## TC 9-10. Abordarea GA pentru problema optimizării portofoliilor de *n* acțiuni. Variantele cu și fără short-selling

Problema pe care ne propunem să o rezolvăm este determinarea portofoliului de risc minim pentru un randament prognozat . Problema este modelată în termenii RISCMIN1M, transformată în problema de maxim.

Maximizează

Pentru comparaţie vor fi folosite datele din tabelul 1.

Tabelul 1.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***S1*** | ***S2*** | ***S3*** | ***S4*** | ***S5*** | ***S6*** | ***S7*** | ***S8*** | ***S9*** | *S10* |
| ***A1*** | 1.2 | 1.3 | 1.4 | 1.5 | 1.1 | 1.2 | 1.1 | 1.0 | 1.0 | 1.1 |
| ***A2*** | 1.3 | 1.0 | 0.8 | 0.9 | 1.4 | 1.3 | 1.2 | 1.1 | 1.2 | 1.1 |
| ***A3*** | 0.9 | 1.1 | 1.0 | 1.1 | 1.1 | 1.3 | 1.2 | 1.1 | 1.0 | 1.1 |
| ***A4*** | 1.1 | 1.1 | 1.2 | 1.3 | 1.2 | 1.2 | 1.1 | 1.0 | 1.1 | 1.2 |
| *A5* | 0.8 | 0.75 | 0.65 | 0.75 | 0.8 | 0.9 | 1.0 | 1.1 | 1.1 | 1.2 |

Vor fi tratate atât situaţiile de tip „short selling”, cât şi situaţiile în care acest lucru este evitat. Aşa cum am menţionat în §4.2, dacă randamentul prognozat este ales în conformitate cu randamentele medii calculate , în general soluţia unei probleme de risc minim nu implică situaţia de tip „short selling” pentru activele care vor fi incluse în portofoliu.

La fiecare moment de timp, populaţia este constituită din *dim* indivizi. Fiecărui fenotip *y* (portofoliu) îi corespunde un cromozom definit de un şir cu *n-1* numere reale, însoţit de informaţia calitatea evaluată în termenii funcţiei obiectiv. Din punct de vedere al reprezentării, este utilizat un vector *x*, cu

unde *y* este un portofoliu:

pentru este fracţiunea investită în acţiunea *i*,

astfel încât şi

şi este valoarea funcţiei obiectiv asociată vectorului (calitatea în termenii funcţiei de evaluare).

Generarea populaţiei iniţiale este realizată aleator. Dacă este aleasă varianta în care nu este admisă o situaţie de tip „short selling”, sunt impuse constrângerile fiecărui individ *x*:

În caz contrar, fiecare alelă este generată aleator uniform între -1 şi 1. Pentru ca să fie în , trebuie ca .

Operatorii de variaţie sunt definiţi astfel. Recombinarea este aplicată cu o rată *pc* şi revine la aplicarea operatorului de recombinare aritmetică totală, cu ponderea *p*, celor perechi de cromozomi aleşi aleator din populaţia de părinţi (fiecare pereche de cromozomi ocupă poziţiile în multisetul părinţilor). Pot fi utilizate două variante:

1. aceeaşi pereche de cromozomi poate fi generată de mai multe ori, respectiv
2. orice pereche de cromozomi aleşi ca părinţi este unică.

Dacă o pereche de cromozomi nu generează progenituri prin recombinare aritmetică (aleator a fost generat un număr sub rata de încrucişare), atunci are loc recombinarea asexuată.

**Observaţie**. Prin aplicarea operatorului de recombinare aritmetică totală, dacă doi indivizi respectă constrângerile (4.28), atunci şi progeniturile lor îndeplinesc (4.28). Această observaţie este importantă în contextul excluderii situaţiei de tip „short selling”.

Operatorul de mutaţie ales este cel de tip fluaj, cu o probabilitate de mutaţie mică şi cu de asemenea foarte mic (aici a fost ales ). Operatorul este implementat în două variante.

Varianta în care este exclusă situaţia de tip „short selling”. Cu o rată *pm*, fiecare genă din fiecare cromozom suferă o mutaţie prin adăugarea unei valori generate din repartiţia normală de medie 0 şi varianţă , dar astfel încât cromozomul rezultat să corespundă unei soluţii fezabile (să fie îndeplinite condiţiile 4.28). Dacă prin modificarea valorii unei gene, *a*, rezultă un cromozom, *x*, care nu aparţine setului soluţiilor fezabile, atunci *a* este ajustată astfel încât *x* să îndeplinească 4.28: dacă *a* rezultă negativă, atunci *a* este setată pe valoarea 0, respectiv dacă suma valorilor genelor lui *x*, *sv*, este supraunitară, *a* este modificată astfel încât *sv* să devină 1 (implicit suma investită în ultima acţiune a portofoliului devine 0).

Varianta generală: cu o rată *pm*, fiecare genă din fiecare cromozom suferă o mutaţie prin adăugarea unei valori generate din repartiţia normală de medie 0 şi varianţă . Este impusă o restricţie prin care valoarea fiecărei alele să fie în intervalul deci, în modul, să fie subunitară, pentru a menţine semnificaţia fiecărei valori de fracţiune investită în acţiunea corespunzătoare. De asemenea, pentru ca să fie în , trebuie ca .

Selecţia părinţilor este realizată pe baza algoritmului SUS, prin utilizarea distribuţiei de probabilitate de selecţie de tip rang liniar, cu presiunea de selecţie *s*. La fiecare generaţie sunt selectaţi *dim* părinţi. Mecanismul de supravieţuire este implementat astfel: generaţia următoare este dată de multisetul progeniturilor, eventual mutante, dar în care cel mai bun individ al generaţiei curente, *bx*, înlocuieşte cel mai slab cromozom copil, dacă nici unul dintre indivizii generaţi în urma aplicării operatorilor de variaţie nu are calitatea cel puţin egală cu cea a lui *bx*.

Este prezentată în continuare implementarea algoritmului în cea de a doua variantă de generare a unei perechi de cromozomi din bazinul de recombinare (orice pereche de cromozomi aleşi ca părinţi este unică). Pentru prima variantă, în funcţia *crossover* sunt generate aleator perechi de cromozomi, pentru care se impune doar condiţia ca cei doi cromozomi să ocupe poziţii diferite în multisetul părinţilor.

function [R]=citeste\_date(nume);

R=load(nume);

end

function [Q,rmed,alpha,B]=parametri(R);

%n actiuni observate in m saptamani

[n,m]=size(R);

rmed=zeros(n,1);

for i=1:n

rmed(i)=mean(R(i,:));

end;

Q=cov(R')\*(m-1)/m;

In1=eye(n-1);

ul=-ones(1,n-1);

B=[In1; ul];

alpha=zeros(n,1);

alpha(n)=1;

end

function [val,V]=fobiectiv(Q,rmed,alpha,B,ro,Rp,x);

val=((alpha+B\*x).')\*Q\*(alpha+B\*x)+(ro/(Rp^2))\*((rmed.')\*

alpha-Rp+(rmed.')\*B\*x)^2;

% disp(val);

V=((alpha+B\*x).')\*Q\*(alpha+B\*x);

% disp(V);

end

function [pop]=gen\_ini(dim,n,Q,rmed,alpha,B,ro,Rp);

pop=zeros(n,dim);

for i=1:dim

x=zeros(n-1,1);

for j=1:n-1

gata=0;

while(~gata)

r=unifrnd(0,1);

if(sum(x)+r<=1)

x(j)=r;

gata=1;

end;

end;

if(sum(x)==1)

break;

end;

end;

pop(1:n-1,i)=x(1:n-1);

[val,V]=fobiectiv(Q,rmed,alpha,B,ro,Rp,x(1:n-1));

pop(n,i)=-val;

end;

end

function [pop]=gen\_ini\_short\_selling(dim,n,Q,rmed,alpha,B,ro,Rp);

pop=zeros(n,dim);

for i=1:dim

x=zeros(n-1,1);

gata=0;

while(~gata)

for j=1:n-1

x(j)=unifrnd(-1,1);

end;

if((sum(x)>=0)&&(sum(x)<=2))

pop(1:n-1,i)=x(1:n-1);

[val,V]=fobiectiv(Q,rmed,alpha,B,ro,Rp,x(1:n-1));

pop(n,i)=-val;

gata=1;

end;

end;

end;

end

function [parinti]=selectie\_SUS\_rang\_l(pop,s);

[n,dim]=size(pop);

p=rang\_l(pop,s);

q=zeros(dim,1);

for i=1:dim

q(i)=sum(p(1:i));

end;

parinti=zeros(n,dim);

i=1;k=1;r=unifrnd(0,1/dim);

while(k<=dim)

while(r<=q(i))

parinti(1:n,k)=pop(1:n,i);

r=r+1/dim;

k=k+1;

end;

i=i+1;

end;

end

function [p]=rang\_l(pop,s);

[n,dim]=size(pop);

p=zeros(1,dim);

for i=1:dim

p(i)=(2-s)/dim+(2\*i\*(s-1)/(dim\*(dim+1)));

end;

end

function [popN]=crossover1(pop,pc,p,Q,rmed,alpha,B,ro,Rp);

[n,dim]=size(pop);

poz=[];

popN=zeros(n,dim);

for i=1:2:dim

ok=0;

while(~ok)

p1=unidrnd(dim-1);

p2=p1;

while(p1>=p2)

p2=unidrnd(dim);

end;

if(~ismember([p1 p2],poz,'rows'))

poz=[poz;[p1 p2]];

ok=1;

end;

end;

% nu pot fi generate aceleasi perechi de parinti

% la treceri diferite

x=pop(1:n-1,p1);

y=pop(1:n-1,p2);

r=unifrnd(0,1);

if(r<=pc)

x1=p\*x+(1-p)\*y;

popN(1:n-1,i)=x1(1:n-1);

[val,V]=fobiectiv(Q,rmed,alpha,B,ro,Rp,x1(1:n-1));

popN(n,i)=-val;

y1=p\*y+(1-p)\*x;

popN(1:n-1,i+1)=y1(1:n-1);

[val,V]=fobiectiv(Q,rmed,alpha,B,ro,Rp,y1(1:n-1));

popN(n,i+1)=-val;

else

popN(1:n,i)=pop(1:n,p1);

popN(1:n,i+1)=pop(1:n,p2);

end;

end;

end

function [popN]=mutatie(pop,pm,t,Q,rmed,alpha,B,ro,Rp,caz);

% mutatia intr-un cromozom, la nivel de gena revine la

% aplicarea operatiei de fluaj

[n,dim]=size(pop);

popN=pop;

for i=1:dim

efectuat=0;

if(caz==0)

% fara short selling

for j=1:n-1

r=unifrnd(0,1);

if(r<=pm)

fluaj=normrnd(0,t/3);

a=pop(j,i)+fluaj;

efectuat=1;

if(a<0)

a=0;

end;

s=sum(popN(1:n-1,i));

if(s+fluaj>1)

xx=s+fluaj-1;

a=a-xx;

end;

popN(j,i)=a;

end;

end;

else

% cu short selling

for j=1:n-1

r=unifrnd(0,1);

if(r<=pm)

gata=0;

while(~gata)

fluaj=normrnd(0,t/3);

if(fluaj>0)

a=min([pop(j,i)+fluaj 1]);

else

a=max([pop(j,i)+fluaj -1]);

end;

s=sum(popN(1:n-1,i));

if((s+fluaj>=0)&&(s+fluaj<=2))

popN(j,i)=a;

efectuat=1;

gata=1;

end;

end;

end;

end;

if(efectuat)

[val,V]=fobiectiv(Q,rmed,alpha,B,ro,Rp,popN(1:n-1,i));

popN(n,i)=-val;

end;

end;

end;

end

function [rezultat]=selectie\_generatie\_urmatoare(pop,popN);

[n,dim]=size(pop);

fob=pop(n,1:dim);

fobN=popN(n,1:dim);

rezultat=popN;

[max1,i]=max(fob);

[max2,j]=max(fobN);

if(max1>max2)

[min1,k]=min(fobN);

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

end;

end

function [x]=GA\_riscmin1(nume,dim,ro,Rp,s,pc,pm,p,t,NM,caz);

% nume=numele fisierului din care sunt prelute datele

% dim=dimensiunea populatiei, numar par

% ro=din functia obiectiv

% Rp=randamentul prognozat

% s=constrangerea de selectie pentru distributia de selectie

% rang liniar

% pc=probabilitatea de recombinare

% p=ponderea la recombinarea prin medie

% pm=probabilitatea de mutatie

% t=pragul de la fluaj

% NM=numarul de iteratii

% caz=0, pentru exemplul1, fara short selling

% caz=1, pentru exemplul2, cu short selling

% GA\_riscmin1('portofoliu1.txt',500,100,1.15,1.8,0.8,0.01,0.7,

0.1,20,0);

% GA\_riscmin1('',500,10,0.25,1.8,0.8,0.02,0.7,0.1,20,1);

if(caz==0)

R=citeste\_date(nume);

[n,m]=size(R);

[Q,rmed,alpha,B]=parametri(R);

pop=gen\_ini(dim,n,Q,rmed,alpha,B,ro,Rp);

else

n=5;

In1=eye(n-1);

ul=-ones(1,n-1);

B=[In1; ul];

alpha=zeros(n,1);

alpha(n)=1;

rmed=[-0.028, 0.366, 0.231, -0.24, 0.535]';

Q=[1.0256 -0.4340 0.0202 -0.1968 -0.0311;

-0.4340 1.1049 -0.0783 0.2347 -0.1776;

0.0202 -0.0783 0.4328 -0.1236 -0.1895;

-0.1968 0.2347 -0.1236 8.0762 1.0093;

-0.0311 -0.1776 -0.1895 1.0093 2.9007];

pop=gen\_ini\_short\_selling(dim,n,Q,rmed,alpha,B,ro,Rp);

end;

pop1=sortrows(pop.',n);

pop=pop1.';

T=[];

T1=[];

for nrit=1:NM

% populatia sortata crescator dupa -fobiectiv

[parinti]=selectie\_SUS\_rang\_l(pop,s);

[popN]=crossover1(parinti,pc,p,Q,rmed,alpha,B,ro,Rp);

% operatia de mutatie este realizata functie de caz –

% cu sau fara short selling

[popNou]=mutatie(popN,pm,t,Q,rmed,alpha,B,ro,Rp,caz);

[rezultat]=selectie\_generatie\_urmatoare(pop,popNou);

pop1=sortrows(rezultat.',n);

pop=pop1.';

[val,V]=fobiectiv(Q,rmed,alpha,B,ro,Rp,pop(1:n-1,dim));

T=[T V];

T1=[T1 val];

end;

disp(['riscul minim:' num2str(T(NM))]);

disp(['minimul functiei obiectiv MINRISC1:' num2str(T1(NM))]);

disp('Fractiunile investite');

disp([pop(1:n-1,dim);1-sum(pop(1:n-1,dim))]);

disp('Randamentul obtinut:');

Rr=(rmed.')\*(alpha+B\*pop(1:n-1,dim));

disp(Rr);

figure

i=1:NM;

plot(i,T1(i),'-rs','LineWidth',2,...

'MarkerEdgeColor','k',...

'MarkerFaceColor','y',...

'MarkerSize',8);

figure

i=1:NM;

plot(i,T(i),'-rs','LineWidth',2,...

'MarkerEdgeColor','k',...

'MarkerFaceColor','m',...

'MarkerSize',8);

end

**Variantă de implementare cu excluderea situaţiei de tip„s*hort selling*”pentru datele din tabelul 4.1**

Rezultatele care pot fi obţinute la apelul

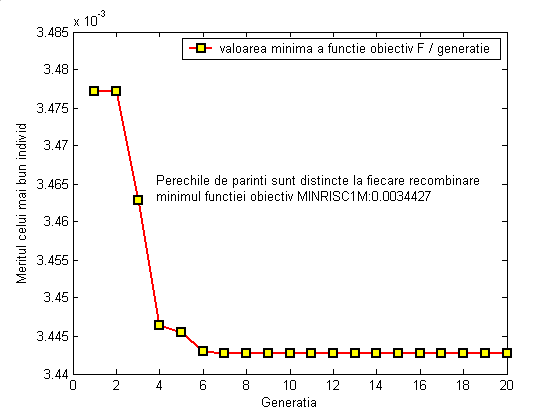
GA\_riscmin1('portofoliu1.txt',500,100,1.15,1.8,0.8,0.01,0.7,0.1,20,0)

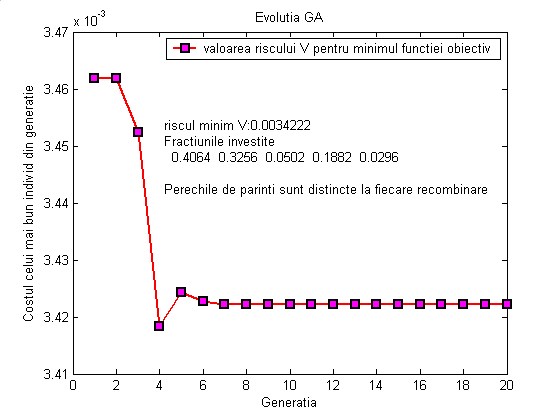
sunt figurate mai jos.

În acest exemplu, minimul funcţiei obiectiv corespunzătoare problemei RISCMIN1M este , riscul calculat după 20 de generaţii este

randamentul este

şi portofoliul obţinut:





**Variantă de implementare, cu considerarea situaţiei de tip „*short selling*” pentru datele din exemplul 2**

Rezultatele care pot fi obţinute la apelul

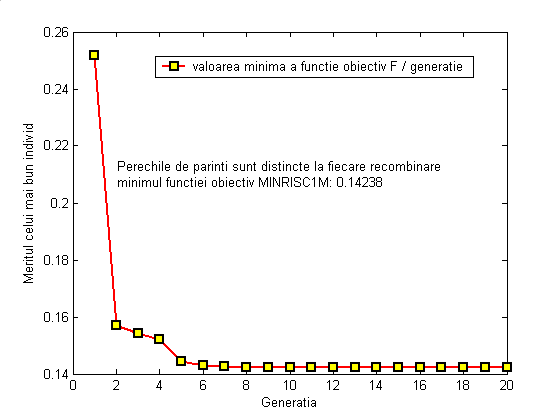
GA\_riscmin1('',500,10,0.25,1.8,0.8,0.02,0.7,0.1,20,1);

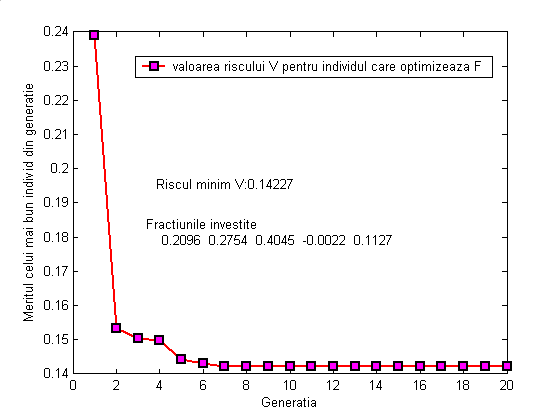
sunt figurate mai jos.

În acest exemplu, minimul funcţiei obiectiv corespunzătoare problemei RISCMIN1M este 0.14238, riscul calculat după 20 de generaţii este

randamentul este

şi portofoliul obţinut:

****

****