STUDIA UNIVERSITATIS MOLDAVIAE, 2019, nr.2(122)
Seria "Ştiinţe exacte şi economice" ISSN 1857-2073 ISSN online 2345-1033 p.64-68

CZU: 519.863

O ABORDARE PRIVIND REZOLVAREA PROBLEMEI COMIS-VOIAJORULUI CU AJUTORUL METODEI POTENŢIALELOR

Dmitri TERZI

Universitatea de Stat din Moldova

Solutia aproximativă a problemei comis-voiajorului se realizează prin modificarea metodei potențialelor de rezolvare a problemei de transport. O condiție necesară se utilizează pentru a verifica optimalitatea soluţiei. În procesul de rezolvare, matricea costurilor este convertită pentru a elimina călătoriile nedorite de la un punct la altul.

***Cuvinte-cheie**: problema comis-voiajorului, metoda potenţialelor, condiţie necesară de optimalitate a soluţiei, transformrea matricei costurilor.*

SOLUTION APPROACH TRAVELING SALESMAN PROBLEM USING THE METHOD OF POTENTIALS

Approximate solution of the traveling salesman problem is carried out by modifying the potential method to solve the transport problem and using the necessary optimality condition. In the process solutions, the cost matrix is converted to eliminate unwanted journeys from one point to another.

***Keywords:** traveling salesman problem, potential method, necessary solution optimality condition, transformation of cost matrix.*

Introducere

Atunci când se utilizează metode cantitative pentru rezolvarea problemelor economice, este necesar să se țină cont de limitele pe care le posedă. Sarcinile de transport sunt astfel de clase, precum: problema selecţie, problema fluxului maximal, problema cu cea mai scurtă cale, distribuirea de produse între întreprinderi, distribuția suprafeței cultivate, optimizarea aprovizionării cu materii prime și materiale pentru întreprinderile producătoare, optimizarea livrărilor de mărfuri de la depozite la magazine de vânzare cu amănuntul, optimizarea traficului de pasageri. Acest lucru permite dezvoltarea unor metode mai eficiente de soluționare a acestora comparativ cu metodele generale [1].

Pentru problema transportului s-au dezvoltat algoritmi de soluții eficiente. Aici, teoria soluționării problemei de transport este utilizată în esență pentru a rezolva problema comis-voiajorului (PCV), luând în considerare limitările sale. În problemă, comis-voiajorul trebuie să viziteze o singură dată fiecare dintre punctele de vizitare și să se întoarcă la punctul de plecare, reducând la minimum costul total al ocolului [2-4].

Adaptăm tehnologia pentru rezolvarea problemei de transport pentru a determina optimalitatea unei soluții admisibile la problema comis-voiajorului.

Datele inițiale ale PCV sunt reprezentate de matricea distanţelor (costurilor)

$$d = \begin{pmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \dots & \dots & \dots. & \dots \\ d_{n1} & d_{n2} & \dots & d_{nn} \end{pmatrix}$$

Trecând de la punctul i în punctul j, comis-voiajorul suferă pierdere d_{ij}. Costul călătoriei prin punctele $i_1, i_2, ..., i_n$ este egal cu

$$f = \sum_{k=1}^{n-1} d_{i_k j_{k+1}} + d_{i_n i_1}.$$

Introducem datele iniţiale ale problemei comis-voiajorului, prezentate (în terminologia problemei de transport) ca vector al stocurilor furnizorilor $a = (a_1, a_{2,}, ..., a_n)$, vectorul de cerințe ale consumatorului $b = (\text{b}_1, \text{b}_{2,}, ..., \text{b}_n)$, unde

$$a_i = 1+\varepsilon,\ i=\overline{1,n},$$
$$b_i = 1,\ i=\overline{1,n-1},\ b_n = 1+n\varepsilon,$$

ε – număr destul de mic.

În sarcină, după cum se poate observa, stocurile totale ale furnizorilor sunt egale cu cerințele totale ale consumatorilor.

Problema comis-voiajorului se rezolvă în felul următor:

1) Formarea unei soluții ciclice valabile. În acest scop se folosesc diferite modalități de generare a unor soluții admisibile.Vom identifica fiecare variabilă x_{ij} cu ruta *(i,j)*. A găsi o soluţie de bază nedegenerată este echivalent cu a găsi cel mult *2n–1* rute (din cele n^2 posibile). Rutele vor fi organizate într-un tabel asemănător celui în care sunt organizate datele problemei, fiecărei rute corespunzându-i o căsuţă *(i,j)*.

Pentru a găsi unele soluţii iniţiale de bază există o multitudine de metode (metoda nord–vest; metoda minimului pe linii; metoda minimului pe coloane; metoda costului minim; metoda diferenţelor maxime; metoda Vogel). Unele metode încearcă nu doar găsirea planului (soluțiilor), ci chiar găsirea unei metode mai bune. În acest plan, vom expune metoda Vogel, adaptată pentru a rezolva problema comis-voiajorului: a) pentru fiecare linie şi fiecare coloană se calculează diferenţa dintre cele mai mici două costuri ale rutelor acesteia (diferenţa poate fi şi 0 dacă minimul este multiplu) şi se găseşte maximul dintre aceste diferenţe; b) dintre toate rutele de pe liniile şi coloanele corespunzătoare acestui maxim se alege ruta de cost minim (dacă minimul este multiplu, se ia una la întâmplare). Se transportă pe această rută maximul posibil. Acest maxim este egal cu minimul dintre cantitatea care mai e disponibilă la furnizorul corespunzător acestei rute şi cantitatea care mai e necesară la consumatorul corespunzător rutei, în momentul alegerii acestei rute. Se transportă în acest fel pentru ca să se folosească cât mai puţine rute şi deci să se obţină o soluţie de bază. Adaptarea metodei Vogel înseamnă că în fiecare etapă a formării unei soluții admisibile $\{x_{ij}\}$ se ia în considerare cerința ciclicității sale.

O altă abordare a formării unei soluții fezabile $\{x_{ij}\}$ este posibilă ca în tabelul următor

M				1+ε
1	M			ε
	1	M		ε
		1-ε	M	2ε
		ε	1	M

unde M – un număr destul de mare, ε – un număr destul de mic, M>0, ε>0.

2) Determinăm potențialele u și v.

3) Mai mult, vom calcula

$$s_{ij} = u_i + v_j - d_{ij}$$

pentru toate valorile zero x_{ij}.

4) Construim următoarea soluție fezabilă. Sunt luate în considerare două cazuri: a) printre toate estimările s_{ij} nu există niciunul pozitiv; b) unele estimări s_{ij} sunt pozitive.

În cazul a) planul este optim. În conformitate cu planul optim (pentru problema de transport construită la ultimul pas) se formează o permutare

$$(\text{xv}, \text{yv}) = \begin{pmatrix} i_1\, i_2 \ldots i_n \\ 1\ 2 \ldots n \end{pmatrix}$$

unde i_k (numărul *k* al părții superioare xv a permutației *(xv,yv))* înseamnă numărul liniei în matricea $\{x_{ij}\}$în care elementul $x_{i_k j}$ este aproape de unul. Dacă permutarea (*xv,yv*) este ciclică, atunci vom obține o soluție admisibilă la problema comis-voiajorului. Dacă permutarea (*xv,yv*) nu este ciclică, atunci se descompune în cicluri și pentru fiecare ciclu de tip $j_1\, j_2 \ldots j_s$ (cu lungimea s) matricea costurilor *d* se corectează, și anume:

elementul $d_{j_s j_1}$ este înlocuit cu un număr suficient de mare M pentru a interzice trecerea de la punctul j_s la punctul j_1.

În cazul b), pentru fiecare evaluare pozitivă s_{ij} corespunzătoare elementului zero al x_{ij} se construiește un ciclu închis (de regulile generale): pornind din această rută, trecând doar prin rutele soluţiei, mergând doar pe verticală sau orizontală şi fiecare trecere de la o rută la alta făcându-se doar perpendicular pe trecerea anterioară. Există un singur circuit cu aceste proprietăţi care trece printr-un număr par de rute. Începând cu "+" din ruta care va intra în bază se notează alternativ cu "+" şi "–" rutele circuitului. Se notează cu θ minimul dintre cantităţile transportate pe rutele notate cu "–" şi ruta pentru care s-a obţinut acest minim este cea care va ieşi din bază. Se scade θ din cantităţile transportate pe rutele notate cu "–" şi se adaugă la cele notate cu "+", rutele care nu sunt pe circuit păstrându-şi valoarea.

O descriere mai detaliată pentru a forma un ciclu închis corespunzător elementului x_{ij} zero este prezentată de următorii operatori (Visual C++).

```
  /*   Program de formare a ciclului pentru celula neocupată   */
#include <stdio.h>

void plan(), cikl(), vivod();
int m,n,i,j,it,k,r,l,r1,k1,i1,j1,i2,j2,i3,j3;
int iv[20],il[20], mu[20], mv[20], ci[20], cj[20];
int xv[20], yv[20], c[20][20];
double f,fopt, s, t, teta, h;
double x[20][20], u[20], v[20];

FILE *ft;

void main()
{
     ft=fopen("ZKcikli.dat","w");
        /*  Construirea unui plan admisibil de dimensiune n */
    x[1][n]=1.01;
    for(i=1; i<=n-3; i++)
       { x[i+1][i]=1.0; x[i+1][n]=0.01; }
    x[n-1][n-2]=0.99; x[n-1][n]=0.02;
     x[n][n-2]=0.01; x[n][n-1]=1.0;

    for(i=1; i<=n; i++)
        { fprintf(ft,"\n");
          for(j=1; j<=n; j++) fprintf(ft,"%8.2f", x[i][j]);
        }

   for(r=1; r<=n; r++)
       for(k=1; k<+n; k++)
      if ((x[r][k]<0.001) && (r!=k))
         {   /*   cikl(r,k) - determinarea unui ciclu pentu celula (r,k)  */
            for(i=1; i<=n; i++) il[i]=0;
            for(j=1; j<=n; j++) iv[j]=0;
            for(i=1; i<=n; i++)
            for(j=1; j<=n; j++)
            if (x[i][j]>0) {  il[i]=il[i]+1;iv[j]=iv[j]+1; }

            il[r]=il[r]+1; iv[k]=iv[k]+1;
              /*    Căutarea lanțului    */
            i3=0;
```

```
      m1:
        for(j=1; j<=n; j++)
         if(iv[j]==1)
        {  iv[j]=0;
           for(i=1; i<=n; i++)
            if(x[i][j]>0 && il[i]>0) il[i]=il[i]-1;
        }

      for(i=1; i<=n; i++)
       if(il[i]==1)
       {  il[i]=0;
          for(j=1; j<=n; j++)
           if(x[i][j]>0 && iv[j]>0) iv[j]=iv[j]-1;
       }

      i1=0; for(i=1; i<=n; i++) if(il[i]==0 || il[i]==2) i1++;
      j1=0; for(j=1; j<=n; j++) if(iv[j]==0 || iv[j]==2) j1++;
      if(i1!=n || j1!=n) { i3++; goto m1; }
      l=1; ci[l]=r; cj[l]=k; i1=r; j1=k;

      i3=0;
      m2:
       for (i=1; i<=n; i++)
       if(i!=i1 && x[i][j1]>0 && il[i]>0)
          {  i1=i; l=l+1; ci[l]=i1; cj[l]=j1; break; }

       for (j=1; j<=n; j++)
       if(j!=j1 && x[i1][j]>0 && iv[j]>0)
         {  j1=j; l=l+1; ci[l]=i1; cj[l]=j1; break; }
        if(i1!=r)
          { i3++; goto m2; }

       teta=x[ci[2]][cj[2]];
       for(i=2; i<=l; i=i+2)
        {   i2=ci[i]; j2=cj[i];
            if(teta>x[i2][j2]) { teta =x[i2][j2]; i3=i2; j3=j2; }
        }

       t=1;
      for(i=1; i<=l; i++)
         {  i2=ci[i]; j2=cj[i];
            x[i2][j2]=x[i2][j2]+teta*t;
           t=-t;
          }

    fprintf(ft, "\n Pri r=%d k=%d l=%d  ",r,k,l);
       for(m=1; m<=l; m++)
                 fprintf(ft,"(%d,%d) ", ci[m], cj[m]);
  }
 fprintf(ft, "\n ");
 fclose(ft);
}
```

Apoi, construim următoarea soluție admisibilă. Calculul continuă conform algoritmului de problemă de transport. Studiul posibilității formării unei aproximări acceptabile ciclice la fiecare s_{ij} pozitiv (pentru zero x_{ij}) este o etapă preliminară specifică a abordării considerate a soluționării problemei comis-voiajorului.

Soluția, reprezentată de traseul m, este verificată printr-un **criteriu necesar al optimalității**:

$$d(k,l)+d(i,j) <= \mathrm{d(k,j)}+\mathrm{d(i,l)},$$

unde tranzițiile $(\mathrm{k,l})$ și (i,j) corespund oricăror două tranziții din traseul m.

5) Din soluțiile ciclice construite admisibile se alege cea mai bună soluție.

Experiment. Au fost efectuate experimente pentru o serie de probleme de test. S-au obținut soluții admisibile care au fost comparate cu cele găsite folosind programele prezetate în [5]. Soluțiile au fost verificate prin criterul necesar al optimalității și au fost formulate concluzii.

Concluzii

Abordarea propusă aici pentru rezolvarea PCV este promițătoare pentru probleme cu diferite dimensiuni şi poate servi drept bază pentru dezvoltarea modificărilor sale și a cercetării experimentale pentru a rezolva în mod eficient problemele discrete de optimizare în economia matematică.

Referinţe:

1. ГОЛЬШТЕЙН, Е.Г., ЮДИН, Д.Б. *Задачи линейного программирования транспортного типа*. Москва: Science, 1969.
2. REINGOLD, E.M., NIEVERGELT, J., DEO, N. *Combinatorial algorithms. Theory and practice.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1977.
3. PAPADIMITRIOU, Ch.H., STEIGLITZ, K. *Combionatorial optimization: Algorithms and Complexity*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982.
4. *Traveling Salesman Problem, Theory and Applications*. Edited by Donald Davendra, 2010.
5. TERZI, D. Realizarea metodei Branch and Bound pentru rezolvarea problemelor de programare discretă. În: *Materialele Conferinţei ştiinţifice internationale jubiliare „Paradigme moderne în dezvoltarea economiei naţionale şi mondiale*". Chişinău, 2018.

Date despre autor:
Dmitri TERZI, doctor în ştiinţe matematice, conferenţiar universitar, Facultatea de Ştiinţe Economice, Universitatea de Stat din Moldova.
E-mail: dgerterzi@yandex.ru
ORCID: 0000-0002-1518-4012

Prezentat la 16.05.2019