意志決定支援のための表計算ソフトウェアの利用

税所幹幸　　津曲　隆　　市村憲治

The Applications of Spread-sheet Software to Mathematical Programming

Motoyuki SAISHO　Takashi TSUMAGARI　Kenji ICHIMURA

表計算ソフトウェア **Lotus 1-2-3** は，データ処理や分析のための多くのツールをもっている。それらの中に，逆解析や最適化問題などへ応用できる「バックソルバー」および「ソルバー」と呼ばれている強力なツールがある。本稿では，これらの機能，利用方法，処理能力について紹介・解説する。バックソルバーについては，１変数関数の逆問題を解かせた例，ソルバーについては，多変数方程式の解法や最適化問題などに使用した例と結果を示す。

１．まえがき

1-2-3 は，コンピュータリテラシー教育に多く使われ，皆になじみの深いものである。このソフトウェアは，ただ単に表計算機能を持っているだけでなく，グラフ機能，データベース機能，マクロ機能を持ち，さらにいくつかの分析機能をも持っている。その分析機能のうちに，「バックソルバー」および「ソルバー」と呼ばれる逆問題を解くための分析ツールがある。バックソルバーが１変数の問題で１つの解を求める場合に使用するのに対し，ソルバーは多変数の方程式を解く場合や数理計画法における複数の解を求める場合に使用する。

本稿では，２章で比較的簡単な関係のある資金計画問題を例にしてバックソ

ルバーの利用方法を説明する。３章では，多変数問題と数理計画法について，その概要を簡単に解説する。そして，そこで述べた例題を使ってソルバーの具体的な利用方法を紹介している。４章では，多変数の非線形方程式の解を求める問題，数理計画法のうちの線形計画問題と非線形計画問題にソルバーを適用し，その処理能力等について実験調査した結果を述べる。その調査から，ソルバーは，線形・非線形計画問題とも問題の規模がさほど大きくなければ，現状レベルのパソコン性能でも使用に耐え得る程度の処理能力を有していることを示す。

社会には，多くの制約条件のもとで最適解を求める必要のある問題に多く遭遇する。それらの問題の解決のために，**1-2-3** の分析機能のうちのバックソルバーとソルバーとを使うことは有効である。

２．バックソルバーの機能と使用例

２．１　バックソルバーの機能

バックソルバーは結果から逆算して解を求めてくれる分析ツールである。一般に結果 y を求める式が $y=f(x)$ と表せた場合，y から x を求めるには，逆関数 $x=g(y)$ を導出すればよい。この逆関数が求まれば大抵は電卓等で計算することもできるが，計算式の中には逆関数が導出できないものもあり，そういった場合にはコンピュータを使ってニュートン法等による数値解法をとらざるを得ない。バックソルバーでは，逆関数を導出できない場合にも結果から解を求めることができる。

２．２　資金計画への適用

バックソルバーの使用例として住宅資金を２つの金融機関（A，B）から借りる場合を考える。[例題１] は求めたい変数に対する逆関数を導出できる場

合，［例題２］は逆関数を導出できない場合の例を取りあげる。

２．２．１　住宅資金の返済問題Ｉ

［例題１］

Ａ金融機関からの借入限度額は1000万円で，年率は３％，返済期間は30年となっている。ここで総返済月額を10万円とした場合，Ｂ金融機関からいくら借りることができるか。Ｂ金融機関の年率は５％で，返済期間は30年とする。

（１）問題の定式化

金融機関から住宅資金を借りる場合の返済方法は，元利均等返済と元金均等返済とがある。ここでは元利均等返済を考える。このとき，mを借入金額，nを返済期間，Rを年率（小数値とする）とすると返済月額kは次式で表せる。

$$k = m \cdot \frac{r}{1-(r+1)^{-12n}} \tag{1}$$

ここで，$r=R/12$.

上式中rは月率である。したがって，［例題１］における総返済月額Kは次式で表せる。

$$\begin{aligned} K &= k_{\mathrm{A}} + k_{\mathrm{B}} \\ &= m_{\mathrm{A}} \cdot \frac{r_{\mathrm{A}}}{1-(r_{\mathrm{A}}+1)^{-12n_{\mathrm{A}}}} + m_{\mathrm{B}} \cdot \frac{r_{\mathrm{B}}}{1-(r_{\mathrm{B}}+1)^{-12n_{\mathrm{B}}}} \end{aligned} \tag{2}$$

ただし，添え字Ａ，Ｂはそれぞれの金融機関の意味である。バックソルバーを［例題１］に適用して，その具体的なデータ設定方法や利用手順などを以下で解説する。なお，この問題は，求める変数の値がＢ金融機関からの借入額m_{B}であるから，式(2)を変形すれば，その値は簡単に求めることができ，

わざわざバックソルバーを使用するまでもない。ただし，こういった簡単な例でもバックソルバーを使えば式を変形することなく原式から直接的に解が得られるため，手間が省けて便利である。

（２）データの設定

1-2-3 を起動すると，ワークシートＡが現れる。ここで［例題１］でのＡ，Ｂ金融機関ごとの借入金額，年率および返済期間の各値と，月率と返済月額の式等をそれぞれ各セルに設定する。各セルの位置は自由に選んでよい。ここでは，図１のように記入した。

A	A	B	C
1		Ａ金融機関	Ｂ金融機関
2	借入金額	10000000	
3	年　率	0.03	0.05
4	月　率	+B3/12	+C3/12
5	返済期間	30	30
6	返済月額	+B2*B4/(1-(B4+1)^(-12*B5))	+C2*C4/(1-(C4+1)^(-12*C5))
7			
8	総借入金額	+B2+C2	
9	総返済月額	+B6+C6	
10			

図１．［例題１］に対するデータの設定

セルB2：Ａ金融機関からの借入額1000万円を設定する。

セルB3：Ａ金融機関の年率0.03を設定する。

セルB4：Ａ金融機関の月率を計算する式（+B3/12）を設定する。

セルB5：Ａ金融機関への返済期間30年を設定する。

セルB6：Ａ金融機関への返済月額の式を設定する。セルの内容は次のようになる。

+B2*B4/(1-(B4+1)^(-12*B5))

セルC3：Ｂ金融機関の年率0.05を設定する。

セルC4：Ｂ金融機関の月率を計算する式（+C3/12）を設定する。

セルC5：Ｂ金融機関への返済期間30年を設定する。

セルC6：Ｂ金融機関への返済月額の式を設定する。セルの内容は次のようになる。

+C2*C4/(1-(C4+1)^(-12*C5))

セルB8：総借入金額（Ａ，Ｂ金融機関の借入金額の和）を設定する。

+B2+C2

セルB9：総返済月額（Ａ，Ｂ金融機関への返済月額の和）を設定する。

+B6+C6

求める解であるＢ金融機関からの借入金額はC2セルに表示される。

（３）バックソルバーの起動と設定

各値と計算式とをセルに設定後，バックソルバーを起動する。マウスで〈範囲〉→〈分析〉→〈バックソルバー〉を選択すると，図２のダイアログボックスが開く。ここで入力セル，仮定セルおよび結果セルを指定する。この例題では，入力セルには逆算したい式のセル（総返済月額：B9セル）を指定する。仮定セルには逆算したい式の結果の値を入れたセルまたはその値（10万円）を設定する。結果セルには解を求めたい変数のセル（Ｂ金融機関からの借入額：C2セル）を指定する。

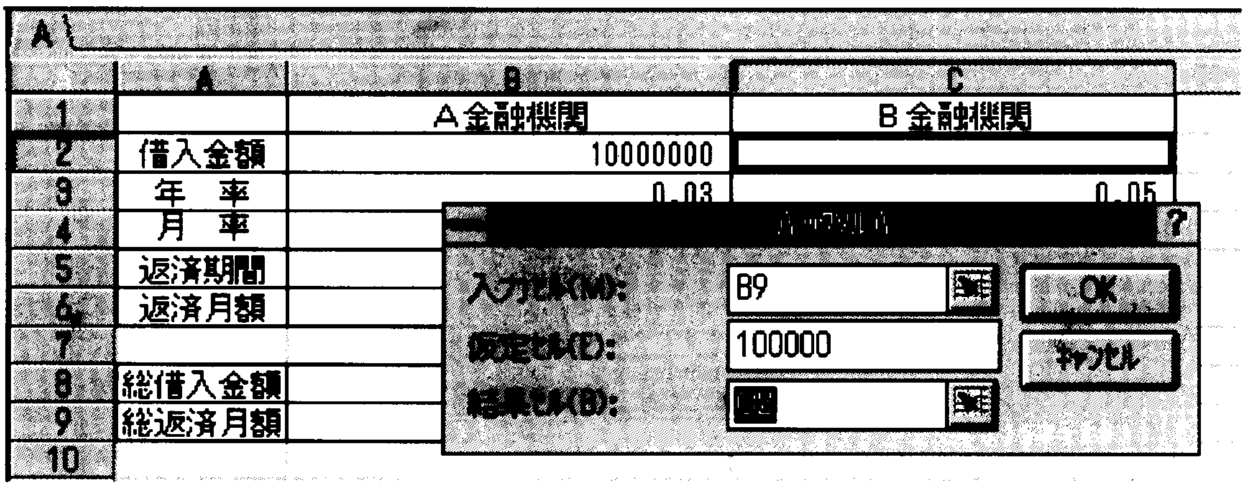

図２．［例題１］に対するバックソルバーの設定

(４)結　果

バックソルバーの設定終了後，図２のダイアログボックスの〈ＯＫ〉ボタンをクリックすると，図３のように結果が表示される。Ｂ金融機関からの借入金額は1077万円まで可能だということがわかる。この結果を吟味し，必要があれば仮定セルの値（総返済月額）を変更して，いろいろな場合を計算する。それらの結果をもとにして最良な場合を選択すればよい。返済期間についてはＡ，Ｂの金融機関とも同じ期間にしたが，この問題では異なる値を設定してもかまわない。

	A	B	C
1		A金融機関	B金融機関
2	借入金額	10000000	10774454
3	年　率	0.03	0.05
4	月　率	0.00250	0.00417
5	返済期間	30	30
6	返済月額	42160	57840
7			
8	総借入金額	20774454	
9	総返済月額	100000	
10			

図３．［例題１］に対するバックソルバーの解答

２．２．２　住宅資金の返済問題Ⅱ

［例題２］

> Ａ金融機関からの借入限度額は1000万円で，年率は３％である。またＢ金融機関からは年率５％で，500万円を借りるものとする。総返済月額を10万円とした場合，返済期間は何年になるか。

［例題２］では式⑵を利用する。この例題では，求める変数の値は両金融機関への返済期間 n である。式⑵からは初等関数の範囲内では逆関数を求められないが，バックソルバーでは式を設定すると式を解釈して解を計算してくれる。

(1)データの設定

図1で示した［例題1］と同じセルを利用する。ただし，B金融機関からの借入金額500万円をC2セルに設定する。

(2)バックソルバーの起動と設定

(1)の設定後，バックソルバーを起動する。バックソルバーのデータを図4のように設定する。入力セルに総返済月額のB9セル，仮定セルに総返済月額の10万円を設定する。結果セルには求める変数の返済期間B5とC5セルを設定する。

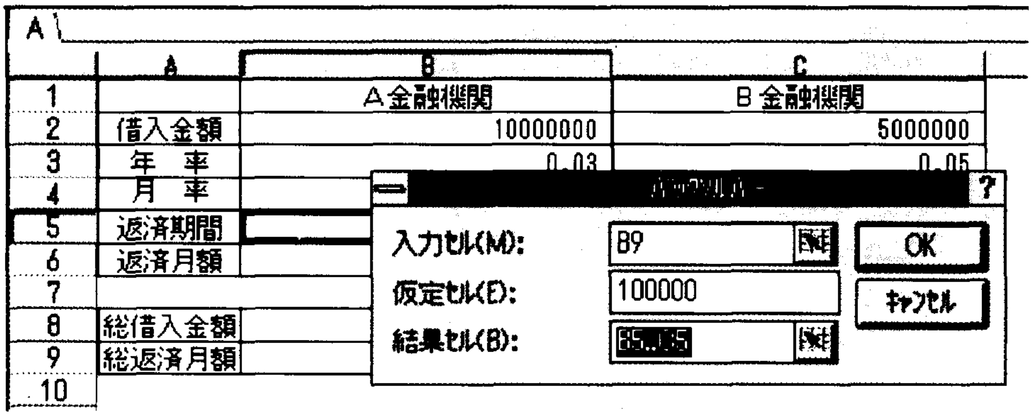

図4．［例題2］に対するバックソルバーの設定

(3)結　果

(2)を設定した後，実行すると図5の結果が得られる。返済期間は16.8年となったので，ほぼ17年かかることがわかる。総返済月額や借入金額等の値を変えてやると返済期間を瞬時に計算してくれる。それらの中から最良な解を選択すればよい。

	A	B	C
1		A金融機関	B金融機関
2	借入金額	10000000	5000000
3	年　率	0.03	0.05
4	月　率	0.00250	0.00417
5	返済期間	16.8	16.8
6	返済月額	63265	36735
7			
8	総借入金額	15000000	
9	総返済月額	100000	
10			

図5. [例題2] に対するバックソルバーの解答

2. 3 まとめ

本章では，バックソルバーの機能と2つの例題に対する操作法とを説明した。最後に，バックソルバーの特徴をまとめておく：

(1) 求める変数の解は1つである。

(2) 通常の問題では，実行時間は1つの解しか求めないので特に問題はない。

(3) 計算式の中には初等関数の範囲では逆関数を求められない問題に対しても，バックソルバーでは数値的に解を求めてくれる。

3. ソルバーとその機能

1-2-3 のソルバーは，連立方程式の解を求める場合，最適化問題の解を求める場合へ応用される。本章では，まず初めに，これら2つの問題について簡単に解説しておく。その後に，ソルバーの具体的な利用方法を述べる。

3. 1 連立方程式

連立線形方程式は，一般に次式で示される。

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases} \tag{3}$$

式(3)は，$\boldsymbol{A}$ を n 次の正則な正方行列，$\boldsymbol{x}$ と $\boldsymbol{b}$ をそれぞれ n 次の列ベクトルとすると次式で表せる。

$$\boldsymbol{Ax} = \boldsymbol{b} \tag{4}$$

このとき，解 $\boldsymbol{x}$ は

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b} \tag{5}$$

により求めることができる。ここで $\boldsymbol{A}^{-1}$ は，$\boldsymbol{A}$ の逆行列である。このように，式(3)の解は解析的に求めることができるが，$\boldsymbol{A}$ の各要素が変数 $x_i(i=1\sim n)$ の関数であると式(3)は非線形方程式となり，解析的に求めるのは困難である。

ソルバーを利用すると，線形および非線形の連立方程式のいずれにたいしても解を求めることができる。

3．2　数理計画法

数理計画法の中には多くの手法がある。すなわち，線形計画法，非線形計画法，確率的計画法，パラメトリック計画法，動的計画法などがあるが[(1)(2)]，ここでは線形計画法および非線形計画法についてその概要を述べる。

線形計画法とは，ある制約条件のもとで，いくつかの変数によって決まる目的関数の値を最大または最小にするような変数の値を求める方法である。このとき，制約条件，目的関数とも線形である。線形とは，ある関数 $g(x)$ で，

$$g(x_1+x_2) = g(x_1) + g(x_2)$$

の関係（重ね合わせの理）が成り立つ場合をいう。ここで線形計画法の説明でよく用いられる代表的な配分問題を考えてみる。

［例題３］

ある花屋にバラが100本，カーネーションが100本入荷した。バラが２本とカーネーション４本で花束「白亜」ができ，バラが３本とカーネーション１本で花束「紫苑」ができる。「白亜」では１束40円，「紫苑」では１束20円の利益があがる。「白亜」と「紫苑」を何束ずつ作ると利益が最大となるか。

「白亜」を x 束，「紫苑」を y 束作るとすると，花の本数の制限から次の制約条件を得る。

バラの本数の制限より

$$2x+3y \leqq 100 \tag{6}$$

カーネーションの本数の制限より

$$4x+y \leqq 100 \tag{7}$$

花束の数は正であるから

$$x \geqq 0 \tag{8}$$

$$y \geqq 0 \tag{9}$$

利益 r は次式で求めることができる。

$$r=40x+20y \tag{10}$$

$r=10R$ と置くと，次のように変換できる。

$$R=4x+2y \tag{11}$$

式(11)が目的関数であり，この R を最大にする x と y を求めるとよい。横軸に x，縦軸に y をとった $x-y$ 平面上に制約条件式(6)〜(9)を描くと図６を得る。この図において，制約条件を満足する x と y の値は四辺形 oacb の内部およびその

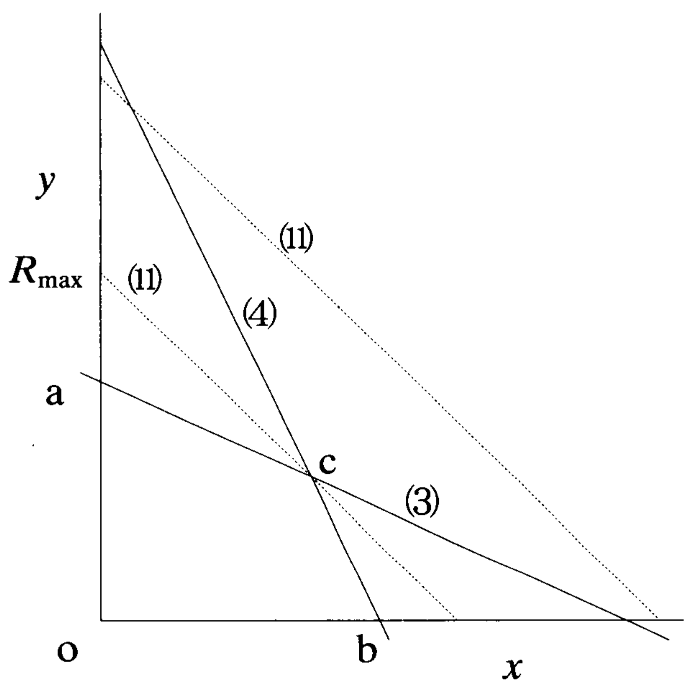

図６．制約条件を示したグラフ

線上になければならない。この四辺形の内部および線上を**実行可能領域**といい，そこに含まれる x と y の組を**実行可能解**という。式⑾の目的関数は，その傾き α（例題では－２）が一定で，利益 R により様々な位置をとる。図６では，２つの例が点線で示してある。式⑾から判るように，利益 R を示しているのは図６の点線の y 切片である。従って R を最大にするのは，図６の点線で表された式⑾が四辺形 oacb の内部または線上を通り，かつその y 切片が最大になるところ（R_{max}）である。利益を最大にする x と y は，点線の傾き α の大きさで変わってくる。式⑶⑷の直線の傾きをそれぞれ α_3，α_4 とし，各種のケースについて以下で説明する。

（１）$\alpha_3 < \alpha < \alpha_4$ の場合

この例題では，$\alpha=-2$，$\alpha_3=-2/3$，$\alpha_4=-4$ であるから，

$$\alpha_3 < \alpha < \alpha_4 \tag{12}$$

の条件を満足する。これが成り立つときは，点線の直線⑾が点 c を通るときに y 切片すなわち利益は最大となり，c 点の x と y の値が求める解となる。この例題では，$x=20$（束），$y=20$（束）である。

（２）$\alpha=\alpha_3$，$\alpha=\alpha_4$ の場合

$\alpha=\alpha_3$ の場合は，直線⑶と⑾が一致する時が y 切片すなわち利益が最大となる。この場合，図６では直線区間 ac 上で x と y の整数（花の本数は整数だから）の組があれば，それが答である。同じように，$\alpha=\alpha_4$ の場合は，直線区間 bc 上の x と y の整数の組が答になる。当然，答は複数個のことがあり得る。

（３）$\alpha < \alpha_3$，$\alpha > \alpha_4$ の場合

$\alpha < \alpha_3$ の場合は，点線が a 点を通るときが，また $\alpha > \alpha_4$ の場合は，b 点を通るときが最大利益を得ることになる。この場合は，「白亜」あるいは「紫苑」のいずれかだけを作ることが最大利益を得ることを示している。

このように，実行可能解の中で目的関数を最大または最小にする変数の値を**最適解**と呼ぶ。次節で，ソルバーを使ってこの例題を解く手順を説明する。

上の例題は簡単な２変数の例であった。一般の n 変数の場合には，a_{ij}，b_i，c_j

を定数として，

制約条件：

$$\sum_{j=1}^{n} a_{ij}x_j \leqq b_i, \quad (i=1,2,\cdots,m) \tag{13}$$

$$x_j \geqq 0, \qquad (j=1,2,\cdots,n) \tag{14}$$

のもとで，

目的関数：

$$f(x_1,x_2,\cdots,x_n)=\sum_{j=1}^{n} c_j x_j \tag{15}$$

を最大または最小にする変数 x_j を求める，といった形式に定式化できる。

上記例題は線形問題であったが，目的関数や制約条件が非線形である場合は非線形問題といわれ，線形計画法のような統一した解法はなく，特別な形に対する解法が示されているにすぎない。非線形問題の例として，次の例題を考えてみる。

［例題４］

制約条件	$x_1+x_2 \leqq 1$,
	$x_1 \geqq 0$，$x_2 \geqq 0$
のもとで，目的関数	
	$f=x_1^2-2x_1+x_2^2-2x_2$
を最小にする x_1，x_2を求めよ。	

制約条件を x_1-x_2平面に表すと図７の網掛け部のようになる。目的関数はこの平面上では点（1,1）を中心とする円になり，目的関数 f の値が大きいほど円の半径は大きくなる。f を最小にする問題であるから，この円と制約条件の外辺とが接する点が最適解となる。この例題から，最適解は線形計画問題では実行可能領域の頂点に存在するのに対し，非線形計画問題では必ずしも領域の頂点には存在しないことが分かる。ソルバーを用いて非線形計画問題を解くと，

与えられた初期値により決まる最良解を探してくれる。しかしそれが必ずしも最適解でないことに注意しておく必要がある。

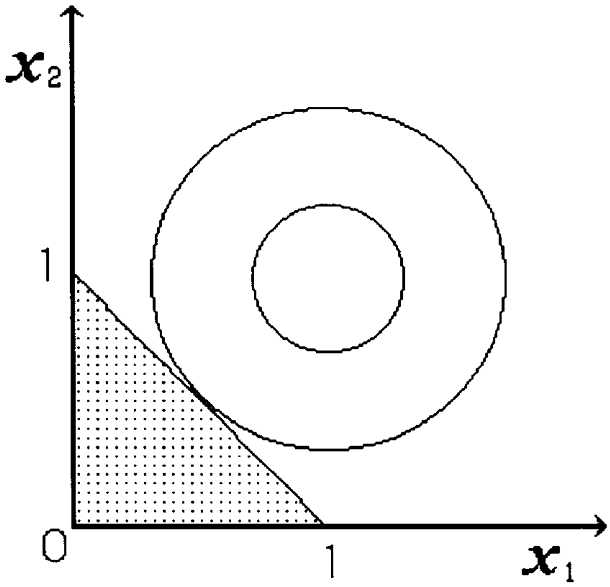

図７．非線形計画問題の解の求め方

３．２　線形計画法へのソルバーの応用

３．２．１　ソルバーの機能

ソルバーは，マニュアルによれば[3]

＊制約条件の範囲内になければならない変数や式がある

＊複数の回答が存在する可能性がある

＊反復計算でないと解けない

などの変数問題に応用する分析ツールとある。このため，ソルバーは，オペレーションズリサーチの１分野である「数理計画法」の問題を解くのに適している。

３．２．２　線形計画問題へのソルバーの応用

本節では，線形計画法の［例題３］をソルバーを使用して解く場合について述べる。

（1）データの設定

1-2-3 を起動すると，ワークシートAが現れる。ここで各セルに制約条件，目的関数，変数（ここでは，「白亜」と「紫苑」の束数）を記入する。各セルの位置は自由に選んでよい。**1-2-3** では，制約条件を入れるセルを「制約セル」，目的関数を入れるセルを「最適セル」，変数を入れるセルを「調整セル」といい，今回の例題では図8のように記入した。図8について説明する。

	A	B	C
1			
2	花束の束数		
3			
4	「白亜」の束数X	0	
5	「紫苑」の束数Y	0	
6			
7	目的関数	0	
8			
9	制約条件	1	
10		1	
11		1	
12		1	
13			

図8．データの設定

セルA2..A9：　B列のセルの説明

セルB4..B5：

「調整セル」である。ここでは，B4セルに x を，B5セルに y を割り付けた。これらのセルには，最初は初期値として適当な値を入れておかなければならない。ここでは　0　を入れた。

セルB7：

「最適セル」である。目的関数は式(11)であるので，ここではセルの内容は次のようになる。

+4*B4+2*B5

B4とB5はそれぞれ変数 x と y の値が入っているセルである。目的関数の中の変数が x と y でなく，セル番地になっていることに注意する必要がある。B7セルの表示が　0　となっているのは，B4とB5セル

の値が現在 0 であり，それによる上式の計算結果が 0 であるからである。

セルB9..B12：

「制約セル」である。「最適セル」への記入方法と同じで，変数をセル番地で表す。例えばB9セルには条件式（論理式）として式(6)を入れるので

+2*B4+3*B5<=100

となる。この論理式を満足しているときは各セルに 1 が，満足しないときは 0 が表示される。

以上で制約条件，目的関数等の入力は終わりである。

（2）ソルバーの起動と設定

データ設定が終了したら，つぎにソルバーを起動する。マウスで〈範囲〉→〈分析〉→〈ソルバー〉を選択すると図９のダイアログボックスが表示される。この例題では，上記に従い「調整セル」にB4..B5を，「制約セル」にB9..B12を，「最適セル」にB7を設定する。最大値，最小値モードの選択では，この問題は目的関数を最大にすることなので，最大値モードを選んでおく。解答の数は，線形の場合は「1」としておくと最適値を選んでくれる。

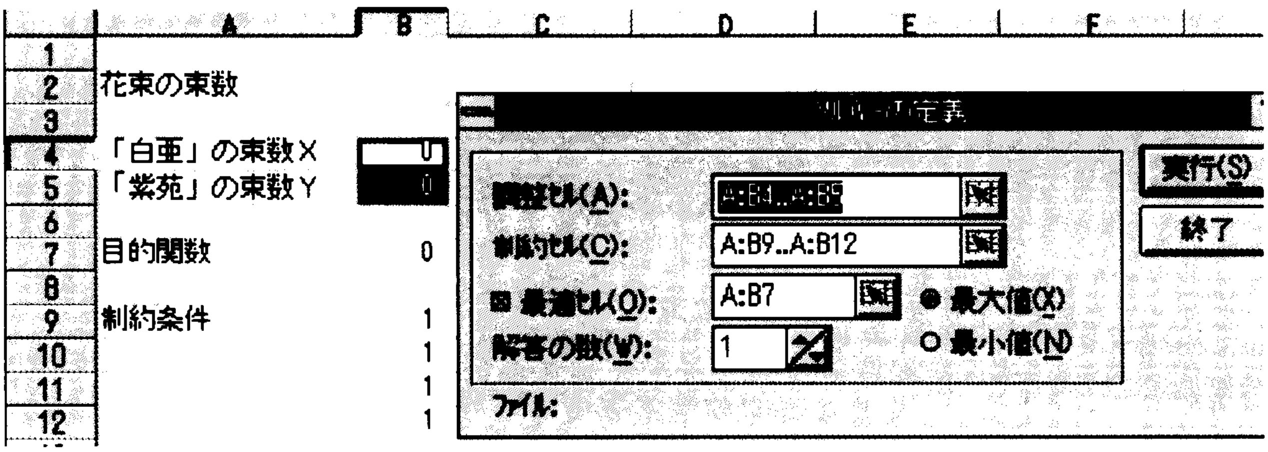

図９．ソルバーの設定（A：は，ワークシートを示す）

(3)結　果

ソルバーの設定終了後，図9のダイアログボックスの実行ボタンをクリックすると，図10のように結果が表示される。x と y の解は，それぞれB4とB5セルに表示され，目的関数の値はB7に表示される。このとき，B7の値は利益額でないことに注意する必要がある。何故なら，式⑾は式⑽を変換したものだからである。B9..B12セルがすべて 1 と表示されているのは，この解は全ての制約条件を満たしていることを示している。

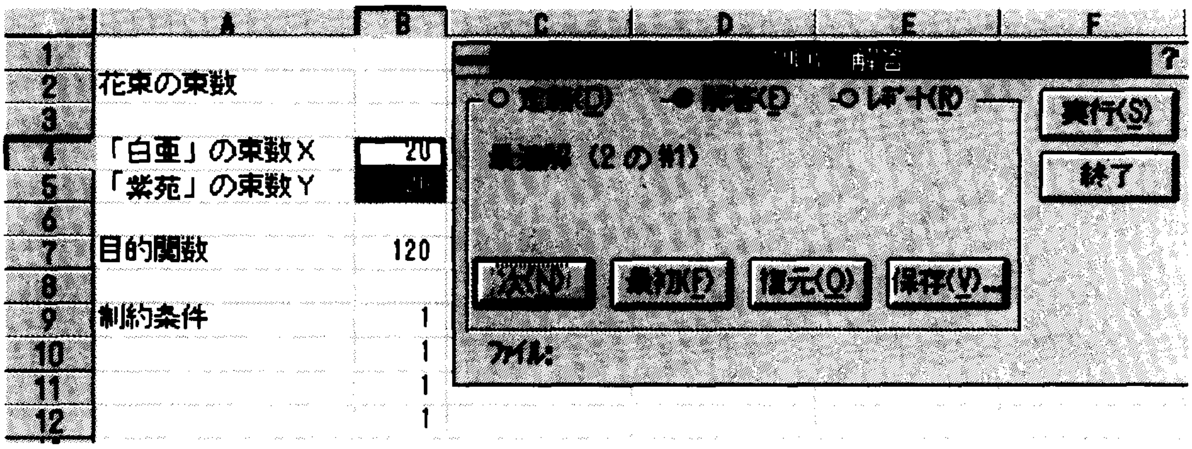

図10. ソルバーの解答

3．2．3　ソルバーの用語

マニュアルなどでソルバーの解説に使用されている重要な用語の意味は以下の通りである。

解　答：ソルバーを実行したとき，制約条件を全て満たしたときの結果を解答という。最適解とは限らない。

試　行：ソルバーを実行したとき，制約条件を1つでも満たさないときの結果を試行という。

最適解：最適セルを真に最大または最小にする解答をいう。

最良解：ソルバーが見つけた解の中で，最適セルの値を最大または最小にする解をいう。必ずしも最適解とは限らない。

4．ソルバーの応用と処理性能

本章では，連立線形・非線形方程式の解法および線形・非線形最適化問題に対してソルバーを適用したいくつかの例を示し，それらを通して得たソルバー使用上の注意点や処理性能に関する具体的な数値等を述べる。なお，本章の計算結果を得るのに使用したコンピュータはEPSON VT512R（CPU: Pentium 120MHz）である。これは現在のパソコンの中では上位に位置する計算能力を有する。

4．1　連立非線形方程式の解法

今，変数$x_i(i=1\sim N)$に対して

$$y_j=f_j(x_1,x_2,\cdot\cdot\cdot,x_N),\ j=1,2,\cdot\cdot\cdot,N \tag{16}$$

なる方程式の組を考える。結果の列y_1，y_2，・・・が与えられたときにx_iを決定する，という形式の問題が多くの分野で現れる。f_jがx_iに関して線形であれば通常の連立線形方程式であるから，このときは，ソルバーよりも**1-2-3**の持つ「逆行列」「行列の積」機能を使った方がよい。ソルバーが威力を発揮するのは，f_jが非線形関数の時である。本節ではこの問題に対する２つの簡単な例を示す。

4．1．1　一意な解が存在する場合

図11は抵抗とダイオードで構成された簡単な電気回路である。定常状態のダイオード電圧V_Dと電流I_Dをダイオードの動作点という。動作点が決まればダイオードの電気回路的性質も決まる。動作点は，ダイオード特性

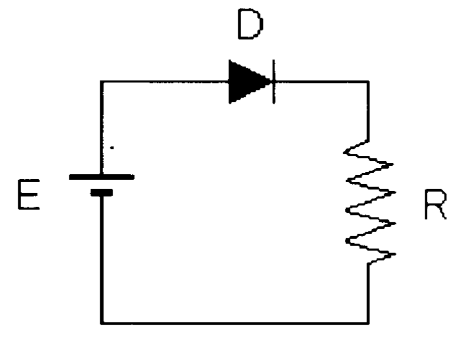

図11．ダイオード・抵抗回路

$$I_D = I_0[\exp(V_D / V_T) - 1] \tag{17}$$

と回路方程式（キルヒホッフの法則）

$$E = V_D + RI_D \tag{18}$$

の二つの方程式に支配される。ここで，E は印加直流電圧，R は抵抗値である。V_T は熱電圧で室温のとき約25mV 程度になる。I_0はダイオードの逆方向飽和電流である。ここでは $I_0=0.1\mu$A と仮定する。

ダイオード特性が指数関数を含んでいるため，動作点を精密に求めるには数値解法を行わねばならない。しかし，実際問題でそれを行うのは面倒なため普通は図式解法や折れ線近似の手法を使って解いている（実際，それで十分であることが多い)。しかし，ここでは，ソルバー適用の一つの例としてこの動作点を精密に求めてみる。

［例題５］

図11の回路において，式⒄⒅を満足する動作点（V_D，I_D）の精密な値を求めよ。

式⒄は単調増加関数ゆえ，この問題では一意な解が存在することは明らかである。こういった，解の一意性が保証された問題は非線形ではあってもかなり解きやすい部類に入る。ソルバーで解くために，式⒄を

$$I_0 = I_D / [\exp(V_D / V_T) - 1] \tag{17'}$$

と変形しておく。こうしておけば，式⒄' と式⒅それぞれの右辺の値が，与えられた E と I_0に一致する変数 I_D と V_D を求める，という変数問題に帰着する。この問題を解くためのワークシート上の具体的なセルの内容を図12に示す。また，ソルバーの設定は次のようにする。

	A	B	C	D
1	ID		+B1/(@EXP(B2/B6)-1)	式⑵'右辺
2	VD		+B2+B5*B1	式⑶右辺
3	E	10	+B3=C1	E=式⑵'右辺（論理式）
4	I0	1E-07	+B4=C2	I0=⑶右辺　（論理式）
5	R	10		
6	VT	0.025		

図12．ダイオード回路の動作点決定のためのセル内容の一例

・調整セル

B1..B2セルを指定する。ソルバーは，このセルの値を初期値として解の探索を行うため，実行前には何らかの値を入力しておく必要がある。

・制約セル

式(17)' (18)それぞれの右辺が E と I_0に等しいという論理式が入力してあるC3..C4セルを指定する。

・最適セル

これは使用しない。

・解答の数

この場合は，解答は１通りしかないから「解答数＝１」としておく。

実際にソルバーを実行したところ約１秒程度で解を得ることができた。いくつかの印加電圧 E に対して得た結果を表１に示しておく。

表１．印加電圧と動作点の関係

E(V)	I_D(A)	V_D(V)
5	0.461637	0.383628
10	0.959807	0.401927
20	1.958025	0.419751

($R=10\Omega, V_T=25\text{mV}, I_0=0.1\mu\text{A}$)

４．１．２　一意な解が存在しない場合

［例題６］

> $f(x,y)=\sin(x^3-y^3)+\cos(x^2+y^2)$ と $g(x,y)=\cos^2(xy)$ において，$f=g$ となるような解 x と y を求めよ。

この問題も $f(x,y)$ や $g(x,y)$ が非線形超越方程式であるから，代数的には解けず数値的に求めざるを得ない。ただし，$f=g$ となるような解は無数に存在することは明らかであるから，［例題５］の時とは異なり，解は一意でなくなる。

この問題に対してソルバーがどのような挙動を示すか調べた。変数に適当な初期値(x_0, y_0)を与え，そのいくつかの組み合わせに対して得られた解を表２に示す。これを見ると，解のほとんどが初期値の近傍に位置していることがわかる。そして，「解答数＞１」に設定していても，ソルバーは初期値近傍の解を見いだしたらそれ以上の探索は行わなかった。他の解も探索しようと実行ボタンをクリックすると，ソルバーは「ほかの解答はありません」というメッセージを表示する。解が複数あるかどうか明確でない問題を解いている場合は，この表示を信じると，他の解を見落としてしまうこともあり得るので注意しなければいけない。また，仮に複数の解の存在をユーザーが認識している場合でも，解を部分的にあるいは全貌を把握しようとすると，試行錯誤的に初期値を色々と変更して計算しなければならない。ユーザーにはかなりの負担が強いられるだろう。多変数の非線形問題を解く場合には，一般に，こういった現象が生じやすいことを念頭においておく必要がある。

表２　［例題６］において各初期値(x_0, y_0)を変化させたときのソルバーの解答

x_0 \ y_0	0	1	2	3	4	5
0	0 0	0 0	0 2.50442	0 4.33884	-4.21265 7.62351	0 5.00882
1	1.23423 0	1.26997 0.49519	1.02669 1.94026	0.54748 2.82702	1.00594 3.97402	1.11574 5.02220
2	8.54558 0.65842	1.94555 1.00834	2.02070 1.97113	2.00141 2.99500	1.99814 4.01644	2.03799 5.01581
3	-0.43680 3.09412	3.05416 1.19572	2.99593 2.00267	3.00340 2.99713	2.99255 4.01603	2.95418 4.99195
4	4.33885 0	4.01017 1.00053	4.00320 1.99855	4.00627 2.99580	4.00082 3.99910	4.015653 5.012751
5	4.33885 0	4.98973 0.99721	5.00397 2.11982	4.87580 3.02496	4.99286 4.04271	5.000230 5

※　表中の値は上段がx，下段がyの解である。

［例題６］で，条件を例えば「$f=g=0.5$」とさらに限定して解を求めようとすると，設定した初期値によっては解に到達できないことがある。ソルバーは，このようなとき，解の推測値をユーザーに要求し，その値から再出発して探索を続行する，といった処理形態を採用している。しかし，解は不明なわけであるから，１つの解を得るだけでも推測値の設定という試行錯誤的な作業が必要になる。

４．２ 輸送問題の最適化Ｉ（線形計画問題）

３章で配分問題を例に線形計画法の原理およびソルバーの使用方法について解説した。ここではそういった問題に対するソルバーの処理能力等について実験した結果を述べる。例は輸送問題とする。この問題は，輸送費が経費の中に占める割合の大きい，例えばコンクリートブロックのような重量物を扱っている企業では極めて重要な問題になっているという事例が報告されている[4]。

４．２．１ 問題設定と定式化

［例題７］

N 個の地点 $A_n(n=1\sim N)$ から M 個の地点 $B_m(m=1\sim M)$ へ何らかの製品を輸送する。各 A_n では a_n[kg] の製品が生産され，B_m ではそれが b_m[kg] 必要とされるものとし，A_n から B_m への輸送量は t_{nm}[kg] とする。A_n と B_m の位置（x_{A_n}, y_{A_n}）と（x_{B_m}, y_{B_m}），および a_n，b_m は全て固定値とする。総輸送コストを最低にする t_{nm} を決定せよ。ただし，総輸送コスト C は，

$$C=\sum_{n=1}^{N}\sum_{m=1}^{M}\gamma_{nm}\,d_{nm}\,t_{nm} \tag{19}$$

で与えられるものとする。ここで，γ_{nm}(円 /kg/km）は単位重量・距離当たりの輸送単価，d_{nm} は輸送距離（A_n-B_m 間直線距離で決まると仮

定）である。また，総生産量と総必要量は一致するものとする：

$$\sum_{n=1}^{N} a_n = \sum_{m=1}^{M} b_m. \tag{20}$$

この問題は，次の形式の最適化問題として定式化される：

目的関数：

$$C \rightarrow 最小 \tag{21}$$

拘束条件（$N+M+N{\cdot}M$ 個）：

$$\sum_{m=1}^{M} t_{nm} = a_n, \qquad (n=1 \sim N) \tag{22}$$

$$\sum_{n=1}^{N} t_{nm} = b_m, \qquad (m=1 \sim M) \tag{23}$$

$$t_{nm} \geqq 0, \qquad (n=1 \sim N,\ m=1 \sim M) \tag{24}$$

４．２．２　ソルバーによる解法

(１)データの設定

γ_{nm}を定数とすればCがt_{nm}の１次式となるから，［例題７］は線形計画問題となる。$N=2$，$M=5$，および

$$\gamma_{nm} = 1 \tag{25}$$

としたときのワークシート上のセル内容の一例を図13に示す。B1セルがγ_{nm}である。生産量a_nと必要量b_mは，それぞれE3..E4とF3..F7セルにあり，これらには式(20)を満たす適当な値を設定した。距離d_{nm}はC3..C12セルである。ここでは，A_nとB_mの位置を範囲名の形式で与えている。実際の座標データは，表３の値を図中にはない別のセルで定義している。

	A	B	C	D	E	F	G	H
1	γnm	1	目的関数	@SUM(D3..D12)				
2	n,m	tnm	dnm	γnm·dnm·tnm	生産量an	消費量bm	制約条件	
3	1,1	0	@SQRT((XA1-XB1)^2+(YA1-YB1)^2)	+$B$1*B3*C3	140	90	@SUM(B3..B7)=E3	式(22)
4	1,2	0	@SQRT((XA1-XB2)^2+(YA1-YB2)^2)	+$B$1*B4*C4	100	20	@SUM(B8..B12)=E4	〃
5	1,3	0	@SQRT((XA1-XB3)^2+(YA1-YB3)^2)	+$B$1*B5*C5		30	+B3+B8 =F3	式(23)
6	1,4	0	@SQRT((XA1-XB4)^2+(YA1-YB4)^2)	+$B$1*B6*C6		70	+B4+B9 =F4	〃
7	1,5	0	@SQRT((XA1-XB5)^2+(YA1-YB5)^2)	+$B$1*B7*C7		30	+B5+B10=F5	〃
8	2,1	0	@SQRT((XA2-XB1)^2+(YA2-YB1)^2)	+$B$1*B8*C8			+B6+B11=F6	〃
9	2,2	0	@SQRT((XA2-XB2)^2+(YA2-YB2)^2)	+$B$1*B9*C9			+B7+B12=F7	〃
10	2,3	0	@SQRT((XA2-XB3)^2+(YA2-YB3)^2)	+$B$1*B10*C10			+B3>=0	式(24)
11	2,4	0	@SQRT((XA2-XB4)^2+(YA2-YB4)^2)	+$B$1*B11*C11			+B4>=0	〃
12	2,5	0	@SQRT((XA2-XB5)^2+(YA2-YB5)^2)	+$B$1*B12*C12			+B5>=0	〃
13							+B6>=0	〃
14							+B7>=0	〃
15							+B8>=0	〃
16							+B9>=0	〃
17							+B10>=0	〃
18							+B11>=0	〃
19							+B12>=0	〃

図13. ［例題７］に対するワークシートの設定例（$N=2$, $M=5$）

表３．A_n と B_m の位置の座標データ

n ＼ 位置	x_{A_n}	y_{A_n}
1	0	0
2	50	0

m ＼ 位置	x_{B_m}	y_{B_m}
1	26	104
2	73	0
3	108	39
4	98	9
5	22	85

（２）ソルバーの設定

・調整セル

各経路の輸送量 t_{nm}，図13のB3..B12セルを指定する。ソルバー実行前に，このセルに入力してあるデータが解探索時の初期値となる。

・制約セル

式(22)～(24)の拘束条件が記述してある図13のG3..G19セルを指定する。

・最適セル

目的関数であるD1セルを指定し，最小値モードを選択する。

・解答の数

この問題は線形である。線形問題であれば，「解答数＝１」としておけば自動的に（もし存在すれば）最適解のみが求まる。最適解以外の解答を必要とするときは「解答数＞１」としておく。ちなみに，この例の全解答数は21個である。

（３）結　果

図13の設定で，ソルバーによって全ての解答を求めた。輸送コストとそれに対応する輸送量の解答を初めの５つまで表４に示す。また，理解を容易にするために最適解については図14にも示しておく。表４から，最適解である解答１および解答２，３のコストを比較すると0.5％程度の差しかないことがわかる。ソルバーは，というよりもコンピュータで計算した結果の全てがそうであるが，結果はあくまでも計算で考慮できる要因だけを対象にしたものであって，意志決定に必要な参考データを提供するだけにすぎない。それゆえ，計算の中に含めにくい他の諸要因を考慮して，最終的な判断は人間の側で行うべきであろう。

表４．ソルバーによる計算結果

	解答（５個のみ示した）				
	1（最適解）	2	3	4	5
C	19156	19249	19257	19601	19622
t_{11}	90	90	90	80	90
t_{12}	0	0	20	0	0
t_{13}	20	0	0	30	30
t_{14}	0	20	0	0	0
t_{15}	30	30	30	30	20
t_{21}	0	0	0	10	0
t_{22}	20	20	0	20	20
t_{23}	10	30	30	0	0
t_{24}	70	50	70	70	70
t_{25}	0	0	0	0	10

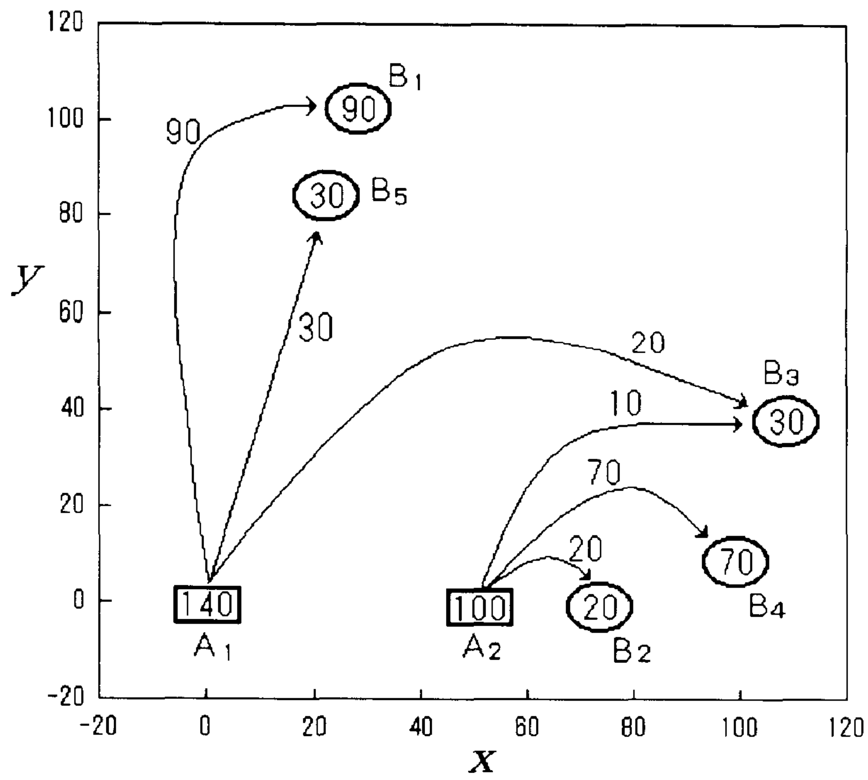

図14. ［例題７］に対する最適輸送量

4．2．3　処理性能と評価

通常の意志決定では各種の条件を変更しながら，コンピュータと対話的に計算を行うケースも多い。そのとき重要になるのが処理速度である。人によってばらつきはあるだろうが，一般的には数分以内にレスポンスがなければ，対話処理に適したシステムとは言い難い。こういった面のソルバーの性能を確かめるために，N と M の組み合わせによって問題の規模を変化させ，最適解の探索時間について測定した。結果を表５に示す。ただし，測定では解答数は１に設定した。また，探索時間の測定にはソルバーの経過時間表示を利用したため，この最小表示単位が１秒しかなく，表５中の処理時間の短いデータにはかなりの誤差が含まれていることに注意してもらいたい。拘束条件の数を横軸に，計算時間を縦軸にして表５のデータをプロットしたのが図15である。探索時間の小さいときの測定精度が悪いことを考慮すると，対数的に良好な直線関係にあることがわかる。ほぼ，

$$探索時間 \propto （拘束条件数)^{p}$$

の形式になっている。表５のデータに対して回帰分析（これも **1-2-3** の機能である）した結果，$p \fallingdotseq 2$ が得られた。すなわち，この例では探索時間は拘束条件の数のほぼ２乗に比例する。こういった直線関係がないと（後で述べる非線形問題がそれに相当する），問題の規模を変えたときにどの程度の計算時間となるかを見積もることができず計画的な作業をやりにくい。特に計算が長時間に及ぶような場合にはこのことは重要である。線形問題は，（もし存在すれば）確実に最適解が得られることが重要なのは言うまでもないが，さらに，上記比例関係が成立することによって計画的な作業をやり易いといった面も見逃せない特徴のひとつである。

表５．［例題７］に対する処理時間の測定結果

N	M	変数の数	拘束条件数	探索時間(sec)
2	5	10	17	1
2	10	20	32	2
2	20	40	62	6
2	40	80	122	31
3	5	15	23	1
3	10	30	43	4
3	20	60	83	11
3	40	120	163	57
5	10	50	65	7
5	20	100	125	22
5	40	200	245	97

変数の数が200という比較的大きな問題でも２分以内に最適解が得られ，少なくともここで取りあげた例題ではソルバーは対話処理に対応しているといえるだろう。さらに，*Windows* 版 **1-2-3** 自体のユーザーインタフェースの良さや他の便利なツール（グラフ化機能やマクロ）なども考慮すると，線形計画問題にソルバーを適用するのは有意義であると判断してよい。最も，計算時間は使用するコンピュータの CPU パワーに大きく依存する。特に，こういった処理では純粋な数値計算が内部で実行されているから，計算時間は CPU と浮動小数点演算プロセッサ（Pentium では内蔵）の性能だけで決まるといっても過言ではない。最適化問題では，より高速のコンピュータに対する要求は必然的に大きくなる。

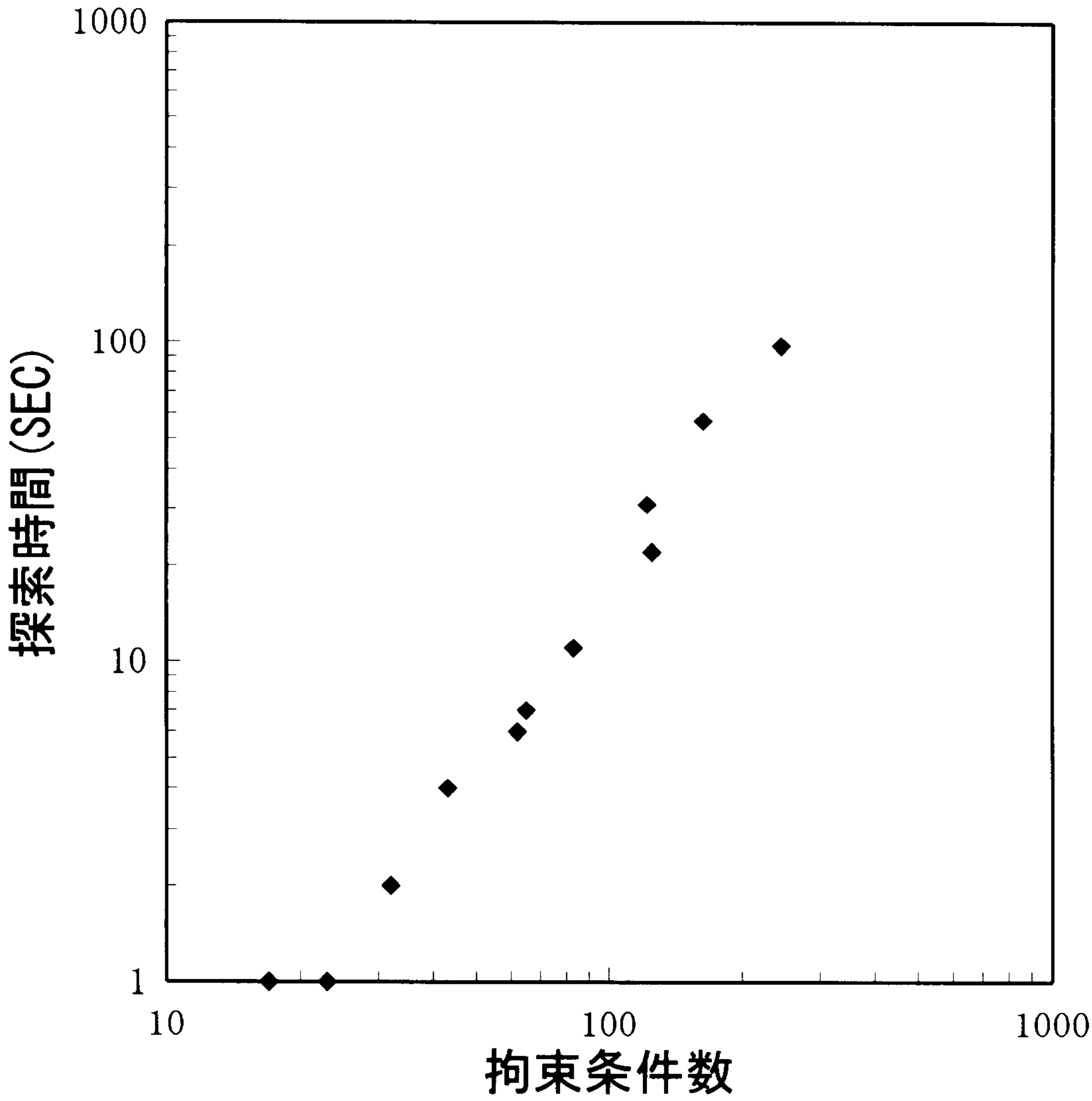

図15. 拘束条件の数と計算時間の関係

4．3　輸送問題の最適化Ⅱ（２次計画問題）

目的関数が変数の２次形式になっている非線形問題を特に２次計画問題と呼ぶ。２次計画問題では，シンプレックス法と同様な手法で最適解に到達するアルゴリズムが知られている。ソルバーが２次計画問題をサポートしているかどうか確かめるために，参考文献[5]にある例題（２変数の２次計画問題）

$$\begin{cases} z=2x_1{}^2-2x_1x_2+x_2{}^2+2x_1-12x_2 \quad \rightarrow \quad 最小 \\ 条件：\ x_1+x_2 \leqq 10,\ x_1 \geqq 0,\ x_2 \geqq 0 \end{cases} \tag{26}$$

を解いてみた。この問題には可能基底解3個があり，これとは別に最適解（$x_1=2.6$，$x_2=7.4$，$z=-53.8$）が存在する。ソルバーは，これら4つの解全てを見つけた。このことから，2次計画問題にもソルバーは対応していることがわかる。一つ不満な点としては，最適解であるはずの解を「最良解」という形で表示することである。これでは，仮に最適解を得ていても，ユーザーは他に良い解があるかのように誤解してしまう恐れがある。

2次計画問題におけるソルバーの処理性能を調べるために，［例題7］で重量・距離単価が重量に比例すると仮定して，

$$\gamma_{nm}=t_{nm}\ /\ T \tag{27}$$

の形式になった場合を考えよう。Tは適当な規格化定数である（ここでは$T=10$とした)。このとき，目的関数Cは

$$C=\frac{1}{T}\sum_{n,m}d_{nm}(t_{nm})^2 \tag{28}$$

とt_{nm}の2次形式となり，［例題7］は2次計画問題となる。この例題に対して問題の規模を4．2節と同様に変化させて処理時間を測定した。線形問題と同様なアルゴリズムとはいっても2次計画問題では探索すべき解の範囲が広く，かなりの計算時間を必要とする。この例題では，$N=2$，$M=5$のとき，全解答（21個）を探索しその中の最良解（全解答を探索しているから，これは最適解のはずである）を得るまでに約20秒ほどかかった。この程度の計算時間であれば何ら問題はない。しかし，$N=2$，$M=10$と変数の数を2倍にすると探索すべき全解答数は一挙に945個にまで増大し，最適解を得るのに約1時間もの計算時間が必要であった。こうなると対話的な処理はもはや不可能である。さらには，ソルバーでは解答の数を最大999個までしか指定できないため，これ以上の規模の問題ではいくら時間をかけても最適解が得られない可能性もある。結局，2次計画問題では最適解を見いだすことが原理的に可能であるとはいっても，問題の規模がある程度以上になると（この例では変数がおおよそ20個以上

程度），処理時間およびソルバーの探索数上限などの要因によって最良解しか得られないことが多いといえる。

筆者らは，非線形問題で対話的にソルバーを使用する際には次のようにしている。解答数を常に最大（999個）に設定しておいて解の探索を実行し，適当な時間（許容時間）がきたらその時点でキャンセルボタンによって計算を中止する。ソルバーは，その時点までに探索した解の中から最良のものを表示してくれる。こういった形態で使用していると，単位時間当たりにどの程度の解探索能力をソルバーが持っているか，が重要になる。なぜなら，探索する解の数が多ければ多いほど最適解に近い最良解を得る確率が高くなるからである。そこで，t_{nm}の初期値をゼロとして，前節の表５と同じNとMの組み合わせに対して，10分間の計算で探索できる解の数を測定してみた。それを１分当たりに換算した値を表６に示す。表６では，表５中にあるいくつかの組み合わせを示していないが，これらは30分程度計算しても１個の解しか探索できなかったため省略した。表６を見ると，線形問題で見られた，問題の規模と処理能力との間のきれいな関係はない。２次計画問題になると必ずしも問題の規模だけでは計算の難易度を予想できないようである。つまり，小さな規模でもある許容時間内ではまともな解が得られない可能性もあるし，逆に，規模は大きくても案外うまく計算できることもあり得るということになる。

表６．非線形問題における単位時間当たりの解の探索数

N	M	１分間当たりの探索数
2	5	78.8
2	10	35.0
2	20	10.4
2	40	51.4
3	5	15.4

４．４　輸送・配置混合問題の最適化（非線形計画問題）

［例題７］において，製品の供給地A_nを新しく増設する計画があると仮定し

よう。このような状況では，輸送コストを最小にするような立地場所と輸送量はどのようになるだろうか，という輸送と配置を混合した問題を考える必要が生じるであろう。

［例題８］

A_1が（0,0）の地点に既にありA_2を新しく建設したい。$N=2$，$M=5$とし，他の条件は［例題７］の場合と全て同じであると仮定して，輸送コストを最小にする輸送量t_{nm}とA_2の位置（x_{A_2}，y_{A_2}）を決定せよ。ただし，γ_{nm}は式(25)で与えられる定数とする。

この例題では，Cは，

$$C=\sum_{m=1}^{5} t_{1m}d_{1m}+\sum_{m=1}^{5} t_{2m}\sqrt{(x_{A_2}-x_{B_m})^2+(y_{A_2}-y_{B_m})^2} \tag{29}$$

の形式となる。右辺第１項はd_{1m}が定数ゆえt_{1m}に関して線形であるが，第２項はt_{2m}，x_{A_2}，y_{A_2}を変数とする非線形関数である。この影響で［例題８］は一般の非線形問題となる。

図13のワークシート設定に位置変数（x_{A_2},y_{A_2}）を調整セルとして加え，他は全て［例題７］と同様な条件で［例題８］にソルバーを適用した。この例題でも，２次計画問題と同様に計算時間さえかければ，輸送コストを最適化するA_2の位置と輸送量の最適解（あるいはそれに近い最良解）が得られるだろうと感覚的には予想される。t_{nm}の初期値をゼロ，そして位置変数（x_{A_2},y_{A_2}）の初期値をいくつか変化させてソルバーで実際に解を求めてみたところ表７のようになった。初期値によって到達する解がばらばらであることがわかる。さらに，他の解を求めようとすると「ほかの解答はありません」というメッセージが出力され，ソルバーはそれ以上の探索を行わなかった。［例題６］で注意したことと同様な現象がここでもおきる。また，表７の目的関数値（輸送コストC）を見ると，［例題７］で我々が適当に設定したA_2の位置（50,0）の時よりもかえって大きくなっていることがわかる（表４参照）。これでは何のために時間をかけて計算しているのかわからない。以上のような結果から，非線形問題の

最適化にはかなり難しい問題が含まれていることを理解できるかと思う。

表７．座標初期値に対するソルバーの解答の変化

初期値 x_{A_2} \ 初期値 y_{A_2}	-100	-50	0	50	100	150	200
-100	-1340.10 1704.50 (226660)	50.56 45.13 (20150)	S	56.97 67.37 (19263)	48.08 43.21 (20211)	S	164.11 -110.68 (37328)
-50	S	S	S	359.33 -257.11 (61051)	S	151.70 85.84 (26529)	S
0	75.71 41.28 (21270)	-113.01 130.24 (29899)	S	S	-45.28 77.93 (22944)	-494.97 2494.28 (260914)	S
50	S	S	S	48.42 42.98 (20230)	S	S	S
100	26.14 85.37 (17477)	S	S	38.87 27.90 (21096)	46.14 42.61 (20216)	55.25 44.03 (20327)	22.62 43.04 (20213)
150	119.32 21.72 (25016)	1979.32 -1703.60 (278081)	S	S	50.56 45.13 (20151)	S	S
200	S	S	39.87 29.96 (20959)	44.56 35.29 (20639)	244.44 173.77 (37266)	49.50 42.94 (20253)	S

（表中の数値は上から順に解 x_{A_2}，y_{A_2}，目的関数値（括弧内）である。S は，解が $x_{A_2}=47.58$，$y_{A_2}=42.52$ に収束し，目的関数値が20243であることを表している。t_{nm} の初期値はゼロとして計算した。t_{nm} の解答はいずれの場合でも，$t_{11}=10$，$t_{13}=30$，$t_{14}=70$，$t_{15}=30$，$t_{21}=80$，$t_{22}=20$，$t_{12}=t_{23}=t_{24}=t_{25}=0$ であった。なお，それぞれの解を求めるには２分程度の時間を要した。）

非線形問題は凸型と非凸型問題に分類できる。これらは，簡単に説明すれば次のようなことである。前者は局所的な最適解が同時に大域的な最適解でもあるような問題である。このときは，コンピュータのメモリを無制限に使用でき，さらに計算時間を気にしないならば最適解を確実に得ることができる。一方，後者は，局所的に最適である解が複数存在する場合で，局所解が大域的な最適解とは限らない，より一般的な問題である。この問題に対する最適化は非常に難しく，現状では能率の良い解析アルゴリズムは見いだされていない。［例題

８］はちょうど非凸型問題に対応していると考えられ，ソルバーが特定の局所最適解に収束した結果，先のような現象が生じたものと思われる。なお，一部の非凸型問題については，実用的な変数規模の問題に対して効率よく大域最適解を求めるアルゴリズムが既に開発されている[6]。そういったアルゴリズムがソルバーに搭載されているかどうかは今のところ調べていない。いずれにしてもこのままでは［例題８］に対する有効な解答は得られそうもない。

非線形における凸型と非凸型という分類は，解の存在空間に対する幾何学形状をイメージして定義された用語である。解の存在空間は制約条件で決まるため，当然であるが，制約条件を変更すれば状況は変わる。表７の計算では A_2 の位置に何ら制約を設けていなかったので，A_2の位置を閉領域内に限定し，さらに A_2や B_m(m=1～5)の近傍に配置してはならないという制約

$$\left\{\begin{array}{l} 0 \leqq x_{A_2} \leqq 120, \quad 0 \leqq y_{A_2} \leqq 120, \\ A_1 \cdot A_2\text{間距離} > 20, \\ A_2 \cdot B_m\text{間距離} > 20, \ (m=1\sim5) \end{array}\right. \tag{30}$$

を設定したところ，特定の解だけに収束してしまうという状況は回避され，複数の解を探索できるようになった。この制約条件下で，全ての変数の初期値をゼロとして，約２時間半計算したところ224個の解を探索できた。本来なら，これ以上の解の探索も可能であるが，メモリ不足によってそれ以上の探索はできなかった（使用環境：メインメモリ16MB，*Windows* 3.1上で **1-2-3** だけ実行)。この計算で得られた最良解を表８に示す。図16は表８の結果を表現したものである。これを見ると，さほど奇抜というほどの結果ではなく，このような解答であれば人間の直感による判断だけでも容易に得られそうで，わざわざ計算する必要もないような印象を受ける。しかし，直感による判断だけでは，それが他の解に比べてどの程度良いのかを客観的に判断することはできない。ソルバーなどを使って計算することの意義はこの点にある。

一般の非線形問題の解法は計算時間が非常に長くなり，手軽に最適値を求めるといった感覚ではなくなる。しかし，それでも，多くの解を自動的に探索で

きるような問題であれば，たとえ時間はかかるにしてもコンピュータに一晩頑張ってもらうだけで，ユーザーは手間をかけることなく，あくる朝にはそれなりの結果を手にすることができる。現状ではこういったやり方でしか非線形問題には対処するしかないと考えられる。もちろん，それも程度の問題である。ここでは$N=2$，$M=5$という簡単な問題だけを取りあげたが，これでもある程度の最良解を得るためには数時間の計算を要した。それゆえ，実用的規模の変数の非線形最適化問題へソルバーを適用し，意味のある時間内で解を求めることは，現在のコンピュータの性能ではほとんど不可能に近いと言わざるを得ないだろう。

表８．［例題８］に対する最良解（解答数＝224）

目的関数	x_{A_2}	y_{A_2}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}	t_{21}	t_{22}	t_{23}	t_{24}	t_{25}
15548	41.11	90.90	0	20	30	70	20	90	0	0	0	10

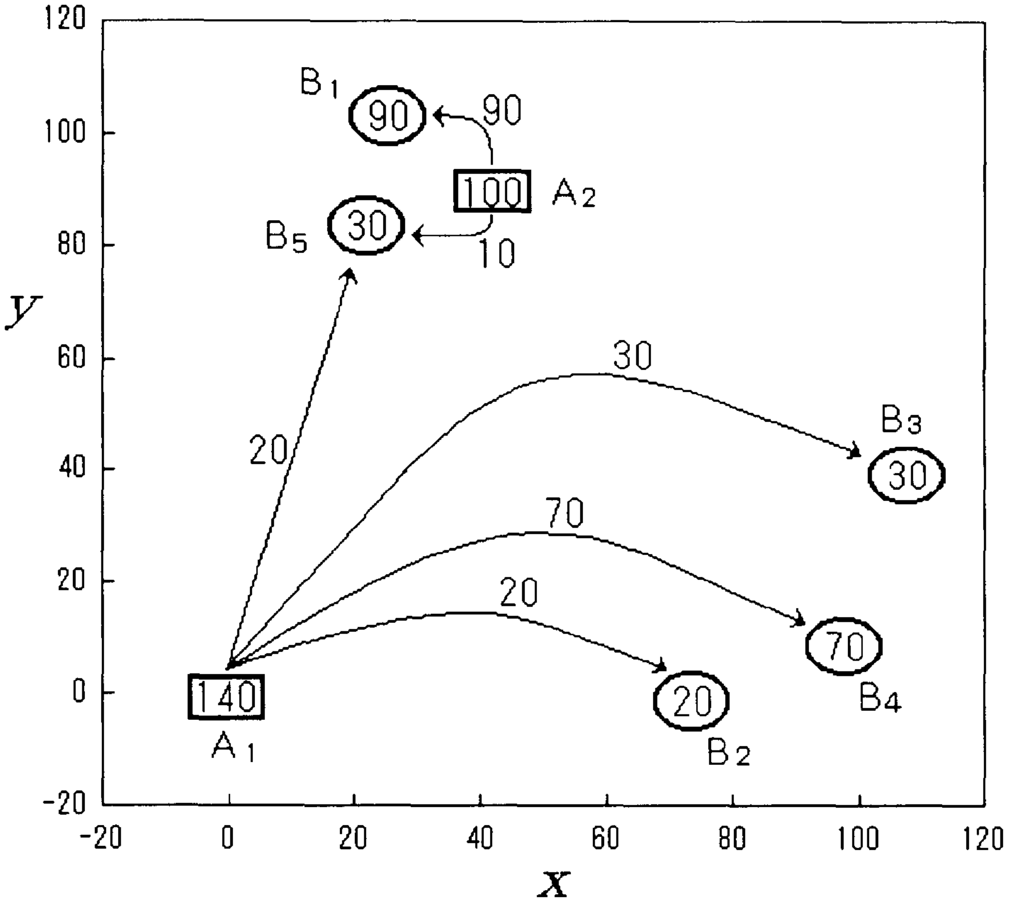

図16　［例題８］に対するA_2の最良配置と輸送量（解答数＝224）

［例題8］だけを対象にすると，これには次のようなアルゴリズムも適用できる。A_2の位置の最適解はB_mの存在する地点からそう離れたところにはないだろうから，立地点の最適解がどの辺りの範囲内に含まれているかを直感的に判断するのは簡単である。それゆえ，その範囲を適当な格子に分割して，A_2の位置を格子上で順次変化させて目的関数値を調べていけば，いずれは最適解に到達できるはずである。ただし，最適解とはいっても，領域の分割に起因する量子化誤差は生じる。近似最適解とでも呼んだ方が適切かもしれない。例えば，(0,0)−(100,100）の四角形領域に範囲を限定して，それを刻み1の格子状に分割して調べるとする。このとき，最高1万回，同じ計算を実行することになる。膨大な作業量ではあるが，この方法では各格子点上でA_2の位置は固定されるわけであるから，実質的に［例題7］と同じ線形計画問題となっていることに注意する。線形問題であれば，$N=2$，$M=5$のときの計算時間は，各格子点当たり約1秒程度（表5参照）の時間で済むわけであるから，約3時間弱で全格子点の計算を完了できることになる。これは，表8の結果を得たのと同程度の時間である。もちろん，この見積もりは，位置の変更作業などを手作業でやるのではなく，マクロを使って処理を自動化した場合の話である。時間が同程度ならば，マクロを作成する分だけ，表8を求めたやり方に比べて面倒なだけで特に有効であるとはいえないだろう。ただ，この方法には非常に魅力的な特徴がある。各格子点での計算に相互干渉が全くなく並列処理に適しているということである。もし将来，並列コンピュータがユーザーに気軽に利用できる環境になれば，これも結構強力な計算手法の一つになるかもしれない。これと似たようなことは他の分野でも多くあり，例えばコンピュータグラフィックス分野におけるレイトレーシング法はアルゴリズム的に優れた並列性を有しており，その特徴を生かすために専用並列コンピュータの研究が試みられている[7]。また，天体物理の分野でも銀河の衝突を解析するために同様の研究がなされ，実機も既に開発されている[8]。最適化問題用にも，安価なこの種のコンピュータの出現が望まれる。

5．あとがき

1-2-3 に搭載されているバックソルバーとソルバーの機能およびその処理能力などについて紹介・解説した。バックソルバーやソルバーを用いると自分のデスク上で手軽に変数問題や最適化問題を扱えるというのが魅力的である。本稿では紹介しなかったが，バックソルバーは，内部に条件分岐を含むような，解析的でない問題にも対応しており，かなり広い応用範囲を持っているので利用価値は高いと考えられる。通常使用する範囲内であれば，バックソルバーの処理性能に関しては特に何ら問題は生じないものと思われる。一方，ソルバーについては，問題の規模が非常に大きくなったり，あるいは小規模でも非線形最適化問題になると適用は難しいだろうが，それでもこの程度の能力があれば社会科学系の研究者にとっては十分な性能であると主張している人もいる[9]。確かに線形計画問題に関しては，**1-2-3** のインターフェースの良さもあって，ソルバーはかなりの能力を発揮する。ただし，近年の線形計画問題で広く普及している考え方である，目的関数が複数存在する場合の最適化（多目的最適化）や意志決定者のあいまい性を考慮したファジー最適化などの問題[10-12]はそのまま自動的には扱えない。今後は，こういった問題にも気軽に対処できるようになればさらに素晴らしいツールとなるであろうが，ここしばらくは，現状のままで我慢するしかなかろう。あるいは，マクロ言語を利用してユーザー独自で新しい機能を付加するという手もある。例えば，ソルバーの搭載されていない **1-2-3 R2.3J**（ソルバーは *Windows* 版の **R4J** 以降組み込まれた）にソルバーと類似の機能をマクロを使って実現した事例などがある[13]。**1-2-3** の利用はこの拡張性も大きな魅力の一つである。ただし，**1-2-3** のマクロは本格的なプログラミングを意識して設計された言語ではないため（簡易言語と呼称されることがある），通常のプログラミング言語に比べると構造化などへの配慮はなされていない。また，一種のインタプリタ型言語であるため処理も遅く，大規模なプログラム開発にはあまり適していないという欠点がある。この辺りは，Microsoft 社 **Excel** Ver.5の方がかなり優れている。**Excel** では，一般の *Windows* アプリケーション開発でよく使用される Visual Basic を用いてマク

ロを記述できる。さらに，Excelは整数計画問題へも応用可能であるようなので，今後，機会があればこれについても紹介したいと考えている。

本解説を足がかりに広くバックソルバーやソルバーが使いこなされ，各人の研究分野での問題解決に役に立ててもらえれば幸いである。

参 考 文 献

(1) 宮川公男：「ＯＲ入門」，日本経済新聞社，1993.

(2) 大鹿譲，一森哲男：「オペレーションズリサーチ」，共立出版，1994.

(3) Lotus社編：「Lotus 1-2-3 R4J ユーザーズガイド」，1993.

(4) 日本ＯＲ学会編：「ＯＲ事例集1991」，日科技連，1991，p. 87.

(5) 古林隆：「線形計画法入門」，産業図書，1984，pp. 167-171.

(6) 今野浩：「大域的最適化法の現状 — 低ランク非凸型最小化問題を中心に — 」，情報処理学会誌，Vol. 36，No. 11，1995，pp. 1062-1069.

(7) 權五鳳，村田誠治，村上和彰，富田眞治：「レイトレーシング法を高速処理する専用並列レンダリング・マシン『熱視線』の要素プロセッサ・アーキテクチャ — VLIW アーキテクチャおよび性能評価 — 」，情報処理学会論文誌，Vol. 34，No. 7，1993，pp. 1650-1662.

(8) 杉本大二郎：「手作りスーパーコンピュータへの挑戦 — テラフロップス・マシンをめざして — 」，講談社ブルーバックス，1993.

(9) 高橋三雄：「勢揃いした意志決定支援ツール」，bit，No. 10，1995，pp. 82-85.

(10) 坂和正敏：「経営数理システムの基礎」，森北出版，1995.

(11) 坂和正敏：「線形システムの最適化」，森北出版，1995.

(12) 坂和正敏，石井博昭，西崎一郎：「ソフト最適化」，朝倉書店，1995.

(13) 松村幸輝：「基礎からのＯＲシミュレーション」，オーム社，1995.