Generarea unei permuatari(generarea unui individ al populatiei)

function y=gen\_perm(m)

% genereaza permutare

%I: m - dimensiune permutare

%E: y - permutare

y=zeros(1,m);

for i=1:m

gata=0;

while(~gata)

x=unidrnd(m);

if(~ismember(x,y))

y(i)=x;

gata=1;

end;

end;

end;

end

Generarea unei populatii reprezentata prin permutari

function Pop=gen\_pop\_perm(dim,m)

%generarea populatiei initiale de permutari pe multimea m

%I: dim - nr. indivizi, m - dimensiune individ (permutare)

%E: pop - populatia

Pop=zeros(dim,m);

for i=1:dim

Pop(i,1:m)=gen\_perm(m);

end;

end

Operatorul de mutatie prin amestec

function y=m\_amestec(x,pm)

% mutatie prin amestec pentru reprezentare cu permutari

% I: x - individul (permutare); pm - probabilitate de mutatie

% E: y - individ mutat

[~,m]=size(x);

y=x;

r=unifrnd(0,1);

if r<pm

p1=unidrnd(m);

p2=p1;

while p2==p1

p2=unidrnd(m);

end;

a=gen\_perm(p2-p1+1);

y(p1:p2)=x(p1-1+a(1:p2-p1+1));

end;

end

Operatorul de mutatie prin amestec intre 2 pozitii

function [PopM] = mutatie\_amestec(pm,dim,m)

% operatorul de mutatie prin amestec intre doua pozitii

% I: pm - probabilitate de mutatie; dim - dim. populatie; m - nr. cromozomi

% E: populatie mutata

pop=gen\_pop\_perm(dim,m);

PopM=pop;

for ind=1:dim

r=unifrnd(0,1);

if(r<pm)

disp(['Mutatie in cromozomul ' num2str(ind)]);

disp(PopM(ind,:));

i=0;j=0;

while(i==j)

poz=unidrnd(m,1,2);

i=min(poz);

j=max(poz);

end;

disp(['Pozitiile ' num2str(i) ' si ' num2str(j)]);

p\_amestec=gen\_perm(j-i+1);

disp('Amestecul este conform permutarii de pozitii:');

disp(p\_amestec+i-1);

PopM(ind,i:j)=PopM(ind,p\_amestec(1:j-i+1)+i-1);

disp('Cromozomul rezultat');

disp(PopM(ind,:));

end;

end;

end

Operatorul de mutatie prin interschimbare

function [y]=m\_interschimbare(x,pm)

% operatorul de mutatie prin interschimbare pe permutari

% I: x - individul (permutarea), pm - probabilitatea de mutatie

% E: y - individul modificat

y=x;

r=unifrnd(0,1);

if r<pm

n=length(x);

poz=unidrnd(n,1,2);

while poz(1)==poz(2)

poz(2)=unidrnd(n);

end;

poz=sort(poz);

y(poz(1))=x(poz(2));

y(poz(2))=x(poz(1));

end;

end

Operatorul de mutatie prin inversiune

function [y]=m\_inversiune(x,pm)

% operatorul de mutatie prin inversiune pe permutari

% I: x - individul (permutarea), pm - probabilitatea de mutatie

% E: y - individul modificat

y=x;

r=unifrnd(0,1);

if r<pm

n=length(x);

poz=unidrnd(n,1,2);

while poz(1)==poz(2)

poz(2)=unidrnd(n);

end;

poz=sort(poz);

y(poz(1):poz(2))=x(poz(2):-1:poz(1));

end;

end

Operatorul de mutatie prin inserare

function [ popN ] = mutatie\_inserare( pop, pm )

%MUTATIE\_INSERARE permutarea care sufera o mutatie

[dim,n]=size(pop);

popN=pop;

for i=1:dim

r=unifrnd(0,1);

if(r<pm)

disp('Mutatie efectuata in cromozomul'); disp(pop(i,:));

p=zeros(1,2);

p(1)=unidrnd(n);

p(2)=unidrnd(n);

while(p(1)==p(2)&&p(2)==p(1)+1)

p(2)=unidrnd(n);

end;

poz=sort(p);

disp('Pozitiile:');

disp(poz);

popN(i,1:poz(1))=pop(i,1:poz(1));

popN(i,poz(1)+1)=pop(i,poz(2));

popN(i,poz(1)+2:poz(2))=pop(i,poz(1)+1:poz(2)-1);

popN(i,poz(2)+1:n)=pop(i,poz(2)+1:n);

disp('Cromozom rezultat');

end;

end;