**Strategii evolutive**

**(Evolution Strategies)**

1. **Introducere**

Strategiile evolutive (ES) constituie clasa algoritmilor evolutivi (EA) în care poate fi ilustrată una dintre cele mai utile caracteristici ale calculului evolutiv, și anume auto-adaptarea strategiei de selectare a parametrilor. În general, auto-adaptarea revine la faptul că o serie de parametri ai EA variază într-o manieră specifică pe parcursul evoluției EA. Din punct de vedere al reprezentării, acei parametri ale căror valori variază sunt incluși în cromozomi ca atribute ale acestora și evoluează împreună cu indivizii populației respective. Caracteristica de auto-adaptabilitate este moștenită de majoritatea procedurilor ES și, pe parcursul ultimilor ani, o serie de alți algoritmi EA au adoptat-o.

Schematic, un algoritm de tip ES este caracterizat astfel

|  |  |
| --- | --- |
| Reprezentarea | Vectori cu componente numere reale |
| Recombinarea | Discretă sau intermediară (aritmetică) |
| Mutația | Perturbare gaussiană (mutație neuniformă) |
| Selecția părinților | Aleatoare, conform distribuției uniforme de probabilitate |
| Selecția supraviețuitorilor | De tipul sau |
| Caracteristică | Auto-adaptarea dimensiunii (pragului) mutației |

Principalele caracteristici ale ES sunt

* reprezentarea cromozomială tipică este prin vectori cu numere din mulțimea ;
* mutația este principalul operator de variație care produce indivizi noi;
* mutația este implementată prin considerarea aditivă a unuei componente zgomot gaussian, de obicei ;
* parametrul este modificat pe parcursul evoluției algoritmului.

*Algoritmul primar, numit strategia evolutivă cu doi membri* (ES2M), este descrisă astfel. Fie funcția obiectiv pentru care există . Problema este de a determina o aproximare

Algoritmul calculează o secvență de termeni, ultimul dintre ei fiind considerat astfel.

**ES2M**

Pas1. ; determină (aleator) un punct inițial

Pas2. Cât timp not(Condiție\_Terminală) execută

2.1. generează aleator din distribuția normală ,

2.2. calculează ,

2.3. dacă

altfel

2.4.

Fiind dată o soluție curentă , este creat un nou punct prin adaugarea la fiecare componentă a câte unui număr generat normal din distribuția . Noul vector calculat devine următorul termen al secvenței, , dacă îmbunătățește valoarea funcției obiectiv. Similar ideilor prezentate în secțiunea 3.3, este utilizată distribuția de probabilitate gaussiană deoarece este simetrică în raport cu media (în acest caz 0) și probabilitatea de a extrage un număr cu o anumită magnitudine a parametrului este funcție rapid descrescătoare de . În algoritmul ES2M parametrul controlează magnitudinea cu care este modificată fiecare componentă prin aplicarea operatorului de mutație neuniformă (fluaj) și, din acest motiv, este de obicei referită drept *dimensiunea pasului de mutație*. Rezultatele studiilor teoretice (Rechenberg, 1973; Reeves, 2002) recomandă o procedură de modificare a dimensiunii pasului de mutație prin bine-cunoscuta regulă de *actualizare de tip auto-adaptare la* *rata de succes* : dacă rata de succes este mai mare decât , atunci valoarea lui este mărită, fiind considerat un spațiu de căutare lărgit, altfel, dacă raportul este inferior lui , valoarea lui este micșorată, căutarea fiind concentrată în jurul punctului curent. Regula este aplicată la intervale egale, de exemplu, după fiecare *k* iterații, astfel. Fie frecvența relativă a succesului efectuării mutației măsurată pe ultimele *k* perioade de timp și *c* un parametru dat, în general (Eibden, 2003).

**Observație**. O operație de mutație este efectuată cu succes dacă, în urma aplicării ei, este îmbunătățită calitatea funcției obiectiv (punctul rezultat are fitness mai mare decât cel pe baza căruia a fost obținut).

Regula de actualizare a parametrului este definită prin

**Exemplu** Implementarea algoritmului ES2M cu regulă de actualizare de tip auto-adaptare cu rata de succes pentru funcția Ackley.

Funcția Ackley este definită prin

Scopul este de a calcula minimul funcției.

Funcția definită prin (2) are valoarea minimă 0, atinsă în punctul . Deoarece funcția Ackley are multe puncte de minim, problema minimizării acestei funcții teoretice susscită interes datorită posibilității studierii diferiților operatori de variație și selecție în scopul alegerii celei mai bune variante (care să conducă la optimul global în cele mai multe situații).

function [y ] = EMS2M\_Ackley(n,a,b,sigma,k,c,MAX)

%% EXEMPLU DE APEL

% y = EMS2M\_Ackley(2,-2,2,0.2,30,0.83,250000);

% y = EMS2M\_Ackley(20,-20,20,0.5,50,0.83,250000);

%% IMPLEMENTARE

% solutia initiala

x=unifrnd(a,b,1,n);y=x;

t=1;

nr=1;ps=0;

a1=a;b1=b;

%construieste secventa de MAX puncte

while t<=MAX

% aplica dupa k pasi regula de actualizare 1/5 a lui sigma

y=mutatie\_gauss(x,n,sigma);

if Ackley(y,n)<=Ackley(x,n)

x=y;

ps=ps+1;

end;

nr=nr+1;

if nr==k

if ps/k>.2

sigma=sigma/c;

else

if ps/k<.2

sigma=sigma\*c;

end;

end;

ps=0;

nr=0;

end;

t=t+1;

a1=min([x a1]);

b1=max([x b1]);

end;

if n==2

plot\_Ackley\_2D(a1,b1,y);

end;

disp(y);

end

function [y] =Ackley(x,n)

% functia de minimizat

sp=0;

sc=0;

for i=1:n

sp=sp+x(i)^2;

sc=sc+cos(2\*pi\*x(i));

end;

y=-20\*exp(-0.2\*sqrt(sp/n))-exp(sc/n)+20+exp(1);

end

function y=mutatie\_gauss(x,n,sigma)

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

% de fluaj

z=normrnd(0,sigma,1,n);

y=x+z;

end

function []=plot\_Ackley\_2D(a,b,y)

figure

[X,Y] = meshgrid([a:0.1:b]);

Z=-20\*exp(-0.2\*sqrt((X.^2+Y.^2)/2))-exp((cos(2\*X.\*pi)+cos(2\*Y.\*pi))/2)+20+exp(1);

surf(X,Y,Z);

hold on;

plot3(y(1),y(2),Ackley(y,2),'s','MarkerFace','y','MarkerEdge','k','MarkerSize',15);

colormap hsv

end

În tabelul următor sunt prezentate câteva variante de soluție obținută prin aplicarea ES2M pentru . Componentele vectorului soluție sunt figurate ca abateri față de optimul global . Soluția inițială este generată aleator, pentru , conform distribuției uniforme. Sunt construiți 250.000 de termeni din șir. Dimensiunea inițială a pasului de mutație este 0.5. Numărul de iterații la care este aplicată regula ratei de succes de este 50. Într-o măsură covârșitoare, este obținut un punct de optim local, valoarea minimă a funcției fiind în jurul valorii 19.9.

|  |  |
| --- | --- |
| Valoarea optimă calculată | Vectorul soluție – optim local. |
| 19.9480 |  |
| 19.9094 |  |
| 19.9482 |  |

În figurile următoare sunt prezentate două variante de soluție furnizate prin aplicarea algoritmului ES2M pentru situația . Soluția inițială este generată aleator, pentru , conform distribuției uniforme. Sunt construiți 250.000 de termeni din șir. Dimensiunea inițială a pasului de mutație este 0.5. Numărul de iterații la care este aplicată regula ratei de succes de este 30. .În cele mai multe situații este furnizat un punct de optim local.

În fiecare caz punctul soluție este figurat printr-un pătrat plasat pe suprafața corespunzătoare graficului funcției.



Valoarea minimă calculată 6.882 – optim local; x=( 0.0000 -2.9646)

****

Valoarea minimă calculată 4.4409e-015 – optim global

x=( 1.0e-015 \*-0.4444 1.0e-015 \*0.4345)

*Strategiile evolutive pot fi exprimate în termenii generali EA* (vezi §2.1) utilizând componentele prezentate în cele ce urmează.

**2. Reprezentarea în spațiul genotipurilor**

Strategiile evolutive sunt utilzate de obiecei în situația în care parametrii problemei de optimizare aparțin unui spațiu continuu, în general funcția obiectiv fiind definită pe cu valori în Dacă nu este luată în calcul componenta de auto-adaptare, codificarea standard este realizată prin vectori cu componente numere reale, spațiul genotipurilor fiind identic cu spațiul fenotipurilor, . Procedurile EA dezvoltate în ultima perioadă utilizează scheme de auto-adaptare mai complexe comparativ cu actualizarea la rata de succes, în conscință, pentru un fenotip , vectorul este doar o parte din genotip, fiecare individ conținând în plus valori ale parametrilor ES, în particular parametri care caracterizează operatorul mutație. Parametrii ES sunt de tip și de tip . Valorile parametrilor reprezintă dimensiunile pasul la implementarea operatorului de mutație, numărul lor, fiind în general 1 sau n. Variabilele sunt utilizate doar în cazul operatorilor de mutație corelată și reflectă interacțiunea dintre dimensiunile fiecărui pas de mutație. În general numărul parametrilor , este definit în termenii variabilei prin

În consecință, forma generală a unui individ din spațiul genotipurilor este descris prin vectorul cu componente,

**3. Mutația**

Operația de mutație utilizată de procedurile ES este cea neuniformă, utilizând distribuția de probabilitate gaussiană *în manieră de auo-adaptare*. Pentru început este considerat un singur parametru al dimensiunii pasului de fluaj. Fie funcția obiectiv, soluția curentă și reprezentarea cromozomială a lui . Genotipul rezultat al mutației este este calculat în două etape, astfel

* modifică dimensiunea pasului de fluaj și obține
* pentru fiecare , cea de-a *i*-a componentă este actualizată independent față de celelalte, prin adăugarea unei valori generată din distribuția normală de medie 0 și deviație standard (parametru calculat la prima etapă)

unde este generat aleator din distribuția

În situația utilizării mai multor parametrii de tip , mutația revine la modificarea valorilor lor în manieră necorelată sau corelată și apoi aplicarea mutației fluaj cu pașii astfel rezultați, conform schemei de mai sus.

**Observații**.

1. În cazul utilizării mutației auto-adaptive, dimensiunile pașilor de mutație nu sunt definite de utilizatori, ci evoluează împreună cu soluția pe care o însoțesc în reprezentarea cromozomială. În acest scop, prima etapă este cea de modificare a parametrilor tip și apoi este aplicată mutația la nivelor componentelor soluției curente. Rațiunea aplicării unui astfel de procedeu este următoarea (Eiben, Smith, 2002). Individul nou creat, , este evaluat de două ori: direct, prin calculul viabilității sale prin evaluarea valorii (la momentul selecției supraviețuitorilor) și indirect, din punct de vedere al abilității de a crea urmași buni. O nouă valoare este evaluată ca fiind mai bună decât valoarea inițială dacă urmașii obținuți pe baza ei sunt mai buni, adică dacă . Mai mult, *dimensiunea pasului de fluaj este viabilă dacă , progenitura rezultată în urma aplicării mutației caracterizate de , este selectat în generația următoare, deci îndeplinește și un criteriu global de calitate* (la nivel de populație). În consecință, un genotip este considerat bun dacă, în același timp, și și sunt buni (o valoare este bună dacă produce urmași buni, fiind unul dintre aceștia).
2. Utilizarea valorilor diferite ale dimensiunii pragului de fluaj la diferite momente de timp ale evoluției ES are la bază ideea că, în circumstanțe diferite, algoritmul se comportă diferit. Auto-adaptarea poate fi, pe de o parte, tratată ca *un mecanism care ajustează strategia de mutație în funcție de evoluția algoritmului de căutare*. Alternativ, în spațiul soluțiilor problemei o formă particulară a vecinătății individului curent măsurată în termenii funcției fitness din acea vecinătate poate defini calitatea operației de mutație, și anume cea mai bună variantă de modificare a soluției curente este pe direcția pe care valorile funcției fitness cresc. Prin asignarea câte unei strategii de mutație particularizată fiecărui individ, care evoluează în termenii acelei strategii și împreună cu ea este deschisă posibilitatea *învățării și utilizarea unui operator de mutație potrivit fiecărei topologii locale* în parte.

**3.1. Mutația cu un singur parametru**

În situația în care mutația utilizează aceeași dimensiune a pasului de fluaj pentru toate variabilele din reprezentarea unui vector soluție , dacă este cromozomul curent, atunci rezultatul mutației este calculat prin

unde este generat aleator din distribuția

unde este generat aleator independent din distribuția , pentru fiecare componentă.

Relația (6.1) corespunde unei reguli de actualizare bazate pe *distribuția lognormală*.

**DE ADĂUGAT CÂTEVA OBSERVAȚII RELATIV LA ACEASTĂ DISTRIBUȚIE ȘI DE REVENIT ȘI AICI ȘI LA 3.2 CU CÂTEVA COMENTARII SUPLIMENTARE**

Relația (6.2) corespunde mutației neuniforme definite pe . Fiecare componentă este mutată *independent de celelalte*, prin generarea câte unei perturbări independente de *n* ori.

Pentru ca aplicarea relației (6.2) să fie efectivă, trebuie ca dimensiunea pasului de fluaj să fie superioară unui prag minim dat. Cu alte cuvinte, dacă în urma aplicării relației (6.1) rezultă sub un prag dat, , atunci setează noua dimensiune la valoarea acelui prag:

Parametrul este setat de utilizator și corespunde ratei de instruire (similar, de exemplu, algoritmilor de instruire dezvoltați pentru rețelele neuronale). Alegerea cea mai utilizată a lui este astfel încât să fie invers proporțional cu

unde simbolizează proporționalitatea directă.

În concluzie, algoritmul de mutație aplicat genotipului pentru calculul rezultatului este descris prin.

1. generează aleator din distribuția
2. calculează
3. if then
4. for i=1:n
   1. generează aleator din distribuția

Motivele utilizării regulii de actualizare (6.1) au la bază ideile expuse în (Back, 1996). Evident, în cele mai multe situații sunt preferate modficările de magnitudine mică în detrimentul celor de magnitudine mare, iar valorile parametrului de tip trebuie să fie pozitive. DE REVENIT!

**3.2. Mutația necorelată utilizând mai mulți parametri de tip**

Motivul utilizării mai multor parametri , în general câte unul pentru fiecare componentă a vectorului soluție, este intenția de a trata diferit mutația fiecărei componente. Modificarea dimeniunii fiecărui pas de mutație este realizată pe baza a doi parametri prestabiliți,

astfel. Fie genotipul curent. Genotipul mutat este calculat pe baza relațiilor

unde este generat aleator din distribuția (o singură valoare pentru toate componentele ) și, pentru fiecare , este generat aleator din distribuția (câte o valoare generată independent de celelalte pentru fiecare )

unde este generat aleator din distribuția (câte o valoare generată independent de celelalte pentru fiecare ).

Similar celor prezentate în §3.1, dacă în urma aplicării relației (10.1) rezultă sub un prag dat, , atunci setează noua dimensiune la valoarea acelui prag:

Motivele utilizării regulii de actualizare (10.1) au la bază ideile expuse în (Ebden, Smith, 2002). DE REVENIT!

În concluzie, algoritmul de mutație aplicat genotipului pentru calculul rezultatului este descris prin.

1. generează aleator din distribuția
2. for i=1:n
   1. generează aleator din distribuția
   2. calculează
   3. if then
3. for i=1:n
   1. generează aleator din distribuția

3.2

**3.3. Mutația corelată utilizând mai mulți parametri de tip**

**4. Selecția părinților**

În cadrul procedurilor ES, selecția părinților nu este influențată de calitatea indivizilor (valorile funcției de fitness), fiecare populație la momentul curent *i* fiind considerată *populație părinte*. Pentru implementarea operatorului de recombinare, părinții sunt aleși aleator uniform din populația curentă (vezi schema generală de recombinare prezentată în §5).

**5. Recombinarea**

Operația de recombinare primară definită pentru dezvoltarea unui algoritm de tip ES implică utilizarea a doi cromozomi părinți pentru crearea unui individ copil. Evident, obținerea unui multiset cu progenituri revine la aplicarea operației de recombinare de ori. Sunt utilizate două modele de recombinare, și anume *recombinarea discretă* *și recombinarea indirectă* (*mediată* sau *prin medie*). În recombinarea discretă fiecare alelă a cromozomului copil este construită prin selecția aleatoare uniformă a alelei corespunzătoare aparținând unuia dintre părinți. Recombinarea intermediară presupune constucția unei progenituri în care fiecare alelă este construită ca medie aritmetică a alelelor corespunzătoare din cromozomii părinte. Cu alte cuvinte, dacă și sunt cromozomii părinte, progenitura *c* este calculată prin,

pentru fiecare

Una dintre cele mai utilizate proceduri de recombinare în cadrul algotimilor ES este o extensie a schemei primare de recombinare *prin utilizarea câte unei perechi de părinți pentru fiecare genă* (Ebden, Smith, 2002). O astfel de procedură este numită *recombinare globală* și presupune, pentru construcția fiecărei alele corespunzătoare genei , alegerea aleatoare uniform din populația curentă , de dimensiune , a câte unei perechi de părinți și efectuarea recombinării discrete sau indirecte a alelelor din pozția *i*, conform următoarei scheme, recombinare\_globală().

for i=1:n

* generează aleator două poziții și consideră cromozomii populației curente din pozițiile
* calculează

Schema generală utilizată pentru generarea multisetului de copii, , pe baza populației curente cu indivizi revine la apelul de ori a procedurii recombinare\_globală() și colectarea cromozomilor rezultați în .

**Observații**

1. Algoitmii din clasa ES, în majoritatea lor, utilizează scheme de recombinare globală. La nivel de genă, tipul de recombinare (discretă sau indirectă) este specific clasei din care face parte gena, și anume dacă aparține zonei vectorului soluție sau zonei vectorului parametrilor (de tip sau ), cele două zone fiind tratate în cele mai multe situații diferit. O alegere comună este aceea în *care alelele zonei soluție sunt recombinate în manieră discretă*, în timp ce, *pentru* *construcția alelelor din zona parametrilor este utilizată recombinarea indirectă*. Intuitiv alegerea este justificată astfel. *Prin recombinare discretă schema de generare a copiilor păstrează diversitatea în spațiul fenotipurilor (corespunzător soluțiilor), în timp ce recombinarea prin medie asigură o strategie precaută de adaptare a parametrilor*.

2. Menținerea caracterului auto-adaptiv al unui algorim ES este asigurată, printre altele, prin generarea unui multiset de copii mai cu dimensiune mai mare decât dimensiunea populației curente pe baza căreia a fost generat.

**6. Selecția supraviețuitorilor**

În continuare vom presupune că populația curentă conține indivizi, iar multisetul copiilor căruia i s-a aplicat procedura de mutație conține copii. Algoritmii din clasa ES utilizează un *mecanism strict determinist* pentru selecția generației următoare, fiind aleși indivizi prin utilizarea uneia din următoarele scheme:

* selecția , care revine la alegerea celor mai buni copii
* selecția , care revine la alegerea celor mai buni indivizi considerând atât membrii multisetului de copii cât și pe cei ai populației curente.

În cele mai multe situații, în special în cazul funcțiilor multimodale cu multe puncte de optim local este preferată selecția , deoarece, la fiecare moment de timp, populația curentă este complet modificată, în acest fel putând fi în principiu depășite punctele de optim local (pentru calculul optimului global). În plus strategia poate împiedica buna funcționare a mecanismului de auto-adaptare, deoarece cromozomi ce conțin parametri puțin viabili pot supraviețui pe parcursul a mai multor generații.

**Observație** În scopul implementării unui mecansim de auto-adaptare efectiv și funcțional în situații generale, trebuie asigurată îndeplinirea următoarelor condiții (Ebden, Smith, 2002)

1. – pentru utilizarea mai multor strategii de mutație;
2. - în general , pentru obținerea unei presiuni de selecție scazute și să fie evitată astfel convergența prea rapidă;
3. folosirea selecției – pentru părăsirea zonelor de optim local, în vederea evoluției către optimul global;
4. utilizarea recombinării indirecte pentru zona parametrică a cromozomilor.

**Exemplu**

Proiectarea și implementarea unui ES pentru minimizarea funcției Ackley. Fie *n* dimensiunea spațiului pe care este definită funcția. Este utilizată mutația necorelată cu câte un parametru pentru fiecare componentă a vectorului soluție.

* Reprezentarea – pentru fiecare componentă a vectorului soluție este păstrată o dimensiune a pragului de fluaj: un cromozom este reprezentat prin
* Generarea populației inițiale – aleator; fiecare alelă din zona soluție este generată uniform în intervalul , în timp ce fiecare valoare a unui parametru este generată uniform în intervalul

function [P] = gen\_ini(dim,n,a,b)

P=zeros(dim,2\*n+1);

for i=1:dim

P(i,1:n)=unifrnd(a,b,1,n);

P(i,n+1:2\*n)=unifrnd(0.25,1,1,n);

P(i,2\*n+1)=Ackley(P(i,1:n),n);

end;

end

* Mutația - mutația necorelată cu mai mulți parametri

function [ MO ] = mutatie(O,tau,taup,eps,dim,n)

%% MUTATIA POPULATIEI DE COPII O

% dim indivizi, cu n (pentru x)+n (pentru sigma) +1 (functia de evaluare) elemente

%% IMPLEMENTARE

MO=O;

for i=1:dim

cromozom=O(i,1:2\*n);

x=cromozom(1:n);

sigma=cromozom(n+1:2\*n);

[x1,y1]=mutatie\_necorelata(x,n,sigma,taup,tau,eps);

cromozomr=[x1 y1];

MO(i,1:2\*n)=cromozomr;

MO(i,2\*n+1)=Ackley(x1,n);

end;

end

function [x1,sigma1]=mutatie\_necorelata(x,n,sigma,taup,tau,eps)

rtau=normrnd(0,tau);

rtaup=normrnd(0,taup,1,n);

sigma1=zeros(1,n);

x1=zeros(1,n);

for i=1:n

sigma1(i)=sigma(i)\*exp(rtau+rtaup(i));

if sigma1(i)<eps

sigma1(i)=eps;

end;

end;

for i=1:n

x1(i)=x(i)+normrnd(0,sigma1(i));

end;

end

* Recombinarea – globală, astfel: alelele zonei soluție sunt recombinate în manieră discretă, iar alelelor din zona parametrilor sunt generate prin recombinare indirectă.

function [O] = genereaza\_copii(P,dim,dimc,n)

O=zeros(dimc,2\*n+1);

for k=1:dimc

% Copil=[xr sigmar]

xr=zeros(1,n);

sigmar=zeros(1,n);

% partea solutie din cromozom - recombinarea globala discreta

%fiecare alela este calculata pe baza a doi parinti generati aleator

for i=1:n

poz=unidrnd(dim,1,2);

rez=unidrnd(2);

xr(i)=P(poz(rez),i);

end;

% partea parametrica din cromozom - recombinarea globala intermediara

%fiecare alela este calculata pe baza a doi parinti generati aleator

for i=1:n

poz=unidrnd(dim,1,2);

sigmar(i)=(P(poz(1),i)+P(poz(2),i))/2;

end;

Copil=[xr sigmar];

O(k,1:2\*n)=Copil;

O(k,2\*n+1)=Ackley(Copil(1:n),n);

end;

end

* Selecția generației următoare – de tip

function [ P] = generatie\_urmatoare(MO,dim,n)

MO\_S=sortrows(MO,2\*n+1);

P=MO\_S(1:dim,:);

end

* Funcția care implementează algoritmul ES

function [sol,val,P] = ES\_Ackley(n,a,b,dim,dimc,MAX,tau,taup,eps,eps1)

%% ES PENTRU MINIMIZAREA FUNCTIEI ACKLEY

%% MODUL DE ALEGERE A PARAMETRILOR

% tau direct prop cu 1/sqrt(2\*sqrt(n));

% taup direct prop cu 1/sqrt(2\*n);

% dimc aproximativ egal cu 7\*dim

%% IMPLEMENTARE

% generarea populatiei initiale

P=gen\_ini(dim,n,a,b);

it=1;

valoare=1;

% de MAX ori sau pana cand abs(valoare)<eps1

while it<=MAX && abs(valoare)>eps1

% evolutia ES pentru o generatie

% genereaza dimc copii

O=genereaza\_copii(P,dim,dimc,n);

% aplica mutatia

MO=mutatie(O,tau,taup,eps,dimc,n);

%calculeaza generatia urmatoare

P1=generatie\_urmatoare\_comma(MO,dim,n);

P=P1;

valoare=P1(1,2\*n+1);

it=it+1;

eps=eps/1.05;

end;

val=valoare;

sol=P(1,1:n);

disp(val);

disp(sol);

end

Parametrii algoritmului

* ,
* – parametru care controlează atingerea optimului global
* – parametru care controlează dimensiunile pașilor de mutație (la momentul inițial – vezi observația de mai jos)
* – numărul maxim de iterații ( evaluări ale funcției obiectiv)

La fiecare iteratie,eps scade conform , valoarea inițială fiind . În tabelul următor sunt prezentate valorile lui eps din 50 în 50 de iteratii.

|  |  |
| --- | --- |
| **Iteratia** | **eps** |
| 0 |  |
| 50 |  |
| 100 |  |
| 150 |  |
| 200 |  |
| 250 |  |
| 300 |  |

În marea majoritate a execuțiilor, eroarea este de ordinul .

Apelul este realizat conform următoarei secvențe.

n=20;

tau=0.5/sqrt(2\*sqrt(n));

taup=0.5/sqrt(2\*n);

MAX=300;

eps=1e-1;

[sol,val,P] = ES\_Ackley(20,-20,20,30,220,MAX,tau,taup,eps,1e-7);

În figura de mai jos este prezentat un exemplu de execuție. Valoarea funcției obiectiv este , în timp ce soluția variază în jurul vectorului optim conform graficului prezentat.



:

1. Back T., *Evolutionary Algoritms in Theory and Practice*, Oxford University Press, 1996
2. Rechemberg, I., *Evolutionstraterie: Optimierung Technisher Systeme nach Prinzipien des Biologishen Evolution*, Fromman-Hozlboog Verlag, 1973
3. Reeves C., Rowe, J., *Genetiv Algorithms: Principles and Perspectives*, Kluwer, Norwell MA, 2002