**ENUNTUL PROBLEMEI:**

1. Elaborați un proiect pentru rezolvarea genetică a următoarei probleme. În procesul de prelucrare a butucilor, o fabrică de cherestea furnizează scândură de două tipuri: finisată, notată cu *Prod1*, şi pentru construcţii, notată cu *Prod2*. Pentru obţinerea a 1000 de unităţi de scândură finisată procedura de tăiere durează 2 ore şi procedura de rindeluire durează 5 ore. Pentru furnizarea a 1000 unităţi de scândură de construcţii, procesul de tăiere durează 2 ore şi cel de rindeluire necesită 3 ore. Fierăstrăul industrial cu care este realizată tăierea poate fi folosit 8 ore pe zi şi rindeaua este disponibilă 15 ore pe zi. Dacă profitul obţinut din producerea a 1000 unităţi de produs este de 120 lei în cazul scândurii finisate, respectiv 100 de lei în cazul scândurii de construcţii, se cere să se determine cantitatea de scândură din fiecare tip, în mii de unităţi, care trebuie produsă zilnic pentru maximizarea profitului fabricii.

**STRUCTURA GENERALA A UNUI ALGORITM GENETIC**

i<- 0

1. **Initializarea populatiei**: Pi este obtinuta prin generarea aleatoare a candidatilor la solutie
2. **Evaluarea candidatilor**: pentru fiecare x∈ Pi, determina f(x)
3. Repeta:
   1. Selecteaza multimea de parinti Bpi
   2. Recombina perchi(sau n-tupluri) de parinti
   3. Efectueaza mutatii asupra progeniturilor rezultate
   4. Evalueaza noii candidati la solutie
   5. Selecteaza indivizii pentru constituirea gneratiei urmatoare, Pi+1
   6. i <- i+1 pana can conditia este satisfacuta

Pentru a rezolva problema data cu ajutorul algoritmilor genetici este necesara dterminarea uni functii obiectiv ce se doreste a fi maximizata. In cazul nostru, functia obiectiv este reprezntata de profitul adus de cantitatea produsa. Functia obiectiv se poate scrie sub forma:

functie\_obiectiv (x1, x2)= x1\*t1+x2\*t2, iar

x1, x2- vectorul linie al cantitatilor produse, iar t transpus- vectorul coloana al profiturilor aduse de cele 2 produse.

Solutia problemei alese o reprezinta un vector y de cantitati format din numere reale, avand functia obictiv maxima. Primul pas spre rezolvarea problemei il constituie generarea populatiei initiale, formata dintr- un numar cunoscut de indivizi, alesi la inceput, generati aleator. Ci din urma trebuie sa respecte conditiile de existenta stabilite in cadrul problemei. Dintre acesstea amintim:

1. Conditiile de negativitate;
2. Conditiile asociate functionalitatii masinilor necesare producerii scandurii. Astfel, facand referire la cunsotintele anterioare de programare liniara, respectiv simplex, observam ca indivizii trebuie sa respecte simultan urmatoarele 2 conditii;

2x1+2x2<=8 && 5x1+3x2<=15

Pasii pe care o sa- i parcurg pentru generarea unui individ sunt urmatorii;

1. Generez un vector aleator de numere cuprinse in intervalul 0 si cantitatea din acel produs care ar putea fi manufacturata intr o zi.
2. Verific daca acesta este fezabil, daca respecta conditiile impuse:

* Calculam dependentele nivelelor de productie pentru fiecare scandura in parte in functie de numarul de ore in care utilajele pot fi disponibile pe parcursul unei zile;
* Daca producerea scandurilor nu depaseste numarul de ore disponibile, individul este valid.

Daca nu este valid, se vor relua pasii prezentati mai sus.

1. Calculam valoarea functiei obiectiv pentru individul valid.

Dupa generarea populatiei initiale, ne vom axa pe procesul de evolutie si vom avea ca scop selectarea unui numar de parinti ce vor reprezenta un bazin de recombinare pentru actiuiile viitoare . Algerea acestora va fi influentata, in cazul nostru, de dimensiunea turneului, k, care poate realiza foarte usor costrangerea selectiei prin crestrea valorii acestuia, insa fara sa depaseaca dimensiunea populatiei initiale. Pe langa acest indicator, selectia indivizilor se bazeaza si pe calitatea acestora, daca sunt sau nu fezabili.

Populatia de parinti nou formata va trece prin procesul de recombinare, unde, in cazul nostru, sunt obtinuti cromozomi cu alele noi, rezultate prin combinarea artmmetica a alelelor cromozomilor parinti. Si in acst caz ne vom asigura ca ci doi copii raman solutii validd ale problemei. In cele din urma le vom determina si calitatea.

Populatia de copii va fi supusa procedeului de mutatie. In cazul nostru, se va realiza mutatia fiecari gene din fiecre cromozom cu o probabilitate definita de noi. Multimea va obtine un noi individ caruia ii testam fezabilitatea.

Din multimea parintilor si copiilor sau progniturilor, o sa selectam generatia urmatoare Procesul de selecția va trebui să asigure perpetuarea celor mai adaptați indivizi, însă să nu excludă posibilitatea ca în următoarea generație să existe și indivizi mai slabi care vor da naștere unor copii foarte buni.

Soluția va fi individul cu valoarea funcției obiectiv maximă din ultima generație obținută.

**FUNCTII ALE ALGORITMULUI GENETIC**

Pentru generarea populatiei am folosit algoritmul pentru generare a unui vctor de numere reale aflate intr un interval, adaugand pe ultima coloana valoarea functiei obictiv sau fitness(in cazul nostru, valoarea profitului obtiuit din maximizarea functiei).

In continuare, am selectat multimea de parinti folosind un aloritm de tip turneu, unde k<numarul de indivizi ai populatiei, urmand urmatorul algoritm:

Pentru i=1...dim executam:

1. Generam aleator k indivizi din populatie;
2. Calculam valoarea functiei obiectiv pentru cei k indivizi, de unde este selectat cel mai bun individ, cel care reuseste sa maximizeze profitul;
3. Ii atribuim parintelui valoarea individului ales.

Probabilitatea de selectie a individului depinde de rangul acestuia in cadrul populatiei, de dimensiuna turneului(cu cat k ste mai mare cu atat exista o probabilitate mai mare de includere a unor indivizi cu calitati uperioar mediei).

Urmeaza procesul de recombinare care presupune selectarea a doua solutii candidat parinte, generarea aleatoare a unui numar din [0, 1) si compararea lui cu rata de recombinare. Daca r<=rata de recombinare, indivizii sunt creati prin recombinara celor doi parinti, altfel sunt creati sexuat. Rata de recombinarea presupune sansa ca perechea de parinti selectata sa fie supusa operatiei de recombinare. Copiii participanti la acest proces sunt determinati print- un proces de recombinarea aritmetica totala, unde sunt obtinuti cromozomi cu alele noi, rezultate prin combinarea alelelor parintilor.

Am ales sa implementez mutatia de tip fluaj care presupune modificarea fiecarei gene din fiecare cromozom cu o probabiitate pm, prin generarea aleatoare a unei valori in multimea valorilor admisibile pe gena respectiva. Daca s a selectat gena x pentru mutatie, individul va fi generat aleator.

Pentru selectia generatiei viitoare am folosit un algoritm de tip elitist care urmareste cea mai buna cantitate din populatia curenta, o notam cu „a”. Daca aceasta este aleasa pentru inlocuire si niciunul din urmasii selectati pentru schimb nu au valoarea functiei obiectiv cel putin egala cu cea lui „a”, atunci cel din urma este mentinut in generatia urmatoare, si este eliminat unul din urmasii selectati pentru inlocuire.

**FUNCTII IMPLEMENTATE**

1. Functia obiectiv

function [ val ] = fitness( x )

val=120\*x(1)+100\*x(2);

end

1. Functia de verificare a conditiilor

function [ ok ] = fezabil( x )

ok=0;

if(2\*x(1)+2\*x(2)<=8)&&(5\*x(1)+3\*x(2)<=15);

ok=1;

end

1. Functia de genrare a populatiei

function [ P ] = genrare\_populatie( dim )

P=zeros(dim,3);

i=1;

x=zeros(1,2);

while i<=dim

x(1)=unifrnd(0,3);

x(2)=unifrnd(0,4);

if fezabil(x)

P(i,1:2)=x;

P(i,3)=fitness(x);

i=i+1;

end;

end;

end

1. Functia de selectie a parintilor

function [ parinti ] = selectie\_turneu(pop,k )

%I:pop-populatia curenta, k-nr de participanti la turneu

%pe ultima coloana e afla valoarea functiei obictiv

% E: parinti- parinti selectati

[m,n]=size(pop);

parinti=zeros(m,n);

turneu=zeros(k,n);

for i=1:m

for j=1:k

t=unidrnd(m);

turneu(j,:)=pop(t,:);

end;

[~,p]=max(turneu(:,n));

parinti(i,:)=turneu(p,:);

end;

end

1. Functia de recombinare

function [ c1,c2 ] = arit\_totala( p1,p2,alpha )

% - parintii(p1, p2), ponderea(alpha), copiii(c1, c2)

[~,n]=size(p1);

c1(1:n)=alpha\*p1(1:n)+(1-alpha)\*p2(1:n);

c2(1:n)=alpha\*p2(1:n)+(1-alpha)\*p1(1:n);

end

function [ popRecombinata ] = recombinare\_populatie( P,dim,alpha,pr )

popRecombinata =P;

[~,n]=size(P);

for i=1:2:dim-1

r=unifrnd(0,1);

if r<pr

[c1,c2]=arit\_totala(popRecombinata(i,1:n-1),popRecombinata(i+1,1:n-1),alpha);

if fezabil(c1)

popRecombinata(i,1:n-1)=c1;

popRecombinata(i,n)=fitness(c1);

end;

if fezabil(c2)

popRecombinata(i+1,1:n-1)=c2;

popRecombinata(i+1,n)=fitness(c2);

end;

end;

end;

end

1. Functia de mutatie

function [ y ] = fluaj( x,pm,a,b,t )

y=x;

r=unifrnd(0,1);

if r<pm

q=normrnd(0,t/3);

y=y+q;

if y<a;

y=a;

end;

if y>b

y=b;

end;

end;

end

function [ y ] = test\_mutatie( x,dim, pm, t )

%x-individul asupra caruia se aplica mutatia, pm- probabilitatea de

%mutatie, a si b captele de interval, y individul obtinut

y=x;

z=zeros(1,2);

for i=1:dim

z(1)=fluaj(y(i,1),1,3,t,pm);

z(2)=fluaj(x(i,2),1,4,t,pm);

if fezabil(z)==1

y(i,1:2)=z;

y(i,3)=fitness(y(z));

end;

end;

end

1. Functia de selectie a generatiei urmatoare

function [ rezultat ] = generatie\_urmatoare( P,desc )

% selectarea elitista a generatiei noi

% I: P - populatia curenta, desc - descendentii generati

% pe ultima coloana se afla valoarea functiei obiectiv

% E: rezultat - generatia noua

rezultat=P;

[dim,n]=size(P);

fctob=P(1:dim,n);

fctobN=desc(1:dim,n);

[max1,i]=max(fctob);

[max2,j]=max(fctobN);

if(max1>max2)

[~,k]=min(fctobN);

rezultat(k,1:n)=P(i,1:n);

end;

end

Scriptul pentru testare

dim=10;

disp('Generare populatie:');

P = genrare\_populatie(dim)

pop=sortrows(P,3);

P=pop;

disp('Selectam multimea de parinti: ');

parinti=selectie\_turneu(P, 9)

alpha=0.2;

pr=0.8;

disp('Populatia recombinata');

popRecombinata=recombinare\_populatie(parinti, dim, alpha,pr)

a=1;

b=3;

pm=1/10;

t=2;

disp('Mutatie: ');

popMutata = test\_mutatie(popRecombinata,dim, pm, t);

disp('Generam populatia urmatoare: ');

rezultat = generatie\_urmatoare(P,popMutata)

P=sortrows(rezultat,3);

maxim=max(P(1:dim,3));

for i=1:dim

if(pop(i,3)==maxim)

disp('Cantitatea de scandura 1: ');

disp(round(pop(i,1)\*1000));

disp('Cantitatea de scandura 2: ');

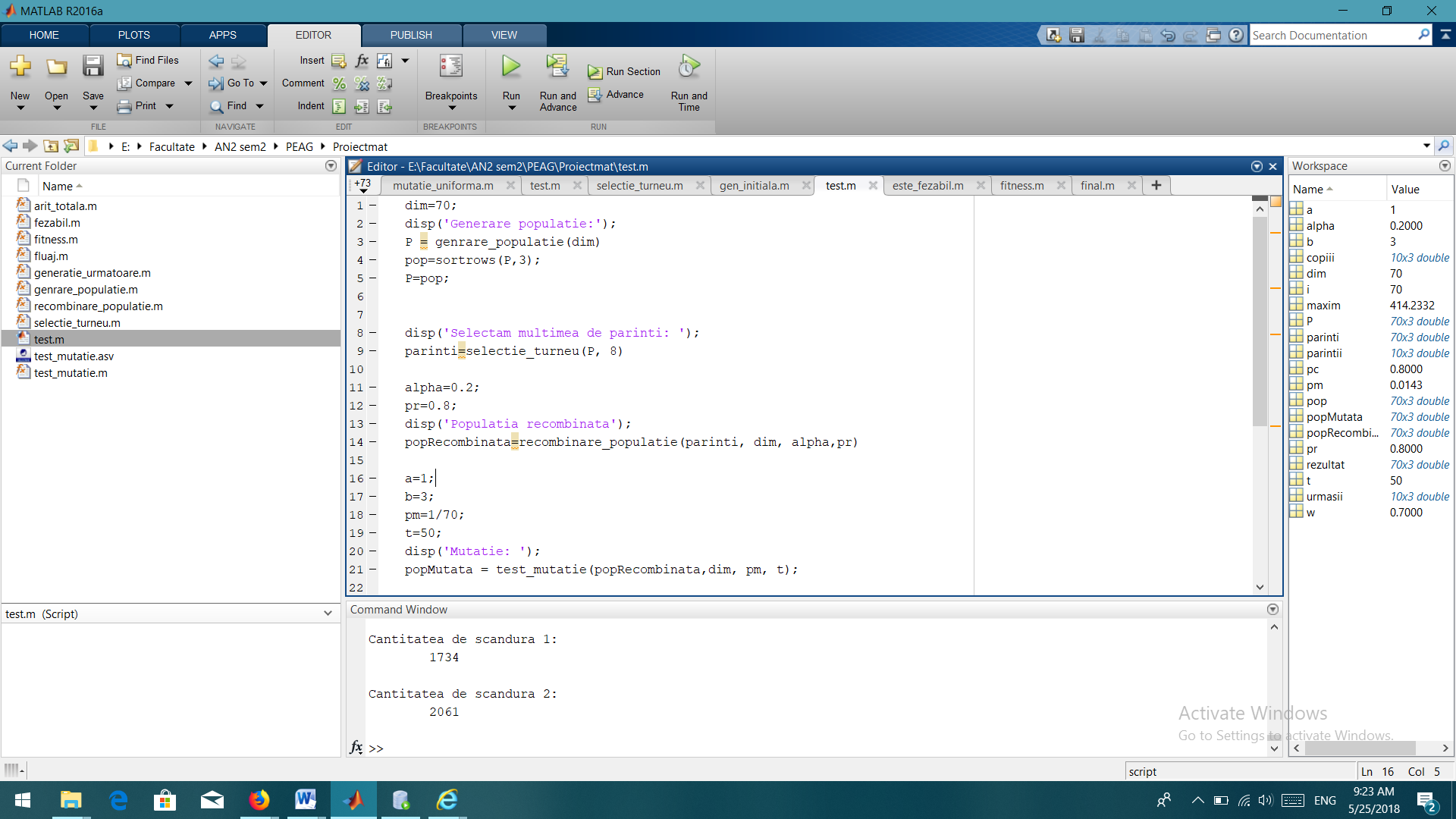
disp(round(pop(i,2)\*1000));

end;

end;

**PARTEA DE TESTARE**

Am ales initial o populatie de 70, k = 8, alpha- ponderea de recombinare – 0.2, rata de recombinare(pr) = 0.8, rata de mutatie=1/70, a=1, b=3, t=50



La un al doilea test, voi seta dimensiunea populatiei =10, k=9 , alpha- ponderea de recombinare – 0.4, rata de recombinare(pr) = 0.6, rata de mutatie=1/10, a=1, b=3, t=6

