

NUOPT による最適化モデルの開発 (1)

齋藤雄志 (専修大学 ネットワーク情報学部)

Developing Linear Programming Models by NUOPT (1)

Takeshi SAITO (School of Network and Information, Senshu University)

This paper is an introduction for students to learn to develop linear programming models using NUOPT, an optimization software. NUOPT is one of few home-manufactured optimization software and is used widely in many universities and research institutes in Japan. The first half of the paper is a beginner's guide for social science students who are not familiar with mathematics and LP. The second half is a user's guide for NUOPT where a simple power generation mix LP model is introduced as an example of optimization model.

キーワード：NUOPT, 線形計画法, 最適電源構成

Key words : NUOPT, Linear Programming, Power Generation Mix

1. まえがき

本稿は、線形計画法やそれを解くソフトウェアに関してあまり知識をもたない学生が、それらを勉強しつつ自分でモデルを開発・利用するために代表的ソフトのひとつである NUOPT (株式会社 数理システム/開発・販売) を利用するための必要最小限の情報を提供することを目的として書かれた。いうなれば NUOPT の簡易マニュアルである。もちろん、線形計画法についての一定の知識は当然のことながら必要であるが、本稿ではごく初歩的な説明をしているだけである。それゆえ、シンプレックス法など線形計画法の解法については触れていないのでそれぞれの学生が別途勉強する必要がある。

本稿の主な目的は、最適化問題というものがどういうものを理解し、実際に解くための情報を提供する。わかりやすくするため、一部の説明では単純な近似的な計算法で代替させている (シャドウプライスの計算)。とはいえ最後に述べる電源計画の最適化などはかなり複雑な構造になっており、それなりに慎重に読む必要がある。

筆者自身はエネルギー問題を分析するツールのひとつとして、1980年代の前半から線形計画法を利用してきたが、現在までにその技術は大きく進歩した。かつては数千の変数、数千の制約の線形計画法の問題を解くのは容易でなく、その計算費用も相当なものだった。現在ではごく普通のパソコンと優れたソフトウェアの組み合わせによりかなり大きな問題を解くこともできる。はじめて線形計画法が考えられてから実に 190 万倍もの速さで解けるようになったという [2]。

線形計画法の問題が現実の問題としてどのくらいありどのくらい応用されているか、またどのような問題があるかについては別稿 [3][4] で詳しく述べる予定である。モデル開発で一番重要なのは定式化である。定式化に問題があれば、意味のない計算を行ったり、興味ある結果を得られなかったりする。目

的関数を構成するコストに関するデータに関して十分な情報がなければ意味のない定式化になるし、そもそも制約条件を構成するデータに大きな不確定性があれば問題の定式化が妥当とはいえないであろう。モデルの右辺の制約条件の値は不確定な情報であることが多い。たとえば、エネルギーモデルにおいて、「原油輸入可能量」というような制約は、それが事実であるというよりは、線形計画法にあわせるためにそのような概念を導入したともいえなくもない。また巨大なモデルはしばしばそのパラメータに関して鈍感な反応を示すことがある。これはモデルの構造に問題があることが原因ということもある。

計算のために内点法を使用している NUOPT は、(株) 数理システムによって開発販売されている国産の線形計画法のソフトウェアである。NUOPT は非線形計画法や整数計画法の問題を解くこともできる。本学では情報科学センターに 2 セット 導入されている。このほかに古くからは MPSX などのソフトウェアが使われてきたし、世界的には Xpress-MP (国内総代理店シェルサービス (株)) が有名なソフトウェアの一つである。また多くの教科書にもさまざまな線形計画法や非線形計画法のプログラムが掲載されている。

2. 線形計画法の簡単な例題 [生産計画]

この章における説明は線形計画法に関して知識を持たない文系学生のための解説である。非常に簡単な例題で説明しており、NUOPT そのものに関心がある場合は 2-5 章は読み飛ばして差し支えない。この例題は筆者が 15 年間程担当した経営学部 1 年次生向けの「情報管理概論」の付属配付資料 (教員用に配布) からとった例題である。

2 種類の製品 P_1 , P_2 を製造している会社がある。それぞれを 1t 製造するために必要な原料 M_1 , M_2 , M_3 の使用量 (原単位で単位は t) と各 1 日あたりの使用可能量 (t) は下記の通りとする。すでに述べたように「使用可能量」というような制約は、契約によってそれが決められている以外にはその妥当性を評価することは容易でない。以下の例に 440t 等という制約が「事実」でなく「信念」や「仮定」である場合もありうる。ここでこの問題を論じていると話はややこしくなるので回避し、あらためて論じたい [3] [4]。多くの定式化において下記の例に示すような利益や原単位は市場あるいは技術的に決定されるので、右辺の制約条件と比べると相対的に信頼性のあるパラメータと見なすことができる。

このとき 2 つの製品を販売することにより得られる一日の利益は

$$P_1 \quad 5 \text{ 万円/t}$$

$$P_2 \quad 8 \text{ 万円/t}$$

として、総利益を最大にする 2 つの製品の 1 日あたりの最適生産量を求める。

[定式化] (モデル LP1)

変数

原材料	原単位 (t/t)		使用可能量 (t)
	P_1	P_2	
M_1	3	4	440
M_2	3	1	270
M_3	2	8	720

X_1 : 製品 P_1 の 1 日あたりの製造量 (t)

X_2 : 製品 P_2 の 1 日あたりの製造量 (t)

制約式

M_1 について $3X_1 + 4X_2 \leq 440$ ①

M_2 について $3X_1 + 1X_2 \leq 270$ ② (1)

M_3 について $2X_1 + 8X_2 \leq 720$ ③

非負条件 $X_1 \geq 0, X_2 \geq 0$ (2)

目的関数 $5X_1 + 8X_2 \rightarrow \text{最大}$ (3)

[説明]

(1) 制約条件の意味

制約条件は 4 つ [①②③+(2)] あるが, (X_1, X_2) は 4 つの条件をすべて満たす範囲だけを動き回ることが出来る。4 つをすべて満たす範囲を可能領域という。

線形計画法に初めて出会った読者 (文系学生) 向きに説明すればつぎのようになる。まず, M_1 を満たす範囲がどのような範囲かを考える。もちろん X_1, X_2 は正であるから可能領域は第 1 象限に限定される。一般に一次式による制約条件の範囲はそれに対応する直線のどちらかである。今の場合直線 $3X_1 + 4X_2 = 440$ の上か下である。 $X_1 = 0$ と $X_2 = 0$ を代入してみると M_1 は満たされるので M_1 の範囲は原点側である。図 1 の斜線の部分はこの制約に $X_1 \geq 0, X_2 \geq 0$ を追加した制約領域である。これは高校まで数学の知識である。

(2) 可能領域

M_2, M_3 についても同様の範囲を作成し, その共通部分をとればそれが可能領域になる。それは図 2 に

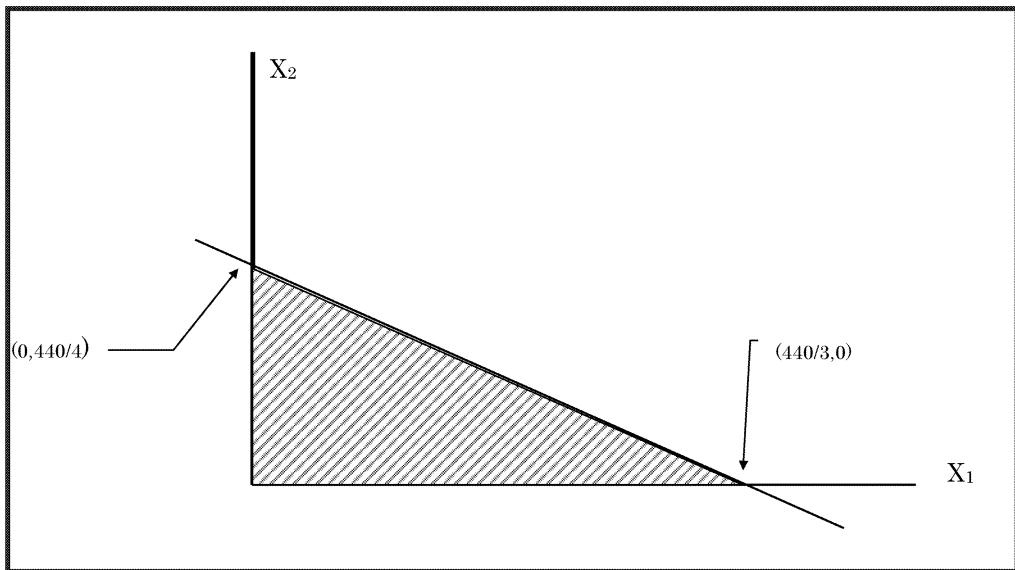


図 1 制約条件 M_1 の満たす範囲

可能領域という四角で示してある 5 角形の領域である。

(3) 目的関数と解について

筆者の経験によれば、高校時代に数学を苦手とした人々に目的関数の意味をわかりやすく説明するのは意外に難しい。一言でいえば、 X_1, X_2 が可能領域を動くとき、(3) が最大になるような X_1, X_2 を見つけることが目的関数最大化の意味である。目的関数を最大にするような解を最適解とよぶ。その時の目的関数の値を最適値 (最大値) とよぶ。最適解がどのようなイメージで計算されるかを筆者なりにやさしく説明すると以下ようになる。

いま目的関数の値を勝手に決めてみる。それはもちろん最適値とはかぎらない。その値を X_0 とおく。この値を定数として固定し、目的関数の式 $X_0 = 5X_1 + 8X_2$ において X_1, X_2 を動かしてみる。 (X_1, X_2) は当然のことながら、直線 $X_0 = 5X_1 + 8X_2$ の上を動く。いま具体的に $X_0 = 415$ とおいてみる。

直線 $415 = 5 \cdot X_1 + 8 \cdot X_2$ は次のような意味を持っている。◎ が特に重要である。

- この直線はたとえば、点 (35,30) を通る。もちろんこの特定な点には特に意味がない。
- ◎ この直線上で目的関数はいつも 415 である。
- この直線は X_2 軸と $X_2 = 415/8$ で交わる。
- この直線は X_1 軸と $X_1 = 415/5$ で交わる。

このことを一般化すると

- X_0 値が任意の値として数値として与えられているとき、これに対応する直線 $X_0 = 5X_1 + 8X_2$ 上ではいつも (3) の目的関数の値は一定である。(もちろん値は X_0)
- この直線は、 X_2 軸と $X_2 = X_0/8$ で交わり X_1 軸と $X_1 = X_0/5$ で交わることがわかる。きわめて原始的に上記のモデルの最適解をもとめるには、つぎのようにすればよい。

① 可能領域の中から任意の点を選ぶ。

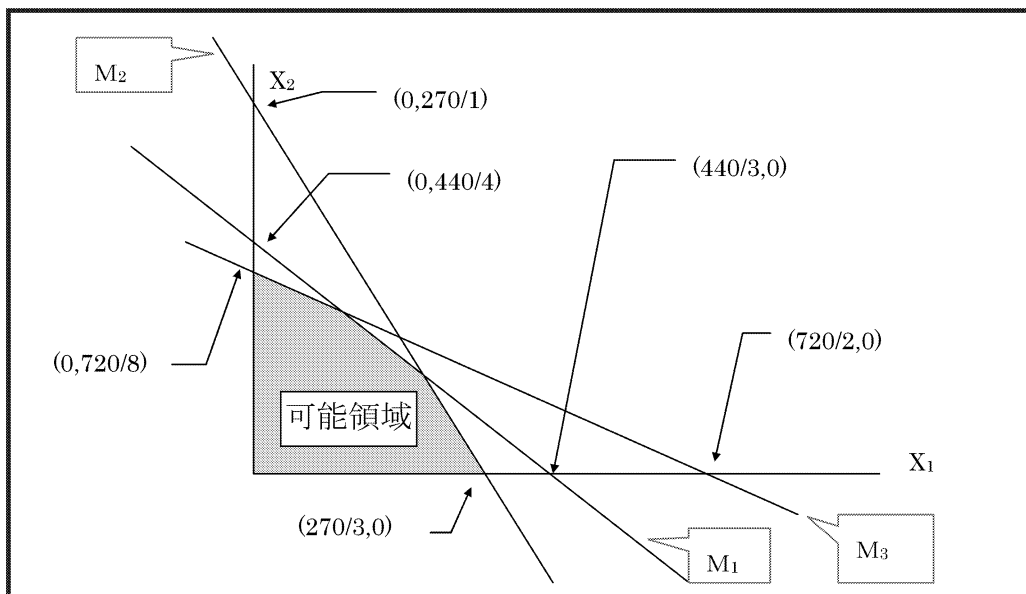


図2 M_1, M_2, M_3 を満たす可能領域

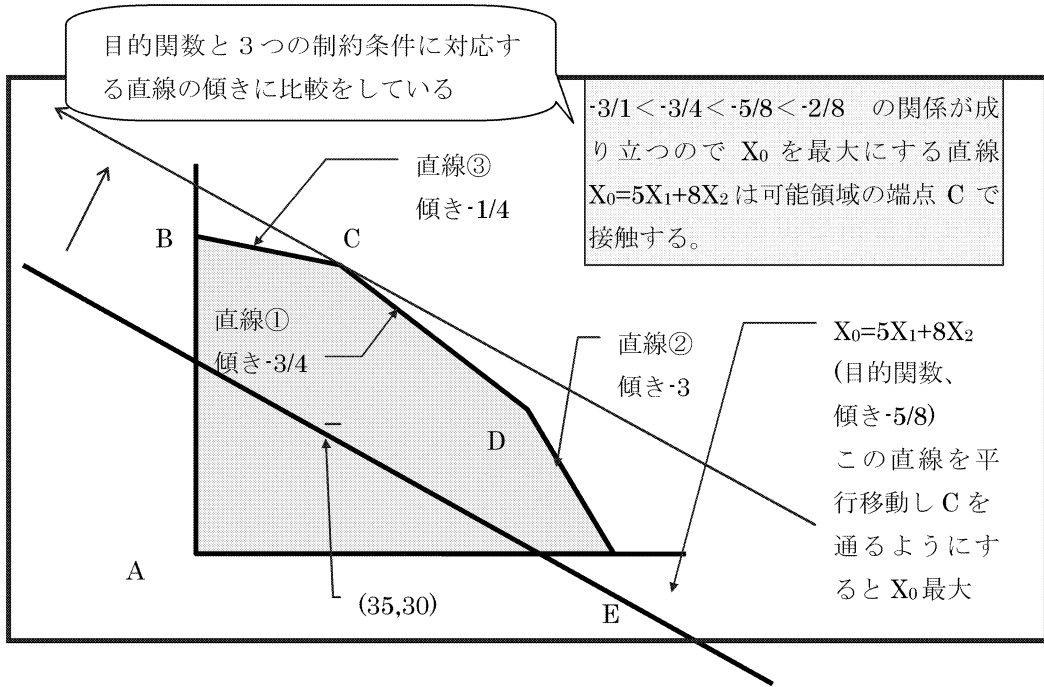


図3 上記の可能領域の拡大図

- ② その点の値を目的関数に代入する。その値を X_0 とするとき、直線 $X_0 = 5X_1 + 8X_2$ の上で目的関数の値は X_0 で一定である。
- ③ このことから①で選んだ点を可能領域の中で徐々に点を動かし直線 $X_0 = 5X_1 + 8X_2$ と直角の方向の右上方を移していけばよい。
- ④ このようにして X_0 が最大になるように選ばれる「点」が最適解であり、それは五角形の角のひとつである。図3でいえば、Cが最適解であることがわかる。この点に対応する X_0 の値が最適値になる。

(4) 目的関数を最大にする端点 (可能領域の角) の見付け方

このような簡単に図示可能な線形計画法の問題では、つぎのようにして制約条件と目的関数を構成する直線の傾きを比較すれば最適解に対応する角の点 (端点という。角の点という意味である) を見つけることができる。上記の例でいえばつぎのように作業を行えばよい。

(4) 直線の傾きの比較

- 1) 目的関数の傾き $-5/8$
- 2) 直線①の傾き $-3/4$
- 3) 直線②の傾き $-3/1$
- 4) 直線③の傾き $-2/8$

これらの傾きの間に $-3/1 < -3/4 < -5/8 < -2/8$ の関係が成り立つので、 X_0 を最大にする直線 $X_0 = 5X_1 + 8X_2$ は可能領域と端点Cで接触することが分る。

(5) 最適解

実際に C 点を 2 つの直線 ① ② の交点として、連立方程式をとけば、解は $X_1=40$, $X_2=80$ となる。この座標を目的関数に代入すると、目的関数の値 = 840 = 最適解となる。

3. Excel による線形計画法の解法

Excel のソルバーを利用すると線形計画を解くことができる。手順はつぎのとおりである。

(1) Excel の「ツール」 「アドイン」 で「ソルバアドイン」をクリックする。この操作で「ツール」に「ソルバー」が現れる。

(2) Excel のシートに図 4 のように数値や式を入力する。

① 「目的関数」のセルには目的関数の係数値を記入する。

② 「目的関数の値」のセルには目的関数の式を記入する。係数は「目的関数の係数」のセルを使用し、変数は X_1 , X_2 のセルを使用する。

③ 係数の値を 6 つのセルに入力する。

④ 「計算値」には次の式を記入する。 X_1 , X_2 は「変数」のセルを、係数の値は「係数」のセルを使用する。

$$=3X_1+4X_2$$

$$=3X_1+1X_2$$

$$=2X_1+8X_2$$

(3) 「ツール」 「ソルバー」 をクリックする。

つぎの図の各セルにはつぎのように指定する。罫線は数値や変数をわかりやすくするために書いてある。また色つきのセルは式を記入するセルであることを示している。

① 目的セル → 図 4 の「目的関数の値」のセル

② 変化させるセル → 図 4 の変数のセル

③ 「追加」をクリックした後、「セル参照」には図 4 の「計算値」のセル、「制約条件」のセルには図 4 の「制約条件」のセルを指定する。「ok」をクリックする。

④ 「実行」をクリックすると、次の図 5 の解が求まる。

目的関数	目的関数の係数		目的関数の値
	5	8	0
変数	X1	X2	
係数	計算値		制約条件
M1	3	4	0 <
M2	3	1	0 <
M3	2	8	0 <
			440
			270
			720

図 4 ソルバーへの入力

目的関数	目的関数の係数		目的関数の値	
	5	8	840	
変数	X1	X2		
	40	80		
係数			計算値	制約条件
M1	3	4	440 <	440
M2	3	1	200 <	270
M3	2	8	720 <	720

図5 ソルバーによる解

4. 線形計画法の一般形

線形計画法はつぎのような一般形で表現できる。現在では、変数の数 n や制約条件の数 m が数万あるいは数十万を超えても PC で解くことが可能である。Excel ではおそらく n や m の値は数十程度が現実の限界であろう。

制約条件

$$\begin{aligned}
 a_{11}X_1 + a_{12}X_2 + \dots + a_{1n}X_n &\leq b_1 \\
 a_{21}X_1 + a_{22}X_2 + \dots + a_{2n}X_n &\leq b_2 \\
 &\dots \\
 a_{m1}X_1 + a_{m2}X_2 + \dots + a_{mn}X_n &\leq b_m
 \end{aligned}
 \tag{4}$$

および非負条件

$$X_1 \geq 0 \quad X_2 \geq 0 \quad \dots \quad X_n \geq 0
 \tag{5}$$

のもとで目的関数

$$c_1X_1 + c_2X_2 + \dots + c_nX_n
 \tag{6}$$

を最大にする X_1, X_2, \dots, X_n を求める問題を線形計画法という。このような問題の解を最適解という。ただし $a_{11}, \dots, a_{mn}, b_1, \dots, b_m, \dots, c_1, \dots, c_n$ は定数, X_1, \dots, X_n は変数である。

5. 影の価格 (シャドウプライス)

最適化問題の優れた点は、最適解そのものを見いだすことができる他にシャドウプライスを計算できることである。

線形計画法の一般形において目的関数を最大にする最適解を $X_1 = X_1^{\max}, X_2 = X_2^{\max}, \dots, X_n = X_n^{\max}$ とする。このときの目的関数の値 O はつぎようになる。

$$O^{\max} = c_1X_1^{\max} + c_2X_2^{\max} + \dots + c_nX_n^{\max}$$

いま制約条件の1つ, たとえば1番目の制約条件の右辺を1単位厳しくする。つまり, b_1 を b_1-1 に置き換える。するとつぎのような問題ができる。ただしここでは b_1 は1よりもずっと大きい数とする。これは b_1 から b_1-1 への変動が微小変動であることを保証するためである。この「-1」は無限小の引き算の代わりに使用している。

制約条件

$$\begin{aligned} a_{11}X_1 + a_{12}X_2 + \cdots + a_{1n}X_n &\leq b_1 - 1 && \leftarrow \text{ここだけが変化する} \\ a_{21}X_1 + a_{22}X_2 + \cdots + a_{2n}X_n &\leq b_2 \\ &\dots \\ a_{m1}X_1 + a_{m2}X_2 + \cdots + a_{mn}X_n &\leq b_m \end{aligned} \quad (7)$$

および非負条件

$$X_1 \geq 0 \quad X_2 \geq 0 \quad \cdots \quad X_n \geq 0 \quad (8)$$

のもとで目的関数

$$c_1X_1 + c_2X_2 + \cdots + c_nX_n \quad (9)$$

を最大にする。

この新問題の解を

$$X_1 = \underline{X}_1^{\max}, X_2 = \underline{X}_2^{\max}, \dots, X_n = \underline{X}_n^{\max}$$

とし, これに対応する目的関数の値を

$$\underline{O}^{\max} = c_1\underline{X}_1^{\max} + c_2\underline{X}_2^{\max} + \cdots + c_n\underline{X}_n^{\max} \quad (10)$$

とする。このとき目的関数の減少量 $O^{\max} - \underline{O}^{\max}$ を一番目の制約条件の影の価格 (シャドウプライス) という。これは利用可能な資源 b_1 の1単位分の価値を表している。 b_1 が b_1-1 に変化することにより, つまり1単位減少することによって発生する利益の減少分 $O^{\max} - \underline{O}^{\max}$ を表しているからである。

同様にして2番目, 3番目, \dots , m 番目の制約条件のシャドウプライスを求めることができる。NUOPTのような専門のソフトでは, 厳密な意味でのシャドウプライスを計算することができる。

つぎに双対問題にふれよう。実は双対問題の解はシャドウプライスでもある。いまつぎのような線形計画法の問題を考える。これを主問題とよぶことにする。

(1) 主問題

制約条件

$$\begin{aligned} a_{11}X_1 + a_{12}X_2 + \cdots + a_{1n}X_n &\leq b_1 \\ a_{21}X_1 + a_{22}X_2 + \cdots + a_{2n}X_n &\leq b_2 \\ &\dots \\ a_{m1}X_1 + a_{m2}X_2 + \cdots + a_{mn}X_n &\leq b_m \end{aligned} \quad (11)$$

および非負条件

$$X_1 \geq 0 \quad X_2 \geq 0 \quad \cdots \quad X_n \geq 0 \quad (12)$$

のもとで目的関数

$$O_1 = c_1 X_1 + c_2 X_2 + \dots + c_n X_n \tag{13}$$

を最大にする X, X_2, \dots, X_n を求める。

(2) 双対問題

主問題に対してつぎのような線形計画法の問題を双対問題という。

制約条件

$$\begin{aligned} a_{11} Y_1 + a_{21} Y_2 + \dots + a_{m1} Y_m &\geq c_1 \\ a_{12} Y_1 + a_{22} Y_2 + \dots + a_{m2} Y_m &\geq c_2 \\ &\dots \\ a_{1n} Y_1 + a_{2n} Y_2 + \dots + a_{mn} Y_m &\geq c_n \end{aligned} \tag{10}$$

および非負条件

$$Y_1 \geq 0 \quad Y_2 \geq 0 \quad \dots \quad Y_m \geq 0 \tag{11}$$

のもとで目的関数

$$O_2 = b_1 Y_1 + b_2 Y_2 + \dots + b_m Y_m \tag{12}$$

を最小にする Y_1, Y_2, \dots, Y_m を求める。

このときつぎの性質が成り立つ。詳しくは専門書をみるとよい。

双対定理

主問題が最適解を持てば双対問題も最適解を持つ。それぞれの最適解を

主問題	X, X_2, \dots, X_n
双対問題	Y_1, Y_2, \dots, Y_m

とすると,

$$c_1 X_1 + c_2 X_2 + \dots + c_n X_n = b_1 Y_1 + b_2 Y_2 + \dots + b_m Y_m \tag{13}$$

が成立する。また Y_1, Y_2, \dots, Y_m は主問題の制約条件のシャドウプライスになっている。いいかえれば双対問題を解けばシャドウプライスは一気に求められる。

6. NUOPT による計算の基本

ここでは最初に説明した 2 変数の線形計画法モデル LP1 を NUOPT によって解く場合の基本的手順を示す。以下に示す手順はもっともシンプルな利用法であってこれにさまざまな手順を追加すると順次高度な利用法となる。

NUOPT のルールに従い上記のミニモデルを作成する。以下にその基本ルールを示す。

(1) 適当なテキストエディタでファイルを作成する。

拡張子は「.smp」とする。ここでは LP1.smp とする。

(2) NUOPT の基本ルール

① 必要に応じてコメントを書く。

```
//簡単な LP の例  工場の最適化
```

② 変数名を指定する。

```
Variable X1 (name="製品 P1 の生産量");
```

この命令は、「変数名は X1 である」ことを指名しているが、このほかに日本語の名前「製品 P1 の生産量 t/日」もつぎの 7.2 で説明する Excel へのデータ転送のための変数名として使用される。コメントを除くすべての命令の後には「;」をつける。

③ 目的関数を指定する。

```
Objective cost (name="全運転コスト", type=maximize);
cost=5*X1+8*X2;
```

「type=maximize」は最大化を表している。最小化するには「type=minimize」と書く

④ 制約条件を記述する。ここでは使用していないが、等号制約条件は「=」で表現する。不等号の表現は「<=」「>=」の 2 つだけである。

```
3*X1+4*X2<=440;    //原材料 M1 の制約 t/日
```

⑤ 変数に対する制約を記述する。上限と下限を一行で指定することができる。

```
0<=X1;            //非負制約
0<=X1<=500;     //変数制約
```

⑥ 線形計画法を実行する。

```
solve ();
```

⑦ 出力する変数を指定する。

```
X1.val.print (); //X1 の最適解の出力
cost.val.print (); //最適値の出力
```

(3) 作成したモデル

上記のルールによってモデルを記述するとつぎのようになる。なお、Variable X1 等における“ ”の中には Windows のファイル名として許されない文字を使用してはいけない。これは、“製品 P1 の生産量 t”等が csv ファイルのファイル名となるためである。使用できない文字としては、「¥/: *?<> |」等がある。

```
//簡単な LP の例  工場の最適化
//単位はすべて 1 日当たり
Variable X1 (name="製品 P1 の生産量"); //t /日
Variable X2 (name="製品 P2 の生産量"); //t /日
```

```

//運転コスト (目的関数)
Objective cost (name="全運転コスト", type=maximize);
cost=5*X1+8*X2;

//原材料制約
3*X1+4*X2<=440; //原材料 M1 の制約 t/日
3*X1+1*X2<=270; //原材料 M2 の制約 t/日
2*X1+8*X2<=720; //原材料 M3 の制約 t/日

//非負制約
0<=X1;
0<=X2;

//求解
solve ();

//結果出力
X1.val.print ();
X2.val.print ();
cost.val.print ();

```

(4) NUOPT による実行

① まず「スタート」→「NUOPT」→「NUOPT GUI」により NUOPT を起動する。ここではインストールの方法の説明は省略する。

NUOPT の画面は図 6 に示すように 3 つの領域から構成されている。

② モデルを NUOPT に移動する。作成したモデルファイル LP1.smp をプロジェクトボードにドラッグすればよい。

③ 計算結果の表示を行う準備をする。まず「オブジェクトブラウザ」より「表示」の中の「表示」アイコンをプロジェクトボードにドラッグする。つぎにマウスポインタを始点となる「モデルアイコン」上に移動する。「モデルアイコン」を右クリックしたまま左クリックしそのまま作成された「表示アイコン」へドラッグする。これでモデルの出力を表示するアイコンができ、図 6 のように 2 つのアイコンが線が結ばれる。

④ 「モデルアイコン」をダブルクリックすると線形計画法が実行される。図 7 に示す実行状態 (status) が表示される。

⑤ 計算結果の表示

「表示」の上部にある「status」の右にある ▼ をクリックし「solfile」をクリックするとつぎのような計算結果が表示される。

「LP1」の実行が開始.....

```
C : ¥WINDOWS¥system32>echo off
```

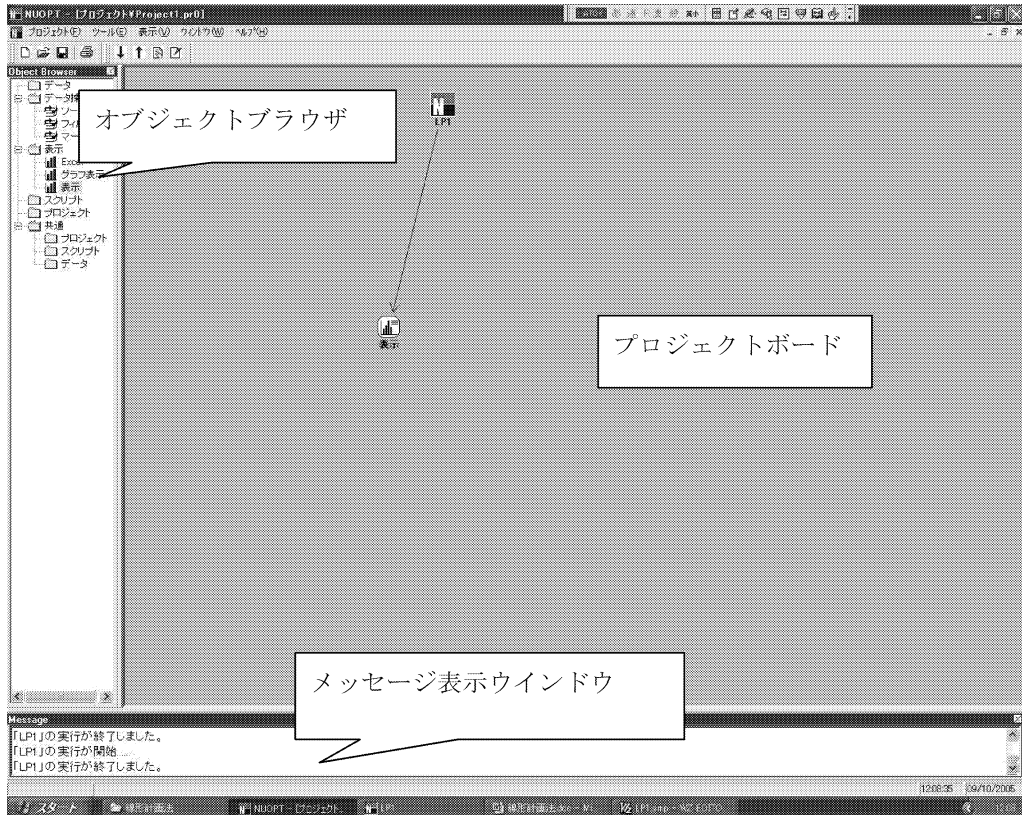


図6 NUOPT の画面

mknuopt for Windows (Ver. 7.0.0), Copyright (C) 2005 Mathematical Systems Inc.

1 個のファイルをコピーしました。

mknuopt: LP1.smp からロードモジュール LP1.exe を作成します。

genClass for Windows (Ver. 7.0.0), Copyright (C) 2005 Mathematical Systems Inc.

LP1.cc

Model description: LP1.smp

LP1.cc

LP1.cc

LP1Control.cc

mknuopt: LP1.exe を作成しました。

展開中 目的関数 (1/6 LP1.smp:8 name="全運転コスト")

展開中 制約式 (2/6 LP1.smp:11)

展開中 制約式 (3/6 LP1.smp:12)

展開中 制約式 (4/6 LP1.smp:13)

展開中 制約式 (5/6 LP1.smp:16)

展開中 制約式 (6/6 LP1.smp:17)

NUOPT 7.0.9a (LP/IP module), Copyright (C) 1991-2005 Mathematical Systems Inc.

	NUOPT状態名	NUOPT状態値
1	バージョン	7.0.9a
2	ステータス	最適化正常終了
3	問題名	LP1
4	変数の数	2
5	制約式の数	3
6	目的	最大化
7	アルゴリズム	高次内点法
8	問題種別	線形計画
9	目的関数値	839.9999993
10	内点法反復	6
11	行列分解回数	7
12	最適性ノルム	4.751645872e-008
13	経過時間(秒)	0.03

図7 実行状態の表示 (NUOPT)

```

PROBLEM_NAME                                LP1
NUMBER_OF_VARIABLES                          2
NUMBER_OF_FUNCTIONS                           4
PROBLEM_TYPE                                 MAXIMIZATION
METHOD                                         HIGHER_ORDER
<preprocess begin> ..... <preprocess end>
<iteration begin>
    res=2.1e+001 ..... 6.3e-005    4.8e-008
<iteration end>
STATUS                                         OPTIMAL
VALUE_OF_OBJECTIVE                            839.9999993
ITERATION_COUNT                               6
FUNC_EVAL_COUNT                               9
FACTORIZATION_COUNT                           7
RESIDUAL                                       4.751645872e-008
ELAPSED_TIME (sec.)                           0.02
SOLUTION_FILE
C:\PROGRA~1\NUOPT\Users\TAKESHI\temp\rad58AA1.tmp\solfile.txt
    
```

製品 P1 の生産量=40

製品 P2 の生産量=80

全運転コスト=840

「LP1」の実行が終了しました。

(5) 計算結果の保存

NUOPT を終了する際にプロジェクト名を指定すればよい。

7. NUOPT による処理の効率化

7.1 パラメータの外部化

つぎのように Parameter コマンドをモデル内に記述することによりパラメータの外部化を行うことができる。外部ファイル（テキストファイル）は拡張子「.dat」で保存し、プロジェクトボードにドラッグし、モデルにリンクする。

```
Parameter c1 (name="製品 P1 の生産コスト"); // 万円/t
Parameter c2 (name="製品 P2 の生産コスト"); // 万円/t
```

最初のモデル LP1 の製品コストと原材料制約のパラメータデータを外部化したモデルはつぎのようになる。

モデル名 LP2.smp

```
//簡単な LP の例 工場の最適化
Variable X1 (name="製品 P1 の生産量"); //t/日
Variable X2 (name="製品 P2 の生産量"); //t/日

Parameter c1 (name="製品 P1 の生産コスト"); // 万円/t
Parameter c2 (name="製品 P2 の生産コスト"); // 万円/t

//運転コスト (目的関数)
Objective cost (name="全運転コスト", type=maximize);
cost=c1*x1+c2*x2;

Parameter b1 (name="原材料 M1 の制約"); // t/日
Parameter b2 (name="原材料 M2 の制約"); // t/日
Parameter b3 (name="原材料 M3 の制約"); // t/日

//製品ノルマ
3*X1+4*X2<=b1; //原材料 M1 の制約
```

```

3*X1+1*X2<=b2; //原材料 M2 の制約
2*X1+8*X2<=b3; //原材料 M3 の制約

//非負制約
0<=X1; //非負制約
0<=X2; //非負制約

//求解
solve ();

//結果出力
X1.val.print ();
X2.val.print ();
cost.val.print ();

```

これにつぎのデータをリンクさせる。“”ではさまれた文字列はモデル内の文字列と完全に一致していなければならない。

データ paramLP2.dat

```

”製品 P1 の生産コスト”=5;
”製品 P2 の生産コスト”=8;
”原材料 M1 の制約”=440;
”原材料 M2 の制約”=270;
”原材料 M3 の制約”=720;

```

7.2 添字の導入

パラメータの外部ファイル化に加えて添字を導入すれば変数やパラメータの記述が楽になる。「i」という添字を導入するには次のように記述する。この添字は変数 X1, X2 のかわりに変数 X[1], X[2] を使用することができるにするためである。同じように複数のパラメータに関して添字を使うことができる。2次元, 3次元の変数も導入することができる。

```

//添字の定義
Set Product (name ”製品集合”);
Product=”1 2”;
Element i (set=Product);

```

以上の知識をベースにモデルを記述すると次のようになる。

LP3.smp 添字の導入とパラメータの外部化をしたモデル

```

//簡単な LP の例 工場の最適化

```

```

//添字の定義
Set Product (name="製品集合");
Product="1 2";
Element i (set=Product);

Set Materials (name="原材料集合");
Materials="1 2 3";
Element j (set=Materials);

Parameter c (name="製品 Pi の生産コスト", index=i);
Parameter A (name="係数", index=(j,i));
Parameter b (name="原材料 Mj の制約", index=j);

Variable X (name="製品 Pi の生産量", index=i);

//運転コスト (目的関数)
Objective cost (name="全運転コスト", type=maximize);
cost=c[1]*X[1]+c[2]*X[2];

//原材料制約
A[j,1]*X[1]+A[j,2]*X[2]<=b[j]; //原材料 Mj の制約

//非負制約
0<=X[i]; //非負制約

//求解
solve ();

//結果出力
X[i].val.print ();
cost.val.print ();

```

これにリンクする外部データファイルは次のようになる。

外部データファイル paramLP3.dat

```

"製品 Pi の生産コスト"=[1] 5 [2] 8;
"原材料 Mj の制約"=[1] 440 [2] 270 [3] 720;
"係数"=[1,1] 3 [1,2] 4 [2,1] 3 [2,2] 1 [3,1] 2 [3,3] 8;

```

これらのファイルはつぎのようにプロジェクトボードでリンクする (右半分)。プロジェクトボード上の LP3 をダブルクリックするとモデルが実行され解が求まる。

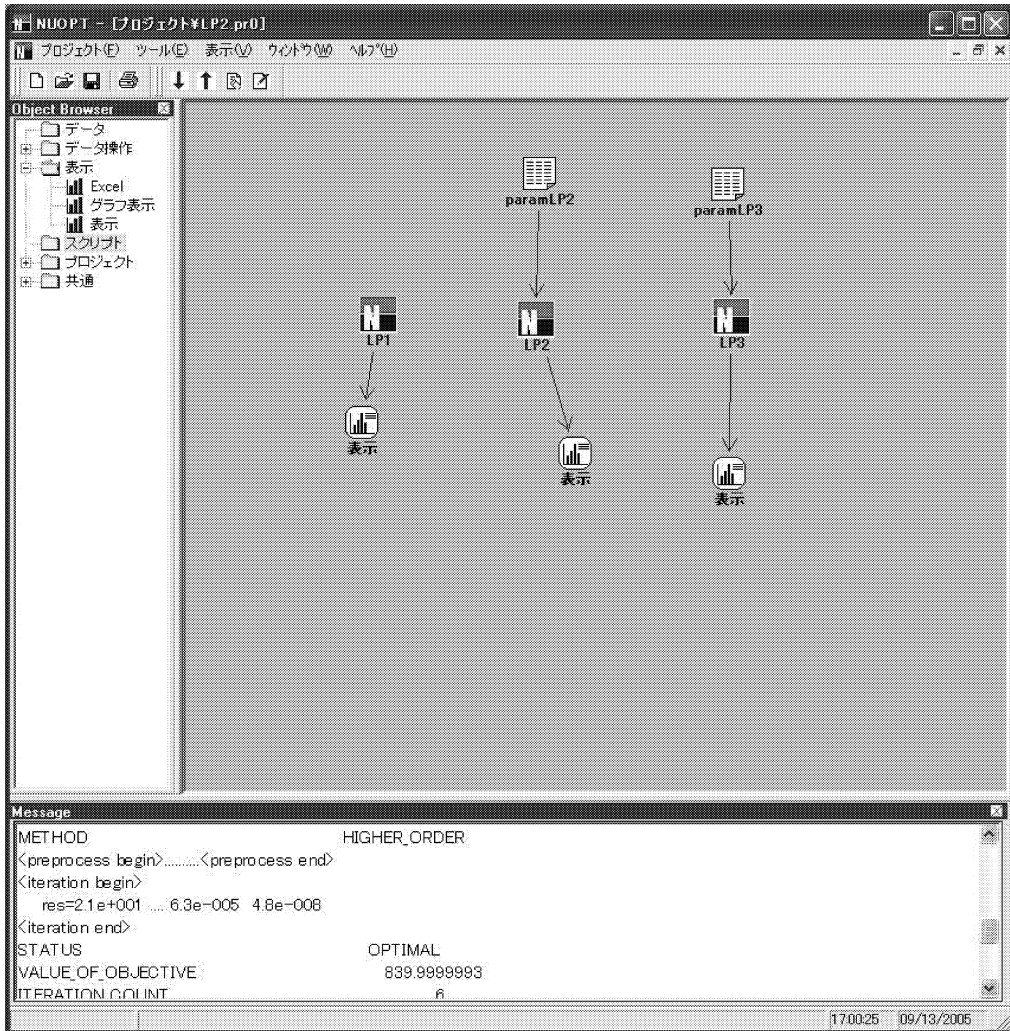


図 8 3つのモデル (NUOPT)

7.3 Excel とのリンク

以上の方法では、モデルのパラメータを指定する外部ファイルはテキストファイルを使用しなければならない。しかしテキストファイルでなく Excel を使うことができれば、データファイルに対する事前のデータ処理、計算結果に対するデータ処理ができるので非常に好都合である。以下では説明の簡略化のために「NUOPY の計算結果→ Excel」の転送方法について説明しよう。Excel → NUOPT については省略する。

事前に「スタート」より「NUOPT」の中の「Excel アドインのインストール」(その説明は NUOPT のマニュアル参照) を実行しておく。これにより Excel に NUOPT のボタンができる。これを利用して、Excel とのリンクや実行を行うことができる。

(1) モデルの修正

モデルの出力を Excel に渡すための命令をモデルに組み込む必要がある。Solve () のあとにつぎの

ような命令をつける。最後の3行がその命令である。

```
(これ以前は省略)
//求解
solve ();

//結果出力
X1.val.print ();
X2.val.print ();
cost.val.print ();

x1.val.dump ();
x2.val.dump ();
cost.val.dump ();
```

(2) 利用するモデルの指示

どのモデルの出力を Excel に出力するかを指定する。モデル (たとえば, LPI) を右クリックし、「Excel 連係の実行対象」をクリックする。V のマークがつく。

(3) Excel を起動する。

まず最初に Excel ファイルに名前をつけ保存しておく。保存しておかないと以下の作業が実行されない。

(4) Excel のひとつのシートに X1, X2, cost に対する日本語の変数名を書く。場所は自由である。たとえば次のように行う。変数名の右側の場所はデータが転送される場所であり、書式を指定しておくともデータが指数形式にならないので見やすい。

製品 p1 の生産量	
製品 p2 の生産量	
全運転コスト	

(5) Excel の「NUOPT」のなかで「選択範囲を名前として使用 (本シート)」をクリックする。

(6) リンク作業

(4) で作成した 3 行 2 列の範囲をドラッグし、Excel の「NUOPT」のなかで「選択範囲と NUOPT」のリンクをクリックする。するとつぎの画面が表示されるので「取り込み」をクリックする。

(7) 「対応されていない範囲」の中に図 9 のようにリンクする範囲が表示される。ここでそれをクリックした後、その下の画面より、「Excel ← NUOPT」となるように転送方法を選び、かつ「次元」は 0 とし、「行」、「先頭行・列をオブジェクトとして使用」をクリックする。この後に上部の 2 つの窓の間にある「→」をクリックし、「ok」を押す。

(8) 実行

Excel の「NUOPT」の中の「実行」をクリックする。これにより計算結果が Excel に転送される。



図9 データの Excel へのリンク (NUOPT)

(9) その他

以上ではスカラー変数のデータ転送を説明したが、複数の多次元変数の計算結果も転送可能である。詳細は NUOPT/SIMPLE/Excel リファレンスマニュアル参照。

8. NUOPT による実際的なモデルの定式化

ここでは電源計画モデルを取り上げ、それを NUOPT で定式化した。本モデルは NUOPT 使用法の説明用モデルであり、シミュレーション結果については[3]で説明する。以下に示すモデルのパラメータやデータはかなり実際のデータに近い架空のデータである。現実性を持たせるためにその設定の理由を示しておく。以下の電源計画モデルの基本的アイデアは筆者が考えたものである [5][8]。

まず図 10 に 1 年間 8,760 時間 (365 日×24 時間) の電力需要 (発電端) を示す。これは同時に設備合計の運転出力でもある。

この需要に対して揚水、石油火力、LNG 火力、石炭火力、原子力、一般水力の運転出力を当てはめるわけであるが、電力需要を大きい順番に並べ替えたつぎの図 11 のような負荷持続曲線の中に当てはめた方がわかりやすい。このような考え方は古くから電力関係では広く利用されている。

線形計画法でモデル化するためにこの負荷持続曲線を図 12 のように近似する。時間帯はつぎのように分ける。h () はモデル内の記号。

第 1 時間帯 1-500

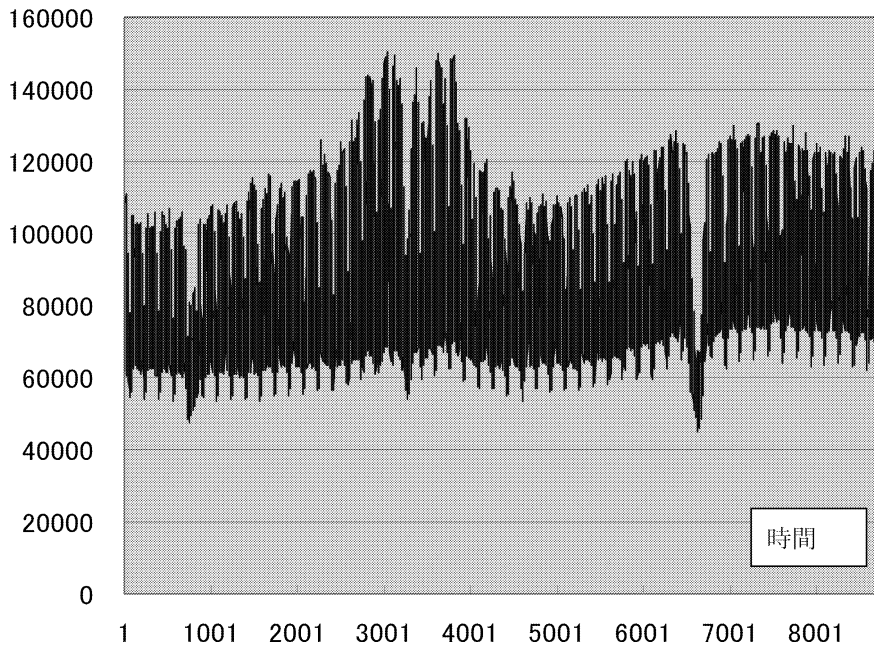


図 10 年間の電力需要 (発電端・架空データ・単位 1,000 kW)

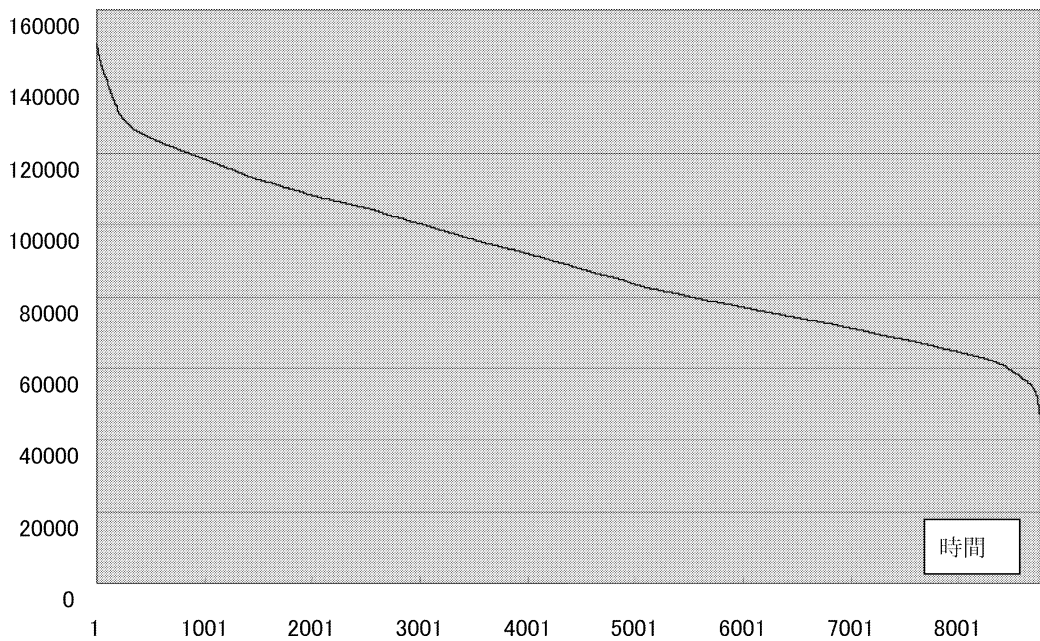


図 11 負荷持続曲線 (架空データ・単位 1,000 kW)

- 第 2 時間帯 501-2,000
- 第 3 時間帯 2,001-4,000
- 第 4 時間帯 4,001-6,000
- 第 5 時間帯 6,001-8,260

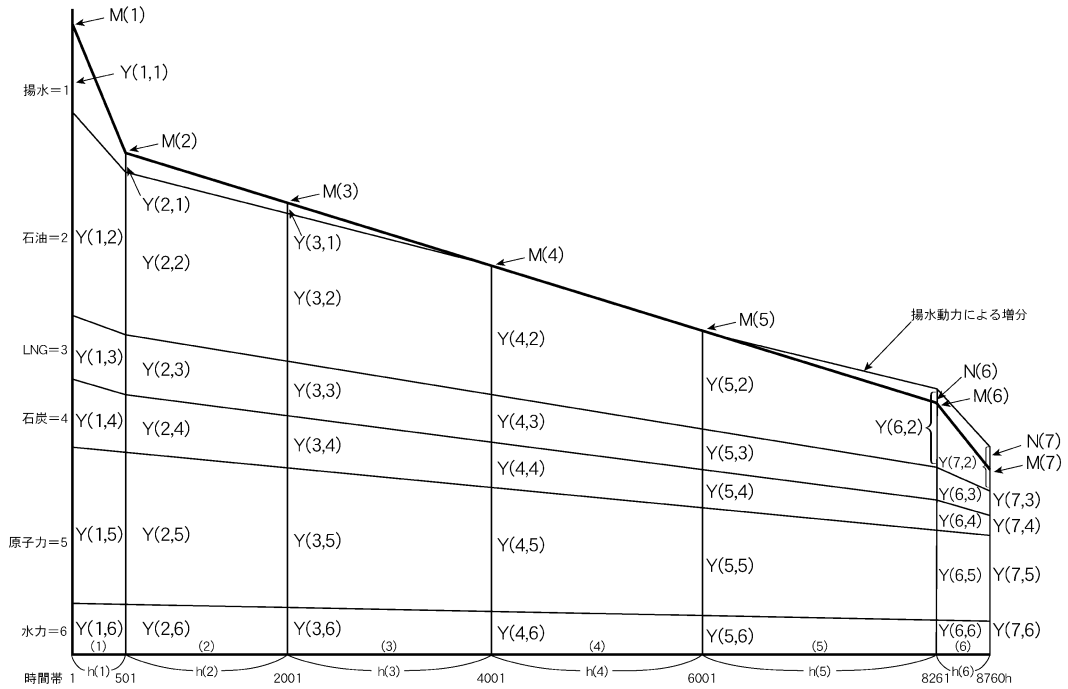


図 12 近似された負荷持続曲線

第 6 時間帯 8,261-8,760

以上の準備のもとで例題としての電源構成モデルをつぎのように定式化する。このモデルは特定の 1 年間だけの最適化を行っている。当然のことながら、現実的なモデルでは 30-40 年間程度を対象期間とする必要がある。もちろん、1 年間の需要の増加と設備の増加は限られているので、1 年ごとにモデルを定式化する必要はなく、標準的には 5 年ごとに定式化すれば十分であろう。

電源構成モデル

```
//最適電源構成モデル
//1 時点モデル (容易に他時点最適化モデルへ拡張可能)
//添字の定義

Set Timex (name="時点");
Timex="1 2 3 4 5 6 7";
Element i (set=Timex);

Set Typex (name="電源種別");
Typex="1 2 3 4 5 6";
Element j (set=Typex);

//パラメータ
Parameter M (name="電力需要", index=i);
```

```
Parameter h (name="時間帯", index=i);
Parameter Z (name="既設設備容量", index=j);
Parameter ZU (name="設備容量上限", index=j);
Parameter a (name="利用可能率", index=j);
Parameter b (name="年経費率", index=j);
Parameter c (name="建設単価", index=j);
Parameter p (name="燃料価格", index=j);
Parameter d (name="供給予備力");
Parameter m (name="発電効率", index=j);
Parameter l (name="設備利用率上限", index=j);

//変数
Variable Y (name="運転出力", index= (i,j));
Variable X (name="新設設備容量", index=j);
Variable R (name="燃料消費量", index=j);
Variable N (name="揚水需要", index=i);

//電力需要充足式
M[i]+N[i]==sum (Y[i,j],j);
Y["4,1"]==0.0;
Y["5,1"]==0.0;
Y["6,1"]==0.0;
Y["7,1"]==0.0;
N[1]==0.0;
N[2]==0.0;
N[3]==0.0;
N[4]==0.0;
N[5]==0.0;

//ピーク時における設備利用可能率
Y[1,j]<=a[j]*(X[j]+Z[j]);

//ピーク時における予備力
sum (a[j]*(X[j]+Z[j]), j)>=(1+d)*M[1];

//運転出力相互関係
Y[1,j]>=Y[2,j];
Y[2,j]>=Y[3,j];
Y[3,j]>=Y[4,j];
Y[4,j]>=Y[5,j];
```

```

Y[5,j] >= Y[6,j];
Y[6,j] >= Y[7,j];

//揚水用動力
0.65*(N[6]*h[5] + (N[6]+N[7])*h[6])
== (Y["1,1"] + Y["2,1"])*h[1] + (Y["2,1"] + Y["3,1"])*h[2] + Y["3,1"]*h[3];

//設備建設上限
X[j] + Z[j] <= ZU[j];

//燃料消費量 (単位 103kl)
R[1] == 0.0; //揚水
R[2] == 0.5*sum ((Y[i+1,2] + Y[i,2])*h[i], (i,i < 7)) * m[2]; //石油火力
R[3] == 0.5*sum ((Y[i+1,3] + Y[i,3])*h[i], (i,i < 7)) * m[3]; //LNG 火力
R[4] == 0.5*sum ((Y[i+1,4] + Y[i,4])*h[i], (i,i < 7)) * m[4]; //石炭火力
R[5] == 0.5*sum ((Y[i+1,5] + Y[i,5])*h[i], (i,i < 7)) * m[5]; //原子力
R[6] == 0.0; //一般水力

//設備利用率上限
sum ((Y[i,j] + Y[i+1,j])*h[i] * 0.5, (i,i < 7)) <= 8.76*(X[j] + Z[j])*I[j];

//運転コスト (目的関数)
Objective cost (name="全コスト", type=minimize);
cost = sum (c[j]*X[j]*b[j],j) + sum (R[j]*p[j],j);

//非負条件
0.0 <= Y[i,j];
0.0 <= X[j];
0.0 <= R[j];
0.0 <= N[i];

//求解
solve ();

//結果出力
Y[i,j].val.print ();
X[j].val.print ();
R[j].val.print ();
N[i].val.print ();
cost.val.print ();

```

```
//Excel への出力
//Y[i,j].val.dump () ;;
//X[j].val.dump () ;
//R[j].val.dump () ;
//N[i].val.dump () ;
//cost.val.dump () ;
```

モデルパラメータ (外生変数) はつぎのように設定した。これは上記の架空のデータ図 11 から作成した。

① 電力需要

図 12 に対応して発電端時間帯別の電力需要 (1,000 kW) をつぎのように与える。時点とは図 12 の時間帯を区切る時点のことである。

第 1 時点 (i=1, 1 番目の需要)	150,474 千 kW
第 2 時点 (i=2, 501 番目の需要)	124,282 千 kW
第 3 時点 (i=3, 2001 番目の需要)	108,259 千 kW
第 4 時点 (i=4, 4001 番目の需要)	91,950 千 kW
第 5 時点 (i=5, 6001 番目の需要)	77,045 千 kW
第 6 時点 (i=6, 8261 番目の需要)	62,629 千 kW
第 7 時点 (i=7, 8760 番目の需要)	44,897 千 kW

② 既設設備容量 (Z[]) はすべて 0 とする。つまり既設の設備に依存しない理想的な電源構成を考える。

③ 設備容量上限 (UZ[]) は設定しない。この意味でも理想的な設備容量を考える。数値としては非常に大きい数値を入れておく (1,000,000)。

④ ピーク時における設備の利用可能率 (a []) はつぎのように与える。

[1] 揚水	1.0
[2] 石油火力	0.85
[3] LNG 火力	0.85
[4] 石炭火力	0.85
[5] 原子力	0.70
[6] 一般水力	0.50

⑤ 年経費率 (b[]) はつぎのように与える。

[1] 揚水	0.1335
[2] 石油火力	0.1807
[3] LNG 火力	0.1807
[4] 石炭火力	0.1807
[5] 原子力	0.1955
[6] 一般水力	0.1335

⑥ 建設単価 (c[]), 百万円/ 10^3 kW) はつぎのように与える。

[1] 揚水	9
[2] 石油火力	16
[3] LNG 火力	18
[4] 石炭火力	20
[5] 原子力	30
[6] 一般水力	54

⑦ 電源別燃料価格 (P[], 単位百万円/1,000 kl [石油換算], 原子力は百万円/百万 kWh) はつぎのように与える。

[1] 揚水	0
[2] 石油火力	25
[3] LNG 火力	30
[4] 石炭火力	20
[5] 原子力	1
[6] 一般水力	0

⑧ 供給予備力 (d) は 8% とする。

⑨ 発電効率 (m[]) はつぎのように与える。

[1] 揚水	0.0
[2] 石油火力	$0.086/0.38/0.975=0.232$
[3] LNG 火力	$0.086/0.38/1.33=0.170$
[4] 石炭火力	$0.086/0.38/0.55=0.411$
[5] 原子力	1.0 (核燃料費は kWh で表現する)
[6] 一般水力	0.0

⑩ 各電源の設備利用率上限 (l[]) をつぎのように設定する。

[1] 揚水	0.15
[2] 石油火力	0.80
[3] LNG 火力	0.80
[4] 石炭火力	0.80
[5] 原子力	0.80
[6] 一般水力	0.45

これに対応するパラメータデータファイルは省略する。このモデルは実際に解くことができる。設備容量上限などいくつかのパラメータをより現実的なものにすればさまざまなシミュレーションを行うことができる。たとえば、今後電力需要の伸びは比較的小さいと見られているが、このような中で原油価格の高騰がどのような影響を与えるかというテーマはひとつの興味あるテーマである。すでに述べたようにそのシミュレーション結果とその解釈についてつぎの報告[3]で説明する予定である。

NUOPT による線形計画法モデルを作成するに当たり、当然のことながらさまざまな初歩的なエラーが発生したが、数理計画 (株) 高橋良徳氏および同社の方々からさまざまな助言を得た。深く感謝したい。NUOPT のルールにはやや特殊なものもあるが、全体とすれば大きなトラブルはなかった。

9. あとがき

本資料では、平成 13 年度情報科学研究所共同研究 (大規模モデルの挙動特性とその役割に関する研究, 代表齋藤雄志・共同研究者蔵下勝行) のために行った準備作業の一部を第三者の便宜を図るために資料として作成した。さまざまな事情で報告が遅れたが、今後、2, 3 件の資料あるいは論文を発表していきたい。第一は 7. に示した電源構成モデルのシミュレーションの結果の報告である。これは、「大規模モデルの挙動特性」を示す例として利用するためのモデルである。大規模モデルは意外に反応が鈍いことがあることを論じたいと考えている。第 2 は世の中で開発されたモデルについてこのような観点から調査を行うことである。エネルギー分野は最も最適化モデルが活用されている分野のひとつであるが、この報告では、エネルギー・資源学会の 21 年間におよぶ「エネルギーシステム・経済・環境コンファレンス」から多くの最適化モデルを取り上げる予定である。

参考文献

- [1] 今野浩: 最適化の時代, <http://www.msi.co.jp/splus/support/salon/userconf/ronbun/4th/konno2004.pdf>.
- [2] 岩間一雄: 新世代の計算限界—その解明と打破—, 科学研究費補助金「特定領域研究」平成 16 年度発足特定領域申請書, <http://keisan-genkai.lab2.kuis.kyoto-u.ac.jp/reports/2003/shinseisho-part1.pdf>.
- [3] 齋藤雄志: NUOPT による最適化モデルの開発 (2), 情報科学研究所所報 (投稿予定).
- [4] 齋藤雄志: さまざまな最適化モデルとその特性, 情報科学 (投稿予定).
- [5] 尾崎敏・黒田昌裕・吉岡完治・桜本光・赤林由雄・大沢悦治・齋藤雄志・阿波田禾積・中村二郎・井澤祐司・伊藤浩吉・木村繁: KEO—電研モデルの構成—経済・エネルギーの相互依存関係の分析—, 電力中央研究所報告 研究報告 583008, 昭和 59 年 4 月.
- [6] 宮川公男: OR 入門, 日経文庫, 日経新聞社, 1992.
- [7] 小山昭雄: 線型計画入門, 日経文庫, 昭和 45 年.
- [8] 齋藤雄志・大庭靖男・七原俊也・伊藤浩吉: 電力需要と電源構成, 電力経済研究, No. 18, pp. 117-141, 1985.
- [9] 日本エネルギー経済研究所計量分析ユニット編: エネルギー・経済統計要覧, (財) 省エネルギーセンター.
- [10] 電気事業連合解統計委員会編: 電気事業便覧, 日本電気協会.
- [11] 資源エネルギー庁長官官房総合政策課編: 総合エネルギー統計, (株) 通商産業研究社.