**PROBLEME PEAG**

* **1 -**

**Pasul 1:** Enuntul problemei

Cele N orașe stat din Grecia antică se luptă între ele pentru dominație, dar în fața unei amenințări externe hotărăsc să se unească. Pentru a stabili planul de apărare, delegații orașelor urmează să se întîlnească, fiecare oraș desemnîndu-și un singur reprezentant. Cunoscînd rivalitățile istorice dintre orașe folosiți un algoritm genetic pentru a găsi o modalitate de așezare a delegaților la masa tratativelor astfel încît delegații din orașe rivale să nu fie vecini (se presupune că acest lucru este posibil). Harta orașelor stat între care există animozități este exprimată printr-o matricea pătratică de ordin N, numită CONFLICT, 𝐶𝑂𝑁𝐹𝐿𝐼𝐶𝑇(𝑖,𝑗) = { 0, 𝑑𝑎𝑐ă 𝑖 ș𝑖 𝑗 𝑛𝑢 𝑠𝑒 𝑎𝑓𝑙ă î𝑛 𝑠𝑡𝑎𝑟𝑒 𝑑𝑒 𝑐𝑜𝑛𝑓𝑙𝑖𝑐𝑡 𝑠𝑎𝑢 𝑖 = 𝑗 1, 𝑑𝑎𝑐ă 𝑜𝑟𝑎ș𝑢𝑙 𝑠𝑡𝑎𝑡 𝑖 𝑒𝑠𝑡𝑒 î𝑛 𝑐𝑜𝑛𝑓𝑙𝑖𝑐𝑡 𝑐𝑢 𝑜𝑟𝑎ș𝑢𝑙 𝑠𝑡𝑎𝑡 𝑗

**Pasul 2:** Reprezentare cromozomiala. Spatiul genotipurilor (spatial solutiilor)

Populatia este formata din indivizi, vectori de permutare de lungime N. Fiecare genă din cromozom reprezintă un oraș, iar valoarea genei indică poziția delegatului din orașul respectiv la masa tratativelor.

Spatiul genotipurilor (sau spațiul soluțiilor) este definit de toate posibilele permutări ale delegaților la masa tratativelor care respectă restricția ca doi delegați din orașe rivale să nu fie așezați unul lângă celălalt.

**Pasul 3:** Constrangeri

Delegatii din orase rivale nu trebuie sa fie vecini.

**Pasul 4:** Evaluare (functia obiectiv)

Functia obiectiv masoara gradul in care solutia respecta restrictia impusa, adica sa numere cate perechi de delegati din orase rivale sunt asezati consecutiv la masa.

fitness(s)=

​∑*i*=1*N*−1{1, daca delegatul din orașul *i* este rival cu delegatul din orașul *i*+1

{0, altfel

**Pas 5:** Generarea populatiei initale

Aleasa aleatoriu din spatiul solutiilor, cu respectarea restrictiilor.

**Pas 6:** Conditii de oprire

1.Numarul limitat de generatii

2.Diversitate genetica: minim 2 calitati diferite in populatie

3.Evolutie: cresterea calitatii in ultimele 50 de generatii

**Pas 7:** Selectarea parintilor

Mecanismul de selectie va fi turneul.

Alegem un nr predefinit de indivizi din populatie pentru turneul de selectie, dintre care se alege castigatorul turneului pe baza valorii functiei fitness.

**Pas 8:** Recombinarea

Recombinarea va fi de tip ordered crossover. (OCX)

Se aleg una sau mai multe pozitii aleatoare in cromozomi, impartindu-I in sectiuni care vor fi schimbate intre parinti. Se copiaza sectiunile intre punctele de crossover din unul dintre parinti in descendent, iar valorille lipsa sunt completate cu informatii din celalalt parinte.

**Pas 9:** Mutatii

Deoarece reprezentarea cromozomilor se face prin permutari de cate N, vom folosi mutatia interschimbare. Aceasta schimba pozitiile, alese aleatoriu, a doua gene dintr-un cromozom .

**Pas 10:** Selectarea supravietuitorilor ( a generatiei urmatoare)

Tipul de selectie pentru generatia urmatoare este elitismul.

**Pas 11:** Conditia de oprire a GA

Algoritmul se va opri atunci cand va atinge numarul maxim de operatii sau daca nu exista macar doua calutati diferite in populatie.

**Pasul 1:** Enuntul problemei

La începutul anului universitar un student trebuie să aleagă cursurile pe care dorește să le urmeze. Bugetul disponibil este de 10.000 lei și pot fi alese trei tipuri de cursuri:

a. Curs cu costul 1000 lei, acordă 5 puncte de credit și necesită 80 ore de studiu individual.

b. Curs cu costul 800 lei, acordă 3 puncte de credit și necesită 40 ore de studiu individual.

c. Curs cu costul 1500 lei, acordă 8 puncte de credit și necesită 100 ore de studiu individual.

Folosiți un algoritm genetic pentru a ajuta studentul să aleagă o combinație de cursuri (cîte din fiecare tip) astfel încît să nu depășească bugetul disponibil, numărul mediu de credite să fie cît mai mare, timpul mediu de studiu individual să nu depășească 70 de ore. Notă: numărul mediu de credite și numărul mediu de ore de studiu individual se calculează ca medie ponderată.

**Pasul 2:** Spatiul solutiilor:

S = S1 x S2 x S3 = [0, 10] x [0, 12] x [0, 6]

Fenotip: x = (a, b, c) cu a = [0, 10], b = [0, 12], c = [0, 6], numere intregi

**Pasul 3:** Reprezentare:

Genotipul = fenotipul (nu se precizeaza codificare)

Populatia este reprezentata de o multime de vectori formati din genotip si rezultatul functiei fitness.

**Pasul 4:** Constrangeri:

* Sa nu depasim bugetul de 10.000 de lei
* timpul mediu de studiu individual să nu depășească 70 de ore

**Pasul 5:** Evaluare:

Functia fitness va avea ca rezultat numarul mediu de credite, avand grija sa respectam constrangerile prezentate mai sus.

F = (a \* 5 + b \* 3 + c \* 8) / (a + b + c);

**Pasul 6:** Generarea populatiei initiale:

se face aleator, din spatiul solutiilor, cu respectarea conditiilor de restrictie.

**Pasul 7:** Selectia parintilor:

Se foloseste mecanismul ruleta cu distributia FPS cu sigma-scalare.

**Pasul 8:** Recombinare:

Schema de recombinare pentru probleme cu constrangeri, reprezentare cu siruri de numere intregi se utiliza operatorul de recombinare uniforma/ uni-punct cu probabilitate 0,8.

**Pasul 9:** Mutatie:

Schema de mutatie pentru problemele cu constrangeri reprezentate cu siruri de numere intregi

Se va utiliza operatorul de mutatie de tip fluaj.

Daca individul este nefezabil, nu se va face mutatia.

**Pasul 10:** Selectia generatiei urmatoare:

Se va face folosind elitismul.

**Pasul 11**: Conditii de oprire

* nr limitat de iteratii de generare
* diversitatea genetica (sa existe 2 calitati diferite in populatie)
* evolutia (cresterea calitatii in ultimele generatii)

**Pasul 1:** Enuntul problemei

Un produs P este obţinut în două fabrici situate în două orașe (Bucureşti şi Craiova) şi este stocat

pentru desfacere în trei depozite, unul situat în Ploieşti, unul în Piteşti şi unul la Cluj. Fabrica din

Bucureşti poate produce săptămînal 120 de tone din produsul P, iar fabrica din Craiova poate produce P în cantitate de 140 tone pe săptămînă. Pentru desfacerea produsului, necesarul săptămînal este: pentru depozitul din Ploieşti 100 de tone, pentru depozitul din Piteşti, 60 de tone, respectiv pentru depozitul din Cluj 80 de tone. În tabelul de mai jos sînt prezentate costurile de transport pe tona de produs.

Ploieşti Piteşti Cluj

Bucureşti 50 70 90

Craiova 60 70 100

Problema de rezolvat: calculul numărului de tone din produsul P care trebuie furnizate de cele două fabrici

fiecărui deposit astfel incat costul de transport sa fie minim si astfel incat sa fie respectate contidiile enuntate mai sus.

**Pasul 2:** Spatiul solutiilor

S = S1 x S2 = [0, 140] x [0, 120]

Fenotipul este format din x = (a, b), unde a = [0, 140], b = [0, 120], numere reale.

**Pasul 3:** Reprezentarea

Genotip = fenotip (nu se precizeaza codificare)

Populatia este reprezentata de o multime de vectori formati din genotip, alaturi de rezultatul functiei fitness.

**Pasul 4:** Constrangeri

Sa se atinga necesarul minim de tone de produs pentru fiecare dintre depozite: din Ploieşti 100 de tone, Piteşti, 60 de tone, respectiv pentru Cluj 80 de tone.

**Pasul 5:** Evaluare

Această funcție fitness reprezintă suma costurilor de transport pentru fiecare rută specifică din fiecare fabrică către fiecare depozit, înmulțită cu cantitatea alocată.

**Pasul 6:** Generarea populatiei initiale:

Se face aleator, din spatiul solutiilor, cu respectarea conditiilor de resprictie prezentate mai sus.

**Pasul 7:** Selectia parintilor

Se face cu operatorul de recombinare SUS cu distributia FPS cu sigma scalare.

**Pasul 8:** Recombinare: operatorul cx

**Pasul 9:** Mutatie: interschimbare

**Pasul 10:** Generatia urmatoare: limita de varsta 1 + elitism

**Pasul 11:** Conditii de oprire: diversitate genetica, nr limitat de cicluri evolutive.

**Pasul 1:** Enuntul problemei

Cele N orașe stat din Grecia antică se luptă între ele pentru dominație, dar în fața unei amenințări externe hotărăsc să se unească. Pentru a stabili planul de apărare, delegații orașelor urmează să se întîlnească, fiecare oraș desemnîndu-și un singur reprezentant. Cunoscînd rivalitățile istorice dintre orașe folosiți un algoritm genetic pentru a găsi o modalitate de așezare a delegaților la masa tratativelor astfel încît delegații din orașe rivale să nu fie vecini (se presupune că acest lucru este posibil). Harta orașelor stat între care există animozități este exprimată printr-o matricea pătratică de ordin N, numită CONFLICT, 𝐶𝑂𝑁𝐹𝐿𝐼𝐶𝑇(𝑖,𝑗) = { 0, 𝑑𝑎𝑐ă 𝑖 ș𝑖 𝑗 𝑛𝑢 𝑠𝑒 𝑎𝑓𝑙ă î𝑛 𝑠𝑡𝑎𝑟𝑒 𝑑𝑒 𝑐𝑜𝑛𝑓𝑙𝑖𝑐𝑡 𝑠𝑎𝑢 𝑖 = 𝑗 1, 𝑑𝑎𝑐ă 𝑜𝑟𝑎ș𝑢𝑙 𝑠𝑡𝑎𝑡 𝑖 𝑒𝑠𝑡𝑒 î𝑛 𝑐𝑜𝑛𝑓𝑙𝑖𝑐𝑡 𝑐𝑢 𝑜𝑟𝑎ș𝑢𝑙 𝑠𝑡𝑎𝑡 𝑗

**Pasul 2:** Spatiul solutiilor

Populatia este formata dintr-o multimer de vectori de tip permutare de lungime N. Fiecare gena din cromozom reprezinta un oras, iar valoarea genei reprezinta locul in care este asezat orasul la masa tratativelor.

**Pasul 3:**  Constrangeri:

Delegatii care sunt in rivalitate (conflict[i][j] = 1) sa nu fie vecini la masa.

**Pasul 4:** Evaluare

Functia fitness consta in numararea perechilor de delegati aflati in rivaltate in asezarea de la masa tratativelor.

**Pasul 5:** Generarea populatiei initiale:

Se alege aleator din spatiul solutiilor avand grija sa respectam restrictiile.

**Pasul 6:** Selectia parintilor:

Se foloseste mecanismul ruleta cu distributia FPS cu sigma-scalare.

**Pasul 7:** Recombinare

Vom folosi operatorul de recombinare ordered crossover(OCX) pentru permutari.

**Pasul 8:** Mutatii

Se va alege operatorul de mutatie prin interschimbare cu probabilitate mica.

Daca individual este nefezabil, acesta nu va mai fi supus mutatiei.

**Pasul 9:** Selectia generatiei initiale:

Se va folosi elitismul.

**Pasul 10:** Conditii de oprire

* atingerea numarului limitat de cicluri evolutive
* diversitatea genetica(cel putin 2 calitati diferite in populatie)
* evolutie(cresterea calitatii in urmatoarele generatii)
* se atinge numarul maxim de perechi distincte posibile n\*(n-1)/2

Pasul 1: Enuntul problemei

Pentru aprovizionarea a n zone defavorizate cu produse de bază, se formează un convoi umanitar, tren avînd n vagoane. Trenul va vizita succesiv fiecare zonă unde va descărca materialele prevăzute pentru acea zonă. Materialele sînt ambalate în m containere speciale. Pentru fiecare container se cunoaște masa totală. Utilizați un algoritm genetic pentru a aloca fiecare container într-un vagon, astfel încît vagoanele să fie încărcate cu mase totale cît mai apropiate. Masele containerelor pot fi preluate dintrun fișier text. Nu se ia în considerare o limită superioară de masă pentru vagoane.

**Pasul 2: Spatiul solutiilor**

S = s1 x s2 = [1, N] x [1, M]

Fenotip: x = (n, m) unde n =….

**Pasul 3: Reprezentare**

Populatia este formata dintr-o multimer de vectori permutare de lungime m, unde fiecare pozitie reprezinta un container, iar valoarea de pe pozitii este vagonul in care este assignat.

**Pasul 4: Constrangeri**

Nu avem.

**Pasul 5: Evaluarea**

Functia fitness consta in calcularea sumei masei totale a fiecarui vagon si a difentei dintre vagoane. Rezultatul functiei obiectiv este 1/media diferentelor dintre vagoane.

**Pasul 6: Generarea populatiei initiale**

Se va face aleator, din spatial solutiilor.

**Pasul 7: Selectia parintilor**

Se foloseste SUS cu distributia FPS cu sigma scalare.

**Pasul 8: Recombinare**

Folosim operatorul ocx.

**Pasul 9: Mutatie**

Interchichimbare. Daca e nefezabil, nu mai sufera mutatii.

**Pasul 10: Selectia generatiei urmatoare prin elitism si limita de varsta 1**

**Pasul 11: Conditii de oprire**

* **se atinge nr max de perechi distincte posibile**
* **evolutia**
* **diversitate genetica**
* **nr max cicluri evolutive**