_{in}x_n = e_i \quad (1-10)$$

人為変数 x_{ia} を導入して次式のように表わします。人為変数は初期解を与えるための非負の変数です。

$$b_{i1}x_1 + b_{i2}x_2 + \cdots + b_{in}x_n + x_{ia} = e_i \quad (1-11)$$

本式で初期の基底解を $x_{ia} = e_i$ で与えます。

上記で基底解とは、制約式の総数を m としたときに、各制約式を満たす m 個の解を言います。基底解を持つ変数を基底変数と言い、その他の変数を非基底変数と言います（非基底変数の値は0）。ただし、上記で各制約式を満たすというのは、変形された各制約式についてのことです。

1.4 シンプレックス法の計算方法

1.3節の準備処理が行なわれた後の計算方法を説明します。上記の人為変数がない場合は主問題だけの1段階のシンプレックス法計算となり、人為変数がある場合は補助問題と主問題の2段階のシンプレックス法計算となります。補助問題でも主問題でも、対象となる目的関数のある行が異なるだけで計算原理は同じです。

[備考] (1-4) 式の目的関数の係数が0の変数がある場合、それらの係数に微小値を与えます。また、それらの変数が有効となる制約式が最初の方に来ないように

調整する必要があります。これらはプログラム内で自動処理されます。なお、上記の微小値の係数は計算上だけで、結果の目的関数には取り込まれません。

<主問題のシンプレックス法計算>

1.3節の準備処理が済んだ問題を次の式で示します。ただし、制約式や変数は整理し直してあります。

$$\begin{aligned} g_{11}x_1 + g_{12}x_2 + \cdots + g_{1n}x_n + x_{1s} &= f_1 \\ g_{21}x_1 + g_{22}x_2 + \cdots + g_{2n}x_n + x_{2s} &= f_2 \\ &\vdots \\ g_{m1}x_1 + g_{m2}x_2 + \cdots + g_{mn}x_n + x_{ms} &= f_m \end{aligned} \quad \text{以上 (1-12)}$$

ただし、 $x_1 \sim x_n$ 、 $x_{1s} \sim x_{ms} \geq 0$

$$f_1 \sim f_m \geq 0$$

また、目的関数は次式で表わします。

$$z - c_1x_1 - c_2x_2 - \cdots - c_nx_n = 0 \quad (1-13)$$

(1-12)～(1-13)式を次のように行列の表式で表わします。同表式に示すように、変数 $x_{1s} \sim x_{ms}$ が初期の基底解となっています。

注) 基底解の変数は、列ベクトルの要素が1つだけ1で他は0の列に対応し、その値は定数項値となります。

	定数項	x_1	x_2	\cdots	x_n	x_{1s}	x_{2s}	\cdots	x_{ms}
z	z_0	r_1	r_2	\cdots	r_n	0	0	\cdots	0
式1	f_1	g_{11}	g_{12}	\cdots	g_{1n}	1	0	\cdots	0
式2	f_2	g_{21}	g_{22}	\cdots	g_{2n}	0	1	\cdots	0
\cdots	\cdots	\cdots	\cdots	\cdots	\cdots	\cdots	\cdots	\cdots	\cdots
式m	f_m	g_{m1}	g_{m2}	\cdots	g_{mn}	0	0	\cdots	1

(1-14)

ただし、 $z_0 = 0$ 、 $r_j = -c_j$

次に、非基底変数 $x_1 \sim x_n$ の中から1変数を選んで基底変数に入れ、その代わりに既存の基底変数の1つを基底変数から外します。そのために、いずれかの g_{pq} を選んで同要素をピボットとして掃き出し計算します。そのピボットは、以下の方法で選定します。

[ピボット g_{pq} の選定方法]

①掃き出し計算による目的関数の増加が最大となるもの

掃き出し計算により、 z_0 は次式の z_0' となります。

$$z_0' = z_0 - r_q f_p / g_{pq} \quad (1-15)$$

従って、 $-r_q f_p / g_{pq}$ が正で最大となるものを選択します。

注) 上記で正のものがなければ、計算終了となります。

ただし、次の②項が維持されることが前提条件となります。

②掃き出し計算結果の定数項（ z_0 を除く）がすべて非負であること

掃き出し計算の結果を'付きで表わすと、定数項は以下の式で表わされます。

$$f_p' = f_p / g_{pq} \geq 0 \quad (1-16)$$

$$f_i' = f_i - g_{iq} f_p / g_{pq} \geq 0 \quad (\text{ただし } i \neq p) \quad (1-17)$$

たとえば、(1-14)式で g_{12} がピボットに選定されたとすると、掃き出し結果は次の表式になります。同表式のように、 x_2 が新たな基底変数になり、逆に x_{1s} が非基底変数になっています。

	定数項	x_1	x_2	...	x_n	x_{1s}	x_{2s}	...	x_{ms}
z	z_0'	r_{11}'	0	...	r_{1n}'	r_{1n+1}'	0	...	0
式1	f_1'	g_{11}'	1	...	g_{1n}'	...	0	...	0
式2	f_2'	g_{21}'	0	...	g_{2n}'	...	1	...	0
...
式m	f_m'	g_{m1}'	0	...	g_{mn}'	...	0	...	1

(1-18)

そして、(1-18)式を新たな表式として、同様の手順を繰り返します。計算の終了は、上記の選定基準による新たなピボットがなくなった時で、そのとき z は最大化されて線形計画の解が得られます。

<補助問題のシンプレックス法計算>

人為変数がある場合は、上述の主問題を解く前に、人為変数による次の式を目的関数（最大化）とする補助問題を解析します。

$$z_a = -x_{ia} - x_{ja} - \dots - x_{ka} \quad (1-19)$$

ただし、 x_{ia} 、 x_{ja} 、 \dots 、 x_{ka} は人為変数

i 、 j 、 \dots 、 k は、人為変数のある式番号

この補助問題を上述と同様に表式で表わすと次のようになります。ただし、わかりやすく説明するため、人為変数は最終列の1つだけを入れてあります。また、同表式の最終行に補助目的関数 z_a の行を追加してあります。

	定数項	x_1	x_2	...	x_n	x_{1s}	...	x_{ms}	x_{ia}
z	z_0	r_1	r_2	...	r_n	0	...	0	0
式1	f_1	g_{11}	g_{12}	...	g_{1n}	1	...	0	0
式2	f_2	g_{21}	g_{22}	...	g_{2n}	0	...	0	0
...
式i	f_i	g_{i1}	g_{i2}	...	g_{in}	0	...	0	1
式m	f_m	g_{m1}	g_{m2}	...	g_{mn}	0	...	1	0
z_a	0	0	0	0	0	0	...	0	1

(1-20)

次に、シンプレックス法計算の前処理として、基底変数 x_{ia} に対応して、 z_a の行にある1を消去します。それには、式iの行を z_a の行から差し引きます。その結果は次のようになります。

	定数項	x_1	x_2	...	x_n	x_{1s}	...	x_{ms}	x_{ia}
z	z_0	r_1	r_2	...	r_n	0	...	0	0
式1	f_1	g_{11}	g_{12}	...	g_{1n}	1	...	0	0
式2	f_2	g_{21}	g_{22}	...	g_{2n}	0	...	0	0
...
式i	f_i	g_{i1}	g_{i2}	...	g_{in}	0	...	0	1
式m	f_m	g_{m1}	g_{m2}	...	g_{mn}	0	...	1	0
z_a	$-f_i$	$-g_{i1}$	$-g_{i2}$...	$-g_{in}$	0	...	0	0

(1-21)

他の人為変数についても、同様の処理を追加します。その後、 z_a を目的関数として、上述のシンプレックス法計算を行ないます。計算終了時には z_a 行の定数項の値が0にならねばなりません。同定数項が0になるということは、各人為変数（非負）が0ということです。この補助問題が終了したら、(1-21)式の最終の z_a 行は除外して、上述の主問題に移ります。

第2章 プログラムの機能・構成

[VBプロジェクト名] L i n e a r P r o g

本ソフトの立ち上げ画面は、図2-1のメニュー画面です。同図のように、基本メニューは「データ」、「計算」、「終了」の3つです。「データ」メニューでは入力データを作成し、「計算」メニューでは線形計画計算を実行します。そして、ソフトを終了させる際は、「終了」をクリックします。

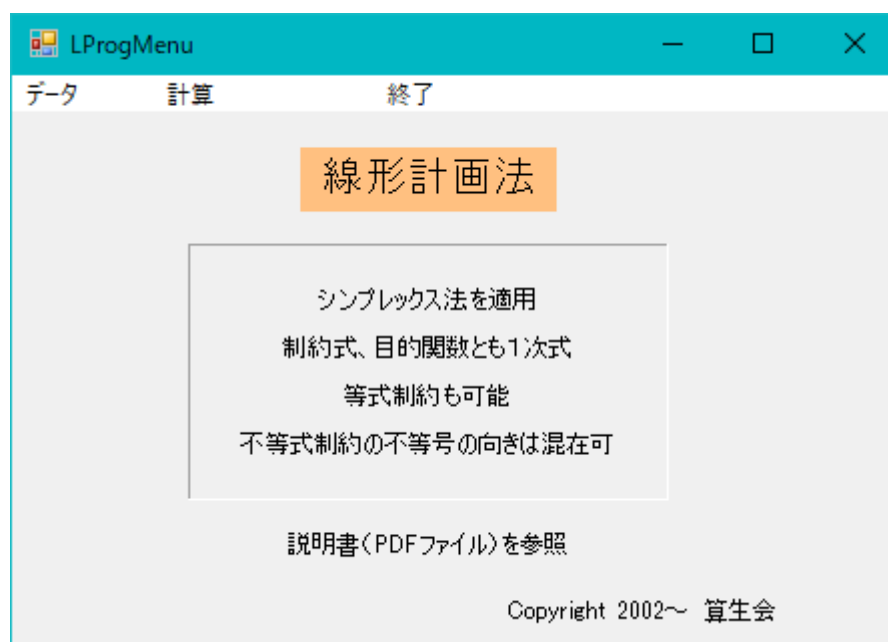


図2-1 「L i n e a r P r o g」ソフトのメニュー画面

次に、図2-2にプログラム構成を示します。同図のように、全体を「メニュー」のLProgMenuモジュール*で制御し、その下に「データ」および「計算」の2つのプログラム群に分かれています。

注) *モジュール名と各フォーム名は、ファイル名と名前を一致させてあります。

データファイルは、「入力データファイル」と計算の「結果データファイル」の2種類があります。ただし、本図には示していませんが、計算プログラム部分で一時記憶ファイル（ファイル拡張子「.tmp」）も使用しています。これらデータファイルはいずれもテキスト形式であり、エディターや表計算ソフトで閲覧・印刷することができます。

表2-1に本ソフトのファイルリストを示します。同表で、[データ]のLProgDat1とLprogDat2は、各種データを画面入力するものです。特に、LProgDat2では、1つのフォームモジュールで、複数種類のデータをグリッド切り替えにより入力できるようにしています。

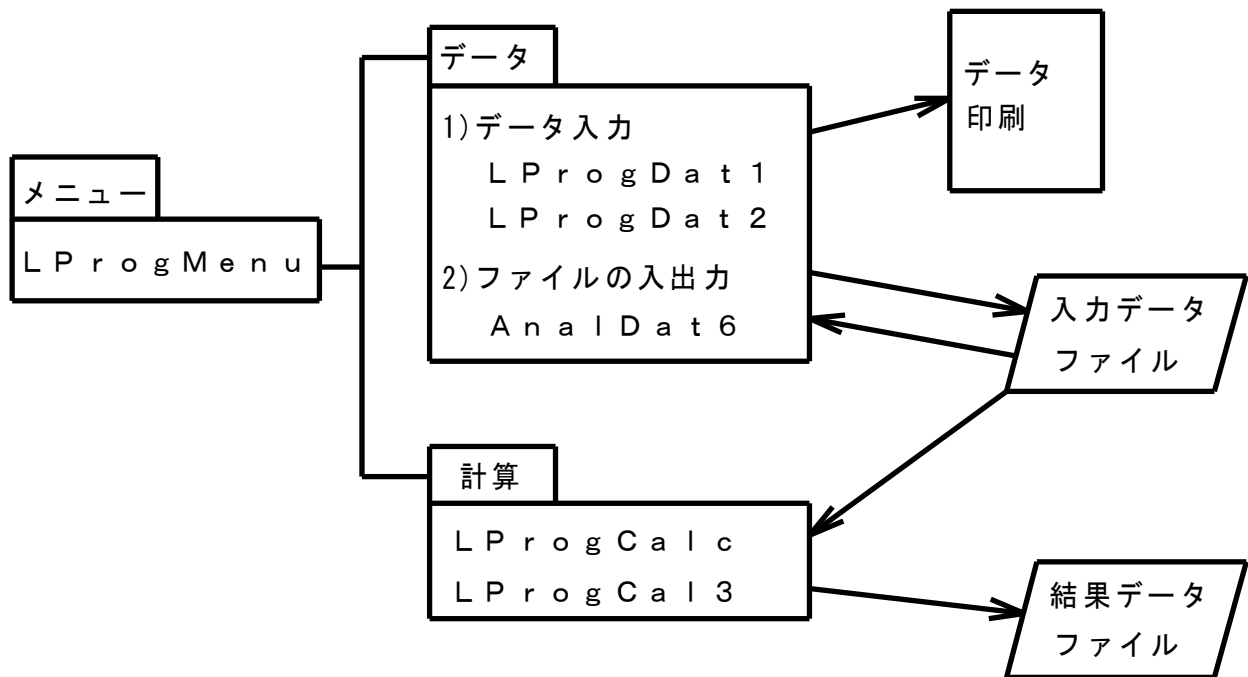


図2-2 「L i n e a r P r o g」ソフトのプログラム構成

表2-1 「L i n e a r P r o g」ソフトのファイルリスト

[全般]

ファイル名	内 容
LProgCom. vb	共通変数の宣言とサブルーチンの定義
LProgMenu. vb	メニューの提示・選択

[データ]

ファイル名	内 容
LProgDat1. vb	データ仕様の入力
LProgDat2. vb	各種データの入力
AnalDat6. vb	データファイルの入出力

[計算]

ファイル名	内 容
LProgCalc. vb	使用するデータファイルの設定
LProgCal3. vb	線形計画計算の実行

第3章 「データ」処理プログラム

3.1 サブメニュー画面

「メニュー」プログラムにおける「データ」のサブメニュー選択画面を図3-1に示します。同図に示すように、5つのサブメニューがあります。全く新たにデータを作成する場合は「データ入力・修正」を選択しますが、既存ファイルを修正して新たなファイルを作成する場合は先に「ファイル読み込み」を行ないます。

また、「オプション」→「バックアップファイル作成」にチェックを入れると、データ読み込み時にバックアップファイル（.bak）が自動作成されます。チェックを外すと作成されなくなります。

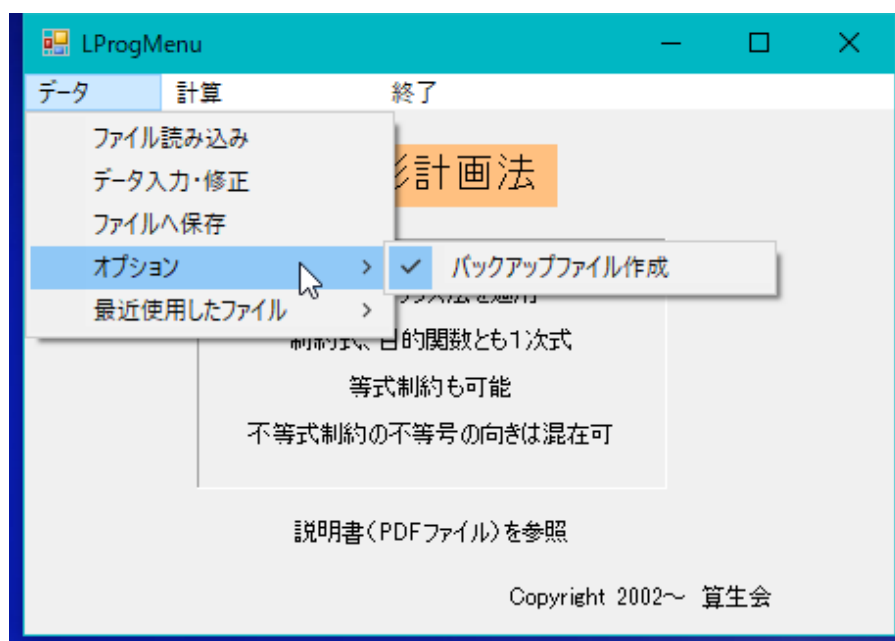


図3-1 「データ」サブメニュー画面

3.2 ファイル読み込み

サブメニューの「ファイル読み込み」を選択すると、コモンダイアログでファイルを開く画面となり、入力データファイルを選定して読み込みます。なお、入力データファイルのデフォルトの拡張子は、「.dat」です。

3.3 データ入力・修正

サブメニューの「データ入力・修正」を選択すると、図3-2のデータ仕様入力画面となります。本画面の入力について、以下に補足説明します。

①タイトルは、任意の題名を入力できます。

②変数の数、不等式制約の数、および等式制約の数をそれぞれ入力します。

◇等式制約の数は0でも構いません。また、これらの個数は、後から変更することができます。

③変数の非負制約の有無は、全変数を一括して指定します。

◇非負制約を有り（1を入力）とすると、全変数を正か0に制約します。

- ④目的関数の種別は、最小化か最大化により、それぞれ0か1を入力します。
- ⑤本画面の「設定」ボタンをクリックすると、次の不等式制約データ入力画面に移ります。
- ⑥各データ入力終了すると、本画面に戻るので、そこで「すべて確定」ボタンをクリックすると入力データが確定します。

図3-2 データ仕様入力画面

3.4 不等式制約データ入力画面

図3-3に不等式制約データ入力画面を示します。

- ①各不等式について、式の種別（不等号の向きにより-1か1）、各変数の係数、および右辺の値を入力します。
- ②「コピー」、「貼り付け」ボタンにより、通常のクリップボード利用ができます。これは以降の画面でも同様です。
◇「取り消し」ボタンを押すと直前の貼り付けを取り消せます。
- ③式の種別に「-99」を入力すると、その行を削除して前詰めします。
- ④すべてのデータを入力し終わったなら、「設定」ボタンをクリックして、次の画面に移ります。

3.5 等式制約データ入力画面

本画面は、等式制約の数が1以上の場合に表示されます。図3-4に等式制約データ入力画面を示します。

- ①各等式について、各変数の係数、および右辺の値を入力します。
- ②係数1に「-99」を入力すると、その行を削除して前詰めします。
- ③すべてのデータを入力し終わったなら、「設定」ボタンをクリックして、次の目的関数データ画面に移ります。

LProgDat2

不等式制約のデータ入力

式の種別に-99を入力すると、その行を削除・前詰めします。

[式の種別]-1: ≤ の不等式、1: ≥ の不等式

	式番号	式の種別	係数1	係数2	右辺の値
▶	1	-1	2	3	12
	2	1	5	4	15
	3	1	1	-1	1
*					

設定 戻る コピー 貼り付け 取り消し

図3-3 不等式制約データ入力画面

LProgDat2

等式制約のデータ入力

係数1に-99を入力すると、その行を削除・前詰めします。

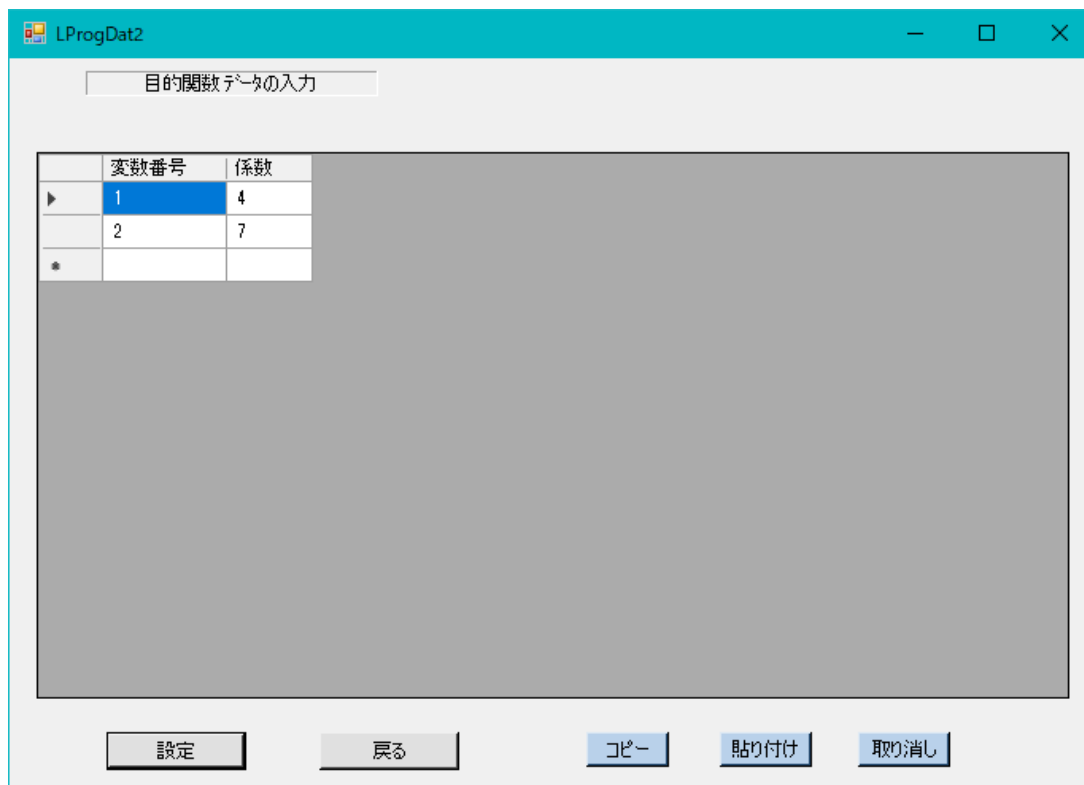
	等式番号	係数1	係数2	右辺の値
▶	1	1	-1	5
*				

設定 戻る コピー 貼り付け 取り消し

図3-4 等式制約データ入力画面

3.6 目的関数データ入力画面

図3-5に目的関数データ入力画面を示します。本画面で、目的関数における各変数の係数を入力します。すべてのデータを入力し終わったなら、「設定」ボタンをクリックして、図3-2のデータ仕様入力画面に戻ります。



	変数番号	係数
▶	1	4
	2	7
*		

図3-5 目的関数データ入力画面

3.7 ファイルへの保存

サブメニューの「ファイルへの保存」を選択すると、コモンダイアログでファイルを保存する画面となり、入力データファイルを設定すると、データが書き出されます。新たなファイル名を入力すると新ファイルが作成されます。先に入力データファイルを読み込んである場合に、同ファイルを選定すると上書きとなります。

図3-6に入力データファイル例を示します。本データファイルは、図1-1の問題事例に対応したもので、同図のように、1行目のタイトル以降、各入力データが書き込まれています。

```

プログラム名, LinearProg
線形計画用テストデータ
変数の数, 2
不等式制約の数, 3
等式制約の数, 0
変数の非負制約の有無 (0:なし、1:有り), 1
目的関数の種別 (0:最小化、1:最大化), 1
<不等式制約> 式番号、不等式種別 (-1:≤、1:≥)、各係数、右辺値
1,    -1,    2,    3,    12
2,    1,    5,    4,    15
3,    1,    1,   -1,    1
<目的関数> 変数番号、係数
1,    4
2,    7

```

図3-6 入力データファイル例

3.8 最近使用したファイルの選択

サブメニュー「最近使用したファイル」のポップアップで、ファイルリストから選択できます。

注) 新版ソフトでファイルが使用されないとリスト表示されません。

第4章 「計算」プログラム

4.1 サブメニュー画面

「メニュー」プログラムにおける「計算」のサブメニュー画面を図4-1に示します。サブメニューは「線形計画計算の実行」と「計算結果の再表示」の2つがあります。「線形計画計算の実行」をクリックすると、次の「計算」ファイル設定画面に移ります。また、「計算結果の再表示」は、計算の実行後にいつでも結果データを再表示するものです。

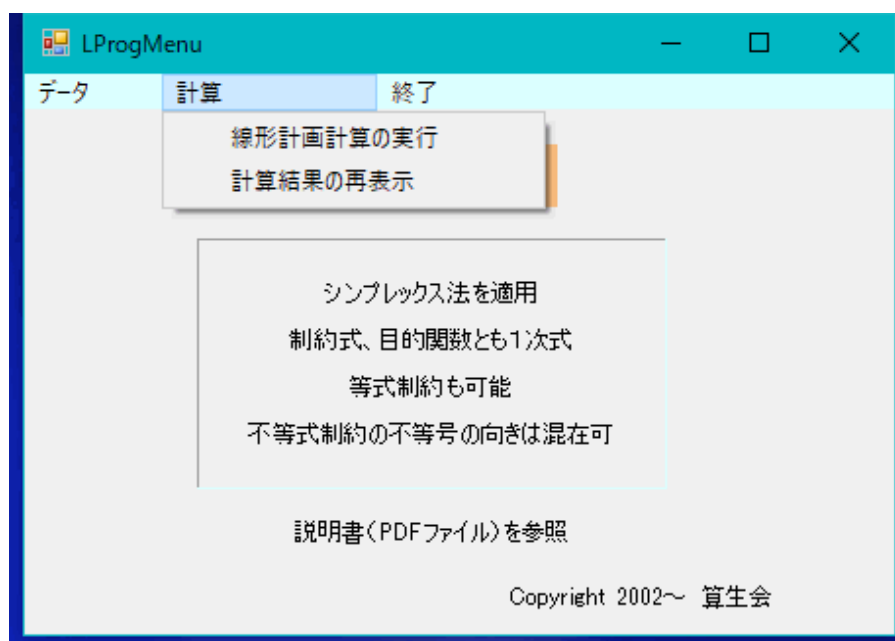


図4-1 「計算」サブメニュー画面

4.2 「計算」ファイルの設定

図4-2に「計算」ファイル設定画面を示します。

- ①「入力ファイル指定」ボタンをクリックすると、コモンダイアログで入力データファイルを選定できます。そして、再び「計算」ファイル設定画面に戻ると、「入力ファイル指定」のテキストボックスには選定済み入力ファイル名が、また「出力ファイル指定」の方には入力ファイル名の拡張子を「.rep」に変更したファイル名が表示されます。もし、他のファイル名に変更したい場合は、テキストボックスで修正します。
- ②本画面で「設定」ボタンをクリックすると、計算が開始されます。

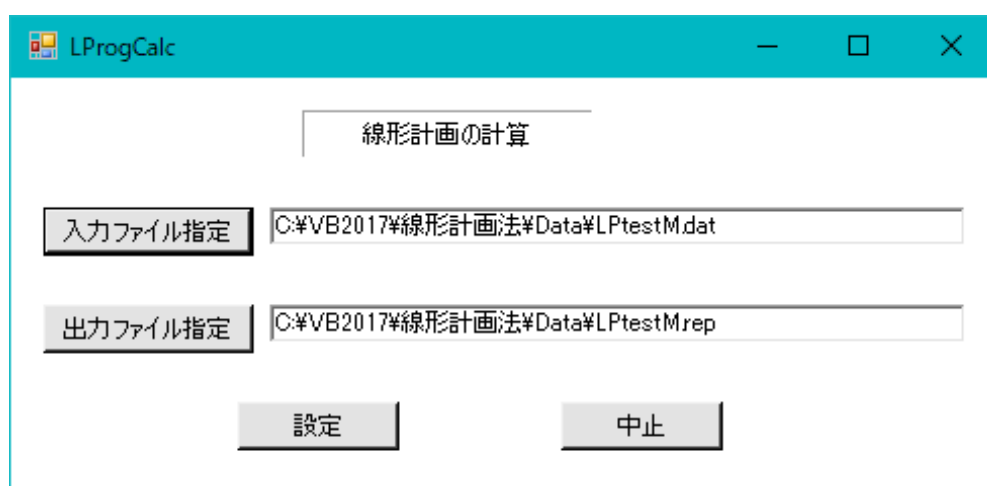


図4-2 「計算」 ファイル設定画面

4.3 計算終了

図4-3に計算終了画面、図4-4に計算確認ダイアログ画面を示します。図4-3では、かかった計算時間も表示されますが、本例では0sとなっており、1s未満で終了しています。また、図4-4の計算確認ダイアログ画面では、最適の目標関数値および各変数解も表示されます。本画面で「OK」ボタンをクリックすると、立ち上げ時のメニュー画面に戻ります。

◇図4-4の各変数解の表示は、最大10変数までで他は省略されます。

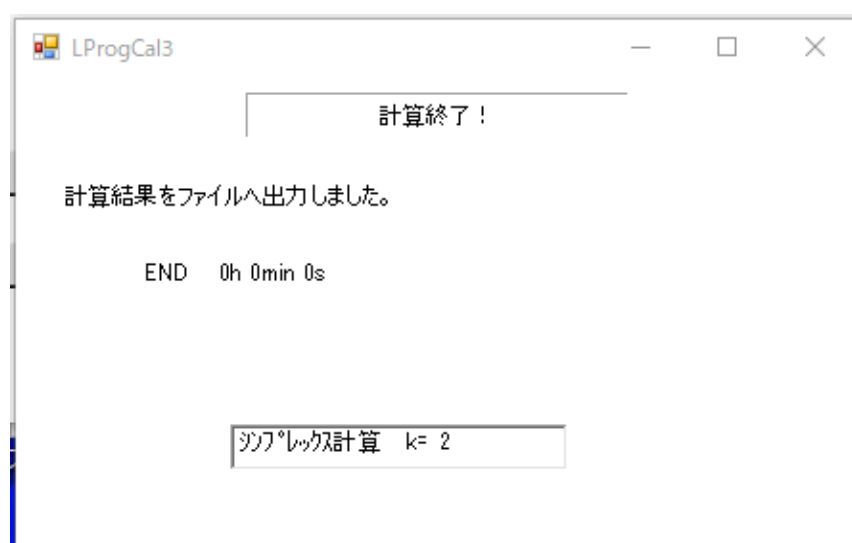


図4-3 計算終了画面

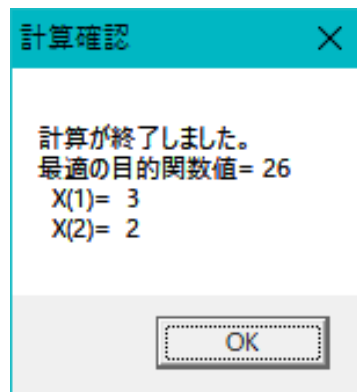


図4-4 計算確認ダイアログ画面

4.4 計算結果

上記4.2節で指定した出力ファイルに計算結果が書き出されます。図4-5に例題の計算結果ファイルへの出力を示します。同図で○印を付した数字の部分について補足説明します。

- ①目的関数値が26と出力されています。
- ②～③そのときの変数値が出力されており、 $X(1)=3$ 、 $X(2)=2$ となっています。

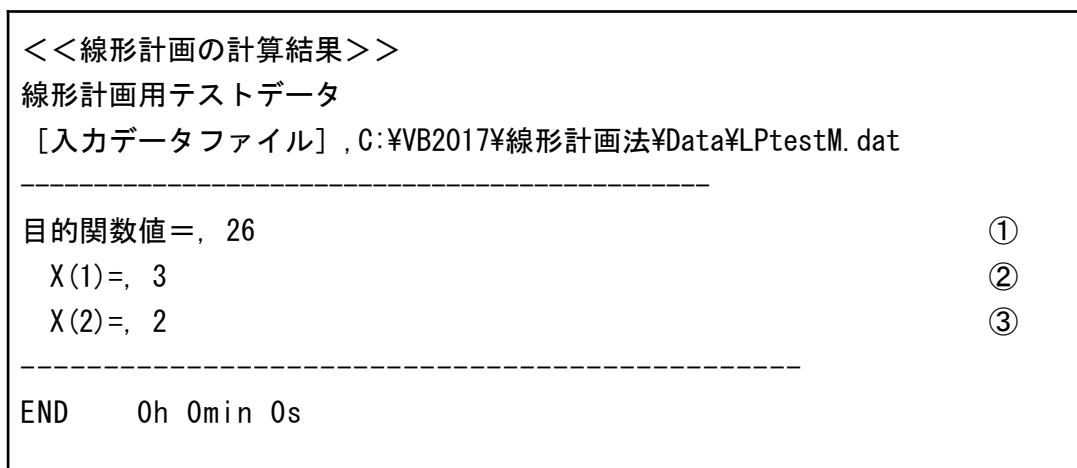
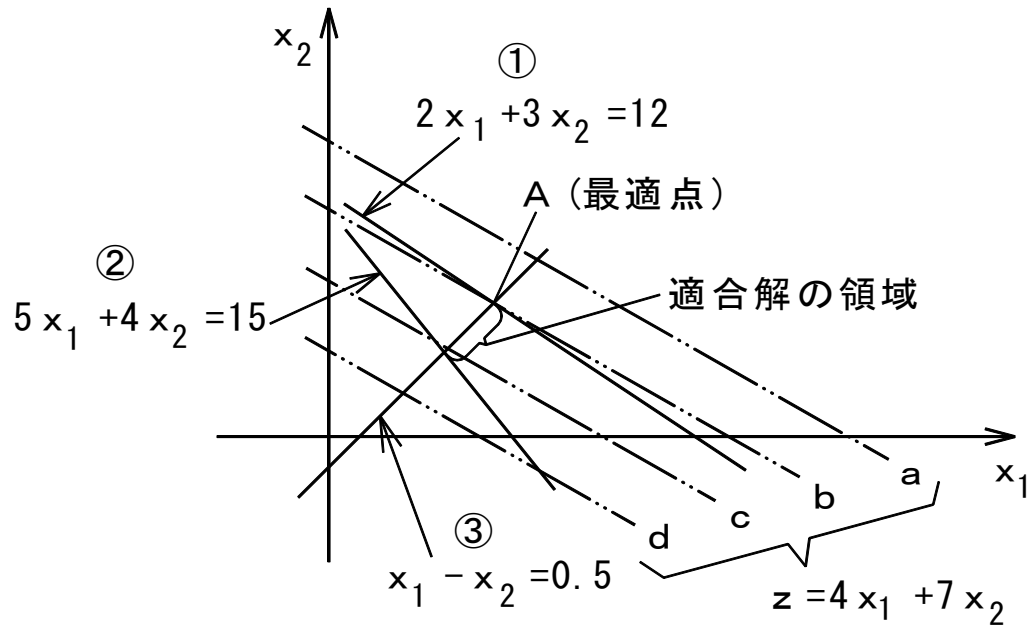


図4-5 計算結果ファイルへの出力例

第5章 等式制約有りの計算事例

5.1 問題事例

図5-1に問題事例を示します。同図のように、③の等式制約が入っています。従って、図の直線③上に適合解の領域に存在します。そして、図のA点が最適点となります。



$$\begin{aligned} \text{[制約式]} \quad & 2x_1 + 3x_2 \leq 12 & \text{①} \\ & 5x_1 + 4x_2 \geq 15 & \text{②} \\ & x_1 - x_2 = 0.5 & \text{③} \end{aligned}$$

$$\text{[目的関数]} \quad z = 4x_1 + 7x_2$$

図5-1 等式制限有りの問題事例

5.2 入力データ

図5-2に入力データファイルを示します。

5.3 計算結果

図5-3に結果データファイルを示します。同図のように、目的関数値は26.2で、そのときの変数値は、 $X(1)=2.7$ 、 $X(2)=2.2$ です。

```

プログラム名, LinearProg
線形計画用テストデータ（等式制約有り）
変数の数, 2
不等式制約の数, 2
等式制約の数, 1
変数の非負制約の有無（0:なし、1:有り）, 0
目的関数の種別（0:最小化、1:最大化）, 1
<不等式制約>式番号、不等式種別（-1:≤、1:≥）、各係数、右辺値
1,    -1,    2,    3,    12
2,    1,    5,    4,    15
<等式制約>式番号、各係数、右辺値
1,    1,    -1,    .5
<目的関数>変数番号、係数
1,    4
2,    7

```

図5-2 入力データファイル

```

<<線形計画の計算結果>>
線形計画用テストデータ（等式制約有り）
[入力データファイル], G:\¥VB2017¥線形計画法¥Data¥LPtestM-REqua.dat
-----
目的関数値=, 26.2
X(1)=, 2.7
X(2)=, 2.2
-----
END      0h 0min 0s

```

図5-3 結果データファイル

[参考文献]

<A> BASIC言語

- A1) 「Microsoft Visual Basic プログラミング ガイド」、
マイクロソフト (株)、1997
- A2) 「実力アップ Visual Basic.NET 基礎編」、川口 輝久、
(株) 技術評論社、2003
- A3) 「Visual Basic 応用プログラミング」、林 晴比古、
ソフトバンク (株)、1997
- A4) 「Visual Basic for Windows プログラミング・パーツ」、河西 朝雄、
(株) ナツメ社、1994
- A5) 「かんたんプログラミング Visual Basic.NET 応用編」、田中 亨、
(株) オーム社、2002
- A6) 「Visual Basic リファレンス・ブック」、相沢 文雄、
(株) ナツメ社、1994
- A7) 「プログラミング Visual Basic.Net Vol.2 活用編」、Francesco Balena、
(株) ドキュメントシステム 訳、日経BPソフトプレス、2002
- A8) 「Visual Basic 2005 [実践] プログラミングテクニック」、谷尻 かおり、
技術評論社、2007

 数値計算

- B1) 「数値計算プログラム BASIC」、J. ハイムホー 編、大矢 建正 訳、
(株) マグロウヒル好学社、1982
- B2) 「応用・BASIC (数値解析とその手法)」、対馬 勝英 他、
(株) パワー社、1981
- B3) 「パソコン BASIC 数値計算Ⅱ」、町田 東一 他、
東海大学出版会、1984
- B4) 「数学公式・数表ハンドブック」、スピーゲル 著、氏家 勝巳 訳、
マグロウヒル ブック (株)、1984
- B5) 「数値解析とその応用」、名取 亮、
コロナ社、1992
- B6) 「線形計算」、名取 亮、
朝倉書店、1993
- B7) 「Visual Basicによる工学計算プログラム」、黒田 英夫、
CQ出版社、2001
- B8) 「Visual Basicによる数値解析プログラム」、黒田 英夫、
CQ出版社、2002