УДК 519.95

Л.В. ЛЮБОВИЦКИЙ

ОДИН ПРИБЛИЖЕННЫЙ АЛГОРИТМ ЦЕЛОЧИСЛЕННОГО ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

В предлагаемом алгоритме реализована одна из возможных модификаций идеи направленного перебора. Алгоритм осуществляет перебор граничных (в некотором смысле) точек, получающихся в пересечении многогранника условий с плоскостями, параллельными координатным плоскостям.

Введем необходимые для дальнейшего обозначения:

$\varphi(x) = \sum_{j=1}^{n} c_j x_j$, $R = \{x : \sum_{j=1}^{n} a_{ij} x_j \geq b_i\ , \ i = \overline{1,m}, \ x_j \geq 0\}$,

$R_1 = \{x : x \in R, \exists_i \ (\sum_{j=1}^{n} a_{ij} x_j = b_i)\}$ – граница многогранника условий.

Пусть Q – некоторое множество точек n -мерного евклидова пространства E_n.

$$\rho(x, Q) = \min_{y \in Q} \rho(x, y), \quad \text{где} \quad \rho(x, y) = \sum_{j=1}^{n} |x_j - y_j| .$$

$$M = \{x : x \in R, \ x_j - \text{ целое } , j = \overline{1,n}\},$$

$$M_1 = \{x : x \in M, \ \rho(x, R_1) < 1\},$$

$$x_j^+ = (x_1, \ldots, x_{j-1}, x_j + 1, x_{j+1}, \ldots, x_n),$$

$$x_j^- = (x_1, \ldots, x_{j-1}, x_j - 1, x_{j+1}, \ldots, x_n) .$$

$M_{j_1, j_2}(x)$ – множество точек y ,удовлетворяющих условиям:

1. $y_j = x_j$, $j \in \{1, \ldots, n\} \setminus \{j_1, j_2\}$, $y_{j_1} \neq x_{j_1}$, $y_{j_2} \neq x_{j_2}$, $j_1 \neq j_2$;

2. $y \in M_1$ или $y \in M \setminus M_1$, причем если $y \in M \setminus M_1$, то $y_{j_1}^{\pm} \in M_1$ или $y_{j_2}^{\pm} \in M_1$. Здесь $y_j^{\pm} = y_j^{+}$ или $y_j^{\pm} = y_j^{-}$.

Очевидно, что $M_{j_1, j_2}(x) = M_{j_2, j_1}(x)$ и $x \notin M_{j_1, j_2}(x)$.

$$\Delta_i(j_1,j_2,x)=\sum_{\substack{j=1\\ j\neq j_1,j_2}}^{n} a_{ij}x_j-b_i\,,\quad \mu_{j_1,j_2}=\max\left\{\left]\frac{\Delta_i(j_1,j_2,x)}{a_{ij_1}}\right[\right\},\ a_{ij_1}\neq 0.$$

Теперь задачу можно сформулировать в виде: найти $\min\limits_{x\in M}\varphi(x)$.

Опишем кратко алгоритм. Пусть известна точка $x^0\in M$.[x)] Вначале полагаем $\alpha:=x^0$, $\beta:=\varphi(x^0)$ (α, x – n-мерные векторы). Пусть уже сделано S шагов.

I) Ищем x^{s+1} такое, что $\varphi(x^{s+1})=\min\varphi(x), x\in M_{j_1,j_2}(x^s)$, $j_1,j_2=\overline{1,n}$;

2) если $\varphi(x^{s+1})<\beta$,то $\alpha:=x^{s+1}$, $\beta:=\varphi(x^{s+1})$[xx)] на 5;

3) если $\varphi(x^{s+1})>\beta$,то на 6 ;

4) если $x^{s+1}=\alpha$,то на 7 ;

5) печать x^{s+1}, $\varphi(x^{s+1})$;

6) $x^s:=x^{s+1}$ на I ,

7) конец.

Алгоритм реализован в виде α – схемы для случая $c_j>0$, $a_{ij}\geq 0$. Вводятся только ненулевые коэффициенты матрицы ограничений.

Замечание I. При вычислении x в п. I для пары (j_1,j_2) достаточно испытать x_{j_1} и x_{j_2} в интервалах $[0,\mu_{j_1,j_2}]$ и $[0,\mu_{j_2,j_1}]$ соответственно. Число различных перебираемых точек для каждой такой пары не превосходит $2(\mu_{j_1,j_2}+\mu_{j_2,j_1})$.

Замечание 2. Число операций, необходимое для выполнения одного шага алгоритма (п.I), не превосходит $4\mu m C_n^2$, где $\mu=\max\mu_{j_1,j_2}$, $j_1,j_2=\overline{1,n}$, $x\in M$.

Замечание 3. Алгоритм применим для приближенного решения задачи:

$$\min\varphi(x)\,,\quad x\in T\,,$$

где $\varphi(x)=\sum\limits_{j=1}^{n}\varphi_j(x_j)$, T – выпуклая область, φ_j – вогнутые функции, x_j – целые. В этом случае должны быть известны верхние границы для x_j, $j=\overline{1,n}$.

```
begin integer m,n,p;
        comment  m - число строк,  n-число столбцов, p -
    число ненулевых элементов матрицы ограничений;
    begin real array x,y,v0,vI,v2[I:n],v[0:m],
          b,d,dI,d2[I:m],a[I:p];
```

x) Для программы, реализующей предлагаемый алгоритм (см.ниже), должна быть задана исходная точка $x\in M$.

xx)Используется один символ АЛГОЛа – „:=“.

```
    integer array aI[I:p+I];
    integer i,j,jI,j2,s,t,q,l,r,kI,k2,kaI,ka2,
    kbI,kb2,tI,t2,t3,t4;
    real g,lI,f,f0,fI,f2,mI,m2;
    procedure PI (s,v,l); value s,l; integer s,l; array v;
    begin for r:=s+I step I until p do
        begin if l=0 then v[aI[r]]:=v[aI[r]]+a[r]
            else v[aI[r]] :=v[aI[r]]-a[r];
            if aI[r+I] ≥ I0000 then go to C2
        end; C2:
    end PI;
comment Прибавляем или вычитаем один столбец, т.е. изменяем
  некоторую компоненту плана на I;
    procedure P2 (j,s); value j; integer j,s;
    begin r:=0; for s:=I step I until p do
          if aI[s] ≥ 10000 then
          begin r:=r+I; if r=j then go to CI end; CI:
    end P2;
 comment   По номеру столбца  j  ищем номер его первой компо-
  ненты в  массивах  a  и  aI ;
    procedure P3(j,s,l); value j,s,l; integer j,s,l;
    begin q:=l; for i:=I step I until m do
          begin if aI[s+q]=i then
                begin g:=x[j]×a[s+q]; if l=0 then
                   begin d2[i]:=d2[i]+g; d[i]:=d[i]+g end
                   else
                   begin d2[i]:=d2[i]-g; d[i]:=d[i]-g end
                end go to C3;
                q:=q+I; if aI[s+q] ≥ I0000 then go to C4; C3;
          end; C4:
    end P3;
```

```
comment      Эта процедура используется при переборе точек,
принадлежащих     M_{j1,j2}(x);
    procedure P4 (x); array x;
    begin for r:=0 step I until m do v[r]:=0; q:=0;
    for j:=I step I until n do
        begin for r:=q step I until p do
            begin s:=if aI[r] ≥ I0000 then aI[r]-I0000
                else aI[r]; v[s]:=v[s]+a[r]×x[j];
                if aI[r+I] ≥ I0000 then go to C5
            end;
            C5:q:=r+I
        end;
        for r:=I step I until m do v[r]:=v[r]-b[r]
    end P4;
```

comment Эта процедура вычисляет $\sum_{i=1} a_{ij} x_j - b_i, \ i=\overline{1,m}$;

ввод (jI,b,x,a,aI);

comment b - вектор правых частей; x - исходная точка, принадлежащая M_1; a -коэффициенты функционала и матрицы ограничений по столбцам (только ненулевые); a_1 - номера строк соответствующих элементов массива a. Коэффициентам функционала соответствует 0 - й номер строки. К элементам массива a_1, соответствующим началам столбцов прибавляется I0000(метка начала столбца). В конце массива ставится I0000(метка конца массива);

```
    P4(x); PI(jI,s); lI:=0;
HI: if x[jI] < 0.5 then go to A;q:=I;for i:=I step I until m do
    begin if aI[s+q]=i then
        begin if v[i]-a[s+q] ≤ -I0 ↑ (-5) then go to A end
        else go to H3;q:=q+I;if aI[s+q]≥I0000 then go to H2;
H3: end;
H2: x[jI] :=x[jI] -I;PI(s,v,I);v[0]:=v[0]-a[s]; go to HI;
```

comment   Вспомогательная часть программы. Работает , пока не придем в точку $x^0 \in M_1$;

```
A: kaI:=ka2:=0; for j:=1 step I until n do v0[j]:=vI[j]:=x[j]
   r0:=v[0];
A0: fi:=I0↑I0; for i:=I step I until m do
   begin d[i]:=0; d2[i]:=0 end;
   for jI:=I step I until n do
   begin P2(jI,t3); P3(jI,t3,0);
      for j2:=jI+I step I until n do
      begin if jI=kaI ∨ j2=ka2 ∧ jI=ka2   j2=kaI
         then go to A6; mI:=m2:=0;
         P2(j2,t4); P3(j2,t4,0); q:=I;
         for i:=I step I until m do
         begin if aI[t3+q]=1 then
            g:=d2[i]/a[t3+q] else go to AI;
            if g < mI then mI:=g; q:=q+I;
            if a [t3+q]≥I0000 then goto A2;
AI: end
A2: mI:=entier(mI)+ 1;q:=I;
   for i:=I step I until I do
   begin if aI[t4+q]=1 then
      g:=d2[i]/a[t4+q] else go toA3;
      if g > m2 then m2:=g; q:=q+I;
      if aI[t4+q]≥ I0000 then go to A4;
A3: end;
A4: m2:=entier (m2)+I;
   f2:=I0↑I0; t:=0; tI:=t3; t2:=t4;
   kI:=jI; k2:=j2;
   for j:=I step I until n do v2[j]:=y[j]:=x[j];
BI: y[kI]:=x[kI] ; y[k2]:=x[k2];
   for i:=1 step I until m do di[i]:=d[i];
   f:=v[0]
  : if y[k2] < 0.5 then go to B2; y[k2]:=y[k2]-I;
```

```
    PI(t3,DI,I); f:=f-a[t2];
    for i:=I step I until m do if dI[i]≤ d2[i]-IO↑(-5)
    then
    begin PI(t2,dI,0); f:=f+a[t2];
         y[k2]:=y[k2]+I; go to B5
    end ;
    if f < f2 then
    begin f2:=f; for j:=I step I until n do
         d2[j]:=y[j]; go to B3
    end;
B4: y[kI]:=y[kI]+I; PI(tI,dI,0);
    f:=f+a[tI]; go to B3;
B5: if y[kI] < mI then go to B4;
B2: if t > 0 then go to A5; t:=I; g:=kI;
    kI:=k2; k2:=g; g:=tI; tI:=t2; t2:=g;
    mI:=m2; go to BI;
A5: if f2 < fI then
    begin fI:=f2;for j:=I step I until n do vI[j]:=v2[j];
         kbI:=kI; kb2:=k2
    end;
    P3(j2,t4,I);
A6: end;
    P3(jI,t3,I)
A7: end;
   comment  Часть АЛГОЛ-схемы между метками  A и  4, соот-
ветствует п. I краткого описания алгоритма;
    lI:=lI+I; kaI:=kbI; ka2:=kb2;
    if abs(fI-f2) < IO↑(-9) then go to O2;
    if fI > f2 then go to O3;
OI: fO:=fI; for j:=I step I until n do vO[j]:=vI[j];
O4:  ВЫВОД (lI,fI,vO);
```

```
03: F4(vI); for j:=I step I until n do x[j]:=vI[j];go to A0;
02: for j:=I step I until n do if abs(v0[j]-vI[j])>0.5 then
    begin if fI< f0 then go to 0I; go to 04 end;
    F4(v0); ВЫВОД (v0,v)
  end
end
```

Поступила в редакцию
I5 апреля I969 г.