

JAERI-M

9 3 1 5

分解原理による大規模線形最適化
プログラム：DEPRI, DEPRIM

1981年2月

藤村統一郎・堀上 邦彦・中原 康明

日本原子力研究所
Japan Atomic Energy Research Institute

この報告書は、日本原子力研究所が JAERI-M レポートとして、不定期に刊行している研究報告書です。入手、複製などのお問合せは、日本原子力研究所技術情報部（茨城県那珂郡東海村）あて、お申しこしください。

JAERI-M reports, issued irregularly, describe the results of research works carried out in JAERI. Inquiries about the availability of reports and their reproduction should be addressed to Division of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken, Japan.

JAERI - M 9315

分解原理による大規模線形最適化
プログラム : DEPRI, DEPRIM

日本原子力研究所東海研究所原子炉工学部
藤村統一郎・堀上邦彦・中原康明

(1981年1月19日受理)

分解原理を応用して大規模な線形システムを解析するプログラム DEPRI を開発した。システム全体の最適性は、部分システムのそれと部分システム間の相互干渉より求めるが、そのアルゴリズムを詳述する。また、このプログラムが適用できる二、三の応用例を掲げるとともに、プログラムの発展性についても検討する。

なお、DEPRIMはこのプログラムを利用し易くするための補助プログラムである。

DEPRI, DEPRIM : Programs for Solving Large Linear
Optimization Problems by Decomposition Principle

Toichiro FUJIMURA, Kunihiko HORIKAMI and Yasuaki NAKAHARA

Division of Reactor Engineering, Tokai Research Establishment, JAERI
(Received January 19, 1981)

DEPRI, a program for analyzing large linear systems, has been developed employing decomposition principle. The optimum condition of the whole system is obtained from the optimum condition of each subsystem and interference among them. The algorithm is described in detail. Furthermore, a few examples to which the program is applicable are given and the possibility of future growth of the program is discussed.

DEPRIM is the auxiliary program which makes it easy to use the DEPRI.

Keywords : Optimization, Large Linear System, Computer Program,
Decomposition Principle, Subsystem, Algorithm

目 次

1. 序 言	1
2. 数値解法	2
2.1 単体法	2
2.2 初期許容解の求め方	5
2.3 分解原理	7
3. 応用例	12
3.1 簡単な問題	12
3.2 エネルギー・モデルへの応用	15
4. プログラムの使用法	18
4.1 DEPRI	18
4.2 DEPRIM	22
5. 結 言	23
謝 辞	24
参考文献	25
付録 DEPRIMの入出力例	26

Contents

1.	Introduction	1
2.	Numerical Method	2
2.1	Simplex Method	2
2.2	Determination of Initial Feasible Solution	5
2.3	Decomposition Principle	7
3.	Some Applications	12
3.1	Simple Problems	12
3.2	Application to an Energy Model	15
4.	How to Use the Programs	18
4.1	DEPRI	18
4.2	DEPRIM	22
5.	Conclusions	23
	Acknowledgements	24
	References	25
	Appendix Sample Input/Output Listing of DEPRIM	26

1. 序 言

我々の日常生活を考えてみると、その行動には必ず目標があり、それを達成するとき、できる限りその労苦（費用、cost）を少なく済ませようとする。また、ほとんどの場合その行動に制約（constraint）がつく。このような事象は、他の社会的活動や工学的システムの挙動についても言えることで、一般に

$$\text{Minimize } F(\mathbf{x}) \text{ subject to } \mathbf{x} \in D \quad (1)$$

と数式化される。ここに、ベクトル \mathbf{x} はすべての制約式を満たす領域（domain）Dにおける変数であり、Fは費用を表わす関数で目的関数（object function）という。この問題は

$$\text{Maximize } -F(\mathbf{x}) \text{ subject to } \mathbf{x} \in D \quad (2)$$

と同等であるが、変数 \mathbf{x} を定めた目的のために最適に選ぶという意味で最適化問題（optimization problem）と呼ばれている^{1)～5)}。

最適化問題は制約式の有無、あるいは目的関数や制約式の型により分類される。線形(linear)最適化問題が有限な(finite)最適値(optimum value)を持つためには何らかの制約が必要であるが、その領域Dが凸多面体(convex polyhedron)の場合は単体法(simplex method)としてその解法が確立されている^{1)～4)}。しかし、送電計画まで含めた原子力発電所の最適運転計画のように制御変数の数が非常に多い問題とか、エネルギー・システムの日本モデルや世界モデルのように数多くの部分システムから構成される大規模なシステムの解析に単純に単体法を適用するには、計算機の性能上の問題も含めて、実行上種々の困難が伴う。このように、システム全体(whole system)が大規模(large scale)で、かつそれが部分システム(subsystem)からなるような場合には、Dantzig-Wolfeによる分解原理(decomposition principle)が一層有効に働く^{1)～7)}。

本稿はこの原理のプログラム化について述べたもので、その基本となる単体法および分解原理の具体的なアルゴリズムが次章で詳述される。また第3章では、製品の輸送の問題^{1) 2) 6)}を初めとする簡単な例題のほか大次元でかつ制約式の多いエネルギー・モデルの例⁸⁾を解くことを考える。開発されたプログラムDEPRIは最適解(optimum solution) $\tilde{\mathbf{x}}$ の探索を進めるに必要な初期許容(initial feasible)解 \mathbf{x}_0 ($\in D$) が自明な場合に限られているが、その使用法が第4章で説明される。このプログラムによる計算結果および許容解が自明でない場合などへの拡張については、第3章および第5章で検討が行われる。

2. 数 値 解 法

「慣習的な記法^{1)～4)}により、一般的な線形最適化問題

$$\begin{aligned} & \text{Minimize } F(\mathbf{x}) = \mathbf{c}^T \mathbf{x} \\ & \text{subject to } \mathbf{A}\mathbf{x} \leq \mathbf{b} \end{aligned} \quad (3)$$

を考えよう。ここに、 $\mathbf{x} = (x_1, \dots, x_n)^T$ は n 個の変数をもつベクトル、 $\mathbf{A} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$ 、

$\mathbf{b} = (b_1, \dots, b_m)^T$ はそれぞれ m 個の制約を表わすための行列およびベクトルで $\mathbf{c} = (c_1, \dots, c_n)^T$ はコスト・ベクトルである。記号 \leq は m 個の制約の各々が

$$a_{11} x_1 + \cdots + a_{1n} x_n \leq b_1 \quad (\text{第1型}) \quad (4)$$

$$\cdots \cdots \cdots \geq b_i \quad (\text{第2型}) \quad (5)$$

$$\cdots \cdots \cdots = b_i \quad (\text{第3型}) \quad (6)$$

のどれか一つで成り立っていることを意味する。従って許容領域 D はこの m 個の制約を同時に満たす \mathbf{x} の集合であり、 n 次凸多面体である。

いま、 $\mathbf{b} \geq \mathbf{0}$ であるとしよう。このとき、制約式が(4), (5), (6)のどの式に相当するかによってそれぞれ第1型、第2型および第3型の制約と呼ぶことにする。また、ある変数 x_j が負になり得るときも

$$x_j = x_j^+ - x_j^- , \quad x_j^+, x_j^- \geq 0$$

と変換できる。以上の議論から明らかなように、我々は一般性を失うことなく線形最適化問題を

$$\text{Minimize } F(\mathbf{x}) = c_0 + \mathbf{c}^T \mathbf{x} \quad \text{subject to } \mathbf{A}\mathbf{x} \leq \mathbf{b}, \quad \mathbf{b} \geq \mathbf{0}, \quad \mathbf{x} \geq \mathbf{0} \quad (7)$$

と記述しうることが分る。ここに、 c_0 はコストの付加であり、これを導入することで解法上の本質的な相違を生じない一方、応用上便利なことが多いので敢えて加えてある。以下、単体法および分解原理アルゴリズムについて述べよう。

2.1 単体法

いま、線形最適化問題(7)式の m 個の制約式が第1型、第2型、第3型の順に並んでおり、それぞれ m_1, m_2, m_3 個あるとしよう。記述の便を考え、 $\alpha = m_1, \beta = m_1 + m_2$ として付加(slack) 変数 $\mathbf{y} = (y_1, \dots, y_\alpha, y_{\alpha+1}, \dots, y_\beta)^T$ を導入すれば、制約式は

$$\begin{aligned}
& a_{11} x_1 + \cdots + a_{1n} x_n + y_1 = b_1 \\
& \dots \\
& a_{\alpha 1} x_1 + \cdots + a_{\alpha n} x_n + y_\alpha = b_\alpha \\
& a_{\alpha+1,1} x_1 + \cdots + a_{\alpha+1,n} x_n - y_{\alpha+1} = b_{\alpha+1} \\
& \dots \\
& a_{\beta 1} x_1 + \cdots + a_{\beta n} x_n - y_\beta = b_\beta \\
& a_{\beta+1,1} x_1 + \cdots + a_{\beta+1,n} x_n = b_{\beta+1} \\
& \dots \\
& a_{m1} x_1 + \cdots + a_{mn} x_n = b_m
\end{aligned} \tag{8}$$

と書ける。従って、問題(7)式は

$$\text{Minimize } F(\bar{\mathbf{x}}) = c_0 + \bar{\mathbf{c}}^t \bar{\mathbf{x}} \text{ subject to } \bar{\mathbf{A}} \bar{\mathbf{x}} = \mathbf{b}, \mathbf{b} \geq \mathbf{0}, \bar{\mathbf{x}} \geq \mathbf{0} \tag{9}$$

に移る。但し、(8)式から明らかなように、 $\bar{\mathbf{c}} = \begin{pmatrix} \mathbf{c} \\ \mathbf{0} \end{pmatrix}$, $\bar{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}$ は $(n + \beta)$ 次のベクトル、

$\bar{\mathbf{A}} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{A} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$ は $(m, n + \beta)$ 行列である。なお、系が縮退した (degenerated) 場合

を除外すれば、問題(7)式において、制約がすべて第3型であり、かつ新しく $m < n$ と置き換えて一般性は失われない。以下、この仮定で議論を進めよう。

まず変数 \mathbf{x} を基底変数 \mathbf{x}_B と非基底変数 \mathbf{x}_N に分割し、 $\mathbf{x} = \begin{pmatrix} \mathbf{x}_N \\ \mathbf{x}_B \end{pmatrix}$ となっているとする。これ

に対応して他の定数も分割すれば、(7)式は

$$\begin{aligned}
& \text{Minimize } F(\mathbf{x}) = c_0 + \mathbf{c}_N^t \mathbf{x}_N + \mathbf{c}_B^t \mathbf{x}_B \text{ subject to } \mathbf{A}_N \mathbf{x}_N + \mathbf{A}_B \mathbf{x}_B = \mathbf{b}, \\
& \mathbf{b} \geq \mathbf{0}, \mathbf{x}_N \geq \mathbf{0}, \mathbf{x}_B \geq \mathbf{0}
\end{aligned} \tag{10}$$

となる。ここに、 \mathbf{x}_N , \mathbf{x}_B はそれぞれ $(n - m)$, m 次のベクトルである。制約式より

$$\mathbf{x}_B = \mathbf{A}_B^{-1} \mathbf{b} - \mathbf{A}_B^{-1} \mathbf{A}_N \mathbf{x}_N \tag{11}$$

として、これを目的関数に代入すれば、

$$F(\mathbf{x}) = c_0 + \boldsymbol{\pi} \mathbf{b} + (\mathbf{c}_N^t - \boldsymbol{\pi} \mathbf{A}_N) \mathbf{x}_N \tag{12}$$

を得る。ここに、 $\boldsymbol{\pi} = \mathbf{c}_B^t \mathbf{A}_B^{-1}$ は m 次の行ベクトルで、評価 (price) ベクトルと呼ばれるものである。

さて、 $\mathbf{h} = \mathbf{c}_N^t - \pi \mathbf{A}_N \geq \mathbf{0}$ の場合を考えてみよう。 $c_0 + \pi b$ が一定であるから、最小値は $\mathbf{x}_N = \mathbf{0}$ で達成され、 $\mathbf{x} = \begin{pmatrix} \mathbf{0} \\ \mathbf{A}_B^{-1} \mathbf{b} \end{pmatrix}$ が解となる。

次に、 $\mathbf{h} = (h_1, \dots, h_{n-m})^t$ として、 $h_1 < 0$ と仮定しよう。このとき、Fは h_1 にかかる \mathbf{x}_N の要素 x_1 を大きくするほど小さくなる。従って、Fを最も効果的に減少させうるのは

$$h_e = \min \{ h_j \mid h_j < 0 \} \quad (13)$$

であり、これに対応する x_e を基底にとり入れる。このとき、 \mathbf{x}_N の他の要素を 0 として x_e がどのくらい大きくとれるかを考えてみる。(11)式において、 $\mathbf{A}_B^{-1} \mathbf{b}$ を $\mathbf{u} = (u_1, \dots, u_m)^t$ 、 $\mathbf{A}_B^{-1} \mathbf{A}_N$ の第 e 列を $\mathbf{v} = (v_1, \dots, v_m)^t$ とすれば、 $\mathbf{x}_B = \mathbf{u} - \mathbf{v} x_e \geq \mathbf{0}$ でなければならないから、 $v_i > 0$ なる要素に対して $u_i / v_i \geq x_e$ という制限を受ける。この最小値

$$w_g = \min \{ w_i = \frac{u_i}{v_i} \mid v_i > 0 \} \quad (14)$$

が x_e の上限であり、 w_g に対応する \mathbf{x}_B の変数 x_f 、即ち x_{n-m+g} に対して、 $x_e = w_g$ ならば $x_f = 0$ となる。このことは、 x_f を基底からとり除けば、

$$F(\mathbf{x}_N^1, \mathbf{x}_B^1) > F(\mathbf{x}_N^2, \mathbf{x}_B^2) \quad (15)$$

を意味する。ここに、 $\mathbf{x}_N^1 = (\dots, x_e, \dots)^t = \mathbf{0}$ 、 $\mathbf{x}_B^1 = (\dots, x_f, \dots)^t$ 、 $\mathbf{x}_N^2 = (\dots, x_f, \dots)^t = \mathbf{0}$ 、 $\mathbf{x}_B^2 = (\dots, x_e, \dots)^t$ である。

このように、Dantzig による単体法は許容領域である凸多面体の端点 (extreme point) を目的関数の値が減少するように探索していく^{1)~4)}。しかし、実際の計算では初期許容解が分っていないと基底の定め方が難しいので注意を要する。なお、既に述べたように $\mathbf{h} \geq \mathbf{0}$ となれば $\mathbf{x}_N = \mathbf{0}$ で最適解が得られるが、 $h_e < 0$ でかつすべての v_i が非正となったとき解が無限大 (infinite)、 $h_e < 0$ でかつ $w_g < 0$ となったときは不能 (nonfeasible) であるということになる。

参考のため、初期許容解が $\mathbf{x} = \begin{pmatrix} \mathbf{x}_N \\ \mathbf{x}_B \end{pmatrix}$ 、 $\mathbf{x}_N = \mathbf{0}$ 、 $\mathbf{x}_B = \mathbf{b}^1$ である場合の単体表 (tableau)

を Table 1 に示しておこう。Table 2 は、基底の入れ替えのために消去を行った後の様子を示している。

Table 1 Simplex tableau for initial feasible solution

\mathbf{x}_N				\mathbf{x}_B					
c_1^1	c_2^1	\cdots	c_{n-m}^1	0	0	\cdots	0	=	$-c_0^1$
a_{11}^1	a_{12}^1	\cdots	$a_{1,n-m}^1$	1	0	\cdots	0	=	b_1^1
a_{21}^1	a_{22}^1	\cdots	$a_{2,n-m}^1$	0	1	\cdots	0	=	b_2^1
\vdots	\vdots		\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots
a_{m1}^1	a_{m2}^1	\cdots	$a_{m,n-m}^1$	0	0	\cdots	1	=	b_m^1

Table 2 Simplex tableau after the base was transformed

\mathbf{x}_N							\mathbf{x}_B			
e							f			
c_1^2	c_2^2	\cdots	0	\cdots	c_{n-m}^2		0	c_1^2	\cdots	$-c_0^2$
a_{11}^2	a_{12}^2	\cdots	0	\cdots	$a_{1,n-m}^2$		1	a_{1f}^2	\cdots	b_1^2
a_{21}^2	a_{22}^2	\cdots	0	\cdots	$a_{2,n-m}^2$		0	a_{2f}^2	\cdots	b_2^2
\vdots	\vdots				\vdots		\vdots	\vdots	\vdots	\vdots
g) a_{g1}^2	a_{g2}^2	\cdots	1	\cdots	$a_{g,n-m}^2$		0	a_{gf}^2	\cdots	b_g^2
\vdots	\vdots				\vdots		\vdots	\vdots	\vdots	\vdots
a_{m1}^2	a_{m2}^2	\cdots	0	\cdots	$a_{m,n-m}^2$		0	a_{mf}^2	\cdots	b_m^2

2.2 初期許容解の求め方

単体法で必要な初期許容解は通常、乗数 (multiplier) 法または多段 (multiphase) 法で求められる。簡単のため

$$\begin{aligned}
 & \text{Minimize } F(\mathbf{x}) = c_1 x_1 + c_2 x_2 \quad \text{subject to} \\
 & a_{11} x_1 + a_{12} x_2 \leq b_1 \geq 0 \\
 & a_{21} x_1 + a_{22} x_2 \geq b_2 \geq 0 \\
 & a_{31} x_1 + a_{32} x_2 = b_3 \geq 0 \\
 & x_1, x_2 \geq 0
 \end{aligned} \tag{16}$$

の例で説明しよう。

まず乗数法の場合、十分大きな正の乗数 $M^3)$ 、付加変数 $\mathbf{y} = (y_1, y_2)^t$ および人為 (artificial) 変数 $\mathbf{z} = (z_1, z_2, z_3)^t$ により(16)式は

$$\begin{aligned}
 & \text{Minimize } F(\mathbf{x}, \mathbf{z}) = c_1 x_1 + c_2 x_2 + M(z_1 + z_2 + z_3) \\
 & \text{subject to} \\
 & a_{11} x_1 + a_{12} x_1 + y_1 + z_1 = b_1 \\
 & a_{21} x_1 + a_{22} x_2 - y_2 + z_2 = b_2 \\
 & a_{31} x_1 + a_{32} x_2 + z_3 = b_3 \\
 & x_1, x_2, y_1, y_2, z_1, z_2, z_3 \geq 0
 \end{aligned} \tag{17}$$

と変形される。前と同様な記法によれば、この問題の初期許容解は $\mathbf{x} = \mathbf{y} = \mathbf{0}$, $\mathbf{z} = \mathbf{b}$ である。この目的関数は、 M が大きいため、まず $\mathbf{z} = \mathbf{0}$ を与える許容解 $\mathbf{x}_1, \mathbf{y}_1$ に対する値 $F(\mathbf{x}_1, \mathbf{0}) = \mathbf{c}^t \mathbf{x}_1$ まで減少し、次に $F(\mathbf{x}, \mathbf{z})$ の最小値 $F(\tilde{\mathbf{x}}, \mathbf{0}) = \mathbf{c}^t \tilde{\mathbf{x}}$ に達する。ここに、 $\mathbf{z} = \mathbf{0}$ となった時点の \mathbf{x}_1 が(16)式の初期許容解であり、最小値を与える $\tilde{\mathbf{x}}$ は(17)式の最適解でもある。

乗数法を、初期許容解を求めるためだけに使う場合、(17)式で $\mathbf{c} = \mathbf{0}$, $M = 1$ とすればよいことは自明であろう。更に、 $b_2, b_3 > 0$ として(17)式を

Minimize z subject to

$$a_{11}x_1 + a_{12}x_2 + y_1 = b_1$$

$$\frac{a_{21}}{b_2}x_1 + \frac{a_{22}}{b_2}x_2 - \frac{1}{b_2}y_2 + z = 1 \quad (18)$$

$$\frac{a_{31}}{b_3}x_1 + \frac{a_{32}}{b_3}x_2 + z = 1$$

と変形すれば、初期許容解は $\mathbf{x} = \mathbf{0}$, $y_1 = b_1$, $y_2 = 0$, $z = 1$ であり、人為変数も少なくて済む。また、 $b_2 = 0$ のときなどは対応する式の z をとり除けばよい。

多段法の場合は次のようにする。まず第1型の制約に注目すると、 $\mathbf{x} = \mathbf{0}$ は自明な初期許容解である。それ故、第1段階として第1型の制約を制約条件とし、第2型の最初の制約式から目的関数を形成し、問題

$$\text{Minimize } F_2(\mathbf{x}) = b_2 - a_{21}x_1 - a_{22}x_2 \quad \text{subject to}$$

$$a_{11}x_1 + a_{12}x_2 \leq b_1 \quad (19)$$

$$x_1, x_2 \geq 0$$

を解く。このとき、最適解 $\tilde{\mathbf{x}}$ に対し $F_2(\tilde{\mathbf{x}}) \leq 0$ となる。もし、 $F_2(\tilde{\mathbf{x}}) > 0$ となれば、それは問題(16)式が不能であったことになり、 $F_2(\tilde{\mathbf{x}}) = -\infty$ となれば、 $F_2(\mathbf{x}) < 0$ となった状態で止める³⁾。従って、我々は(16)式の初めの2つの制約をみたす \mathbf{x} を見つかったことになり、この \mathbf{x} に対して $b_3 - a_{31}x_1 - a_{32}x_2 \geq 0$ ならば第2段階の問題

$$\text{Minimize } F_3(\mathbf{x}) = b_3 - a_{31}x_1 - a_{32}x_2 \quad \text{subject to}$$

$$a_{11}x_1 + a_{12}x_2 \leq b_1 \quad (20)$$

$$a_{21}x_1 + a_{22}x_2 \geq b_2$$

$$x_1, x_2 \geq 0$$

を解けばよい。なお、第1型の制約が無い場合は $\mathbf{x} = \mathbf{0}$ を初期許容解として

$$\text{Minimize } F_2(\mathbf{x}) = b_2 - a_{21}x_1 - a_{22}x_2 \quad \text{subject to} \quad (21)$$

$$a_{21}x_1 + a_{22}x_2 \leq b_2, \quad x_1, x_2 \geq 0$$

として解くこともできよう。

多段法が乗数法と異なる点として

- ① 人為変数が不要である。
- ② 単体法を繰り返し使うことによって解ける。
- ③ 計算中の桁あふれを心配しなくて済む。
- ④ 計算時間がかかる。

などがあげられる³⁾。

2.3 分解原理

変数や制約の多い、いわゆる大規模な線形システムでは単体法で解くのに困難な場合が生じて来る。大規模では特に非零要素の少ない疎な(sparse)システムには相応の工夫が必要であろう。システム全体が各部分システムから構成されているような場合には Dantzig-Wolfe による分解原理が有効に働く^{1)~7)}。その一般形は

$$\begin{aligned} \text{Minimize } F(\mathbf{x}) &= c_0 + \sum_{k=1}^l c_k^t \mathbf{x}_k \quad \text{subject to} \\ \sum_{k=1}^l A_k \mathbf{x}_k &\leq b \geq \mathbf{0}, \quad B_k \mathbf{x}_k \leq b_k \geq \mathbf{0} \quad (k = 1 \sim l), \\ \mathbf{x}_k &\geq \mathbf{0} \quad (k = 1 \sim l) \end{aligned} \quad (22)$$

であり、全体制約(constraint of whole system)および部分制約(constraint of subsystem)において、各制約はそれぞれ第1型から第3型の順に並んでいるものとする。いま、その個数を $p_1 \sim p_3$ 個、 $p_1^k \sim p_3^k$ 個 ($k = 1 \sim l$)、 $q = p_1 + p_2 + p_3$ 、 $q_k = p_1^k + p_2^k + p_3^k$ ($k = 1 \sim l$)、 $m = q + \sum_{k=1}^l q_k$ とし、変数の数を r_k ($k = 1 \sim l$)、 $n = \sum_{k=1}^l r_k$ とすれば、(22)式対応する単体表は Table 3 のようになる。ここでも(8)式と同様に付加変数を導入すれば、(22)式は

$$\begin{aligned} \text{Minimize } F(\mathbf{x}) &= c_0 + \sum_{k=1}^l c_k^t \mathbf{x}_k \quad \text{subject to} \\ \sum_{k=1}^l A_k \mathbf{x}_k + J \mathbf{y} &= b, \quad B_k \mathbf{x}_k + J_k \mathbf{y}_k = b_k, \\ \mathbf{x}_k, \mathbf{y}, \mathbf{y}_k &\geq \mathbf{0} \end{aligned} \quad (23)$$

となる。ここに、 J は全体制約の付加変数に対応する ($q, p_1 + p_2$) 行列であり、 J_k は各部分制約の付加変数に対応する同様の行列である。

さて、この問題を各部分システムの許容領域 D_k の頂点(cornor point) \mathbf{x}_{kj} ($j = 1 \sim s_k$) を使って変形してみよう。この頂点と拡張された表現(23)式の部分システムの許容領域 \overline{D}_k の頂点 $\overline{\mathbf{x}}_{kj}$ の間には次の関係があることに気付く。

- ① \overline{D}_k の頂点の \mathbf{x} 成分と D_k の頂点は対応している。
- ② 従って、頂点の数はともに s_k に等しい。
- ③ $\overline{\mathbf{x}}_{kj}$ は \overline{D}_k の境界を表わすすべての等式の上にあるが、 \mathbf{x}_{kj} は D_k の境界を表わす一部の式の上にある。

以上のことから、 D_k および \overline{D}_k の内点 $\mathbf{x}_k, \overline{\mathbf{x}}_k$ は、それぞれの頂点により

$$\mathbf{x}_k = \sum_{j=1}^{s_k} \alpha_{kj} \mathbf{x}_{kj}, \quad \overline{\mathbf{x}}_k = \sum_{j=1}^{s_k} \alpha_{kj} \overline{\mathbf{x}}_{kj}, \quad \sum_{j=1}^{s_k} \alpha_{kj} = 1, \quad \alpha_{kj} \geq 0 \quad (24)$$

Table 3 Simplex tableau for original problem

	x_1	x_2	\dots	x_ℓ	n	1
1	c_1	c_2	\dots	c_ℓ	$= -c_0$	
p_1					\leq	
p_2	A_1	A_2	\dots	A_ℓ	\geq	b
p_3					$=$	
p_1^1					\leq	
p_2^1	B_1				\geq	b_1
p_3^1					$=$	
p_1^2					\leq	
p_2^2	B_2				\geq	b_2
p_3^2					$=$	
p_1^ℓ					\leq	
p_2^ℓ				B_ℓ	\geq	b_ℓ
p_3^ℓ					$=$	

と表わせる。従って、(23)式は

$$\begin{aligned} \text{Minimize } F(\bar{\mathbf{x}}) &= c_0 + \sum_{k=1}^l \sum_{j=1}^{s_k} \bar{\mathbf{c}}_k^T \bar{\mathbf{x}}_{kj} \alpha_{kj} \quad \text{subject to} \\ \sum_{k=1}^l \sum_{j=1}^{s_k} \bar{\mathbf{A}}_k \bar{\mathbf{x}}_{kj} \alpha_{kj} + \mathbf{J} \mathbf{y} &= \mathbf{b}, \quad \sum_{j=1}^{s_k} \bar{\mathbf{B}}_k \bar{\mathbf{x}}_{kj} \alpha_{kj} = \mathbf{b}_k \\ \sum_{j=1}^{s_k} \alpha_{kj} &= 1 \quad (k = 1 \sim l), \quad \alpha_{kj} \geq 0 \quad (k = 1 \sim l, j = 1 \sim s_k) \end{aligned} \quad (25)$$

となり、 $\bar{\mathbf{c}}_k$ および $\bar{\mathbf{A}}_k$ の拡張された部分が 0 であることや $\bar{\mathbf{B}}_k \bar{\mathbf{x}}_{kj} = \mathbf{b}_k$ を考慮すれば、更に

$$\begin{aligned} \text{Minimize } F(\alpha) &= c_0 + \sum_{k=1}^l \sum_{j=1}^{s_k} d_{kj} \alpha_{kj} \quad \text{subject to} \\ \sum_{k=1}^l \sum_{j=1}^{s_k} \mathbf{p}_{kj} \alpha_{kj} + \mathbf{J} \mathbf{y} &= \mathbf{b}, \quad \sum_{j=1}^{s_k} \alpha_{kj} = 1 \quad (k = 1 \sim l), \\ \alpha_{kj} &\geq 0 \quad (k = 1 \sim l, j = 1 \sim s_k) \end{aligned} \quad (26)$$

と変形される。ここに、 $d_{kj} = \bar{\mathbf{c}}_k^T \bar{\mathbf{x}}_{kj}$ は定数、 $\mathbf{p}_{kj} = \bar{\mathbf{A}}_k \bar{\mathbf{x}}_{kj}$ は q 次の定ベクトルであり、 α_{kj} が変数である。これを元の問題に対して端点問題 (extremal problem) と呼ぶことにしよう。

Table 4 に示されるように、端点問題では、付加されたものを除く変数の数が n から $s = \sum_{k=1}^l s_k$

に増えているが、制約式の数は逆に m 個から $(q+l)$ 個に減っている。線形最適化問題を解く労苦が主として制約式の数に依存することを思うと、このことは計算の簡略化にかなり役立つものと思われる²⁾。

次に端点問題を解くアルゴリズムを考察しよう。まず問題(26)式を解くにあたり必要なことは、独立な $(q+l)$ 個の基底を見つけることである。しかし、注意しなければならないのは、係数 \mathbf{p}_{kj} 、即ち頂点 $\bar{\mathbf{x}}_{kj}$ が初めから陽に与えられていないことである。説明を簡明にするため、一般的な議論は後の章に譲るとして、ここではすべての制約が第 1 型であると仮定しよう。

第 1 型の定義から自明なように、元の問題(22)式に対して $\mathbf{x}_k = \mathbf{0}$ ($k = 1 \sim l$) は各部分システムの許容領域 D_k の頂点であり、かつ全体制約と部分制約を同時にみたす 1 つの許容解である。制約が第 1 型のものばかりということから $q = p_1, p_2 = p_3 = 0, \mathbf{y} = \mathbf{y}_1$ であり、 $\mathbf{x}_{k1} = \mathbf{0}$ とすれば $\mathbf{p}_{k1} = \mathbf{0}$ ($k = 1 \sim l$) である。従って、Table 5 に示されるように、Table 4において独立な $(q+l)$ 個の $(q+l)$ 次ベクトルが見つかったことになり、単体法による最適値の探索が開始できる。この α_{k1} に対応する l 個の基底は、Table 1 のように \mathbf{y} に対応する基底の右に置くと単体表の作成上都合が良い。

以下、ある探索の段階において s 個の変数 α_{kj} のうち、基底に採用されたものを $(k, j) \in B$ 、そうでないものを $(k, j) \in N$ などと表わすこととする。また、 α と \mathbf{y} の $(s+q)$ 個の変数のうち、基底を形成するものを β_B 、そうでないものを β_N とし、それに対応して定数ベクトル、係

Table 4 Simplex tableau for extremal problem

s_1	s_2	\dots	s_ℓ	p_1	p_2	1
α_1	α_2	\dots	α_ℓ	y_1	y_2	
1	d_1	d_2	\dots	d_ℓ	0	0
p_1				I	0	=
p_2	p_1	p_2	\dots	p_ℓ	0	-I
p_3					0	=
1	\dots	1				= 1
ℓ		1	\dots	1		= 1
					0	= 1
						= 1

Table 5 Independent vectors of order $(q+\ell)$ which compose the base

ℓ	\dots	q		
α_{11}	α_{21}	\dots	$\alpha_{\ell 1}$	y
1	0	0	\dots	0
q	0	0	\dots	0
ℓ	0	0	\dots	1
	1	0	\dots	0
	0	1	\dots	0
	0	0	\dots	1

数行列を分割したものをそれぞれ \mathbf{g}_N , \mathbf{g}_B , \mathbf{Q}_N , \mathbf{Q}_B とすれば、問題(26)式の単体表は、その段階で問題

$$\begin{aligned} \text{Minimize } F(\beta) - d_0 &= \mathbf{g}_N^t \beta_N + \mathbf{g}_B^t \beta_B \text{ subject to} \\ \mathbf{Q}_N \beta_N + \mathbf{Q}_B \beta_B &= \mathbf{f}, \quad \beta_N, \beta_B \geq 0 \end{aligned} \quad (27)$$

を表わしている。ここに、 $-d_0$ および \mathbf{f} の出発値はそれぞれ $-c_0$, $(\mathbf{b}^t, 1, \dots, 1)^t$ であり、 \mathbf{g}_B , \mathbf{Q}_B のそれは、それぞれ \mathbf{o} , \mathbf{I} である。

次のステップは最適性の判定である。前にも記したように、もし

$$\mathbf{h} = \mathbf{g}_N^t - (\pi, \tilde{\pi}) \mathbf{Q}_N \geq 0 \quad (28)$$

ならば、 $\beta_N = \mathbf{o}$ で最小値 $F(\beta) = d_0 + (\pi, \tilde{\pi}) \mathbf{f}$ をとる。ここでも行ベクトル $(\pi, \tilde{\pi}) = \mathbf{g}_B^t \mathbf{Q}_B^{-1}$ は評価ベクトルと呼ばれる。 $\mathbf{h} \geq \mathbf{o}$ でないときは、この最小要素 $\min_{(k,j) \in N} (d_{kj} - \pi p_{kj} - \tilde{\pi} e_k)$ に対応する列を非基底から除いてやる。そして、その組 $d_{kj}, \left(\begin{array}{c} p_{kj} \\ e_k \end{array} \right)$ を基底

側に移すことによって \mathbf{h} を段々大きくしていく。ここに e_k は k 番目の要素が 1 である l 次の単位ベクトルである。しかし、実際の計算ではこの基底に入れる添字 (k, j) は N と B の合併集合 $N \cup B$ の中から選んでよい。というのは、既に基底にあるものと一次従属になった場合、目的関数 F が改良されることに反するからである。従って、

$$\begin{aligned} \min_{(k,j) \in N} & (-\pi p_{kj} - \tilde{\pi} e_k + d_{kj}) \\ = \min_{(k,j) \in N \cup B} & (-\pi p_{kj} - \tilde{\pi} e_k + d_{kj}) \\ = \min_k & \{ \min_j (-\pi p_{kj} - \tilde{\pi} e_k + d_{kj}) \} \quad (k, j) \in N \cup B \end{aligned} \quad (29)$$

であり、

$$\begin{aligned} \text{Minimize } & r_k = (-\pi A_k + c_k^t) x_k - \tilde{\pi}_k \\ \text{subject to } & B_k x_k \leq b_k, \quad x_k \geq \mathbf{o} \end{aligned} \quad (30)$$

を $k = 1 \sim l$ について解いて

$$r_e = \min_{1 \leq k \leq l} r_k \quad (31)$$

とすればよい。ここに、 $\tilde{\pi}_k$ は $\tilde{\pi}$ の k 番目の要素であり、 r_e は \mathbf{h} の最小要素である。従って、 $r_e \geq 0$ ならば $\mathbf{h} \geq \mathbf{o}$ であり、 $r_e < 0$ なら e に対応している列を基底に入れる。このステップは、部分システム(30)式の許容領域 D_k がすべて有界 (bounded) のとき問題はないが、全制約に関して有界でありながら、ある r_k が無限大になることがある。そういう場合は

$$(-\pi \mathbf{A}_k + \mathbf{c}_k^t) \mathbf{x}_k < 0, \quad \mathbf{B}_k \mathbf{x}_k \leq \mathbf{b}_k \quad (32)$$

なる解 $\mathbf{x}_k \geq 0$ が存在するので、組 $\mathbf{c}_k^t \mathbf{x}_k, (\mathbf{A}_k \mathbf{x}_k, \mathbf{o})^t$ を基底に入れればよい³⁾。

第3のステップは基底から除く列を決めることがある。単体法と同様に、 $-r_e$ にかかる β_N の要素 $(\beta_N)_e$ を大きくとって目的関数 F を減らしたい。(11)式に対応する式

$$\beta_B = \mathbf{Q}_B^{-1} \mathbf{f} - \mathbf{Q}_B^{-1} \mathbf{Q}_N \beta_N \quad (33)$$

において、 \mathbf{Q}_N の列のうち必要なのは e に対応するものだけなので、 $\mathbf{u} = \mathbf{Q}_B^{-1} \mathbf{f} = (u_1, \dots, u_{q+l})^t$,

$\mathbf{Q}_B^{-1} \mathbf{Q}_N$ の第 e 列を $\mathbf{v} = (v_1, \dots, v_{q+l})^t$ として u_i, v_i より決められる。いま、

$w = \max_{1 \leq i \leq q+l} v_i \leq 0$ とすると、 β_B は自由なので解は無限大となる。 $w > 0$ のときは、

$i = 1 \sim q+l$ のうちの $v_i > 0$ なるものに対して $\delta_i = u_i / v_i$ としたとき

$$(\beta_N)_e \leq \delta_f = \min \delta_i \quad (34)$$

の制限を受ける。従って、f に対応する組を基底から取り除く。具体的には、単体法と同じように、 \mathbf{Q}_N の e に対応する組 $(g_N)_e, (\mathbf{Q}_N)_e^t$ を用い、その f に対応する要素を軸として単体表を消去すればよい。

以上より、 $r_e \geq 0$ となった場合、最小値は $\beta_N = \mathbf{o}$ により $F = d_0 + (\pi, \tilde{\pi}) \mathbf{f}$ として求まるが、元の問題の解 \mathbf{x} の復元は厄介である。まず $\beta_B = \mathbf{Q}_B^{-1} \mathbf{f}$ より基底に採用された α_{kj} ($k, j \in B$) を求める。そして、採用された頂点 \mathbf{x}_{kj} を記憶しておき、(24)式から

$$\mathbf{x}_k = \sum_{(k, j) \in B} \alpha_{kj} \mathbf{x}_{kj} \quad (35)$$

としなければならない。

3. 応用例

分解原理のアルゴリズムをより具体的に示し、その有効性を実証するために、簡単な例題および実用例を掲げよう。

3.1 簡単な問題

[例 1]

すべての制約が第1型であり、かつ変数の少ない例として Künzi の例³⁾は適切である。問題

$$\text{Maximize } F(\mathbf{x}) = 18 + x_1 + 8x_2 + \frac{1}{2}x_3 + \frac{3}{2}x_4$$

subject to

$$(-\pi \mathbf{A}_k + \mathbf{c}_k^t) \mathbf{x}_k < 0, \quad \mathbf{B}_k \mathbf{x}_k \leq \mathbf{b}_k \quad (32)$$

なる解 $\mathbf{x}_k \geq 0$ が存在するので、組 $\mathbf{c}_k^t \mathbf{x}_k, (\mathbf{A}_k \mathbf{x}_k, \mathbf{o})^t$ を基底に入れればよい³⁾。

第3のステップは基底から除く列を決めることがある。単体法と同様に、 $-r_e$ にかかる β_N の要素 $(\beta_N)_e$ を大きくとって目的関数 F を減らしたい。(11)式に対応する式

$$\beta_B = \mathbf{Q}_B^{-1} \mathbf{f} - \mathbf{Q}_B^{-1} \mathbf{Q}_N \beta_N \quad (33)$$

において、 \mathbf{Q}_N の列のうち必要なのは e に対応するものだけなので、 $\mathbf{u} = \mathbf{Q}_B^{-1} \mathbf{f} = (u_1, \dots, u_{q+l})^t$,

$\mathbf{Q}_B^{-1} \mathbf{Q}_N$ の第 e 列を $\mathbf{v} = (v_1, \dots, v_{q+l})^t$ として u_i, v_i より決められる。いま、

$w = \max_{1 \leq i \leq q+l} v_i \leq 0$ とすると、 β_B は自由なので解は無限大となる。 $w > 0$ のときは、

$i = 1 \sim q+l$ のうちの $v_i > 0$ なるものに対して $\delta_i = u_i / v_i$ としたとき

$$(\beta_N)_e \leq \delta_f = \min \delta_i \quad (34)$$

の制限を受ける。従って、f に対応する組を基底から取り除く。具体的には、単体法と同じように、 \mathbf{Q}_N の e に対応する組 $(\mathbf{g}_N)_e, (\mathbf{Q}_N)_e^t$ を用い、その f に対応する要素を軸として単体表を消去すればよい。

以上より、 $r_e \geq 0$ となった場合、最小値は $\beta_N = \mathbf{o}$ により $F = d_0 + (\pi, \tilde{\pi}) \mathbf{f}$ として求まるが、元の問題の解 \mathbf{x} の復元は厄介である。まず $\beta_B = \mathbf{Q}_B^{-1} \mathbf{f}$ より基底に採用された $\alpha_{kj} (k, j) \in B$ を求める。そして、採用された頂点 \mathbf{x}_{kj} を記憶しておき、(24)式から

$$\mathbf{x}_k = \sum_{(k, j) \in B} \alpha_{kj} \mathbf{x}_{kj} \quad (35)$$

としなければならない。

3. 応用例

分解原理のアルゴリズムをより具体的に示し、その有効性を実証するために、簡単な例題および実用例を掲げよう。

3.1 簡単な問題

[例 1]

すべての制約が第1型であり、かつ変数の少ない例として Künzi の例³⁾は適切である。問題

$$\text{Maximize } F(\mathbf{x}) = 18 + x_1 + 8x_2 + \frac{1}{2}x_3 + \frac{3}{2}x_4$$

subject to

$$\begin{aligned}
 & x_1 + 4x_2 + \frac{7}{2}x_3 + \frac{1}{2}x_4 \leq 1 \\
 & 2x_1 + 3x_2 \leq 6 \\
 & 5x_1 + x_2 \leq 5 \\
 & 3x_3 - x_4 \leq 12 \\
 & -3x_3 + x_4 \leq 0 \\
 & x_3 \leq 4 \\
 & x_1 \sim x_4 \geq 0
 \end{aligned} \tag{36}$$

を最小問題に直し、 $\mathbf{x}_{11} = (0, 0)^t$, $\mathbf{x}_{21} = (0, 0)^t$ が初期許容解であることに気付けば、最初の基底は Table 6 のようになる。この問題では $\mathbf{c}_1^t = (-1, -8)$, $\mathbf{c}_2^t = (-\frac{1}{2}, -\frac{3}{2})$,

$$\mathbf{A}_1 = (1, 4), \quad \mathbf{A}_2 = (\frac{7}{2}, \frac{1}{2}), \quad \mathbf{B}_1 = \begin{pmatrix} 2 & 3 \\ 5 & 1 \end{pmatrix}, \quad \mathbf{B}_2 = \begin{pmatrix} 3 & -1 \\ -3 & 1 \\ 1 & 0 \end{pmatrix} \text{ であるので,}$$

$\boldsymbol{\pi} = (0)$, $\tilde{\boldsymbol{\pi}} = (0, 0)$ を使って各部分システム(30)式を解くと $\mathbf{x}_{12} = (0, 2)^t$, $r_1 = -16$, $\mathbf{x}_{22} = (4, 12)^t$, $r_2 = -20$ となる。従って、 \mathbf{x}_{22} の方を基底に採り入れる。一方、 $d_{22} = \mathbf{c}_2^t \mathbf{x}_{22} = -20$, $(\mathbf{Q}_N)_e = (\mathbf{p}_{22}, 0, 1)^t = (20, 0, 1)^t$ であるから $\mathbf{Q}_B^{-1} = \mathbf{I}$ より $\mathbf{u} = (1, 1, 1)^t$, $\mathbf{v} = (20, 0, 1)^t$ となる。このとき、 $\delta_f = 1/20$ となり、 \mathbf{y} を基底から除く。従って単体表は組 $(-20, 20, 0, 1)^t$ の 20 を軸に消去され Table 7 を得る。この表では、組の交換を行い、消去直前の状態を

$$\left(\begin{array}{ccc|c} -20 & 0 & 0 & 18 \\ \hline 20 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{array} \right) = \left(\begin{array}{cc|c} \mathbf{g}_B^t & -d_0 \\ \hline \mathbf{Q}_B & \mathbf{f} \end{array} \right) \tag{37}$$

としたとき、 $\mathbf{R} = \begin{pmatrix} 1 & \mathbf{g}_B^t \\ \mathbf{o} & \mathbf{Q}_B \end{pmatrix}$ の逆行列 $\mathbf{R}^{-1} = \begin{pmatrix} 1 & -\mathbf{g}_B^t \mathbf{Q}_B^{-1} \\ \mathbf{o} & \mathbf{Q}_B^{-1} \end{pmatrix}$ により

$$\left(\begin{array}{cc|c} -(\boldsymbol{\pi}, \tilde{\boldsymbol{\pi}}) & \mathbf{R}^{-1}(-d_0) \\ \hline \mathbf{Q}_B^{-1} & \mathbf{f} \end{array} \right)$$

次の反復では、 $(\boldsymbol{\pi}, \tilde{\boldsymbol{\pi}}) = (-1, 0, 0)$ として 2 つの部分システムを解くと、 $\mathbf{x}_{13} = (0, 2)^t$, $r_1 = -8$, $\mathbf{x}_{23} = (0, 0)^t$, $r_2 = 0$ を得る。 \mathbf{x}_{13} から組 $(-16, 8, 1, 0)^t$ を採用して $\mathbf{u} = (1/20, 1, 19/20)^t$, $\mathbf{y} = \mathbf{Q}_B^{-1}(8, 1, 0)^t = (8/20, 1, -8/20)^t$ より取り除く変数を計算すれば α_{22} となる。 $\mathbf{R}^{-1}(-16, 8, 1, 0)^t = (-8, 8/20, 1, -8/20)^t$ より $8/20$ を軸に消去すれば、Table 8 となる。

更に次の反復では、 $(\boldsymbol{\pi}, \tilde{\boldsymbol{\pi}}) = (-2, 0, 0)$ として計算するが、その結果は $r_l = 0$ であり、反復が収束したことが分る。従って、最小値は $F = -20$ である。また、 $\alpha_{13} = 1/8$, $\alpha_{23} = 7/8$, $\alpha_{21} = 1$ と、対応する頂点 \mathbf{x}_{13} , \mathbf{x}_{23} , \mathbf{x}_{21} を(35)式に代入すると、 $\mathbf{x}_1 = (0, 1/4)^t$, $\mathbf{x}_2 = (0, 0)^t$ を得る。以上の過程は付録にプログラム DEPRIM の出力としても掲げてある

で参照されたい。

Table 6 Starting base for Künzi's problem

y	α_{11}	α_{21}	
0	0	0	18
1	0	0	1
0	1	0	1
0	0	1	1

Table 7 Simplex tableau after the first elimination was performed

f	e	α_{22}	α_{11}	α_{21}	
y					
0	1	0	0	0	19
1	1/20	0	0	0	1/20
0	0	1	0	0	1
0	-1/20	0	1	0	19/20

Table 8 Simplex tableau after the last elimination was performed

f	e	α_{22}	α_{13}	α_{11}	α_{21}	
0	2	0	0	0	0	20
1	1/8	0	0	0	0	1/8
0	-1/8	1	0	0	0	7/8
0	0	0	1	0	0	1

(例 2)

次に、現実的な例として輸送問題を考えてみよう。Fig. 1 に示されるように²⁾、工場 i の製品を消費地 j に輸送する量を x_{ij} とする。このとき、全生産量を a_i 、全消費量を b_j 、輸送費用の単価を c_{ij} とすると、費用を最小にする輸送方法は

$$\text{Minimize } F(\mathbf{x}) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad \text{subject to } \sum_{j=1}^n x_{ij} = a_i, \quad (38)$$

$$\sum_{i=1}^m x_{ij} = b_j, \quad x_{ij} \geq 0 \quad (i = 1 \sim m, j = 1 \sim n)$$

から求まる。いま、変数 x が、 $((x_{ij}, j = 1 \sim n), i = 1 \sim m)$ の順に並んでいるとすれば、最初の生産量に関する制約が部分制約となり、後の消費量の制約が全体制約となる。もっともこの例では、 x を $((x_{ij}, i = 1 \sim m), j = 1 \sim n)$ の順に並べれば、制約の役割が逆になる。

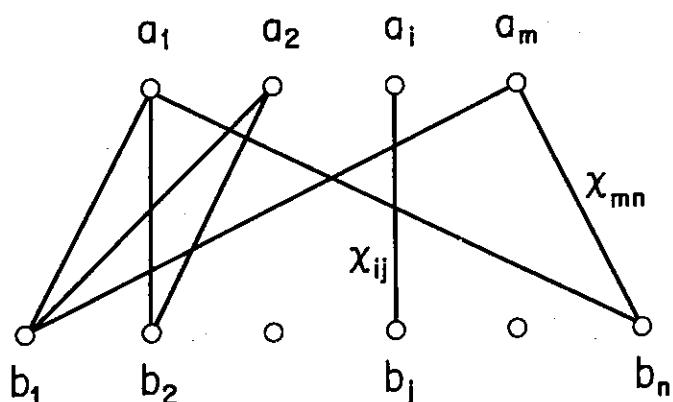


Fig. 1 Transportation problem of products.

The x_{ij} is the number of the products that are transported from the i -th factory to the j -th consuming place.

3.2 エネルギー・モデルへの応用

我々が手にすることことができた実用的な例として、日本における一つのエネルギー・モデルを取上げる。モデルの作成および試算は、既にNIRA(National Institute for Research Advancement, 総合研究開発機構)で行われているが⁸⁾、ここではそのデータを多少修正して用いている。

詳しくは文献8)を参照することにして、エネルギー・フローから考察しよう。一般に、エネルギー供給源として、水力、地熱、核燃料、石炭、石油、LNG(液化天然ガス)の6種類が考えられる。これらは、変換や精製を経て電力や石油製品等となり、産業や民生部門の需要を満たす。例えば、核燃料の一部はLWR(軽水炉)という変換手段を通して電力という中間エネルギー(intermediate energy)となり、その一部は照明や動力源として使われる。ここに、中間エネルギーの電力は、その発電設備建設量が供給量を制約する要因となるため、発電形態ごとに独立の供給源と考える。そして、その電力セクターを石炭火力、石油火力、LNG火力、LWR、水力、揚水、ガス・タービン、地熱、FBR(高速増殖炉)、HTGR(高温ガス炉)の10種類、非電力セクターを、石炭、石油、LNG、LPG(プロパン・ガス)、都市ガスの5種類とし、合計15種類に分類する。

次に、エネルギー需要は、産業部門の電力需要、産業部門の熱需要、民生部門、輸送部門に大別する。産業部門の電力需要は、産業別にしないで、電力負荷率により、基底負荷、中間負荷、尖峰負荷の3つと電力貯蔵に分類する。一方、産業部門の熱需要は、消費の大きい鉄鋼、化学、

そしてエネルギー多消費産業、中消費産業、少消費産業の5種類とする。また、民生部門は暖房、冷房、給湯・厨房、照明・動力の4種類、輸送部門は船舶、航空、バス・トラック、鉄道、乗用車の5種類とする。従って、エネルギー需要は合計18種類である。

以上より、(15, 18)のエネルギー・マトリックスが考えられるが、このうち実際に供給が行われる120種類のフローを中間エネルギー変数とする。そして、1977・5年から2002・5年の25年間を対象期間とし、全期間を各期5年の5つのタイム・ステップに分割し、各期の代表年を1980年、1985年、1990年、1995年、2000年とする。このとき、各期の中間エネルギー変数を x_j^k ($k = 1 \sim 5, j = 1 \sim 120$)、発電設備建設量を x_j^k ($j = 121 \sim 130$) コスト係数を c_j^k ($j = 1 \sim 130$) とすれば、線形モデルにより

$$\text{Minimize } F(\mathbf{x}) = \sum_{k=1}^5 \mathbf{c}_k^t \mathbf{x}_k \quad \text{subject to} \quad \sum_{k=1}^5 \mathbf{A}_k \mathbf{x}_k \leq \mathbf{b} \geq \mathbf{0} \quad (39)$$

$$\mathbf{B}_k \mathbf{x}_k \geq \mathbf{b}_k \geq \mathbf{0}, \quad \mathbf{x}_k \geq \mathbf{0} \quad (k = 1 \sim 5)$$

と書ける。ここに、 $\mathbf{x}_k = (x_1^k, \dots, x_{130}^k)^t$, $\mathbf{c}_k = (c_1^k, \dots, c_{130}^k)^t$ であり、 F はコストの評価関数、 \mathbf{A}_k , \mathbf{B}_k はそれぞれ、全体および部分制約の係数行列、 \mathbf{b} , \mathbf{b}_k はそれぞれ、全体および部分制約量である。全体制約は資源供給量制約を中心とし、部分制約は需要量、資源供給量、発電設備建設量、最大電力、電力量、揚水発電バランス、機器分担率、代替の各制約である。

今回の計算では、全体制約の数を q 、各部分制約の数を q_k 、そしてそれらを各々第1型、第2型、第3型の制約に分けて $p_1 \sim p_3$, $p_1^k \sim p_3^k$ としたとき、 $p_1 = 10$, $p_2 = p_3 = 0$, $p_1^1 = 36$, $p_2^1 = 27$, $p_3^1 = 17$, $p_1^k = 6$, $p_2^k = 6$, $p_3^k = 17$ ($k = 2 \sim 5$) としている。即ち、 $q = 10$, $q_1 = 80$, $q_k = 29$ ($k = 2 \sim 5$) である。

第2章第3節では、分解原理が応用可能な一般的な問題を(22式)で掲げ、制約が第1型のみからなる場合に限ってその解法を説明した。(39式)で与えられるエネルギー・モデルの問題は、確かに(22式)の一般形に含まれるが、部分制約が第1型でないため、様々な解法上の違いが生じる。

最も大きな相違は、初期許容解が不明なことである。そこで、(18式)による2段(two phase)法の概念を応用するため、(39式)の各部分制約に対して、付加変数 y_j^k ($j = 1 \sim p_2^k$) と人為変数 z_k を導入し、問題

$$\begin{aligned} \text{Minimize } G(\mathbf{z}) &= \sum_{k=1}^5 z_k \quad \text{subject to} \quad \sum_{k=1}^5 \mathbf{A}_k \mathbf{x}_k \leq \mathbf{b} \geq \mathbf{0}, \\ \mathbf{B}_1^k \mathbf{x}_k &\leq \mathbf{b}_1^k \geq \mathbf{0}, \quad \mathbf{B}_2^k \mathbf{x}_k - \mathbf{y}_k + \mathbf{E}_2^k \mathbf{z}_2^k = \mathbf{e}_2^k, \quad \mathbf{B}_3^k \mathbf{x}_k + \mathbf{E}_3^k \mathbf{z}_3^k = \mathbf{e}_3^k \\ \mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_2^k, \mathbf{z}_3^k &\geq \mathbf{0} \quad (k = 1 \sim 5) \end{aligned} \quad (40)$$

を考える。ここでは、 \mathbf{B}_j^k ($j = 1 \sim 3$) はそれぞれ \mathbf{B}_k の第1型、第2型、第3型に対応する部分、 \mathbf{y}_k は付加変数 y_j^k のベクトル、 \mathbf{z}_j^k ($j = 2 \sim 3$) は人為変数 z の第2型および第3型に対応する部分である。また、 \mathbf{E}_2^k は次数 p_2^k の対角行列、 \mathbf{e}_2^k は p_2^k 次のベクトルであり、部分制約量を表わすベクトル \mathbf{b}_k の第2型の制約を表わす部分 \mathbf{b}_2^k において、 i 番目の要素が0であれば、 \mathbf{E}_2^k および \mathbf{e}_2^k の*i*番目の要素を0、正であれば、それぞれ1とおいたものである。 \mathbf{E}_3^k , \mathbf{e}_3^k も同様である。(40式)から明らかのように、この初期許容解は $\mathbf{x}^k = \mathbf{0}$, $\mathbf{y}_k = \mathbf{0}$, $\mathbf{z}_j^k = (1, \dots, 1)^t$ ($k = 1 \sim 5$)

$1 \sim 5, j = 2 \sim 3$) であり、この問題自身にも分解原理が適用できて、最適解 $\mathbf{z}_j^k = \mathbf{0}$ ($k = 1 \sim 5, j = 2 \sim 3$) が得られる。従って、このときの解を \mathbf{x}_{kj} ($k = 1 \sim 5$) として(26)式の \mathbf{p}_{kj} を計算すれば、前と同様に、Table 4で (10 + 5) 個の基底が得られ、問題(39)式の探索が開始できる。

いま一つの相違は、部分システムの解法である。第1型の制約ばかりのときは、単体法を変形した改良(revised)単体法^{2) 3)}のプログラム RESEX^{3) 9)}で解かれていたが、問題(39)式では、一般的な単体法のプログラム SIMPLX^{3) 9)}で解かれる。

このエネルギー・モデルのデータは、Table 9に示されるように、大部分が0であり、数値の大きさに大きなひらきはない。しかし、変数の数が650、全ての制約の数が206と大規模なため、FACOM M 200計算機¹⁰⁾の单精度計算では有意な解が得られず、部分システムのみ SIMPLD⁹⁾による倍精度計算が行われた。

Table 9 Coefficients of problem derived from an energy model in Japan

Coefficient	Sign	Range of order	Nonzero percent
\mathbf{c}_k	0 +	$10^1 \sim 10^2$	< 95
\mathbf{A}_k	- 0 +	$10^{-1} \sim 10^0$	< 5
\mathbf{b}	0 +	$10^{-1} \sim 10^3$	< 50
\mathbf{B}_k	- 0 +	$10^{-2} \sim 10^1$	< 10
\mathbf{b}_k	0 +	$10^{-2} \sim 10^2$	< 60

この問題は、部分制約に第2型や第3型の制約を含んでおり、DEPRIやDEPRIMで解くことができないので、別のプログラムを開発して解かれた。そのプログラムは、現在最適に設計されているとは言えず、公表に至っていないが、参考までにこの問題に対する計算コストや精度を示しておこう。分解原理を応用しないで、倍精度プログラム SIMPLDでこの問題を解いたとき、記憶容量は1192 kB(キロバイト)、計算時間は1分21秒、解の最小値は 9.6549×10^4 であった。一方、分解原理により、部分システムの部分をSIMPLD、その他の部分を单精度計算で解いた場合、コストはそれぞれ、832 kB、1分14秒、最小値は 9.6548×10^4 であった。单精度計算のプログラム SIMPLXではこの問題が解けないと考えると、最適値に大きな差がなく、かつ少ない記憶容量で計算するために、分解原理の一つの特徴を活かせたと言えよう。

4. プログラムの使用法

線形システム(2)式において、全体制約および部分制約が第1型である場合にこの式を解くプログラム、DEPRI および DEPRIM が開発された。これらはともに副プログラムの形式をとっており、特に前者は計算の途中でも呼び出せるよう一般的にできている。一方、後者はこの型の問題のみを計算する場合に便利であり、入力データをカードまたはファイルで繰り返し与えれば、配列宣言文、CALL文のみで何ケースもの計算ができる。

以下、JSSL（原研版・科学用サブルーチン・ライプラリ）マニュアル¹¹⁾の書式に従い、これらの使用法を示す。特に、項目(2)の説明では、変数の型はすべて暗黙の型に従い、実数型はすべて単精度とする。また、入力と記された変数はすべて、計算後も内容が変わらないものとする。

4.1 DEPRI

(1) 機能と特徴

問題

$$\text{Minimize } F(\mathbf{x}) = c_0 + \sum_{k=1}^l \mathbf{c}_k^T \mathbf{x}_k \quad \text{subject to} \quad \sum_{k=1}^l \mathbf{A}_k \mathbf{x}_k \leq \mathbf{b} \geq \mathbf{0},$$

$$\mathbf{B}_k \mathbf{x}_k \leq \mathbf{b}_k \geq \mathbf{0}, \quad \mathbf{x}_k \geq \mathbf{0} \quad (k = 1 \sim l)$$

を解く。但し、 $\mathbf{c}_k = (c_j^k)$ 、 $\mathbf{x}_k = (x_j^k)$ は r_k 次のベクトル、 $\mathbf{A}_k = (a_{ij}^k)$ は (q, r_k) 行列、 $\mathbf{b} = (b_i)$ は q 次のベクトル、 $\mathbf{B}_k = (b_{ij}^k)$ は (q_k, r_k) 行列、 $\mathbf{b}_k = (b_i^k)$ は q_k 次ベクトルとする。このプログラムを使うと、一般の改良单体法のプログラムと比べ、記憶容量や計算時間が節約できる。

(2) 呼び出し方

CALL DEPRI (A, JA, B, JB, X, JX, NPAR, NCWS, IZA, ISA, NCRT, JNCRT, NVAR, JNVAR, IZB, JIZB, ISB, JISB, LCB, JLBC, IPRT, JIPRT, IFALL, AL, JAL, DT, JDT, IPRT 1, JIPRT 1, IPRT 2, JIPRT 2, PT, JPT, CT, JCT, LCX, JLCX, EPS, AR, JAR, BR, JBR, CR, JCR, LW1, JLW1, LW2, JLW2, LW3, JLW3, UN, JUN, BASE, JBASE, BASEL, JBASEL, ICCT, JICCT, IPRINT)

A ————— 行列 \mathbf{A}_k やコスト・ベクトル \mathbf{c}_k などを格納する 1 次元配列の入力。 下の表の行列を行または列に沿って入れる。

$$\begin{matrix} & 1 & r_1 & r_2 & \cdots & r_l \\ q & \left(\begin{array}{c|c|c|c|c} c_0 & c_1^t & c_2^t & \cdots & c_l^t \\ \hline b & A_1 & A_2 & \cdots & A_l \end{array} \right) \end{matrix}$$

JA —— 配列Aの大きさを示す入力。

$$JA \geq (q + 1) \times (1 + \sum_{k=1}^l r_k) \text{ とする。}$$

B —— 部分制約を表わす行列 B_k などを格納する1次元配列の入力。下の表の行列を、部分制約ごとに行または列に沿って入れる。

$$\begin{matrix} & 1 & r_k \\ q_k & \left(\begin{array}{c|c} b_k & B_k \end{array} \right) \end{matrix}$$

JB —— 配列Bの大きさを示す入力。

$$NB \geq \sum_{k=1}^l q_k \times (1 + r_k) \text{ とする。}$$

X —— 解 x_k を格納する1次元配列の出力。X(1)に最小値 $F(x)$ 、X(2)～X($n + 1$)に解ベクトル $x_1 \sim x_l$ が outputされる。ここに、

$$n = \sum_{k=1}^l r_k \text{ である。}$$

JX —— 配列Xの大きさを示す入力。JX $\geq n + 1$ とする。

NPAR —— 部分形式の数 i を示す入力。NPAR ≥ 1 とする。

NCWS —— 全体制約式の数 q を示す入力。NCWS ≥ 1 とする。

IZA —— 合成された行列 $\left(\begin{array}{c|c|c|c} c_0 & c_1^t & \cdots & c_l^t \\ \hline b & A_1 & \cdots & A_l \end{array} \right)$ の要素を、配列Aに格納したときの行の隔りを示す入力。従って、行に沿って格納したとき $n + 1$ 、列に沿って格納したとき 1 とする。

ISA —— IZAと同様に列の隔りを示す入力。行に沿って格納したとき 1 、列に沿って格納したとき $q + 1$ とする。従って、合成された行列の (i, j) 要素は配列Aの $(i - 1) \times IZA + (j - 1) \times IZS + 1$ 番目に格納される。

NCRT —— 部分制約式の数 q_k を格納する1次元配列の入力。NCRT(k) = q_k とする。

JNCRT —— 配列 NCRT の大きさを示す入力。JNCRT $\geq l$ とする。

NVAR —— 部分制約における変数 x_j^k の数 r_k を格納する1次元配列の入力。NVAR(k) = r_k とする。

JNVAR —— 配列 NVAR の大きさを示す入力。NVAR $\geq l$ とする。

IZB —— 部分制約を表わす、合成された行列 (b_k, B_k) を、配列Bに格納したときの行の隔たりを示す1次元配列の入力。 k 番目の部分形式に関して、行に沿って格納したとき IZB(k) = $1 + r_k$ 、列に沿って格納したとき 1 とする。

JIZB —— 配列 IZB の大きさを示す入力。JIZB $\geq l$ とする。

ISB —— IZB と同様に列の隔たりを示す1次元配列の入力。行に沿って格納したとき 1 、列

に沿って格納したとき q_k とする。

JISB —— 配列 ISB の大きさを示す入力。 $JISB \geq l$ とする。

LCB —— 部分制約を表わす、合成された行列 ($\mathbf{b}_k, \mathbf{B}_k$) を配列 B に格納したとき、この先頭の要素 b_1^k の入っている 1 つ前の位置を示すための 1 次元配列の作業領域。従つ

$$\text{て}, LCB(1) = 0, LCB(k) = \sum_{j=1}^{k-1} q_j \times (1 + r_j) \quad (k = 2 \sim l) \text{ となっている。}$$

JLCB —— 配列 LCB の大きさを示す入力。 $JLCB = l$ とする。

IPRT —— 単体表の基底に導入されたベクトルの次元数を格納するための 1 次元配列の作業領域。 k 番目の部分形式に対応するベクトルが、表の i 番目の行に入れられたとき、

$$IPRT(i) = r_k \quad (1 \leq i \leq q + l, 1 \leq k \leq l) \text{ となっている。}$$

JIPRT —— 配列 IPRT の大きさを示す入力。 $JIPRT \geq q + l$ とする。

IFALL —— 計算の終了状態を示す出力。有限な解をもつとき 0、無限大の解をもつとき 1 となる。

AL —— 制約量を表わすベクトル \mathbf{b} などの変換のために使う 1 次元配列の作業領域。

JAL —— 配列 AL の大きさを示す入力。 $JAL \geq 1 + q + l$ とする。

DT —— 1 次元配列の作業領域。

JDT —— 配列 DT の大きさを示す入力。 $JDT \geq \max_{1 \leq k \leq l} r_k$ とする。

IPRT 1 — 部分形式を解くときの 1 次元配列の作業領域。

JIPRT 1 — 配列 IPRT 1 の大きさを示す入力。 $JIPRT 1 \geq \max_{1 \leq k \leq l} r_k$ とする。

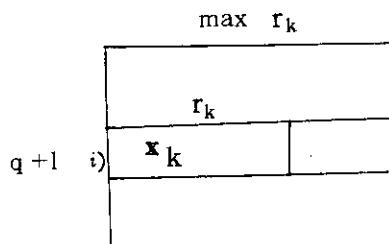
IPRT 2 — IPRT 1 と同様な 1 次元配列の作業領域。

JIPRT 2 — 配列 IPRT 2 の大きさを示す入力。 $JIPRT 2 \geq \max_{1 \leq k \leq l} q_k$ とする。

PT —— 評価ベクトルのための 1 次元配列の作業領域。

JPT —— 配列 PT の大きさを示す入力。 $JPT \geq q + l$ とする。

CT —— 頂点 \mathbf{x}_k を格納するための 1 次元配列の作業領域。 単体表の i 行目に入ってくるベクトルに対応する頂点 \mathbf{x}_k は下のように格納される。



JCT —— 配列 CT の大きさを示す入力。 $JCT \geq (q + l) \times (\max_{1 \leq k \leq l} r_k)$ とする。

LCX —— 解 \mathbf{x}_k を配列 X に格納するとき、その位置を計算するための 1 次元配列の作業領域。

LCB と同様に、 $LCX(1)=1, LCX(k)=1 + \sum_{j=1}^{k-1} r_k$ ($k = 2 \sim l$) となっている。

JLCX —— 配列 LCX の大きさを示す入力。 $JLCX \geq l$ とする。

EPS —— 付属ルーチン RESEX で使われる零判定値 ϵ を与える入力。通常、 c_k, A_k, b, B_k, b_k の全要素の絶対値の和を s としたとき、 $\epsilon = 5 \times s \times 10^{-5}$ くらいにとる。

それで解が求まらないとき、更に大きくとると解が求まることがある。

AR —— 付属ルーチン RESEX に部分形式の係数などを送るための 1 次元配列の作業領域。

JAR —— 配列 AR の大きさを示す入力。 $JAR \geq \max_{1 \leq k \leq l} (1 + q_k) \times (1 + r_k)$ とする。

BR —— 付属ルーチン RESEX で必要な 1 次元配列の作業領域。

JBR —— 配列 BR の大きさを示す入力。 $JBR \geq \max_{1 \leq k \leq l} (1 + q_k)^2$ とする。

CR —— BR と同様な 1 次元配列の作業領域。

JCR —— 配列 CR の大きさを示す入力。 $JCR \geq 1 + \max_{1 \leq k \leq l} q_k$ とする。

LW1 —— BR と同様な 1 次元配列の作業領域。

JLW1 —— 配列 LW1 の大きさを示す入力。 $JLW1 \geq \max_{1 \leq k \leq l} r_k$ とする。

LW2 —— BR と同様な 1 次元配列の作業領域。

JLW2 —— 配列 LW2 の大きさを示す入力。 $JLW2 \geq \max_{1 \leq k \leq l} \{ \min(q_k, r_k) \}$ とする。

LW3 —— BR と同様な 1 次元配列の作業領域。

JLW3 —— 配列 LW3 の大きさを示す入力。 $JLW3 \geq \max_{1 \leq k \leq l} \{ \min(q_k, r_k) \}$ とする。

UN —— 単体表に用いる 1 次元配列の作業領域。

JUN —— 配列 UN の大きさを示す入力。 $JUN \geq (l + q + l) \times (q + l)$ とする。

BASE —— 基底に導入されたベクトルなどのために用いる 1 次元配列の作業領域。

JBASE —— 配列 BASE の大きさを示す入力。 $JBASE \geq l + q + l$ とする。

BASEL —— 基底に導入されるベクトルを変換したときに用いる 1 次元配列の作業領域。

JBASEL —— 配列 BASEL の大きさを示す入力。 $JBASEL \geq l + q + l$ とする。

ICCT —— 採用された頂点を部分形式ごとに数えるための、 1 次元配列の作業領域。

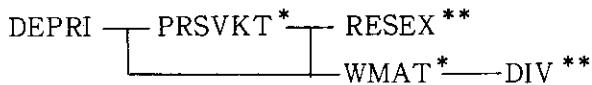
JICCT —— 配列 ICCT の大きさを示す入力。 $JICCT \geq l$ とする。

IPRINT —— 計算結果の印字のレベルを示す入力。 0 のとき、 計算の終了状態、 最小値 $F(x)$ およびそれを与える x_k を印字する。 1 のとき、 以上のほか、 最終回の単体表を印字する。 また、 2 のときは、 1 で印字されるもののほか、 途中の単体表も印字する。

(3) プログラムに関するその他の情報

使用言語は FORTRAN であり、 プログラムの大きさは 23156 バイト (5789 語) である。

使用されている付属ルーチンの呼び出し関係は下記による。



ここに、 * 印は DEPRI に固有な付属ルーチンであることを示し、 ** 印は既存の JSSL のルーチンであることを示す。

4.2 DEPRIM

(1) 機能と特徴

DEPRI と同様の機能をもつが、DEPRI との違いについては、この章の初めに記されている。

(2) 呼び出し方

```
CALL DEPRIM (A, JA, B, JB, X, JX, NCRT, JNCRT, NVAR, JNVAR,
IZB, JIZB, ISB, JISB, LCB, JLBC, IPRT, JIPRT, AL, JAL, DT, JDT,
IPRT1, JIPRT1, IPRT2, JIPRT2, PT, JPT, CT, JCT, LCX, JLCX, AR,
JAR, BR, JBR, CR, JCR, LW1, JLW1, LW2, JLW2, LW3, JLW3, UN,
JUN, BASE, JBASE, BASEL, JBASEL, ICCT, JICCT, AM, JAM1, JAM2,
BM, JBM1, JBM2)
```

DEPRIM の引数のうち、先のDEPRI にもあるものの役割は既に記した通りである。但し、データの入力、結果の出力の方法は若干異なるので注意を要する。まず、新しい引数の説明をしよう。

AM —— 配列 A に入れるべきデータを一時的に読み込むための、2 次元配列の作業領域。

JAM1 —— 配列 AM の第 1 の寸法を示す入力。 $JAM1 \geq 1 + q$ とする。

JAM2 —— JAM1 と同様に、第 2 の寸法を示す入力。 $JAM2 \geq 1 + n$ とする。

BM —— 配列 B に入れるべきデータを一時的に読み込むための、2 次元配列の作業領域。

JBM1 —— 配列 BM の第 1 の寸法を示す入力。 $JBM1 \geq \max_{1 \leq k \leq l} q_k$ とする。

JBM2 —— JBM1 と同様に、第 2 の寸法を示す入力。 $JBM2 \geq \max_{1 \leq k \leq l} (1 + r_k)$ とする。

次に、データの入力形式を説明する。

入力 ① FORMAT (20 A 4)

タイトル・カードを一枚用意する。

入力 ② FORMAT (3 I 5)

全体制約式の数 q 、部分形式の数 l 、印字レベルを示す IPRINT を入力する。

入力 ③ FORMAT (16 I 5)

部分制約式の数 q_k ($k = 1 \sim l$) を入力する。

入力 ④ FORMAT (16 I 5)

部分形式の変数の数 r_k ($k = 1 \sim l$) を入力する。

入力 ⑤ FORMAT (8 F 10.3)

配列 A に入れるべきデータを入力する。入力は行に沿って行い、1 行分ずつ入力する。従って、最初に c_0 , c_k^t ($k = 1 \sim l$), 次に (b, A_k) ($k = 1 \sim l$) の第 1 行, ..., 最後に (b, A_k) ($k = 1 \sim l$) の第 q 行の順に与える。

入力 ⑥ FORMAT (8 F 10.3)

配列 B に入れるべきデータを入力する。入力は行に沿って、 (b_1, B_1) の第 1 行、次に

$(\mathbf{b}_1, \mathbf{B}_1)$ の第 2 行, …, 最後に $(\mathbf{b}_1, \mathbf{B}_1)$ の第 q_1 行の順に与え, これを $(\mathbf{b}_k, \mathbf{B}_k)$ ($k = 2 \sim l$) について繰り返す。

以上の入力①～⑥を繰り返せば, 何ケースでも計算できる。

次に, 計算結果の出力を説明する。

- | | |
|------|---|
| 出力 ① | DEPRIM の入力のベタ打ちリスト |
| 出力 ② | DEPRIM の入力の, 書式を変えたリスト |
| 出力 ③ | DEPRIM で計算された零判定値 ϵ |
| 出力 ④ | オプション IPRINT のレベルによる, DEPRI の出力 |
| 出力 ⑤ | 計算時間, それに終了状態を示す IFALL |
| 出力 ⑥ | 最小値 $F(\mathbf{x})$ およびそのときの \mathbf{x} |

(3) プログラムに関するその他の情報

言語は FORTRAN であり, プログラムの大きさは 34484 バイト (8621 語) である。DEPRIM は DEPRI のほか, 既存の JSSL のルーチンの DTLIST, 組込みルーチンの CLOCKM, ABS, FLOAT を呼んでいる。DEPRI の呼び出し関係は前節を参照されたい。

5. 結 言

本稿では, まず線形最適化問題を解く上で基本となる単体法について概観し, それに必要な初期許容解の求め方についても述べた。それらは, 我々が主目的とする分解原理の応用上不可欠のものであった。

分解原理が応用可能な線形最適化問題の基本式は(22)式で与えられるが, そのうち制約がすべて第 1 型の場合は初期許容解が自明であり, 解き易い。このため, 分解原理の説明に用いられるとともに, 第一段階としてプログラム DEPRI が開発された。

このプログラムの原案は Künzi のテキスト³⁾にある DECOMP である。しかし, DECOMP には未完の部分があり, また, 作成済の部分にも誤りがいくつか見られることから, 我々は別途開発した。DEPRI では, 全変数の数 n が 10, 全制約の数 m が 10 の問題まで解かれている。しかし, この程度の大きさの問題でも, 部分形式を解くとき, 零判定値 ϵ を DEPRI の使用手引きで示した標準値にとると, 単体法の規準が満たされず, 無限ループに陥いることがあるので注意を要する。上の例では, 標準値 ϵ がおよそ 10^{-5} と計算されたが, 無限ループに陥いったので $\epsilon = 10^{-3}$ で計算すると解が求まり, かつ 4 桁以上正確であった。

DEPRI で解ける問題は, 当然ながら, 分解原理を使わない通常の単体法のプログラムでも解ける。例えば, SIMPLX によると, 主な記憶容量は $(1 + m) \times (1 + n)$ で足りるが, 入力した係数は保存されない。一方, DEPRI では, 每回部分システムを解くために, 止むを得ず係数行列の \mathbf{A}_k や \mathbf{B}_k などを保存するという事情もあるが, 現在必要としている, 配列 A, B, CT, AR, BR, UN などの記憶領域のうち, いくらかは減ずることができよう。

$(\mathbf{b}_1, \mathbf{B}_1)$ の第 2 行, …, 最後に $(\mathbf{b}_1, \mathbf{B}_1)$ の第 q_1 行の順に与え, これを $(\mathbf{b}_k, \mathbf{B}_k)$ ($k = 2 \sim l$) について繰り返す。

以上の入力①～⑥を繰り返せば, 何ケースでも計算できる。

次に, 計算結果の出力を説明する。

- | | |
|------|---|
| 出力 ① | DEPRIM の入力のベタ打ちリスト |
| 出力 ② | DEPRIM の入力の, 書式を変えたリスト |
| 出力 ③ | DEPRIM で計算された零判定値 ϵ |
| 出力 ④ | オプション IPRINT のレベルによる, DEPRI の出力 |
| 出力 ⑤ | 計算時間, それに終了状態を示す IFALL |
| 出力 ⑥ | 最小値 $F(\mathbf{x})$ およびそのときの \mathbf{x} |

(3) プログラムに関するその他の情報

言語は FORTRAN であり, プログラムの大きさは 34484 バイト (8621 語) である。DEPRIM は DEPRI のほか, 既存の JSSL のルーチンの DTLIST, 組込みルーチンの CLOCKM, ABS, FLOAT を呼んでいる。DEPRI の呼び出し関係は前節を参照されたい。

5. 結 言

本稿では, まず線形最適化問題を解く上で基本となる単体法について概観し, それに必要な初期許容解の求め方についても述べた。それらは, 我々が主目的とする分解原理の応用上不可欠のものであった。

分解原理が応用可能な線形最適化問題の基本式は(22)式で与えられるが, そのうち制約がすべて第 1 型の場合は初期許容解が自明であり, 解き易い。このため, 分解原理の説明に用いられるとともに, 第一段階としてプログラム DEPRI が開発された。

このプログラムの原案は Künzi のテキスト³⁾にある DECOMP である。しかし, DECOMP には未完の部分があり, また, 作成済の部分にも誤りがいくつか見られることから, 我々は別途開発した。DEPRI では, 全変数の数 n が 10, 全制約の数 m が 10 の問題まで解かれている。しかし, この程度の大きさの問題でも, 部分形式を解くとき, 零判定値 ϵ を DEPRI の使用手引きで示した標準値にとると, 単体法の規準が満たされず, 無限ループに陥いることがあるので注意を要する。上の例では, 標準値 ϵ がおよそ 10^{-5} と計算されたが, 無限ループに陥いったので $\epsilon = 10^{-3}$ で計算すると解が求まり, かつ 4 行以上正確であった。

DEPRI で解ける問題は, 当然ながら, 分解原理を使わない通常の単体法のプログラムでも解ける。例えば, SIMPLX によると, 主な記憶容量は $(1 + m) \times (1 + n)$ で足りるが, 入力した係数は保存されない。一方, DEPRI では, 每回部分システムを解くために, 止むを得ず係数行列の \mathbf{A}_k や \mathbf{B}_k などを保存するという事情もあるが, 現在必要としている, 配列 A, B, CT, AR, BR, UN などの記憶領域のうち, いくらかは減ずることができよう。

より大規模で実用的な例題を捜しているうち、我々はエネルギー・モデルの問題⁸⁾を手にした。しかし、この問題は部分制約に第2型や第3型のものがあり、DEPRIでは扱えない。そこで、第2段階として、任意の初期許容解を与えれば最適解が求まる別のプログラムを作成した。既に記したように、この問題ではまず、元の変数と付加変数を0とし、人為変数を1として初期許容解を求め、次にそれを使って最適解を求める2段法で求められた。これと同時に、多段法による解法も試みたが、各段階の計算時間が、与えられた元の問題を解くくらい必要なので、付加変数が不要などという利点はあるが棄却した。

このプログラムによるエネルギー・モデルの問題の計算結果は先に記述した通りであるが、そこでは付加変数や人為変数のための領域を余分にとるなど、プログラムとしての最適化は全く行われていないので、現在のところ公表に至っていない。

分解原理を応用できる問題の一般式は(22)式であるが、全体制約が第2型や第3型を含む場合の議論をしておこう。まず、第1型の制約が p_1 個、第2型の制約が p_2 個ある場合には、Table 4 から想像し得るように、初期許容解さえ分かれれば、独立な $(q + p_1 + p_2)$ 個の基底ベクトルが見つかったことになるので、探索を始めることができる。そして、その初期許容解は2段法などで求めることができる。

次に、第3型の制約が p_3 個加わった場合を考えると、初期許容解が分っただけでは解けないので、一案として、最初から p_3 個の人為変数を入れ、乗数法で最適値が求まるまで探索を行うなどの方法をとらなければならないだろう。これらのプログラムは現在開発中なので、完了し次第、JSSL¹¹⁾などで公表する予定である。

以上のように、分解原理のアルゴリズムは簡明である。しかし、これを大規模システムに応用する場合、次の点は重要である。一つは、計算の誤差集積による零判定の問題である。これは、倍精度、4倍精度計算というふうに精度を上げていけば解決するが、記憶容量や計算時間の点で難点がある。従って、積和計算の部分など必要な所のみ倍精度にするなどの工夫が必要であろう。もう一つの問題は付加変数や人為変数の導入による記憶領域の問題である。現在のプログラムは分り易さを重点に設計されているが、単体表の作り方の工夫や、補助記憶装置¹⁰⁾の利用を行えば、もっと規模の大きい計算も可能であろう。

分解原理の概念はシステムが非線形の場合でも応用可能であり^{2) 6) 7)}、また、各部分システムが独立に計算されるので、並列演算¹²⁾を行う場合にも適している。このような意味で、将来、もっと多方面で活用されるよう望んで止まない。

謝　　辞

NIRA エネルギー・モデルの問題を計算するにあたり、計算センタ、藤井実氏には、データや計算結果の検討に協力をいただいた。また、プログラム DESOMJ の使用にあたり、原子炉システム研究室、鈴木忠和氏の助力を得た。ここに謝意を表します。

より大規模で実用的な例題を捜しているうち、我々はエネルギー・モデルの問題⁸⁾を手にした。しかし、この問題は部分制約に第2型や第3型のものがあり、DEPRIでは扱えない。そこで、第2段階として、任意の初期許容解を与えれば最適解が求まる別のプログラムを作成した。既に記したように、この問題ではまず、元の変数と付加変数を0とし、人為変数を1として初期許容解を求め、次にそれを使って最適解を求める2段法で求められた。これと同時に、多段法による解法も試みたが、各段階の計算時間が、与えられた元の問題を解くくらい必要なので、付加変数が不要などという利点はあるが棄却した。

このプログラムによるエネルギー・モデルの問題の計算結果は先に記述した通りであるが、そこでは付加変数や人為変数のための領域を余分にとるなど、プログラムとしての最適化は全く行われていないので、現在のところ公表に至っていない。

分解原理を応用できる問題の一般式は(22)式であるが、全体制約が第2型や第3型を含む場合の議論をしておこう。まず、第1型の制約が p_1 個、第2型の制約が p_2 個ある場合には、Table 4 から想像し得るように、初期許容解さえ分かれれば、独立な $(q + p_1 + p_2)$ 個の基底ベクトルが見つかったことになるので、探索を始めることができる。そして、その初期許容解は2段法などで求めることができる。

次に、第3型の制約が p_3 個加わった場合を考えると、初期許容解が分っただけでは解けないので、一案として、最初から p_3 個の人為変数を入れ、乗数法で最適値が求まるまで探索を行うなどの方法をとらなければならないだろう。これらのプログラムは現在開発中なので、完了し次第、JSSL¹¹⁾などで公表する予定である。

以上のように、分解原理のアルゴリズムは簡明である。しかし、これを大規模システムに応用する場合、次の点は重要である。一つは、計算の誤差集積による零判定の問題である。これは、倍精度、4倍精度計算というふうに精度を上げていけば解決するが、記憶容量や計算時間の点で難点がある。従って、積和計算の部分など必要な所のみ倍精度にするなどの工夫が必要であろう。もう一つの問題は付加変数や人為変数の導入による記憶領域の問題である。現在のプログラムは分り易さを重点に設計されているが、単体表の作り方の工夫や、補助記憶装置¹⁰⁾の利用を行えば、もっと規模の大きい計算も可能であろう。

分解原理の概念はシステムが非線形の場合でも応用可能であり^{2) 6) 7)}、また、各部分システムが独立に計算されるので、並列演算¹²⁾を行う場合にも適している。このような意味で、将来、もっと多方面で活用されるよう望んで止まない。

謝　　辞

NIRA エネルギー・モデルの問題を計算するにあたり、計算センタ、藤井実氏には、データや計算結果の検討に協力をいただいた。また、プログラム DESOMJ の使用にあたり、原子炉システム研究室、鈴木忠和氏の助力を得た。ここに謝意を表します。

参 考 文 献

- 1) Dantzig, G. B. : "Linear Programming and Extensions", Princeton University Press (1963)
- 2) 関根泰次 : "数理計画法 I, II", 岩波書店 (1968)
- 3) Künzi, H. P., Tzsachach, H. G., Zehner, C. A. : "Numerical Methods of Mathematical Optimization", Academic Press (1971)
- 4) 鈴木忠和 : "最適化手法の評価と最適化コード・システム SCOOP の開発", JAERI 1263 (1979)
- 5) Dantzig, G. B., Wolfe, Ph. : "Decomposition Principle for Linear Programs", Operations Res. 8, 101 (1960)
- 6) Orchard-Hays, W. : "Advanced Linear-Programming Computing Techniques", McGraw-Hill Book Company (1968)
- 7) Wismer, D. A. (Ed.) : "Optimization Methods for Large-Scale Systems", McGraw-Hill Book Company (1971)
- 8) 安成弘他 : "NIRA エネルギー・モデルの開発", NRO-53-4, 総合研究開発機構・自主研究 (1978)
- 9) 堀上邦彦, 藤村統一郎, 中原康明 : "線形最適化プログラム・パッケージ使用説明書", JAERI - M 9048 (1980)
- 10) 富士通 : "FACOM 230 M - VII FORTRAN N - H 使用手引書 (75 SP - 0280 - 1)", 富士通 (1976)
- 11) 藤村統一郎, 西田雄彦, 浅井清 (編) : "JSSL (原研版・科学計算用サブルーチン・ライブラリ) 第2版", JAERI - M 8479 (1979)
- 12) Kuck, D. J. : "The Structure of Computers and Computations (Vol. 1)", John Wiley & Sons, Inc. (1978)

付録 DEPRIMの入出力例

FACOM CSIV/F4 FORTRAN IV (HE) V04L16 DATE 81.01.07 TIME 10.37.23

SPECIFIED OPTIONS: NONAME,FLAG(I),SOURCE,NOMAP,BYNAME,GOSTMT,NOSTATIS

```

C      EXAMPLE FOR DEPRIM
C      FROM KUNZI,S TEXT BOOK (1971) P46
ISN 00001  DIMENSION A(10),B(15),X(5)
ISN 00002  DIMENSION L1(2),L2(2),L3(2),L4(2),L5(2)
ISN 00003  DIMENSION IPRT(3),AL(4),D(2),IPRT1(2),IPRT2(3)
ISN 00004  DIMENSION P(3),C(6),LL(2),AR(12),BR(16),CR(4)
ISN 00005  DIMENSION LW1(2),LW2(3),LW3(3)
ISN 00006  DIMENSION UN(12),BASE(4),EASEL(4),ICCT(2)
ISN 00007  DIMENSION AM(2,5),BM(3,3)
ISN 00008  CALL DEPRIM (A,10,B,15,X,5,L1,2,L2,2,L3,2,L4,2,L5,2
1     ,IPRT,3,AL,4,D,2,IPRT1,2,IPRT2,3
2     ,P,3,C,6,LL,2,AR,12,3R,16,CR,4
3     ,LW1,2,LW2,3,LW3,3,UN,12,BASE,4,BASEL,4,ICCT,2
4     ,AM,2,5,BM,3,3)
ISN 00009  STOP
ISN 00010  END

```

PAGE-0001

```

***** *
* INPUT DATA LIST *
*
***** *

```

	EXAMPLE FROM KUNZI,S TEXT BOOK (1971) P46						
1	1	2	2			1	
2	2	3				2	
3	2	2				3	
4						4	
5	-18.0	-1.0	-8.0	-0.5	-1.5	C	5
6	1.0	1.0	4.0	3.5	0.5	A	6
7	6.0	2.0	3.0			E1	7
8	5.0	5.0	1.0			S1	8
9	12.0	3.0	-1.0			S2	9
10	0.0	-3.0	1.0			S2	10
11	4.0	1.0	0.0			S2	11

*** INPUT DATA END ***

TITLE ===== EXAMPLE FROM KUNZI'S TEXT BOOK (1971) P46

NUMBER OF CONSTRAINTS IN COMMON (M) = 1

NUMBER OF PARTIAL FORMS (N) = 2

OUTPUT CONTROL KEY WORD (IPRINT) = 2

NUMBER OF CONSTRAINTS IN EACH PARTIAL FORM (L1) =

2 3

NUMBER OF VARIABLES CONTAINED IN THE CONSTRAINTS FOR EACH PARTIAL FORM (L2) =
2 2----- COEFFICIENT OF THE MATRIX A -----
ROW NO.= 1 -0.180000E+02 -0.100000E+01 -0.800000E+01 -0.500000E+00 -0.150000E+01
ROW NO.= 2 0.100000E+01 0.100000E+01 0.400000E+01 0.350000E+01 0.500000E+00

----- COEFFICIENTS OF THE PARTIAL MATRICES B(I)-----

PARTIAL NO. = 1

ROW NO.= 1 0.600000E+01 0.200000E+01 0.300000E+01

ROW NO.= 2 0.500000E+01 0.500000E+01 0.100000E+01

PARTIAL NO. = 2

ROW NO.= 1 0.120000E+02 0.300000E+01 -0.100000E+01

ROW NO.= 2 0.0 -0.300000E+01 0.100000E+01

ROW NO.= 3 0.400000E+01 0.100000E+01 0.0

EPS= 0.17200E-03

M = 1 N = 2

M(K)

2 3

N(K)

2 2

ZSCHR(K)

3 3

SSCHR(K)

1 1

LOCATION OF B(K)

0 6

LOCATION OF X(K)

0 2

MATRIX A (1 - 5 COLUMN)

-1.8000E+01 -1.0000E+00 -8.0000E+00 -5.0000E-01 -1.5000E+00
1.0000E+00 1.0000E+00 4.0000E+00 3.5000E+00 5.0000E-01

JAERI-M 9315

MATRIX B PART (1)

MATRIX (1 - 3 COLUMN)

6.0000E+00	2.0000E+00	3.0000E+00
5.0000E+00	5.0000E+00	1.0000E+00

MATRIX B PART (2)

MATRIX (1 - 3 COLUMN)

1.2000E+01	3.0000E+00	-1.0000E+00
0.0	-3.0000E+00	1.0000E+00
4.0000E+00	1.0000E+00	0.0

1 -TH ITERATE 1 -TH PARTIAL FORM

MATRIX B(K) (1 - 3 COLUMN)

0.0	1.0000E+00	8.0000E+00
6.0000E+00	-2.0000E+00	-3.0000E+00
5.0000E+00	-5.0000E+00	-1.0000E+00

LINEAR

1	2
1	3

1	1.600E+01	-4.333E+00	-2.667E+00
2	2.000E+00	-2.000E+00	-3.000E+00
4	3.000E+00	-5.000E+00	-1.000E+00

MATRIX B

-2.667E+00	0.0
-3.333E-01	0.0
3.333E-01	-1.000E+00

1 -TH ITERATE 2 -TH PARTIAL FORM

MATRIX B(K) (1 - 3 COLUMN)

0.0	5.0000E-01	1.5000E+00
1.2000E+01	-3.0000E+00	1.0000E+00

0.0	3.0000E+00	-1.0000E+00
4.0000E+00	-1.0000E+00	0.0

LINEAR

	1	2
	1	4
0.0	5.000E+00	-1.500E+00
3 1.200E+01	-3.000E+00	1.000E+00
2 0.0	3.000E+00	-1.000E+00
5 4.000E+00	-1.000E+00	0.0

MATRIX B

0.0	-1.500E+00	0.0
-1.000E+00	-1.000E+00	0.0
0.0	-1.000E+00	0.0
0.0	0.0	-1.000E+00

LINEAR

	1	2
	5	4
2.000E+01	-5.000E+00	-1.500E+00
3 1.200E+01	-3.000E+00	1.000E+00
2 1.200E+01	3.000E+00	-1.000E+00
1 4.000E+00	-1.000E+00	0.0

MATRIX B

0.0	-1.500E+00	-5.000E+00
-1.000E+00	-1.000E+00	0.0
0.0	-1.000E+00	-3.000E+00
0.0	0.0	-1.000E+00

GAMMA = -2.0000E+01

2 -TH COLUMN WAS ENTERED

BASE (ENTERED)

2.0000E+01 0.0 1.0000E+00

BASE (TRANSFORMED)

2.0000E+01 0.0 1.0000E+00

2 -TH ROW WAS LEFT

TABLEAU

MATRIX	UN	(1 -	3	COLUMN)
1.0000E+00	0.0	0.0			
5.0000E-02	0.0	0.0			
0.0	1.0000E+00	0.0			
-5.0000E-02	0.0	1.0000E+00			

APPR. MAX (-Y) = 1.9000E+01

CONSTANT VECTOR

5.0000E-02 1.0000E+00 9.5000E-01

CORNER POINT (2 , 2)

4.0000E+00 1.2000E+01

2 -TH ITERATE 1 -TH PARTIAL FORM

MATRIX B(K) (1 - 3 COLUMN)

0.0	5.9605E-08	4.0000E+00
6.0000E+00	+2.0000E+00	-3.0000E+00
5.0000E+00	-5.0000E+00	-1.0000E+00

LINEAR

1	2
1	3

2	8.0000E+00	-2.667E+00	-1.333E+00
2	2.0000E+00	-2.0000E+00	-3.0000E+00
4	3.0000E+00	-5.0000E+00	-1.0000E+00

MATRIX B

-1.333E+00	0.0
-3.333E-01	0.0
3.333E-01	-1.000E+00

2 -TH ITERATE 2 -TH PARTIAL FORM

MATRIX B(K) (1 - 3 COLUMN)

0.0	-3.0000E+00	1.0000E+00
1.2000E+01	-3.0000E+00	1.0000E+00
0.0	3.0000E+00	-1.0000E+00
4.0000E+00	-1.0000E+00	0.0

LINEAR

1	2
1	4

3	0.0	9.537E-07	-1.000E+00
3	1.2000E+01	-3.000E+00	1.000E+00
2	0.0	3.000E+00	-1.000E+00
5	4.0000E+00	-1.000E+00	0.0

MATRIX B

0.0	-1.0000E+00	0.0
-1.0000E+00	-1.0000E+00	0.0
0.0	-1.0000E+00	0.0
0.0	0.0	-1.0000E+00

GAMMA = -8.0000E+00

1 -TH COLUMN WAS ENTERED

BASE (ENTERED)

8.0000E+00 1.0000E+00 0.0

BASE (TRANSFORMED)

4.0000E-01 1.0000E+00 -4.0000E-01

2 -TH ROW WAS LEFT

TABLEAU

			MATRIX	UN	(1 -	3	COLUMN)
2.0000E+00	0.0	0.0						
1.2500E-01	0.0	0.0						
-1.2500E-01	1.0000E+00	0.0						
-7.4506E-09	0.0	1.0000E+00						

APPR. MAX (-Y) = 2.0000E+01

CONSTANT VECTOR

1.2500E-01 8.7500E-01 1.0000E+00

CORNER POINT (1 , 2)

0.0 2.0000E+00

3 -TH ITERATE 1 -TH PARTIAL FORM

			MATRIX	B(K)	(1 -	3	COLUMN)
0.0	-1.0000E+00	3.8147E-06						
6.0000E+00	-2.0000E+00	-3.0000E+00						
5.0000E+00	-5.0000E+00	-1.0000E+00						

3 -TH ITERATE 2 -TH PARTIAL FORM

MATRIX B(K) (1 - 3 COLUMN)

0.0	-6.5000E+00	5.0000E-01
1.2000E+01	-3.0000E+00	1.0000E+00
0.0	3.0000E+00	-1.0000E+00
4.0000E+00	-1.0000E+00	0.0

LINEAR

1	2
1	4

0.0	-5.000E+00	-5.000E-01
3	1.200E+01	-3.000E+00
2	0.0	3.000E+00
5	4.000E+00	-1.000E+00
		0.0

MATRIX B

0.0	-5.000E-01	0.0
-1.000E+00	-1.000E+00	0.0
0.0	-1.000E+00	0.0
0.0	0.0	-1.000E+00

----- LAST SITUATION -----

PREVIOUS GAMMA = -8.0000E+00

BASE (TRANSFORMED)

4.0000E-01 1.0000E+00 -4.0000E-01

2 -TH ROW WAS LEFT

PRESENT GAMMA = 0.0

TABLEAU

MATRIX UN (1 - 3 COLUMN)

2.0000E+00	0.0	0.0
1.2500E-01	0.0	0.0
-1.2500E-01	1.0000E+00	0.0
-7.4506E-09	0.0	1.0000E+00

MAXIMUM Y = 2.0000E+01

CONSTANT VECTOR

1.2500E-01 8.7500E-01 1.0000E+00

CORNERS POINTS

K C(K,1) --- C(K,N(K))

1 0.0 2.0000E+00

END CONDITION (0 = FINITE SOL., 1 = NO FINITE)

0

SOLUTION X =

0.0 2.5000E-01 0.0 0.0

MINIMUM Y = -2.0000E+01

===== EXECUTION TIME (MILI-SEC.) = 40 FALL= 0

OPTIMUM VALUE (MIN.) = -0.1999998E+02

X(I) = 0.0 0.2499999E+00 0.0 0.0