北海道大学工学部研究報告
第79号（昭和51年）

Bulletin of the Faculty of Engineering,
Hokkaido University, No. 79 (1976)

有界変数法と積形式を用いた L. P. コードの作成

大堀隆文* 藤田克孝* 戸田 悟* 大内 東* 加地郁夫*

（昭和50年9月30日受理）

L. P. code with the Upper-Bounded Technique and the Product Form of Inverse

Takahumi OOHORI, Katsuyuki FUJITA, Satoshi TODA, Azuma OOUCHI and Ikuo KAJI

(Received September 30, 1975)

Abstract

The authors programmed an L. P. code with the upper-bounded technique and the product form of inverse for the user's subroutine at the Hokkaido University Computing Center.

This report describes this L. P. program-code and its specifications.

1. まえがき

線形計画法 (LP) は，経営科学や OR のみならず，工学上の各分野において，著しい成果をあげている。しかし，システムの巨大化と複雑化により，従来の LP では，計算機の容量及び計算時間の壁にぶつかる。Orchard Hays は，この問題を解決するために，上下限付変数に対する解法（有界変数法）と逆行列の積形式を用いた解法（積形式法）を取り入れた LP を提案した[1]。また，平本らによって，この考えに基づいた流れ図 (flow chart) が発表された[3]。しかし，この流れ図には，いくつかの誤まりが発見されたので，著者らは，これを修正し，Fortran によるプログラムコードを作成した。

本報告は，この Fortran プログラムと，その使用方法について述べたものである。

2. LP 問題の定式化

LP 問題の典型的な形は，次のように書ける。

$$\max z = \boldsymbol{c}\boldsymbol{x}$$

$$\text{sub. to.;}\ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \qquad \boldsymbol{x} \geq 0$$

ここで，$\boldsymbol{A}$ は $m \times n$ の係数行列，$\boldsymbol{b}$ は m 成分ベクトル，$\boldsymbol{c}$ と $\boldsymbol{x}$ は n 成分ベクトルである。

3. 有界変数法について

変数の上限に対する制約は，現実問題を考える場合に，よく現われる。これらの制約をすべて制約式として定式化を行なうと，問題のサイズが大きくなって，計算時間が多くかかることにもなり，また，モデル化の際の手間を増加させ，ひいては誤りの原因にもなると思われる。

そこで，変数の型（タイプと呼ぶ）を設けて，変数の値のとり得る範囲を限定した上で，この種の制約を制約式から除外して，L. P. 問題を解くことを考える。そのような手法として，有界

変数法がある。

さて，$a \leq b$ とした時，変数 x の取り得る値の範囲として，次の4つの場合が考えられる。

$$(1) \quad -\infty \leq x \leq \infty$$
$$(2) \quad a \leq x \leq \infty$$
$$(3) \quad -\infty \leq x \leq b$$
$$(4) \quad a \leq x \leq b$$

(2) と (3) は，それぞれ，$x-a$，$-(x-b)$ をあらためて x とおくことにより，

$$0 \leq x \leq \infty$$

という形にすることができる。

(4) も，$x-a$ をあらためて x とおくと，

$$0 \leq x \leq b-a$$

と書けるが，$a<b$ と $a=b$ の場合を区別して，$a<b$ の場合は，$(x-a)/(b-a)$ を x とおいて，

$$0 \leq x \leq 1,$$

$a=b$ の場合は，$x-a$ を x とおいて，

$$x=0$$

と書くことができる。

以上をまとめると，変数を次の4つのタイプに再分類できる。

(1)′ 自由変数型	(タイプ3)	$-\infty \leq x \leq \infty$
(2)′ 正変数型	(タイプ2)	$0 \leq x \leq \infty$
(3)′ 上界付き変数型	(タイプ1)	$0 \leq x \leq 1$
(4)′ 0変数型	(タイプ0)	$x=0$

このようにして再分類された変数，特に上界付き変数を用いた単体法が，文献1)に詳しく述べられているので，それを参照されたい。

4. 改訂単体法について

本来の単体法で L. P. 問題を解こうとすると，最新のコンピュータでも息づまりする。例えば，m 行 n 列のモデルでは，約 mn の記憶領域を必要とする。m, n が大きくなると，主記憶だけでは取り扱えないので，磁気テープ，デイスク，ドラム等の補助記憶装置を使用しなければならない。しかし，これらはアクセスするのにほぼ2桁のオーダーぐらい遅いので，主記憶と補助記憶の間を行き来するデータの量を減らす工夫を必要とする。

単体法は，適当な基底行列，例えば，単位行列から出発して，それを更新していって，ついには，最適解の条件を満足するような基底行列を求める方法である。しかし，そこで行なわれる演算は，与えられた係数行列 $\boldsymbol{A}$ に，基底逆行列 $\boldsymbol{B}^{-1}$ を掛ける演算の繰り返しである。したがって，$\boldsymbol{B}^{-1}$ が利用できるならば，必要な時に，単体表の任意の部分を作り出すことができる。この考え方を用いたのが，いわゆる，改訂単体法である。

改訂単体法では，もとの係数行列 $\boldsymbol{A}$ を補助記憶に格納することにすれば，主記憶に記憶する行列の大きさは約 m^2 でよいので，$m(n-m)$ だけの節約になる。しかし，制約式の個数 m が大きくなる時は，改訂単体法を用いても容易ではない。

5. 逆行列の積形式について

そこで，基底逆行列をピボット演算行列の積で表現する方法，いわゆる，積形式法が考案され

た。ピボット演算行列というのは，基底変換前の基底逆行列から，変換後の基底逆行列を求めるための変換行列である。

単体法の反復計算において，第 K ステージの基底行列を $\boldsymbol{B}_K$ とし，基底から追い出す変数 $x_{i(r_K)}$ (i は変数固有の番号，r_K は基底での位置) と，基底に取り入れる変数 x_S (S は変数固有の番号) の決定により作られるピボット演算行列を $\boldsymbol{E}_K$ とする。

そうすると，次のステージにおける基底逆行列は，次の積形式で求めることができる。

$$\boldsymbol{B}_{K+1}^{-1}=\boldsymbol{E}_K\boldsymbol{E}_{K-1}\cdots\boldsymbol{E}_1$$

ピボット演算行列 $\boldsymbol{E}_K$ は単位行列と，第 r 列 (これを η ベクトルと呼ぶ) だけが異なる行列であるので，$\boldsymbol{B}_{K+1}^{-1}$ を求めるためには，η ベクトルとその位置を示す r の列 (r_1, η_1; r_2, η_2; …; r_K, η_K) がわかるだけでよい。

単体法の反復ごとに得られる η ベクトルを，その位置 r といっしょに，補助記憶に記録しておき，評価ベクトル (基底に入る変数を決めるためのベクトル，$\boldsymbol{u}'\boldsymbol{B}^{-1}$ と，基底から出る変数を決めるためのベクトル，$\boldsymbol{B}^{-1}\boldsymbol{a}_S$ のこと；但し，$\boldsymbol{u}$ は基底変数に対する価格ベクトルで，$\boldsymbol{a}_S$ は基底に入るのに選ばれたベクトルである) を求める時には，η ベクトルを1本ずつ主記憶内に読み込んできて，計算すればよい。

したがって，主記憶内には，η ベクトルを読み込むための，作業用の m 次元ベクトルを1本用意しておけばよいので，制約式の多い L. P. 問題 (例えば，$m=1000$) をコンピュータで解くことが可能となる。

6. アルゴリズム

上記で述べた第3章～第5章の考え方に基づいたアルゴリズムが，流れ図 (flow chart) の形で，平本らによって，与えられている[3]。しかし，この流れ図には，著者らの考察により，いく

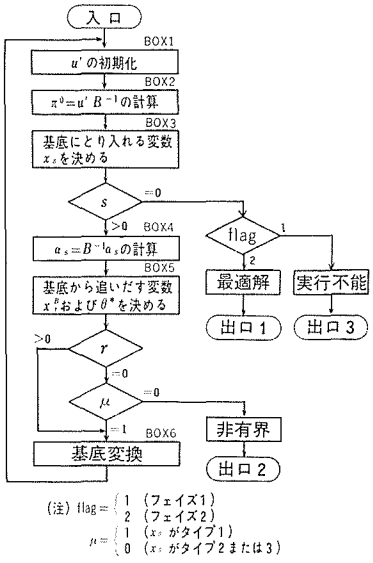


図1　積形式法と有界変数法による単体法の概略流れ図

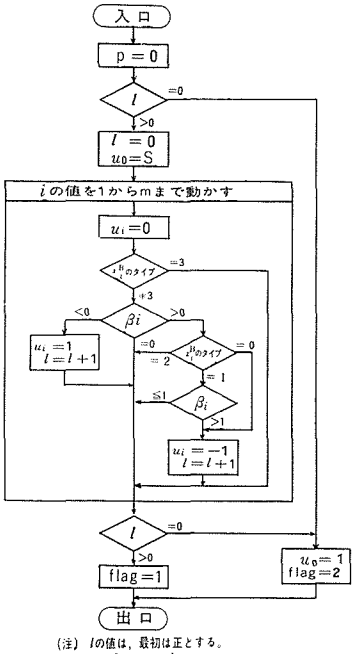


図2　Box 1 の流れ図

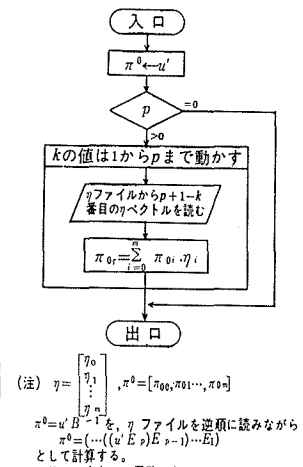
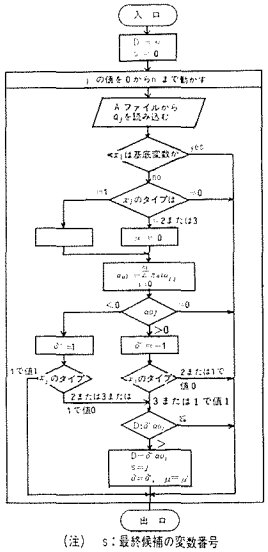
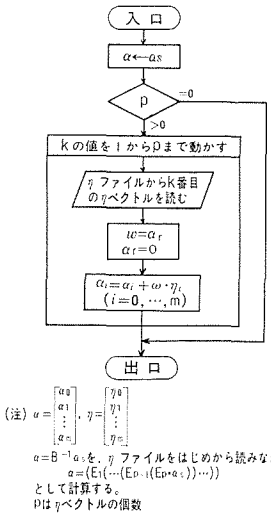


図3　Box 2 の流れ図



(注) s：最終候補の変数番号
$\delta = -\mathrm{sign}(a_{0s})$
$D = \delta a_{0s}$
$\mu = \begin{cases} 1 & (x_s \text{がタイプ1変数}) \\ 0 & (\text{その他の場合}) \end{cases}$

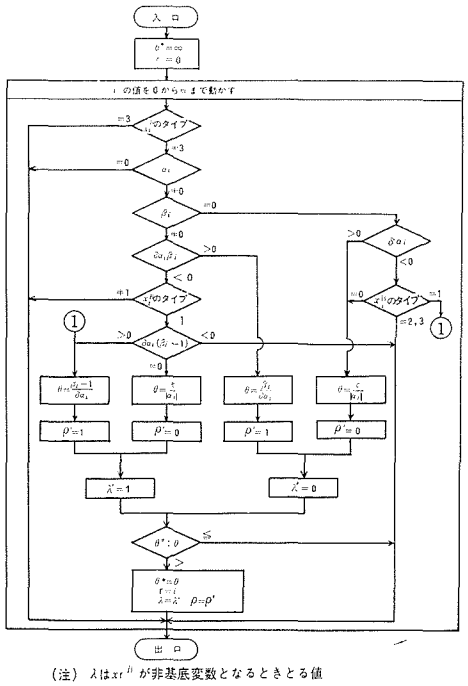
図4 Box 3 の流れ図



(注) $\alpha = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}$, $\eta = \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_m \end{bmatrix}$

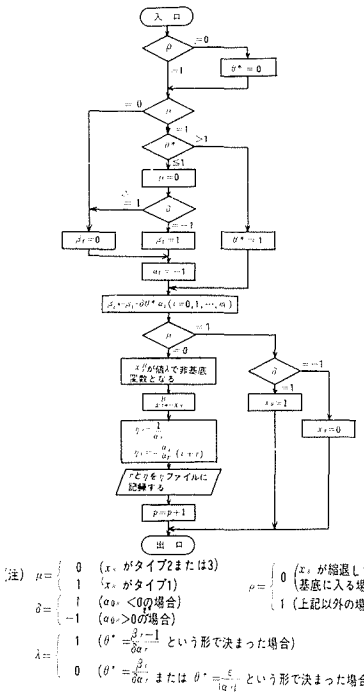
$\alpha = B^{-1}a_s$ を，η ファイルをはじめから読みながら
$a = (E_1\{\cdots(E_{p-1}\{(E_p \cdot a_s)\}\cdots)\}$
として計算する。
p は η ベクトルの個数

図5 Box 4 の流れ図



(注) λ は $x_r^{(l)}$ が非基底変数となるときとる値

図6 Box 5 の流れ図



(注) $\mu = \begin{cases} 0 & (x_r \text{がタイプ2または3}) \\ 1 & (x_r \text{がタイプ1}) \end{cases}$

$\delta = \begin{cases} 1 & (a_{0r} < 0 \text{の場合}) \\ -1 & (a_{0r} > 0 \text{の場合}) \end{cases}$

$\lambda = \begin{cases} 1 & (\theta^* = \frac{\beta_r - 1}{\delta a_r} \text{という形で決まった場合}) \\ 0 & (\theta^* = \frac{\beta_r}{\delta a_r} \text{または} \theta^* = \frac{\xi}{|a_{rj}|} \text{という形で決まった場合}) \end{cases}$

$\rho = \begin{cases} 0 & (x_s \text{が縮退して基底に入る場合}) \\ 1 & (\text{上記以外の場合}) \end{cases}$

図7 Box 6 の流れ図

つかの誤まりが発見されている。それを修正し，general flow-chart を図1に，detail flow-chart を図2～図7に示す。

7. プログラムの構成

本報告のプログラムは，図1～図7のアルゴリズムに基づいて，作成されたものであり，補助

記憶装置（ファイル）を用いるか否かにより，LPFILE と LPCORE に分けられている。いずれも，約 300 枚のカードデックからなる，サブルーチン形式の Fortran プログラムで，図 8 のような構成をしている。

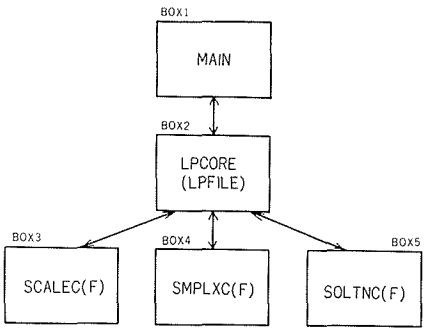


図 8　プログラムの構成

図 8 において，各 Box は 1 つのプログラム単位をなしており，それぞれ，以下に述べるような役割を持つ。

Box 1　MAIN：サブルーチン LPCORE（または，LPFILE）を呼ぶためのメインプログラム。ここでは，LP 問題を解くのに必要な入力情報が与えられる。また，出力として，実行不能，非有界，最適解のいずれかが指示される。

Box 2　LPCORE：必要な記憶領域が主記憶内に十分入り得る場合に使われるサブルーチン。LPFILE に比べて，計算時間が短かくてすむ。
LPFILE：補助記憶装置を用いる場合のサブルーチン。LPCORE に比べて，大きな問題を扱える。

LPCORE, LPFILE のいずれを用いる場合でも，さらに，SCALEC (F), SMPLXC (F), SOLTNC (F) の 3 つのサブルーチンが呼ばれる。但し，（ ）内は，LPFILE を用いる場合に使用されるサブルーチンである。

Box 3　SCALEC (F)：第 3 章で述べたように，変数をスケール変換し，タイプ別変数に再分類するためのサブルーチン。このスケール変換は，計算機の桁落ちによる誤差を少なくする効果をもつ。

Box 4　SMPLXC (F)：このプログラムの主要部分を占め，有界変数法と逆行列の積形式を用いた改訂単体法を実行するサブルーチン。

Box 5　SOLTNC (F)：Box 4 で求まった解を逆スケール変換して，メインプログラムに送るためのサブルーチン。Box 3 とは，逆の手続きを踏む。

8. プログラムの使用方法

本プログラムは，整合寸法を用いているため（つまり，記憶容量を可変にできる），メインプログラム作成に際しては，弱干の注意を要する。以下，入力として必要な変数と，出力情報の説明を行なう。

8.1 入 力 情 報

NCONS：制約式の本数
NVAR：変数の個数
MP：NCONS+1
NP：NVAR+1
INPL：NCONS+NVAR+1
IPMAX：LP の繰り返し回数の上限値。通常，MP の 2 倍程度の大きさにとる。
EPS：0 判定に用いる正定数。通常，10^{-4} 程度にとる。
IXTYPE(i)：i=2～NP に変数のタイプ（第 3 章を参照）を入れる。
XLOW(i)：i=2～NP に変数の下限値を入れる。タイプ 1 以外の時は 0.0 を入れる。
XUP(i)：i=2～NP に変数の上限値を入れる。タイプ 1 以外の時は 0.0 を入れる。

INEQ(i)：i=2～MP に，制約式が等号ならば0を，制約式が右向き不等号（左向き不等号は右向きに変換しておく）ならば2を入れる。

C(i)：i=2～NP に変数のコストを入れる。

BETA(i)：i=2～MP に制約式の右辺の値を入れる。

A(i, j)：i=2～MP, j=2～NP に制約式の左辺の係数行列を入れる。但し，LPFILE を用いる場合はファイルに入れる。

8.2 出 力 情 報

IOPT：最適解の時は1，実行不能の時は2，非有界の時は3が入る。

XSOLUT(i)：最適解が求まった時，i=1 には評価関数の値，i=2～NP には変数の値が入る。

8.3 必要な配列宣言

1次元配列　大きさ INPL：IXTYPE

〃　NVAR：NBV, IX

〃　NP：XLOW, XUP, C, XSOLUT

〃　MP：INEQ, BETA, IBV, U, PAI, H, A*, YETA*

〃　IPMAX：IR**

2次元配列　大きさ MP×NP：A**

〃　MP×IPMAX：YETA**

* LPFILE でのみ必要とする行列。

** LPCORE でのみ必要とする行列。

8.4 必要なファイル定義文

LPFILE を用いる場合，必ず，次の FD 文でファイルを定義しなければならない。

```
*FD FO1,FILE=(TEMP,AF),UNIT=DPW  ,VOL=WORK,DEVD=DA,SPACE=(TRK,100,20)
*FD FO2,FILE=(TEMP,YETA),UNIT=DPW  ,VOL=WORK,DEVD=DA,SPACE=(TRK,100,20)
```

図 9

9. FORTRAN プログラム

サブルーチン LPCORE のプログラムを，以下に掲げる。

```
      SUBROUTINE LPCORE (NCONS,NVAR,EPS,IXTYPE,XLOW,XUP,INEQ,C,BETA,A,
     *                   XSOLUT,IOPT,MP,NP,INPL,IPMAX,IX,IBV,NBV,IR,U,PAI,
     *                   YETA,H)
      DIMENSION IXTYPE(INPL),IX(NVAR),IBV(MP),NBV(NVAR),XLOW(NP),XUP(NP)
     *         ,BETA(MP),INEQ(MP),C(NP),A(MP,NP),XSOLUT(NP)
      DIMENSION IR(IPMAX),U(MP),PAI(MP),YETA(MP,IPMAX),H(MP)
      IXTYPE(1)=3
      XLOW(1)=XUP(1)=0.
      DO 20 J=2,NP
   20 A(1,J)=-C(J)
      A(1,1)=1.
      DO 30 I=2,MP
   30 A(I,1)=0.
      BETA(1)=0.0
      CALL SCALEC(NVAR,NP,IXTYPE,IBV,NBV,XLOW,XUP,BETA,A,INEQ,MP,IX,
     *            INPL)
      CALL SMPLXC (NCONS,NVAR,MP,NP,INPL,EPS,IXTYPE,IX,IBV,NBV,
     *             BETA,A,IOPT,IPMAX,IR,U,PAI,YETA,H)
```

図 10

```
      WRITE(6,8000) IOPT
 8000 FORMAT(6H IOPT=I5)
      GO TO (400,500,600),IOPT
  500 WRITE(6,2600)
 2600 FORMAT(3H0   ,'INFEASIBLE SOLUTION')
      RETURN
  600 WRITE(6,2700)
 2700 FORMAT(3H0   ,'UNBOUNDED SOLUTION')
      RETURN
  400 CONTINUE
      CALL SOLTNC (NVAR,NP,MP,IXTYPE,IX,IBV,NBV,XLOW,XUP,BETA,C,
     *                         XSOLUT,INPL)
      RETURN
      END

      SUBROUTINE SCALEC(NVAR,NP,IXTYPE,IBV,NBV,XLOW,XUP,BETA,A,INEQ,MP,
     *          IX,INPL)
      DIMENSION IXTYPE(INPL),IX(NVAR),IBV(MP),NBV(NVAR),XLOW(NP)
     *         ,XUP(NP),BETA(MP),INEQ(MP),A(MP,NP)
      DO 100 J=1,NP
      IF(IXTYPE(J).NE.1) GO TO 100
      XY=XUP(J)-XLOW(J)
      DO 150 I=1,MP
      BETA(I)=BETA(I)-A(I,J)*XLOW(J)
      A(I,J)=A(I,J)*XY
  150 CONTINUE
  100 CONTINUE
      DO 200 J=1,NVAR
      NBV(J)=J+1
      IX(J)=0
  200 CONTINUE
      IBV(1)=1
      DO 300 I=2,MP
      INP=I-1+NP
      IBV(I)=INP
      IXTYPE(INP)=INEQ(I)
  300 CONTINUE
      RETURN
      END

      SUBROUTINE SMPLXC (NCONS,NVAR,MP,NP,INPL,EPS,IXTYPE,IX,IBV,NBV,
     *           BETA,A,IOPT,IPMAX,IR,U,PAI,YETA,H)
      DIMENSION IXTYPE(INPL),IX(NVAR),IBV(MP),NBV(NVAR),BETA(MP),
     *        A(MP,NP),IR(IPMAX),U(MP),PAI(MP),YETA(MP,IPMAX),H(MP)
      L=1
      S=0.0
      IP=0
 1111 CONTINUE
      IF(L.EQ.0) GO TO 110
      L=0
      U(1)=S
      DO 100 I=2,MP
      U(I)=0.0
      IBVAR=IBV(I)
      IF(IXTYPE(IBVAR).EQ.3) GO TO 100
      BET=BETA(I)
      IF(BET.GE.-EPS.AND.BET.LE.EPS) GO TO 100
      IF(BET.LT.-EPS) GO TO 150
      IF(IXTYPE(IBVAR)-1) 140,120,100
  120 IF(BET.LE.1.+EPS) GO TO 100
  140 U(I)=-1.0
      L=L+1
      GO TO 100
  150 U(I)=1.0
      L=L+1
  100 CONTINUE
      IF(L.NE.0) GO TO 160
  110 U(1)=1.0
      IFLAG=2
      GO TO 190
  160 IFLAG=1
  190 CONTINUE
      DO 200 I=1,MP
      PAI(I)=U(I)
```

図 11

```
200 CONTINUE
    IF(IP.EQ.0) GO TO 210
    DO 220 K=1,IP
    PAIR=0.0
    DO 230 I=1,MP
    PAIR=PAIR+PAI(I)*YETA(I,IP+1-K)
230 CONTINUE
    IRW=IR(IP+1-K)
    PAI(IRW)=PAIR
220 CONTINUE
210 CONTINUE
    D=1.0E8
    IS=0
    DO 380 K=1,NVAR
    J=NBV(K)
    IXTYP=IXTYPE(J)
    II=K
    IF(IXTYP-1) 380,352,370
370 MYU1=0
    GO TO 320
352 MYU1=1
320 IF(J-NP) 375,375,376
375 ALPHA=0.0
    DO 330 I=1,MP
    ALPHA=ALPHA+PAI(I)*A(I,J)
330 CONTINUE
    GO TO 390
376 ALPHA=PAI(J+1-NP)
390 CONTINUE
    IF(ALPHA.GE.-EPS.AND.ALPHA.LE.EPS) GO TO 380
    IF(ALPHA.LT.-EPS) GO TO 356
    IDEL1=-1
    IF(IXTYP.EQ.2) GO TO 380
    IF(IXTYP.EQ.3) GO TO 355
    IF(IX(II).EQ.0) GO TO 380
    GO TO 355
356 IDEL1=1
    IF(IXTYP.NE.1) GO TO 355
    IF(IX(II).EQ.1) GO TO 380
355 D1=IDEL1*ALPHA
    IF(D.LE.D1) GO TO 380
    D=D1
    IS=J
    INS=II
    IDEL=IDEL1
    MYU=MYU1
380 CONTINUE
    DELTA=FLOAT(IDEL)
    IF(IS.EQ.0) GO TO 222
    IF(IS.LE.NP) GO TO 365
    DO 366 I=1,MP
366 H(I)=0.
    H(IS+1-NP)=1.
    GO TO 367
365 DO 360 I=1,MP
360 H(I)=A(I,IS)
367 CONTINUE
    IF(IP.EQ.0)GO TO 420
    DO 400 K=1,IP
    IRW=IR(K)
    Z=H(IRW)
    H(IRW)=0.0
    DO 400 I=1,MP
    H(I)=H(I)+Z*YETA(I,K)
400 CONTINUE
420 CONTINUE
    THETA=9.99E+50
    IRW=0
    DO 500 I=1,MP
    IBVAR=IBV(I)
    IF(IXTYPE(IBVAR).EQ.3) GO TO 500
    IF(H(I).GE.-EPS.AND.H(I).LE.EPS) GO TO 500
    IF(BETA(I).GE.-EPS.AND.BETA(I).LE.EPS) GO TO 52
    IF(DELTA*H(I)*BETA(I).LT.0.) GO TO 53
```

図 12

```
      THETA1=BETA(I)/(DELTA*H(I))
      IEPS1=0
      GO TO 55
   52 IF(DELTA*H(I).GT.0.) GO TO 54
      IF(IXTYPE(IBVAR)-1) 54,56,500
   53 IF(IXTYPE(IBVAR).NE.1) GO TO 500
      IF((BETA(I)-1.).GE.-EPS.AND.(BETA(I)-1.).LE.EPS) GO TO 58
      DHB=DELTA*H(I)*(BETA(I)-1.)
      IF(DHB.GT.0.) GO TO 56
      GO TO 500
   54 THETA1=EPS/ ABS(H(I))
      IEPS1=1
   55 LAMDA1=0
      GO TO 57
   56 THETA1=(BETA(I)-1.)/(DELTA*H(I))
      IEPS1=0
      LAMDA1=1
      GO TO 57
   58 THETA1=EPS/ ABS(H(I))
      IEPS1=1
      LAMDA1=1
   57 IF(THETA.LE.THETA1) GO TO 500
      THETA=THETA1
      IEPS=IEPS1
      IRW=I
      LAMDA=LAMDA1
  500 CONTINUE
      IF(IRW.GT.0) GO TO 666
      IF(MYU.EQ.0) GO TO 444
  666 CONTINUE
      IF(IEPS.EQ.1) THETA=0
      IF(MYU.EQ.0) GO TO 61
      IF(THETA.GT.1.) GO TO 62
      MYU=0
  630 CONTINUE
      IF(IDEL.EQ.1) GO TO 61
      BETA(IRW)=1.
      GO TO 64
   61 BETA(IRW)=0.
   64 YETR=1./H(IRW)
      H(IRW)=-1.
      GO TO 63
   62 THETA=1.
   63 DO 600 I=1,MP
  600 BETA(I)=BETA(I)-DELTA*THETA*H(I)
      IF(MYU.EQ.0) GO TO 65
      IF(IDEL.EQ.1) GO TO 66
      IX(INS)=0
      GO TO 69
   66 IX(INS)=1
      GO TO 69
   65 NBV(INS)=IBV(IRW)
      IX(INS)=LAMDA
      IBV(IRW)=IS
      IP=IP+1
      DO 620 I=1,MP
  620 YETA(I,IP)=-H(I)*YETR
      YETA(IRW,IP)=YETR
      IR(IP)=IRW
   69 CONTINUE
      GO TO 1111
  222 IF(IFLAG.EQ.1) GO TO 333
      IOPT=1
      RETURN
  333 IOPT=2
      RETURN
  444 CONTINUE
      IOPT=3
      RETURN
      END

      SUBROUTINE SOLTNC (NVAR,NP,MP,IXTYPE,IX,IBV,NBV,XLOW,XUP,BETA,C,
     *     XSOLUT,INPL)
      DIMENSION  IBV(MP),IXTYPE(INPL),XSOLUT(INPL),BETA(MP),XUP(NP),
```

図 13

```
     1XLOW(NP),NBV(NVAR),IX(NVAR),C(NP)
      DO 100 J=1,NP
  100 XSOLUT(J)=0.
      DO 410 I=1,MP
      J=IBV(I)
      IF(J.GT.NP) GO TO 410
      IF(IXTYPE(J).NE.1) GO TO 430
      XSOLUT(J)=BETA(I)*(XUP(J)-XLOW(J))+XLOW(J)
      GO TO 410
  430 XSOLUT(J)=BETA(I)
  410 CONTINUE
      DO 420 I=1,NVAR
      J=NBV(I)
      IF(IXTYPE(J).NE.1) GO TO 420
      IF(IX(I).EQ.1) GO TO 440
      XSOLUT(J)=XLOW(J)
      GO TO 420
  440 XSOLUT(J)=XUP(J)
  420 CONTINUE
      ZOBJ=0.
      DO 500 J=2,NP
  500 ZOBJ=ZOBJ+C(J)*XSOLUT(J)
      XSOLUT(1)=ZOBJ
      RETURN
      END
```

図 14

10. 使　用　例

第 9 章に掲げた Fortran プログラム LPCORE を用いた使用例を次に示す。

10.1 メインプログラム

```
C     MAIN PROGRAM   ( USER'S MAIN )
      DIMENSION IXTYPE(21),IX(13),IBV( 8),NBV(13),
     1          XLOW(14),XUP(14),BETA( 8),C(14),INE@( 8),U( 8),PAI( 8),
     2          H( 8),A( 8,14),YETA( 8, 30),XSOLUT(14),IR( 30)
      READ(5,1500) NCONS,NVAR,EPS
 1500 FORMAT(2I5,F10.0)
      IPMAX=30
      MP=NCONS+1
      NP=NVAR+1
      INPL=MP+NVAR
      READ(5,1000) (IXTYPE(J),C(J),XLOW(J),XUP(J),J=2,NP)
 1000 FORMAT(I10,3F10.4)
      READ(5,1200) (INE@(I),BETA(I),I=2,MP)
 1200 FORMAT(I5,F15.2)
      DO 10 J=2,NP
      READ(5,1300) (A(I,J),I=2,MP)
 1300 FORMAT(7F10.5)
   10 CONTINUE
      CALL LPCORE (NCONS,NVAR,EPS,IXTYPE,XLOW,XUP,INE@,C,BETA,A,XSOLUT,
     *     IOPT,MP,NP,INPL,IPMAX,IX,IBV,NBV,IR,U,PAI,YETA,H)
      WRITE(6,2400) XSOLUT(1)
 2400 FORMAT(1H ,'OPTIMAL VAL=',F10.4)
      WRITE(6,2500) (XSOLUT(J),J=2,NP)
 2500 FORMAT(1H ,'OPTIMAL SOL=',/,(1H ,7F10.4))
      STOP
      END
```

図 15

10.2 デ　ー　タ

```
7   13    0.0001
       1        0.        10.       100.
       0        0.         0.         0.
       1        0.        10.       100.
       0        0.         0.         0.
       1        0.        10.       100.
       0        0.         0.         0.
       1        0.        10.       100.
       1       -2.         0.       200.
       1       -2.         0.       200.
       1       -2.         0.       200.
       1       -2.         0.       200.
       1       -2.         0.       200.
       1       -2.         0.       200.
0              50.
0              50.
0              50.
0              50.
0              50.
0              50.
0              50.
    1.         0.         0.         0.         0.         0.         0.
    0.         1.         0.         0.         0.         0.         0.
    0.         0.         1.         0.         0.         0.         0.
    0.         0.         0.         1.         0.         0.         0.
    0.         0.         0.         0.         1.         0.         0.
    0.         0.         0.         0.         0.         1.         0.
    0.         0.         0.         0.         0.         0.         1.
   -1.         1.         0.         0.         0.         0.         0.
    0.        -1.         1.         0.         0.         0.         0.
    0.         0.        -1.         1.         0.         0.         0.
    0.         0.         0.        -1.         1.         0.         0.
    0.         0.         0.         0.        -1.         1.         0.
    0.         0.         0.         0.         0.        -1.         1.
```

図 16

10.3 結　　果

```
 IOPT=    1
 OPTIMAL VAL= -300.0000
 OPTIMAL SOL=
   100.0000     0.0      100.0000     0.0      100.0000     0.0       50.0000
    50.0000     0.0       50.0000     0.0       50.0000     0.0
```

(注)　この例題の使用 CORE 量は 0.5 K, 計算時間は 100 MS であった。

図 17

11. あ と が き

本報告のプログラムコードは，著者らによって，数多くの制御または経済の問題に適用され，いずれも成功をおさめている。したがって，プログラムの虫もなくなり，非常に信頼度の高いものになっている。

なお，使用計算機は，北大大型計算機センター Facom 230-75 である。

終りに，本プログラムコードは，北大大型計算機センタープログラム開発により，作成された。お世話を受けた北大計算機センターの方々に，深く謝意を表する。

文　　献

1) Orchard-Hays, W.: Advanced Linear-Programming Techniques, (1968) McGraw-Hill.
2) W. オーチャード・ヘイズ：コンピュータによる線形計画法（上記訳本），(昭和 48)，培風館。
3) 平本厳，長谷彰：線形計画法，(昭和 48)，培風館。