Principii privind algoritmii genetici pentru rezolvarea unei probleme tridimensionale de transport

Lect. Dorina MOANȚĂ
Catedra de Matematică, A.S.E. București

Se definește problema tridimensională de transport, modelul cu sume duble [MIH-70] pentru care există algoritmi de optimizare [COR-72],[MOA-76]. Cu caracter de noutate propunem pentru această problemă principii pentru algoritmi genetici. Algoritmi de căutare pentru o problemă deja rezolvată cu algoritmi de optimizare, algoritmii genetici constituie o abordare modernă prin prisma inteligenței artificiale și, deși nu asigură obținerea soluțiilor optime, conduc la timpi de calcul rezonabili.

Cuvinte cheie: *operatori genetici, cromozomi, funcție de evaluare, inversiune, mutație, încrucișare.*

1. Problema tridimensională de transport

Practica economică a impus rezolvarea unei probleme de genul: în ***m*** centre de producție (depozite) se fabrică (se găsește) un anumit produs ce se poate transporta cu ***p*** mijloace de transport în ***n*** centre de consum (destinații) care-l solicită; dacă se cunoaște fiecare centru de producție i (i = 1, ..., m), cantitatea disponibilă a_i , în fiecare centru de consum j (j = 1, ..., n), cantitatea necesară b_j, pentru fiecare mijloc de transport k (k = 1, ...,p) cantitatea ce poate fi transportată c_k și costul transportului unității din acel produs de la producătorul i la destinatarul j cu mijlocul de transport k, se cere o distribuție a cantităților care urmează a fi transportate cu cheltuieli totale minime, astfel încât cererea să fie satisfăcută. Formularea matematică a problemei enunțate dă modelul problemei de programare tridimensională numită "de transport", care, conform clasificării date în [MIH-70], este o problemă triplanară de transport cu trei categorii de restricții exprimate prin sume duble:

$$\min \sum_{i\in M}\sum_{j\in N}\sum_{k\in P} c_{ijk}x_{ijk} = \min L(X) \qquad (1.1)$$

$$\sum_{j\in N}\sum_{k\in P} x_{ijk} = a_i \quad i \in M \qquad (1.2)$$

$$\sum_{i\in M}\sum_{k\in P} x_{ijk} = b_j \quad j \in N \qquad (1.3)$$

$$(\mathrm{I}) \qquad \sum_{i\in M}\sum_{j\in N} x_{ijk} = c_k \quad k \in P \qquad (1.4)$$

$$x_{ijk} \geq 0, i\in M, j\in N, k\in P \qquad (1.5)$$

$$a_i > 0,\ b_j > 0,\ c_k \geq 0,\ c_{ijk} \geq 0 \qquad (1.6)$$

$$\sum_{i\in M} a_i = \sum_{j\in N} b_j = \sum_{k\in P} c_k = T \qquad (1.7)$$

unde $M = (1 \ldots m)$, $N = (1 \ldots n)$, $P = (1 \ldots p)$ și $I = MxNxP$

Se cunosc următoarele date: **m** - numărul centrelor de producție; **n** - numărul centrelor de consum; **p** - numărul mijloacelor de transport; a_i - cantitatea de produs livrată de centrul de producție **i**; b_j - necesarul din acel produs în centrul de consum **j**; c_k - cantitatea care poate fi transportată de mijlocul de transport **k**; c_{ijk} - costul transportului unității de produs de la sursa **i** la destinația **j** cu mijlocul de transport **k**.
Se determină: x_{ijk} - cantitatea din produsul considerat ce urmează să fie transportată de

la sursa **i** la destinaţia **j** cu mijlocul de transport **k**.
După interpretarea care se dă elementelor cunoscute, modelul poate avea multiple aplicaţii: de exemplu, dacă se înlocuiesc cantităţile cu număr transporturi, costurile cu timpuri, atunci soluţia optimă va da minimul de autovehicule-oră.
Modelul matematic al acestei probleme este adaptabil unor probleme de distribuţie, afectare de personal, probleme de alegere. Condiţia (1.7) de echilibru neîndeplinită cere introducerea unei variabile fictive care să o echilibreze, ca în [COR-72]. Numim matricea costurilor $\mathbf{C} = \{(c_{ijk})/(i,j,k) \in \mathbf{I} \}$. Matricea $\mathbf{X} = \{(x_{ijk}) / (i,j,k) \in \mathbf{I} \}$ care verifică (1.2) - (1.6) se numeşte soluţie admisibilă sau plan de transport. Dacă realizează şi (1.1 se numeşte plan optim de transport .
Exemplu: Considerăm problema de transport cu 4 depozite, de unde se solicită transportul unui singur produs cu 3 mijloace de transport la 4 centre de consum ($m = 4$, $n = 4$, $p = 3$). Se dau vectorii a, b, c: disponibil, necesar, capacitate vehicul. Cunoaştem cantităţile disponibile în depozite, capacitatea fiecărui mijloc de transport şi în ce cantităţi este cerut.
$a_1 = 24$; $a_2 = 8$; $a_3 = 18$; $a_4 = 10$.
$b_1 = 11$; $b_2 = 19$; $b_3 = 21$; $b_4 = 9$.
$c_1 = 17$; $c_2 = 31$; $c_3 = 12$. Matricea costurilor (C) este:

k=1

i\j

15	9	10	8
19	15	20	6
14	24	17	11
24	14	13	9

k=2

i\j

18	14	4	10
20	21	24	19
11	20	16	13
28	19	21	10

k=3

i\j

7	11	12	13
17	14	18	23
8	13	15	18
31	21	14	20

Metodele tradiţionale şi algoritmii corespunzători eşuează în rezolvarea completă a problemelor de mare complexitate.

2. Algoritmi genetici

Metodele genetice sunt metode de căutare a optimului în două direcţii: explorarea (căutarea) soluţiei mai bune şi exploatarea mai bună a spaţiului soluţiilor.
Algoritmii genetici clasici operează cu şiruri binare şi cer modificarea problemei, ce nu este deloc simplă.
Parametrii problemei sunt codificaţi prin algoritmul genetic, într-un şir de caracteristici (de obicei binare), analoage cu *cromozomii* biologici. În cazul uzual al mai multor parametri şirul conţine mai multe subşiruri numite *gene*, câte unul pentru fiecare parametru. Fiecare cromozom reprezintă o posibilă soluţie a problemei.
Un algoritm genetic efectuează operaţii specifice în cadrul unui ***proces de reproducere*** guvernat de către ***operatorii genetici***. Noile soluţii sunt create prin selecţia şi recombinarea cromozomilor existenţi, în vederea optimizării unei ***funcţii de evaluare*** (sau de performanţă) aleasă pentru fiecare problemă în parte.
Plecând de la o populaţie de cromozomi generată aleator, fiecare nouă populaţie (generată prin reproducere) înlocuieşte gene-raţia anterioară.
Indiferent de formularea problemei, *principiul comun* al algoritmilor genetici este: ***o populaţie de indivizi este supusă la transformări şi în timpul acestui proces indivizii luptă pentru supravieţuire.***
Funcţia de evaluare globală se îndreaptă spre optim şi oferă soluţii din ce în ce mai bune problemei originale.
Indiferent de formularea problemei, structura algoritmilor genetici este:
Iniţial: *iniţializarea* populaţiei de cromozomi - soluţii posibile care *verifică restricţiile.*
Iterativ: 1) *evaluarea* cromozomilor din populaţia curentă - *evaluarea funcţiei obiectiv* pentru aceste soluţii; 2) *selectarea* acelora pentru care valoarea funcţiei obiectiv este mai "bună"; 3) *recombinarea* cro-

mozomilor selectați folosind *operatori genetici* (se creează astfel o nouă generație).
Stop: timpul de căutare s-a terminat.

3. Principii privind algoritmii genetici pentru o problemă de transport

Algoritm GENETIC-1.
Algoritmul genetic ***clasic*** operează cu cromozomi, cu soluții care sunt șiruri de biți, (de valoare 0 și 1). A defini un vector bit pentru o soluție înseamnă a crea un vector $(v_1, v_2, \dots v_r)$, $r = m*n*p$, a cărui componentă v_q, $(q = 1,r)$ este un vector bit $(w_0^q, \dots w_s^q)$ și reprezintă un întreg asociat al poziției (i,j,k) din matricea restricțiilor dată de:

plan: $k = [(q-1)/n\cdot m + 1]$ pentru
$l = (q-1) \bmod m\cdot n$ (2.1)
coloană: $j = [(l-1)/m + 1]$
linie: $i = (l-1) \bmod m + 1$

Lungimea vectorului w (parametrul s) determină cel mai mare întreg $(2^{s+1} - 1)$ care poate fi reprezentat.

Satisfacerea restricțiilor. Fiecare vector soluție trebuie să verifice:

$$v_q \geq 0 \quad q = 1, 2, \dots m\cdot n\cdot p \qquad (2.2)$$

$$\sum_{k=1}^{p} \sum_{i=(k-1)\cdot m\cdot n+(u-1)\cdot n+1}^{(k-1)m\cdot n+(u-1)\cdot n+n} v_i = a_u \quad u = 1, m \qquad (2.3)$$

$$\sum_{j=t,\ \text{pas}\, n}^{mnp} v_j = b_t \quad t = 1,n \qquad (2.4)$$

$$\sum_{k=(s-1)m\cdot n+1}^{(s-1)m\cdot n+m\cdot n} v_k = c_s \quad s = 1,p \qquad (2.5)$$

Inegalitatea (2.2) este verificată întotdeauna, deoarece o secvență de 0 sau 1 se consideră ca un întreg pozitiv. Celelalte restricții dau totalizarea pentru fiecare sursă, destinație, capacitate auto, deși aceste formule nu pot fi simetrice.

Evaluarea funcției obiectiv exprimă costul total de transport de la surse la destinații cu mijloace de transport, care este dat de:

$$\text{eval}((v_1, v_2, \dots v_r)) = \sum_{q=1}^{r} v_q \cdot \text{cost}_{ijk} \qquad (2.6)$$

unde i,j,k sunt dați de (2.1).

Operatori genetici. Folosim operatori genetici clasici. *Mutația*: schimbarea unui singur bit în vectorul soluție corespunde aici la o schimbare a unei valori întregi v_q. Aceasta, în schimb pentru problemele de transport va declanșa o multitudine de transformări în sensul menținerii egalității restricțiilor.

Rezultă că reprezentarea vectorială menționată nu este cea mai indicată pentru a defini operatorii genetici în problemele de transport.

Algoritm GENETIC-2.
Părăsim reprezentarea vectorială a soluțiilor și folosim *structura matriceală în trei dimensiuni* a datelor. Orice matrice soluție $X=\{(x_{ijk}) / (i, j, k) \in I\}$ trebuie să verifice:

$$x_{ijk} \geq 0, \forall (i, j, k) \in I \qquad (3.1)$$

$$\sum_{j\in N}\sum_{k\in P} x_{ijk} = a_i, \ i \in M \qquad (3.2)$$

$$\sum_{i\in M}\sum_{k\in P} x_{ijk} = b_j, \ j \in N \qquad (3.3)$$

$$\sum_{i\in M}\sum_{j\in N} x_{ijk} = c_k, \ k \in P \qquad (3.4)$$

Prezentăm în continuare un procedeu de a obține o soluție care satisface toate restricțiile (3.1) - (3.4). Numim aceasta procedură de *inițializare* care este de fapt o componentă fundamentală a operatorilor genetici care sunt prezentați în continuare.

```
INIŢIALIZARE:
input: array a ( m ), b ( n ), c ( p )
output: an array x ( m, n, p ) astfel că sunt satisfăcute ( 3.1. ) - ( 3.4. )
procedure
begin
        set toate numerele de la 1 la m·n·p nevizitate
        repeat
                select un număr aleator q nevizitat de la 1 la m·n·p
                set k = [ ( q - 1 ) / ( n · m ) + 1 ]
                set l = ( q - 1 ) mod ( n · m )
```

```
            set j = [ ( l - 1 ) / m + 1 ]
            set i = ( l - 1 ) mod m + 1
            set val = min ( a_i, b_j, c_k )
            set x_ijk = val
            set a_i = a_i - val
            set b_j = b_j - val
            set c_k = c_k - val
        until toate numerele sunt vizitate
end
```

Pot exista secvenţe de numere pentru care procedura de INIŢIALIZARE produce soluţia optimă. Procedura generează orice soluţie posibilă care conţine cel mult $m+n+p-2$ elemente întregi nenule. Aceasta nu generează alte soluţii care, deşi posibile, nu au această caracteristică, adică prin acest procedeu se creează numai soluţii admisibile de bază, cu componente întregi.

Exemplu: (pentru cazul exemplificat)

Există cu totul 4 x 4 x 3 = 48 numere, toate sunt nevizitate la început.

Selectăm la întâmplare numere: fie q = 32. Atunci k = 2, j = 4, i = 4, val = min(10, 9, 31) = 9, x_{442} = 9, apoi a_4 = 1, b_4 = 0, c_2 = 23. Repetăm aceste calcule cu următoarele numere la întâmplare: 15, 6, 2, 19, 27, 25, 41, 42, *, *, ... Evident, după 9 iteraţii toate componentele a_i, b_j, c_k sunt "0" şi rezultă urmatoarele matrice din X_1:

k=1

I\j

	8		
	8		
		1	

k=2

i\j

		16	
2		4	
			9

k=3

i\j

9	3		

Selectând întâmplător numerele: 1, 2, 18, 22, 23, 27, 43, 47, 48, *, *, ... rezultă matricea X_2:

k=1

i\j

11	6		

k=2

i\j

	7		
	6	2	
		16	

k=3

i\j

		2	
		1	9

În *acest caz reprezentăm o soluţie* a problemei tridimensionale de transport *ca un vector*: o suită de m*n*p întregi diferiţi, dintre (1, m*n*p), care conform procedurii de iniţializare va produce o soluţie admisibilă. Cu alte cuvinte vom vedea un vector soluţie ca o permutare de numere şi urmărim permutările care corespund soluţiei optimale.

Satisfacerea restricţiilor: Orice permutare a m*n*p numere distincte produce o soluţie unică care satisface toate restricţiile (3.1) - (3.2). Ne garantează aceasta procedura de iniţializare.

Evaluarea funcţiei: Deoarece orice permutare corespunde unei matrici unice, $X=\{x_{ijk}/(i,j,k)\in I\}$, funcţia de evaluare este expresia funcţiei obiectiv.

$$\mathrm{eval}(x_{ijk}) = \sum_{i\in M}\sum_{j\in N}\sum_{k\in P} c_{ijk}x_{ijk} \qquad (3.8)$$

Operatorii genetici. Asupra datelor aranjate astfel pot acţiona operatorii genetici uzuali: inversiunea, mutaţia, încrucişarea (o încrucişare mai complexă).

- *inversiunea*: orice vector soluţie (y_1, y_2, ... y_r), ($r = m \cdot n \cdot p$) poate fi uşor inversat în alt vector soluţie (y_r, y_{r-1}, ... y_1);
- *mutaţia*: oricare două elemente y_i şi y_j ale vectorului soluţie (y_1, y_2, ... y_r), pot fi uşor schimbate, rezultând un alt vector;
- *încrucişarea (crossover)*: Un operator de încrucişare arbitrar va conduce şi la vectori care nu pot fi acceptaţi soluţii. De aceea în [MIC-92] pentru problema bidimensională de transport, se propune un operator euristic de încrucişare, din familia PMX, pe care îl utilizăm şi noi pentru această problemă tridimensională.

Din doi părinţi se obţine un urmaş astfel:

1) se face o copie a părintelui secundar,
2) se alege o parte arbitrară a primului părinte,
3) se fac schimbări minime pentru a obţine structura căutată.

Exemplu: Pentru părinţii (ale căror roluri pot fi inversate):

P1	1	2	3	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	8	9	10	11	12
P2	<u>7</u>	3	1	11	<u>4</u>	12	<u>5</u>	2	10	9	<u>6</u>	8

- alegem arbitrar partea 4 5 6 7 din poziţia a patra, de lungime 4.
- toate poziţiile selectate sunt marcate în P2.
- se construieşte urmaşul folosind doar poziţiile nemarcate din P2, intercalând din poziţia a patra, partea aleasă de lungime 4.

Urmaşul rezultat este:

U	3	1	11	4	5	6	7	12	2	10	9	8

4. Concluzii

Algoritmii genetici pentru problema bidimensioanlă de transport au fost testaţi [MIC-92] atât pentru cazuri reale cât şi artificiale. Algoritmii genetici merită să fie folosiţi în cazul neliniar, când funcţia obiectiv este neliniară. Pentru compararea acestor algoritmi cu algoritmii de optimizare trebuie decis criteriul care să fie utilizat: numărul de generaţii care să atingă optimul, timpul în care obţinem optimul, numărul de operaţii cerute să completeze fiecare generaţie.

Bibliografie

[COR-72] - CORBAN A., Un model multidimensional de transport, Studii şi cercetări matematice, volumul. 24,7/1972, Bucureşti;

[MIC-92] - MICHALEWICZ Z., *Genetic Algorithms+Data Structures=Evolution Programs,* Springer Verlag, 1972, Berlin;

[MIH-70] - MIHU C., *Programarea tridimensională de transport,* Editura Tehnică, 1970, Bucureşti;

[MOA-76] - MOANŢĂ D., An Algorithm for Solving The transportation tridimensional Problem, Ec. Comp. and Ec. Cybernetics Studies and Research, 2/1976, Bucureşti.