УДК 519.95

С.М.АНЦЫЗ, В.И. ХОХЛЮК

АЛГОРИТМ I ГОМОРИ И МОДИФИКАЦИЯ МАРТИНА ДЛЯ ЭТОГО АЛГОРИТМА

(Две процедуры для численного решения линейной полностью целочисленной задачи)

Линейная полностью целочисленная задача формулируется следующим образом.

Найти максимум линейной функции

$$L(Y) = c_0 + \sum_{j=1}^{n} c_j y_j$$

при условиях:

$$1) \quad \sum_{j=1}^{n} d_{ij} y_j \leq d_i \qquad (i \in I_1),$$

$$\sum_{j=1}^{n} d_{ij} y_j = d_i \qquad (i \in I_2),$$

$$2) \quad y_j \geq 0 \qquad (j=1,\dots,n),$$

$$3) \quad y_j \equiv 0 \pmod 1 \qquad (j=1,\dots,n),$$

(последнее условие есть условие целочисленности), где I_1 и I_2 - два взаимно дополнительных подмножества множества индексов $I=\{1,\dots,m\}$, m - число ограничений (равенств и неравенств), n - число переменных. Будем считать, что все параметры c_j, d_{ij} и d_i - целые числа, а многогранное множество, задаваемое условиями I), 2), ограничено.

Рассматриваемой задаче ставится в соответствие матрица $E=\| e_{ij} \|$ ($i = 0,1,\dots,m+1$; $j=1,\dots,n+2$), где

$$e_{i1} = 0 \qquad (i=0,\ i \in I_1),$$

$$e_{i1} = 1 \qquad (i=m+1,\ i \in I_2),$$

$$e_{02} = c_0,\ e_{i2} = d_i,\ e_{m+1,2} = 1 \qquad (i \in I),$$

$$\left.\begin{aligned} &e_{0,j}=c_{j-2},\ e_{ij}=d_{i,j-2} \\ &e_{m+1,j}=0 \end{aligned}\right\}\ (i\in I,\ j=3,\dots,n+2).$$

$i \backslash j$	1	2	3 . . . $n+2$
0	0	c_0	c_1, . . . , c_n
1 I_1 i I_2 m	⋮ 0 ⋮ 1 ⋮ 0 ⋮ 1 ⋮	d_i	d_{i1} . . . d_{in}
$m+1$	1	1	0, . . . , 0

Алгоритм I Гомори и модификация Мартина для этого алгоритма, реализующие метод отсечения, записаны в виде процедур на α - варианте языка АЛГОЛ-60. Параметрами процедуры являются переменные N, m, n, массивы b и a, где N -номер задачи, m - число ограничений (равенств и неравенств), n - число переменных.

Массив b заполняется значениями величин: ε_1 - точность (пусть Z - элемент симплексной таблицы; если $|Z|<\varepsilon_1$ то $Z:=0$); ε_2 -точность для дробной части $f(Z)$ числа Z (пусть $\delta=\begin{cases}1+[\lg|Z|] & |Z|\geq 1,\\ 0 & |Z|<1;\end{cases}$

если $f(Z)<\varepsilon_2\cdot 10^{\delta}\vee f(Z)>1-\varepsilon_2\cdot 10^{\delta}$, то $f(Z):=0$); p - наибольшее возможное число ненулевых элементов симплексной таблицы ($p = \min(P/2, (m+n+4)\cdot(n+2))$), где P - свободная часть машинной памяти); Q - число итераций (преобразований симплексной таблицы)для вынужденного останова; p_1 - период выдачи (если число итераций кратно p_1 - происходит выдача); p_2 - период применения критерия выбора строки, порождающей отсечение (если число решенных задач линейного программирования кратно p_2, для построения отсечения выбирается первая сверху строка, содержащая в столбце 2 элемент с отличной от нуля дробной частью); R - вид выдачи.

При обращении к процедуре массив a размерности $p\times 2$ заполняется отличными от нуля элементами e_{ij} и номерами i их

позиций в столбце матрицы E , а оставшиеся ячейки -нулями. Каждому ненулевому элементу соответствует строка массива a , в первом столбце которой стоит номер позиции элемента, а во втором - величина элемента. В массиве a начало каждого столбца матрицы E помечается : к номеру позиции первого ненулевого элемента прибавляется M (считаем, что число переменных и ограничений в сумме меньше $M - 2$, например, $M = 10000$). В процессе выполнения процедуры в массиве a находятся только ненулевые элементы симплексной таблицы и номера их позиций в столбце.

Общая схема процедуры

Блок В0 - построение начальной симплексной таблицы (начального базисного решения), с учетом заданных параметров целочисленной задачи линейного программирования.

Блок Р0 - вспомогательные процедуры.

Блок В1 - построение двойственно допустимой симплексной таблицы (решения).

Блок В2 - построение оптимальной симплексной таблицы (решения) задачи линейного программирования.

Блок В3 - построение дополнительного ограничения (ограничений) и связанное с ним преобразование симплексной таблицы.

В процедуре предусмотрены разнообразные выдачи. Вид выдачи определяется двузначным натуральным числом R . Первая цифра этого числа может принимать значения 1,..., 7, которым соответствуют: 1 - вывод параметров наблюдения за счетом и вывод плана после решения задачи линейного программирования, останов; 2 - вывод параметров наблюдения и вывод плана после решения целочисленной задачи, останов; 3- 1 без останова и 2; 4 - вывод параметров наблюдения, плана и матрицы (симплексной таблицы) после решения задачи линейного программирования, останов; 5-вывод параметров наблюдения, плана и матрицы после решения целочисленной задачи,останов; 6-4 без останова и 5; 7 - вывод параметров наблюдения после решения целочисленной задачи, останов.

Вторая цифра R может принимать значения 1,...,8, которым соответствуют: 1 -выдача через период не происходит; 2 -через период выдаются параметры наблюдения; 3-через период выдаются параметры наблюдения и план; 4- через период выдаются параметры наблюдения, план и матрица; 5 - вывод вводимых данных(для контроля) и 1; 6 - вывод вводимых данных и 2 ; 7 - вывод вводимых данных и 3; 8- вывод вводимых данных и 4. Если вторая циф-

ра $R \geq 5$, в начале блока ВО $b[7]$ уменьшается на 4.

Кроме выдач, определяемых числом R, всегда выдаются: перед началом работы алгоритма величины i (номер алгоритма), N, m и n; после блока ВО- величины $c[4]$, $c[5]$, $c[6]$ (см. ниже) и массив b; после решения задачи линейного программирования и в случае вынужденного останова - величины C (вид останова) и массив c (параметры наблюдения за счетом).

Под планом здесь понимается второй столбец матрицы. В позиции с номером 0 находится значение целевой функции, в позициях с номерами I,..., m, $m+2$ - значения слабых переменных, в позиции с номером $m+$ I всегда стоит I. В позициях с номерами $m+3,\ldots,m+n+2$ стоят значения основных переменных $y_1,\ldots,y_n$. При выдаче плана номера позиций основных переменных уменьшаются на $m+2$. Выдаются только ненулевые элементы.

Параметры наблюдения за счетом вычисляются после каждой итерации и заполняют массив C следующим образом: $C[1]$ -число итераций; $C[2]$ и $C[3]$ связаны со значением целевой функции ($L(Y) = \Omega \cdot C[2] + C[3]$, где Ω - достаточно большое положительное число); $C[4]$ - число ненулевых элементов матрицы, максимальное по всем предыдущим итерациям; $C[5]$ и $C[6]$ - абсолютная величина элемента матрицы, соответственно, минимальная и максимальная по всем предыдущим итерациям; $C[7]$ - абсолютная величина произведения центральных элементов, максимальная по всем предыдущим итерациям; $C[8]$ - число итераций для построения двойственно допустимой симплексной таблицы (до выдачи после решения задачи линейного программирования включительно), далее - число решенных задач линейного программирования.

Вид останова C принимает значения I,...,6, которым соответствуют: I - решена задача линейного программирования; 2 -решена целочисленная задача; 3- задача линейного программирования несовместна; 4- целочисленная задача несовместна; 5 - вынужденный останов из-за нехватки памяти; 6- вынужденный останов по заданному числу итераций.

АЛЬФА - программа алгоритма I Гомори.

```
procedure GOMORY I (N,m,n,b,a);
   value N,m,n; integer N,m,n;
   array a,b;
  begin
   integer p,pI,i,j,k,l,q,r,s,t;
   real    d,g,h,u,v,x,y,z,D,M;
   real array c[I:8];
   function Ø(z)=if abs(z)< b[I]  then 0 else z;
   function e(z)=if z > M then z-M else z;
   procedure A(gI,g2);
     value gI,g2; real gI,g2;
    begin
       if Ø(g2)≠0 then  {a[q,I]:=gI;a[q,2]:=g2; q:=q-I}
    end A;
   procedure P;
    begin
      i:=0;
      for l:=q+I,...,p do  {i:=i+I;
        a[i,I]:=a[l,I]; d:=a[l,2]; a[i,2]:=d;
        d:=abs(d); if c[5] > d then c[5]:=d;
        if c[6]< d then c[6]:=d} ;
      pI:=p-q; if c[4]< pl then c[4]:=pI;
      a[pl+I,I]:=M
    end P;
BO:Построение начальной симплексной таблицы:
     i:=I; p:=b[3]; M:=I0000; c[ ]:=0; c[5]:=M;
     for r:=I,...,p do if a[r,I]=0 then go to BOI;
BOI: if frac (b[7]/10)> 0. 45 then  {b[7]:=b[7]-4;
      СП  0I76  (1I53,a[I,I], a[r,2],0,0)} ;
```

```
    j:=N; k:=m; l:=n; вывод  (i,j,n,l);
    for k:=2,...,p do if a[k,I] > M  then go to B02;
B02: l:=r:=r-I; q:=p; k:=k-2;
    for i:=n+2 step-I until 2 do
    begin
      v:=0; if i≠2 then A(m+1,-I);
      for j:=k step-I until I do  {d:=e(a[j,I]);
B03:    g:=e(a[l,I]); if g=d then v:=v-a[l,2] ;
        if g > d then  {if a[l,I] > g then go to B04;
                        l:=l-I; go to B03}}   ;
B04:  for s:=l step-I until I do
                      if a[s,I] > M THEN go to B05;
B05:  if a[s,I]=M then a[s,2]:=-a[s,2]; A(m+2,v);
      for j:=r step-I until s do A(a[j,I],a[j,2]);
      r:=l:=s-I
    end I;
    A(m+I000I,I); P;
    вывод   (c[4],c[5],c[6],b);
PO:ПРОЦЕДУРЫ:
  begin
   procedure f(z);
      value z;
    begin
      real bI;
       d:=frac(z); bI:=b[2]; z:=abs(z);
       while z >I do  {z:=z/I0; bI:=bI × I0} ;
       if d < bI ∨ d >I-bI then d:=0
    end f;
   procedure print(T);
    comment  производится выдача через период ( T=I ), после
    решения задачи линейного программирования (  T=2 ),после
```

```
решения целочисленной задачи ( T=3 );
  value T; integer T;
 begin
    if T≠I then {r:=N; ВЫВОД(r)} ; d:=b[7]/10;
    d:=if T=I then frac(d) else entier(d);
    if d<0.I5 then go to PII;
    for r:=2,...,p do if a[r,I]≥M then go to PIO;
PIO:    c[2]:=if a[I,I]=M then a[I,2] else 0;
    c[3]:=if a[r,I]=M then a[r,2] else 0;
    if N=0 then c[8]:=c[I];
      ВЫВОД (c);
    if d<0.25∨d=7 then go to PII;
    if T=2 then {if d=2∨d=5 then go to PII} ;
    k:=M; u:=a[r,I]:=a[r,I]-M; s:=r;
    while u<m do {if u>m+2 then
        {a[s,I]:=u-m-2; if k>s then k:=s} ;
     s:=s+I; u:=a[s,I]} ; s:=s-I;
      СП 0 176(II53,a[r,I],a[s,2],0,0);
    a[r,I]:=a[r,I]+M;
    for r:=k,...,s do a[r,I]:=a[r,I]+m+2;
    if d<0.35∨d=3 then go to PII;

    if d=1∨d=2 then go to END;
      СП 0 176 (II53,a[I,I],a[pI,2],0,0);
    if d=4 then go to END;

PII: end printa;
  procedure lex(X);
   comment определяется номер j центрального столбца αj
  симплексной таблицы,номера S и t его начала и конца в
```

массиве a. $(1/d_j)\alpha_j = \min\limits_{\substack{\ell = s, n+1 \\ d_\ell > 0}} (1/d_\ell)\alpha_\ell$ (сравнение векторов производится в лексикографическом смысле). При $X=1$ $d_\ell = 1$, при $X=2$ $(-d_\ell)$ – величина ℓ-го элемента y-ой строки симплексной таблицы (центральной), при $X=3$ d_ℓ – дробная часть этого элемента. В x запоминается d_j;

```
    value X; integer X;
  begin
      s:=0; q:=pI; if X=I then y:=-I;
      for l:=n+2 step-I until 3 do
      begin
        d:=2-X;
        for r:=q step-I until I do
           {if a[r,I]>K then go to P20;
            if a[r,I]=y  then d:=-a[r,2]} ;
P20:    a[r,I]:=a[r,I]-K; if a[r,I]=y then
          d:=-a[r,2]; if X=3 then f(-d);
        if Ø(d)>0 then {if s>0 then go to P2I;
           go to P22} ; go to P23;
P2I:    k:=s-r; v:=a[s,I]-k;
        for i:=r,...,q do
        begin
          g:=a[i,2]/d; h:=a[i+k,2]/x;
          if a[i,I]<v then
            {if g<0 then go to P22; go to P23} ;
          if a[i,I]>v then
            {if h>0 then go to P22; go to P23} ;
          if g  h then go to P22;
          if g>h then go to P23; v:=a[i+k+i,I]
        end 3;
        if a[q+k+I,2]/x<0 then go to P23;
```

```
P22:        s:=r; j:=1; x:=d; t:=q;
P23:        a[r,I]:=a[r,I]+M; q:=r-I
         end 2;
         if s=0 then {N:=if N=I then 3 else 4;
           go to K}
      end lexa;
      procedure trans (X);
```

comment массив a заполняется ненулевыми значениями элементов симплексной таблицы, преобразованной по формулам
$\alpha'_j = \sigma(1/a_j)\alpha_j$, $\alpha'_\ell = \alpha_\ell + d_\ell \alpha'_j$ $(\ell = \overline{1, n+2})$.
В качестве центральной выбирается: строка, состоящая из 1 $(X=1)$, y-ая строка ($X=2$), строка, состоящая из взятых со знаком минус дробных частей элементов y-ой строки, $(X=3)$, s, t, j, x, y те же, что и в lex;

```
       value X; integer X;
      begin
        k:=pI; d:=if X=I then x else-x; D:=D × d;
        r:=p; l:=p-t; q:=p:=s+l-I;
        for i:=s,...,t do {a[i+1,I]:=a[i,I];
           a[i+1,2]:=a[i,2]:=-a[i,2]/d} ;
       s:=s+1; a[s,I]:=a[s,I]-M;
       for i:=n+2 step-I until I do
       begin
        if i=j then {l:=s-r+k; go to P33} ;
        d:=0;
        for l:=k step-I until I do
         {if a[l,I]≥M then go to P30;if a[l,I]=y then d:=a[l,2]};
P30:   a[l,I]:=a[l,I]-M; if a[l,I]=y then
       d:=a[l,2]; if X=3 then {f(d); d:=-d} ;
       if X=I then {if i=2 then go to P33; d:=I} ;
       if d=0 then go to P33;
       for t:=r step -I until s do
```

```
        begin
           g:=a[t,1]; u:=_[t,2] * d;
P9I:       if k<1 then go to P32;
           h:=a[k,I]; v:=a[k,2];
           if g ≤ h then {k:=k-I; if g=h then
          A(h,u+v) else  {A(h,v); go to P3I}}  else {
P 32      A(g,u)}  end 5;
        if k > q then {N:=5; go to K} ;
P33:    for t:=? step -I until 1 do A(a[t,1],a[t,2]);
        k:=1-I; a[q+I,I]:=e(a[q+I,I])+M
      end 4;
      P; p:=r; c[I]:=c[I]+I;
      if c[7] < abs(D) then c[7]:=abs(D);
      if c[I]> b[4] then {N:=6; go to K} ;
      if frac (c[I]/b[5])=0 then print(I)
     end transa;
    procedure comp (F);
   comment  выполняется преобразование симплексной таблицы;
           value F; integer F;
      begin
P40:       i:=F-I;
```

P4I: определение номера y отрицательного элемента F -го столбца:

```
           i:=i+I; if a[i,2] < 0 ^ a[i,I]≠M then go to P42;
           if a[i+I, ] < M then go to P4I;
           if F=I then {if i=I then go to B2;N:=if N≤I then 3
                                     else 4; go to K} ;
           go to B3;
P42 :      y:=e(a[i,I]); lex(2); trans(2); go to P40
      end compa;
```

BI: построение двойственно допустимой симплексной таблицы (решения):

```
      N:=0; D:=I; lex (I);
      if a[s,2]> 0 then go to B2; trans (I); comp (I);
```

B2: построение оптимальной симплексной таблицы:

```
    N:=N+I; if N=I then c[8]:=c[I]; comp(2);
B3: построение дополнительного ограничения:
    if N=I then print(2); c[8]:=N; g:=0; i:=2;
    u:=a[2,I]-M;
    while u < M do {f(a[i,2]); if d > g then
      {y:=u; if frac(N/b[6])=0 then go to B3I;
       g:=d} ; i:=i+I; u:=a[i,I]} ;
    if g=0 then {N:=2; go to K} ;
B3I:  lex(3); trans(3); go to B2;
K:    print(3);
END:  end PO;
   end GOMORY I;
```

АЛЬФА -программа алгоритма I Гомори с модификацией Мартина.

```
   procedure MARTIN (N,m,n,b,a);
      value N,m,n; integer N,m,n;
      array a,b;
   begin
     integer p,pI,i,j,k,l,q,r,s,t;
     real    d,g,h,u,v,x,y,z,D,M;
     real array c[I:8];
     function Ø(z)=if abs(z) < b[I] then 0 else z;
     function e(z)=if z ≥ M then z-M else z;
     procedure A(gI,g2);
        value gI,g2; real gI,g2;
      begin
         if Ø(g2)≠0 then {a[q,I]:=gI
                          a[q,2]:=g2; q:=q-I}
      end A;
     procedure P;
```

```
begin
  i:=0;
  for l:=q+I,...,p do {i:=i+I;a[i,I]:=a[l,I];
        d:=a[l,2];a[i,2]:=d;d:=abs(d);
        if c[5]>d then c[5]:=d; if  c[6]<d then c[6]:=d} ;
  pI:=p-q; if c[4] < pI then c[4]:=pI;
  a[pI+I,I]:=M
end P;
BO: Построение исходной информации:
  i:=II; p:=b[3]; M:=I0000; c[ ]:=0; c[5]:=M;
  for r:=I,...,p do if a[r,I] =0 then go to B
BOI: if frac(b[7]/I0)>0.45 then {b[7]=b[7]-4;
  СПОI76(II53,a[I,I], a[r,2], 0,0)} ;
  j:=N; k:=m; l:=n;  ВЫВОД (i,j,k,l);
  for k=2,...,p do if a[k,I] ≥ M then go to BO2;
BO2: l:=r:=r-I; q:=p; k:=k-2;
  for i:=n+2 step -I until 2 do
  begin
    v:=0; if i≠2 then A(m+i,-I);
    for i:=k step -I until I do {d:=e(a[j,I]);
BO3:    g:=e(a[l,I]); if g=d then v:=v-a[l,2] ;
      if g > d then {if a[l,I] > g then go to BO4;
                     l:=l-I; go to BO3)} ;
BO4:  for s:=i step-I until I do
                if a[s,I] ≥ M then go to BO5;
BO5:  if a[s,I]=M then a[s,2]:=-a[s,2] ; A(m+2,v);
      for j:=r step -I until s do A(a[j,I],a[j,2]);
      r:=l:=s-I
  end I;
  A(m+I000I,I); P;
   ВЫВОД  (c[4],c[5],c[6],b);
PO: ПРОЦЕДУРЫ
  begin
    procedure f(z);
```

```
    value z;
  begin
    integer iI; real bI;
      d:=frac(z); bI:=b[2]; z:=abs(z);
    for iI:=I,...,IO do {if z<I then go to POI;z:=z/IO;bI:=bI×IO};
POI:    if d < bI ∨ d > I-bI then d:=O
    end f;
    procedure print (T);
    comment производится выдача через период ( T=I ), после ре-
    шения задачи линейного программирования ( T=2 ), после ре-
    шения целочисленной задачи (T=3); value T; integer T;
    begin
      if T=I then вывод (N); d:=b[7]/IO;
      d:=if T=I then frac(d) else entier(d);
      if d < O,I5 then go to PII;
      for r:=2,...,p do if a[r,I] ≠ M then go to PIO;
PIO:  c[2]:=if a[I,I]=M then a[I,2] else O;
      c[3]:=if a[r,I]=M then a[r,2] else O;
      if c[7]< abs(D) then c[7]:=abs(D);
      if N=O then c[8]:=c[I];
           вывод (c);
      if d < O,25 ∨ d=I then go to PII;
      if T=2 then {if d=2 ∨ d=5 then go to PII};
      k:=M; u:=a[r,I]:=a[r,I]-M; s:=r;
      while u< M do {if u > m+2 then
               {a[s,I]:=u-m-2; if k > s then k:=s};
        s:=s+I; u:=a[s,I]}; s:=s-I;
       СПОI76(II53,a[r,I],a[s,2],O,O);
      a[r,I]:=a[r,I]+M;
      for r:=k,...,s do a[r,I]:=a[r,I]+m+2;
      if d < O,35 ∨ d=3 then go to PII;
      if d=I ∨ d=2 then go to PII;
```

```
        CR 0176  (1153,a[I,I],a[pI,2],0,0);

        if d=4 then go to END;

PII: end printa;
    procedure lex(X);
```

comment определяется номер s в массиве a начала и номер j столбца α_j симплексной таблицы, для которого $\frac{1}{d_\ell}\alpha_\ell$ -лексикографически наименьший. При $X=1$, $d_\ell=1$, при $X=2$, d_ℓ -величина ℓ-го элемента i_*-ой строки матрицы, при $X=3$, d_ℓ - дробная часть этого элемента. В x запоминается d_j. $y = i_*$ - номер центральной строки;

```
      begin
        s:=0; q:=pI; if X=I then y:=-I;
        for l:=n+2 step -I until 3 do
        begin
          d:=2-X;
          for r:=q step -I until I do
           {if a[r,I] ≥ M then go to P20;
            if a[r,I]=y then d:=-a[r,2]} ;
P20:      a[r,I]:=a[r,I]-M; if a[r,I]=y then
          d:=-a[r,2]; if X=3 then f(-d);
          if Ø(d)>0 then  {if s>0 then go to P2I;
                  go to P22} ; go to P23;
P2I:      k:=s-r;
          for i:=r,...,q do
          begin
            g:=Ø(a[i,2]/d); h:=Ø(a[i+k,2]/x);
            v:=e(a[i+k,I]);
            if a[i,I] < v ∨ v ≥ M then
              {if g<0 then go to P22; go to P23} ;
            if a[i,I] > v then
              {if h>0 then go to P22; go to P23} ;
```

```
        if g < h then go to P22;
        if g > h then go to P23;
      end 3;
      if a[q+k+I,I] < M ∧ a[q+k+I,2]/x ≤ 0
                    then go to P23;
P22:  s:=r; j:=l; x:=d;
P23:  a[r,I]:=a[r,I]+M; q:=r-I
    end 2;
    if s=0 then {N:=if N ≤ I then 3 else 4;
                 go to K}
  end lexa;
  procedure trans (X);
```

comment массив a заполняется ненулевыми значениями элементов симплексной таблицы, преобразованной по формулам $\alpha'_j = -\frac{1}{d_j}\alpha_j$, $\alpha'_\ell = \alpha_\ell + d_\ell \alpha'_j$ $(\ell = 1, \ldots, n)$. В качестве преобразующей выбирается строка, состоящая из I(X=I), i. -ая строка (X=2), строка, состоящая из дробных частей элементов i. -ой строки (X=3), строка $\frac{\lambda[i] - d_\ell}{\lambda_j}$, где d_ℓ то же, что при X=2 (X=4).
s,j,x,y те же, что и в lexe;

```
    begin
      i:=pI; d:=if X=I then x else -x;
      if X=4 then {for i:=s,...,p do
        if y=e(a[i,I]) then {d:=a[i,2]; go to P30}
P30:  d:=-d/λ[j]} ; D:=D × d;
      for r:=s,...,p do {a[r,2]:=-a[r,2]/d;
                 if a[r+I,I] ≥ M then go to P3I} ;
P3I:  q:=p;
      for i:=n+2 step -I until I do
      begin
        u:=0; if i=j then {r:=q; go to P34} ;
        if X=I then {if i=2 then go to P34;
                             go to P32} ; d:=0;
```

```
     for l:=k, l-I while u<M do
      {if e(a[l,I])=y then {d:=a[l,2];go to P32};u:=a[l,I]};
P32: u=0; if X=3 then {f(d); d:=-d};
     if X=4 then d:=Ø((λ[i]-d/λ[j]);
     if d=0 then go to P34;
     for l:=r step -I until s do
     begin
       g:=e(a[l,I]); v:=a[l,2] x d;
P33:   if u<M then {u:=a[k,I]; t:=h:=e(u);w:=a[k,2]};
       if g>t then A(g,v) else {k:=k-I;
          if u≠t then t:=-I; if g=h then A(h,w+v)
             else {A(h,w); go to P33}};
     end 5;
     if t=-I then a[q+I,I]:=a[q+I,I]+M;
     if k>q then {N:=5; go to K};
P34:   while u<M do {u:=a[k,I];A(u,a[k,2]);k:=k-I};
                           if i=j then s:=q+I;
     end 4;
     P; c[I]:=c[I]+I; if c[I]>b[4] then {N:=6;
        go to K}; if frac(c[I]/b[5])=0 then print(I)
    end transa;
    procedure comp(F);
    comment выполняется преобразование симплексной таблицы;
      begin
P40:     j=F-I;
P4I: определение номера y отрицательного элемента F-го столб-
    ца:
         i:=i+i; if a[i,2]<0 ∧ a[i,I]≠M then go to P42;
         if a[i+I,I]<M then go to P4I;
         if F=I then {if i=I then go to B2;
            N:=if N≤I then 3 else 4; go to K};
         go to B3;
P42:     y:=e(a[i,I]); lex(2); trans(2); go to P40
      end compa;
```

```
BI: построение двойственно допустимого решения:
    N:=0; D:=I, lex(I);
    if a[s,2] >0 then go to B2; trans(I); comp(I);
B2: построение оптимального решения задачи линейного програм-
    мирования:   N:=N+I;  if N=I then c[8]:=c[I];comp(2);
B3: построение дополнительного ограничения:
    if N=I then print(2); c[8]:=N;g:=u:=0;i:=2;
    while u<M do {f(a[i,2]);if d>g then
      {if frac(N/B[6])=0 then go to B3I; g:= l; j:=i} ;
      i:=i+I; u:=a[i,I]} ;
    if d>0 then {i:=j; go to B3I} ; N:=2; go to K;
B3I: y:=e(a[j,I]); lex(3);
построение отсечения Гомори:
    if frac(N/b[6])=0 then  {trans(3); go to B2} ;
построение отсечения Мартина:
     λ[ ]:=0; i:=0;
    for k:=I,...,pI do {if a[k,I]≥M then i:=i+I;
     if e(a[k,I])=y then  λ [i] :=a[k,2]} ;
B32: f(λ[j]); if d=0 then go to B33; λ[j]:=λ[j]/d;
   for i:=I,...,n+2 do if i≠j  then
     {f(λ[i]); λ[i]:=λ[i]-d × λ [j]} ; go to B32;
B33: trans(4); for i:=I,.. ,pI do  {f(a[i,2]);
                    if d>0 then go to B3I} ;
   for k:=I,...,pI do a[k,2]:=okp(a[k,2]);
   go to BI;
K:   print(3);
END: end PO;
  end MARTIN;
```

Рассмотрим иллюстративный пример.

Найти максимум $y_3 - y_2$

при условиях: $y_j \geq 0$, $y_j = 0 \ (mod\, 1)$ $(j = 1, 2, 3)$ и

$$4y_1 + 5y_2 \leq 10,$$
$$-3y_1 + 2y_2 \leq 7,$$
$$2y_1 + y_3 = 5.$$

Для проведения расчета с помощью процедуры можно использовать, например, следующую α - программу.

```
begin integer  NN,mm,nn,real array bb[I:7];
      ввод  (NN,mm,nn,bb);
begin real array aa[I:bb[3],1:2];
      procedure GOMORY 1 (N,m,n,b,a);
      ввод  (aa); GOMORY I (NN,mm,nn,bb,aa);
end; end *
```

Для рассматриваемого примера параметры процедуры принимают значения: NN=I (например); mm=3; nn=3; массив bb -10^{-6}, 10^{-7}, 50, 100, 5, 10, 68 (например, это параметры управления, которые можно менять); массив aa - 10003, I,4,I; 10001,10,2, ',3,5,4, I; 10000, I,I,4,2, -3,3,2; 10000, - I,I,5,2,2;10003, I; 0, 0.

Замечание. После ненулевых элементов в массиве a должен обязательно стоять хотя бы один 0 (признак конца).

Поступила в редакцию
15 ноября 1968 г.