

УДК 512.25 + 519.1

А.А. АНИКЕИЧ, А. Б. ГРИБОВ

ПРОГРАММА РЕШЕНИЯ ЗАДАЧИ ОТСЫКАНИЯ ОПТИМАЛЬНЫХ  
МАРШРУТОВ ПЕРЕВОЗОК

В последнее время в связи с проектированием автоматизированных систем управления отраслей и производственных комплексов усилился интерес к задачам оптимизации перевозочного процесса на автомобильном транспорте. Некоторым из этих задач посвящены работы [1 - 5].

В настоящей статье приводится программа решения задачи отыскания оптимальных маршрутов, которая является частным случаем более общей задачи, рассмотренной в [4]. Эта задача достаточно полно удовлетворяет современным условиям планирования и организации перевозочного процесса.

Данная статья является логическим продолжением работы [3]. По сравнению с работой [3] в ней учитываются новые моменты. Ставится дополнительное ограничение на длину маршрута с тем, чтобы маршруты укладывались в рабочую смену шофера (вместо выбора эталонной линии и представления остальных линий через нее).

Решается также вопрос о получении интенсивностей использования маршрутов в целых числах. Программа на ЭВМ написана на АЛГОЛ-60 с учетом изобразительных средств альфа-транслатора [8].

## § 1. Постановка задачи и метод решения

Задача маршрутизации перевозок формулируется так. Дано  $S$  линий перевозок,  $\Gamma$  автопарков. Известны объемы перевозок по линиям  $\beta_i$  ( $i=1, 2, \dots, S$ ) и мощности автопарков  $\beta_i$  ( $i=S+1, \dots, S+\Gamma$ ).

Известны все интересующие нас расстояния. Назовем маршрутом  $H = (h_0, h_1, \dots, h_k)$  упорядоченный набор из номера автопарка  $h_0$  и  $k$  номеров линий перевозок  $h_1, \dots, h_k$ . Положим

$$a_{in} = |\{h_\alpha \in H \mid h_\alpha = i\}|, \quad i=1, \dots, s+r,$$

где  $|A|$  — число элементов множества  $A$ . Таким образом,  $a_{in}$  — это кратность линии  $i$  в маршруте  $H$ .

На маршруте  $H$  осуществляются груженные пробеги  $\ell_n$ , погрузо-разгрузочные работы  $\rho_n$  и порожние пробеги  $c_n$  (все перечисленные величины выражаются в едином измерителе, например, километрах). Сумма трех определенных выше величин составляет длину маршрута.

Множество всех мыслимых маршрутов, для которых имеет место

$$\begin{aligned} \ell_n + \rho_n + c_n &\leq \mu \\ k &\leq \bar{k}, \end{aligned}$$

будем обозначать через  $\mathcal{H}$ . Здесь  $\mu$  — ограничение на длину маршрута, а  $\bar{k}$  — максимальная звенность маршрута.

Будем искать систему оптимальных маршрутов, обеспечивающую

$$\min \sum_{n \in \mathcal{H}} c_n x_n$$

и удовлетворяющую условиям:

$$x_n \geq 0, \quad (I)$$

$$\sum_n a_{in} x_n = \beta_i, \quad i=1, \dots, s, \quad (2)$$

$$\sum_n a_{in} x_n \leq \beta_i, \quad i=s+1, \dots, s+r. \quad (3)$$

Идеи метода решения этой задачи изложены в [3]. В результате решения задачи получаем систему оптимальных маршрутов и интенсивности их применения.

По своей физической сути интенсивность применения маршрута должна быть равна или кратна грузоподъемности машин, направляемых на маршрут. Нас интересует получение приближенного целочисленного решения для рассматриваемой задачи.

Для этого поступим следующим образом. Зафиксируем целые части значений переменных, полученных в оптимальном плане, а остатки постараемся распределить между маршрутами таким образом, чтобы приращение целевой функции за счет округления было минимальным.

Обозначим через  $d_i$  остаток объема перевозки по линии  $i$ , возникший при округлении интенсивностей применения маршрутов до их целых частей

$$d_i = \sum_n a_{in} \{x_n\}, \quad i = 1, \dots, S,$$

где  $\{x_n\}$  - дробная часть  $x_n$ . Освобождающиеся мощности автопарков (с учетом оставшегося резерва) обозначим через  $\rho_\alpha$  ( $\alpha = 1, \dots, r$ ).

Найдем такие  $y_n$ , равные 0 или 1, которые обеспечивают минимум

$$\sum_n c_n y_n$$

при условиях:

$$\sum_n a_{in} y_n \geq d_i, \quad i = 1, \dots, S,$$

$$\sum_{\{n: \beta_n = \alpha\}} y_n \leq \rho_\alpha, \quad \alpha = 1, \dots, r,$$

где  $\beta_n$  - номер автопарка, в котором начинается и кончается маршрут  $n$ .

Получилась задача целочисленного линейного программиро-

вания. Задача решается методом одностороннего перебора, идея которого предложил авторам И.В.Романовский.

## § 2. Программа решения задачи на АЛГОЛ-60

Прежде чем приступить к записи программы на АЛГОЛ-60 с использованием изобразительных средств Альфа-транслятора, объясним смысл встречающихся в ней обозначений и дадим краткую характеристику её работы. В программе:

- $S$  - число линий перевозок ;
- $Г$  - число автопарков ;
- $K$  - максимальная звенность маршрута ;
- $e$  - задаваемая точность счёта ;
- $mГ$  - большое число ;
- $b$  - барабанный массив для хранения обратной матрицы ;
- $c$  - барабанный массив для хранения матриц расстояний:  $(i)$  между линиями ;  $(ii)$  между автопарками и началами линий ;  $(iii)$  между концами линий и автопарками ;
- $bc$  - массив в МОЗУ, в который попеременно записываются обратная матрица и матрицы расстояний ;
- $d$  - вектор двойственных оценок ;
- $fu$  - значение функционала ;
- $q$  - матрица оптимальных переходов ;
- $bh$  - векторы маршрутов, входящих в очередной план ;
- $h$  - маршрут - "кандидат" для ввода в  $bh$  ;
- $co$  - вектор затрат на выезд из автопарков ;
- $rs$  - вектор,  $i$ -ый элемент которого является расстояние груженого прооега по линии  $i$  плюс "километрах" на погрузо-разгрузочные работы на обоих концах линии ;
- $t$  - ограничение на длину маршрута ;
- $Г, Г_j$  - для накопления длин маршрутов ;
- $x$  - вектор интенсивностей использования маршрутов ;
- $a$  - вектор, вводимый в базис ;
- $p$  - коэффициенты разложения вектора  $a$  по базису ;
- $g, f$  - рабочие массивы.

Характеристика других массивов и переменных или употребление уже известных в другой роли будет дана ниже при описании

блоков программы.

В программе имеется семь барабанных блоков:

$l_{100}, l_{10}, l_4, l_3, l_p, l_{p1}, l_{p2}$ .

Блок  $l_{100}$  перерабатывает часть входной информации в форму, пригодную для дальнейшего использования. В этом блоке формируются матрицы расстояний между линиями, между автопарками и началами линий, между концами линий и автопарками. В нем

$p$  - число пунктов погрузки,  $a$  - число пунктов разгрузки,  
 $pl[i]$  - номер пункта погрузки, в котором начинается линия  $i$ ,  
 $rl[i]$  - номер пункта разгрузки, в котором кончается линия  $i$ ;  
 $ap[i, j]$  - расстояние от автопарка  $i$  до пункта погрузки  $j$ ;

$ra[i, j]$  - расстояние от пункта разгрузки  $i$  до автопарка  $j$ ;

$rp[i, j]$  - расстояние от пункта разгрузки  $i$  до пункта погрузки  $j$ .

Заметим, что в тех случаях, когда число линий велико, придется отказаться от формирования матрицы расстояний между линиями. При этом нужно будет хранить матрицу расстояний между точками (пунктами погрузки и разгрузки) и получать необходимые расстояния между линиями непосредственно из нее.

Блок  $l_{10}$  формирует начальный план. Сюда входит: построение начальной системы маршрутов, её базиса, обратной матрицы, соответствующей этому базису. Попутно вычисляется значение функционала при построенной системе маршрутов. В конце работы блока обратная матрица записывается на МБ.

Блок  $l_4$  производит построение маршрута, подлежащего вводу в очередной план с учетом ограничения на его длину, используя схему динамического программирования.

В блоке  $l_3$  происходит построение вектора, соответствующего найденному маршруту, коэффициентов разложения последнего по базису и определение маршрута, подлежащего выводу из плана. Кроме того, здесь пересчитываются интенсивности использования маршрутов, двойственные оценки, обратная матрица и функционал, соответствующие новому плану. Этот блок реализует основную схему модифицированного симплекс-метода.

Блок  $l_p$  готовит исходную информацию для решения целочисленной задачи линейного программирования.

В массив  $bc$  теперь записывается матрица  $\|a_{in}\|$ , а мас -

симв  $\rho$  величины  $\alpha_i$  и  $\rho_\infty$  (см. постановку целочисленной задачи).

Блок  $lp1$  производит формирование и выдачу результатов в печать. В выдачу входят: число итераций, значение функционала, список маршрутов, соответствующие им интенсивности, двойственные оценки, длины, груженные и порожние пробеги по маршрутам, коэффициент использования пробега.

Блок  $lp2$  производит решение вспомогательной целочисленной задачи линейного программирования.

Результатом работы блока является выдача списка маршрутов, подлежащих округлению в большую сторону, и значения функционала в окончательном плане исходной задачи.

В этом блоке вектор  $plan$  представляет собой список номеров способов, входящих в текущий план, а вектор  $copy$  - список номеров способов, которые обеспечивают текущее "рекордное" значение функционала ( $\min$ ).

В программе происходит обращение к стандартной программе СП 0177, производящей запись на барабан (условное число 1037) и чтение с барабана (условное число 1032) соответствующих массивов.

```
begin integer m,n,r,s,k,11; real e,mr,fn;  
  array b,c [1:50,1:50]; drum b,c,1100;  
  ввод (r,s,k,e,mr); m:=r+s;  
1100: begin integer a,p,i,j; ввод (a,p);  
  begin integer array pl,rl [1:s];  
  array ap [1:r,1:p],ra [1:a,1:r],rp [1:a,1:p],bc [1:m,1:m];  
  ввод (pl,rl,rp,ap,ra);  
  for i:=1 step 1 until r do  
    begin for j:=1 step 1 until r do bc[i,j]:=0;  
      for j:=1 step 1 until s do  
        begin bc[i,r+j]:=ap[i,pl[j]];  
          bc[r+j,1]:=ra[rl[j],i]  
        end  
      end;  
    end;
```

```

    for i:=1 step 1 until s do for j:=1 step 1 until s do
        bc [r+i,r+j]:=rp[r1[i],p1[j]];
    CN 0177 (1037,bc[1,1],bc[m,m],0,c[1,1])
    end
end;

begin integer i,i0,u,v,v1,u1,j,j1,il;
    real m1,y,m2,t,t1; array x,bu,s,d,p[1:m],
    bc[1:m,1:m],co[1:r],rs[1:s];
    integer array q[1:s,2:k],bh[1:m,o:k],h[o:k];
    drum l10,l4,l3,lp,lp1,lp2;
l10: begin real r1;
    BBOA (x,rs,d,t,co,il);
    io:=1; fu:=ni:=0;
    for i:=1 step 1 until m do
        begin bh[i,o]:=0;
            for v:=1 step 1 until m do
                bc[i,v]:=0; bc[i,i]:=1
            end;
        for i:=s+1 step 1 until m do bh[i,o]:=s-i;
        for i:=1 step 1 until s do fu:=fu+x[i];
        fu:=fu*mr;
    CN 0177 (1037,bc[1,1],bc[m,m],o,b[1,1])
    end;
l1: i:=io;
l4: begin Boolean sigma; array g,f,ri,rj[1:s];
    CN 0177 (1032,bc[1,1],bc[m,m],o,c[1,1]);
    m1:=0; u:=1;
    for v:=1 step 1 until s do
        begin g[v]:=co[i]+d[s+i]-bc[i,v+r];
            rj[v]:=bc[i,v+r]+rs[v]

```

```

end;
12: for v:=1 step 1 until s do
    begin f[v]:=g[v]+d[v]; y:=f[v]-bc[v+r,i];
        if y>m1 then
            begin v1:=v;
                u1:=u;
                m1:=y;
                t1:=rj[v]+bc[v+r,i]
            end
        end;
    if u=k then go to 13; u:=u+1; sigma:=false;
    for v:=1 step 1 until s do
        begin m2:=-mr;
            for j:=1 step 1 until s do
                begin if rj[j]+bc[j+r,v+r]+rs[v]+bc[v+r,i]≤t
                    then
                        begin y:=f[j]-bc[j+r,v+r];
                            if y>m2 then begin m2:=y; j1:=j end
                        end
                    end
                end;
            if m2≠-mr then
                begin ri[v]:=rj[j1]+bc[j1+r,v+r]+rs[v];
                    sigma:=true; g[v]:=m2; q[v,u]:=j1
                end
            else begin ri[v]:=-mr;
                g[v]:=-mr
            end
        end;
    if sigma then begin for v:=1 step 1 until s do
        rj[v]:=ri[v]; go to 12 end

```

```

end;
13: begin real r3;
    if m1 > e then
        begin for v:=1 step 1 until m do a[v] := 0;
            h[u1] := -v1; a[s+1] := a[v1] := 1;
            for v:=u1 step -1 until 2 do
                begin v1 := h[v-1] := q[v1, v]; a[v1] := a[v1] + 1 end;
            h[0] := 1;
            i0 := if i < r then i+1 else 1;
            CH 0177 (1032, bc[1,1], bc[m,m], o, b[1,1]);
            for v:=1 step 1 until m do
                begin p[v] := 0;
                    for j:=1 step 1 until m do
                        p[v] := p[v] + bc[v,j] * a[j]
                    end
                end
            end
        else
            begin i := if i < r then i+1 else 1;
                if i ≠ i0 then go to 14
            else begin for v:=s+1 step 1 until m do
                    begin if d[v] > m1 then
                        begin m1 := d[v]; h[0] := s-v; v1 := v end
                    end;
                    if m1 > e then
                        begin CH 0177(1032, bc[1,1], bc[m,m], o, b[1,1]);
                            for v:=1 step 1 until m do
                                p[v] := bc[v, v1]
                            end
                        end
                    end
                end
            end;
        end;
    end;

```

```

if  $m1 \leq e$  then go to lp;
m2:=mr;
for i:=1 step 1 until m do
  begin if  $p[i] > 10^{-6}$  then
    begin  $y := x[i]/p[i];$ 
      if  $y < m2$  then begin  $m2 := y; u := i$  end
    end
  end;
if  $m2 = mr$  then begin БМВОД (h,p,m1,q); go to lp end;
 $y := 1/p[u];$ 
for v:=1 step 1 until m do
  begin  $bc[u,v] := bc[u,v] \times y; d[v] := d[v] - bc[u,v] \times m1$  end;
  for i:=1 step 1 until m do
    begin if  $i \neq u \wedge p[i] \neq 0$  then
      begin for v:=1 step 1 until m do
         $bc[i,v] := bc[i,v] - bc[u,v] \times p[i];$ 
         $x[i] := x[i] - p[i] \times m2;$ 
        if  $x[i] < 0$  then  $x[i] := 0$ 
      end
    end;
     $x[u] := m2;$ 
     $fu := fu - m2 \times m1;$ 
    ЦН 0:77 (1037, bc[1,1], bc[m,m], 0, b[1,1])
  end;
for v:=0 step 1 until k do  $bh[u,v] := h[v];$ 
   $bu[u] := t1;$ 
   $n := n + 1;$ 
go to l1;
lp: begin real r4; БМВОД (n,fu,bh,x,bu,d);  $y := 1;$ 
  for i:=1 step 1 until m do

```

```

begin for j:=1 step 1 until m do bc[i,j]:=0;p[i]:=0 end;
for i:=1 step 1 until m do
begin i1:=abs(bh[i,0]); bc[i,1]:=i1;
if i1=0 then begin BNB0X (i1); stop end;
if bh[i,0]<0 then begin p[i1]:=p[i1]+entier(x[i]);
y:=y+entier(x[i]) end;
j:=0;for j:=j+1 while x[i]>10-6 do x[i]:=x[i]-1;
t1:=if x[i] ≥ -ε then 0 else 1+x[i];
a[i]:=t1;y:=y+t1; p[i1]:=p[i1]+t1;
for u:=1 step 1 until k do
begin if bh[i,u-1]<0 then go to l1;
i1:=abs(bh[i,u]);
p[i1+r]:=p[i1+r]+t1;
bc[i,i1+1]:=bc[i,i1+1]+1
end;
l1: end;
CII 0177 (1037,bc[1,1],bc[m,m],0,b[1,1])
end;
lp1: begin real r5; m1:=0;
CII 0177(1032,bc[1,1],bc[m,m],0,c[1,1]);
for v:=1 step 1 until m do
begin if bh[v,0]>0 then
begin x[v]:=bc[bh[v,0],abs(bh[v,1])+r];
d[v]:=0;
for i:=1 step 1 until k do
begin if bh[v,i]<0 then
begin i1:=1; go to l end;
x[v]:=x[v]+bc[bh[v,i]+r,abs(bh[v,i+1])+r];
d[v]:=d[v]+bc[bh[v,i]+r,bh[v,i]+r]
end;
l: end;

```

```

1:  $x[v] := x[v] + bc[r-bh[v,1], bh[v,0]]$  ;
    $d[v] := d[v] + bc[r-bh[v,1], r-bh[v,1]]$  ;
    $m1 := m1 + x[v] \times s[v]$  ;
    $bu[v] := a[v] / (x[v] + d[v])$ 

   end

end;

BUBOX (x,d,bu,m1)

end;

12: begin integer array copy,plan[1:y];
    integer t,no,w,u1,u2; real min,sum;
    CII 0177(1032,bc[1,1],bc[m,m],o,b[1,1]);
    sum:=no:=0; v:=1; min:=mr;
    mk1: no:=no+1;

    if no  $\leq$  m then
        begin u1:=p[bc[no,1]] ; if u1=0 then go to mk1;
            v:=1;
            for i:=1 step 1 until s do
                begin u1:=bc[no,i+1]; u2:=p[r+1];
                    rs[i]:=u2-u1;
                    if u2 > 0  $\wedge$  u1 > 0 then v:=0
                end;
            if v=1 then go to mk1;
            p[bc[no,1]] := p[bc[no,1]] -1;
            plan[v]:=no;
            sum:=sum+x[no];
            for i:=1 step 1 until s do p[r+1]:=rs[i];
            t:=t+1;
            go to mk1
        end;

    end;

t:=t-1;

```

```

if t=0 then go to mk2;
for i:=r+1 step 1 until m do if p[i]>e then go to mk3;
if sum < min then
    begin min:=sum; w:=t; вывод (min,w,plan);
        for i:=1 step 1 until y do copy[i] :=plan[i]
    end;
mk3: no:=plan[t]; sum:=sum-x[no];
    p[bc[no,1]] :=p[bc[no,1]] +1;
    for i:=1 step 1 until s do p[r+i] :=p[r+i]+bc[no,
                                                i+1];

    go to mk1;
mk2: min:=fu-m+min;
    вывод (copy,min,w);
    stop
end
end
end

```

Программа отлажена на "М-20." Она позволяет решать задачи маршрутизации перевозок, размерность которых удовлетворяет условию:

$$S^2 + 10S + 2SK + 2SR + rK + 7r + K \leq 2800.$$

### § 3. Пример использования программы (перевозка мебели в Ленинграде)

Рассмотрим использование программы на примере перевозки промышленных (дискретных) грузов, ибо именно в этом случае остро стоит вопрос о получении решения в целых числах.

Ленинград снабжается мебелью 25 предприятиями Ленмебельдревпрома. Перевозки осуществляет грузовой автопарк № 67 Главнавтотранса. Потребителями являются 62 магазина, расположенные в Ленинграде и некоторых городах Ленинградской области.

Ниже будет показано решение задачи маршрутизации перевозок на один из дней. В этот день рассматривалось 40 линий перевозок. Расстояния перевозок между поставщиками и потребителями, между автопарком и поставщиками, а также между потребителями и автопарком приводится в табл. 1.

Линии перевозок (поставщик-потребитель) отмечены в табл. I звездочками. Будем нумеровать их по строке таблицы слева направо. Например, линия перевозки № 29 означает, что нужно осуществить перевозку груза от поставщика № 7 потребителю № 28.

Объем перевозок по линиям задается в виде числа груженых ездов мебелевоза, которое нужно сделать от поставщика по данной линии до потребителя. Одна груженная езда - это перевозка 120 стульев, или 10-12 диванов, или 5-6 шкафов, или 2-3 гарнитуров. Ежедневно автомобильным парком совершается до 250 рейсов. В нашей задаче число груженых ездов по линиям в порядке их нумерации запланировано в количестве:

3, 1, 2, 3, 4, 1, 3, 3, 3, 1, 3, 1, 3, 4, 2, 1, 1, 1, 4, 2, 1, 1, 4, 3, 1, 2, 2, 1, 1, 1, 2, 3, 1, 1, 2, 2, 5, 4, 3, 3.

Задача решалась при следующих ограничениях. Требовалось, чтобы длина маршрута (с учетом погрузо-разгрузочных работ) не превышала 150 км, а число линий в маршруте было не более 5. "Километраж", затрачиваемый на погрузо-разгрузочные работы на линиях, начинающихся у поставщиков в порядке их нумерации, принимался равным 12, 15, 13, 15, 12, 15, 18, 16, 12, 13, 12 км.

После 90 итераций симплекс-метода, потребовавших 10 мин. времени "М-20", был осуществлен переход на блок решения целочисленной задачи линейного программирования. Получена за 30 мин. следующая система маршрутов (см. табл. 2).

Приведенные в табл. 2 маршруты несколько изменены по сравнению с машинным решением. Это связано с тем, что при решении целочисленной задачи не гарантируется точного выполнения задания по линиям перевозок. По некоторым линиям задание может быть перевыполнено. Поэтому из состава некоторых маршрутов исключены линии, по которым есть "перебор". Эти маршруты помечены звездочкой.

#### Заключение

По приведенной выше программе "М-20" могут решаться средние по размеру задачи. Использование машины "ЕСМ-6" позволило бы решать любые практически встречающиеся задачи маршрутизации перевозок грузов в городах.

Расстояния между объектами, в км

Таблица I

Потребители	Авто-парк	Поставщики										
		I	2	3	4	5	6	7	8	9	10	11
Авто-парк		11	3	5	9	3	8	8	4	9	9	13
1	11	21*	16*	15	15	13	10	10	10	10	10	9
2	3	9*	6*	5	6	3	7	7	4	9	9	12
3	4	11	8	7	8	5	5	5	2	7	7	11
4	7	6	7*	9	3	6	10	10	7	11	11	14
5	4	11	8*	7*	8	5	5	5	2	7	7	11
6	4	13	10	9*	10*	6	6	6	3	6	6	10
7	2	11	5	5	9*	3	8	8	4	9	9	13
8	3	9*	6*	5	6	3	7	7	4	9	9	12
9	7	11	11	11	6	6	3	3	4	6	6	9
10	11	20	12	9	20	12	15	15	12	17	17	17
11	11	24	15	12	20	14	15	15	13	18	18	16
12	10	15	14	14	11	11	4	4	8	3	3	5
13	13	17	17	17	14*	11	6	6	11	4	4	2
14	9	14	13	12	10*	10	3	3	7	2	2	4
15	8	17	12	12	13*	10	7	7	7	8	8	11
16	13	17	17	17	14	11*	6	6	11	4	4	2
17	6	6*	8	9	4	7*	8	8	7	12	12	15
18	9	5	9	11	2	9*	8	8	8	10	10	14
19	9	5	9	11	2	9*	8	8	8	10	10	14
20	14	10*	15*	17	7	15*	8	8	10	10	10	14
21	10	7	11	13	4	10*	6	6	6	7	7	10
22	10	7	11	13	4	10*	6*	6	6	7	7	10
23	9	9	6	8	11	8	13*	13	10	14	14	18
24	5	11	2	4	9	4	12	12	8	13	13	17
25	12	16	17	16	13	11	5	5	11	4	4	3
26	11	2	11	13	5	10	12	12	11	14	14	17
27	8	11	12	12	7	8	2*	2	5	5	5	7
28	8	7	8	10	3	8	8	8*	6	9	9	12
29	17	10	16	18	11	16	19	19*	17	20	20	25
30	6	16*	9	9	14	8	9	9	6	11	11	15
31	5	13	4	2	11	3	12	12*	7*	12*	12	17
32	13	6	13	14	8	13	15	15	13	17*	17	21
33	5	11	2	4	9	4	12	12	8	13*	13*	17
34	10	20	13	10	18	12	12	12	11	16	16*	15
35	11	3	11	13	5	10	12	12	11	14	14	17*
36	9	3	9	10	5	9	12	12	10	14	14	18*

Характеристика оптимальных маршрутов перевозок

Таблица 2.

№ маршрута	Порядок объезда линий	Количество машин на маршрут	Грузовые пробеги, км	Порожние пробеги, км	Длина маршрута, км	Коэффициент использования пробега
1	A24-14-39-14-36	1	60	23	149	0,72
2	A-4-19-14-37-4	1	44	15	129	0,75
3	A-5-15-37-5-11	1	49	25	144	0,66
4*	A-5-15-37-12-11	1	48	28	146	0,63
5	A-24-32-7-32-7	1	48	22	144	0,69
6	A-19-14-28-32-7	1	40	16	129	0,71
7	A-16-39-11-19-9	1	56	17	136	0,77
8	A-23-35-21-38	1	58	39	146	0,60
9	A-19-13-36-5-9	1	53	18	140	0,74
10*	A-20-13-40-17-9	1	57	26	149	0,68
11	A-1-30-1	1	61	42	145	0,59
12*	A-22-35-6-10	1	49	36	142	0,57
13	A-23-27-23-27	1	56	36	146	0,61
14.	A-4-20-13-39-3	1	55	16	137	0,77
15.	A-23-1-40-31	1	70	31	149	0,69
16.*	A-24-33-8-34-8	1	47	30	136	0,61
17.*	A-2-40-8-18	1	50	36	125	0,58
18.*	A-8-26-37-25-37	1	51	34	151	0,60
19.	A-38-38-38	1	48	51	138	0,48
20.*	A-26-29-31	1	30	27	102	0,53

Имеется и другая перспектива работы. Она заключается в том, что вместо хранимой в памяти ЭВМ обратной базисной матрицы можно использовать её алгоритмичное восстановление, как это делается, например, в [6], или получать непосредственное решение системы линейных алгебраических уравнений, как, например, это предложено в [7]. Экономия памяти в этих случаях позволит решать задачи маршрутизации перевозок на машинах средней мощности в полном объеме.

#### Л и т е р а т у р а

1. И.Н.Бернштейн, С.А.Панов. Алгоритм решения задачи маршрутизации - Экономика и математические методы, № 5, 1966.
2. К.В.Ким. Задача отыскания оптимальных маршрутов - Экономика и математические методы, № 5, 1965.
3. А.А.Аникеич, А.Б.Грибов, И.В.Романовский. Об одной задаче маршрутизации перевозок - Кибернетика, № I, 1966.
4. Сб. "Применение математики в экономике", вып.4, Л., изд. ЛГУ, 1967.
5. Сб. "Постановка и решение задач на ЭВМ в области автомобильного транспорта", М., 1966.
6. А.Б.Грибов. Метод рекуррентного пересчета характеристического многочлена - ДАН СССР, т.177, № 3, 1968.
7. С.С.Сурин. Решение задач линейного программирования с большим числом нулевых элементов в исходной симплексной таблице. Сб. "Оптимальное планирование", вып.10, Новосибирск, 1968.
8. Г.И.Бабецкий, М.М.Безанова, Ю.М.Волошин, А.П.Ершов, Б.А.Загацкий, Л.Д.Зимевская, Г.И.Кожухин, С.К.Кожухина, Ю.И.Михалевич, Р.Д.Мишкович, И.В.Поттосин, Л.К.Трохан. АЛФА- система автоматизации программирования, Новосибирск, 1967.

Поступила в редакцию  
15.XI.1967 г.