*дата :*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **НУЛП, ІКНІ, САП** | | Тема | оцінка | підпис |
| КНC-13 | 4 (номер лаб.) | Запрограмувати ГА для пошуку максимального і мінімального значення цільової функції |  |  |
| Скалецький З.О. | |
| № залікової: 1608404 | |
| Методи нечіткої логіки та еволюційні алгоритми при автоматизованому проектуванні | | Викладач: | |
| Кривий Р. З. | |

**Мета:** Ознайомитися з основними теоретичними відомостями за темою роботи. Вивчити роботу функції ga пакету Matlab.

**Теоретичні відомості**

* *Турнірна селекція*

Турнірна селекція - спочатку випадково вибирається встановлену кількість особин (зазвичай дві), а потім з них вибирається особина з кращим значенням функції пристосованості

* *Одноточечне схрещування*

Впорядковуючий оператор схрещування (Order crossover, OX) був запропонований Д. Девісом у 1985 році для негомологічних числових хромосом. Схрещування може виконуватися по одній або по двох точках. Точки схрещування вибираються випадково.

При одноточечному схрещуванні в хромосому першого нащадка копіюється хромосома першого батька, а потім гени нащадка, що розташовані правіше точки схрещування, перевпорядковуються у послідовність, що відповідає другому батькові. При цьому другий батько переглядається від початку до кінця, зліва направо, і елементи, яких не вистачає у нащадку, додаються, починаючи від точки схрещування, один по одному.

* Крок 1. Випадковим чином вибрати точку схрещування.
* Крок 2. Скопіювати в хромосому першого нащадка сегмент хромосоми першого батька, що розташований лівіше точки схрещування.
* Крок 3. Інші гени в нащадку копіюються із другого батька в упорядкованому вигляді зліва направо, крім елементів, які вже увійшли до нащадка.

Для створення другого нащадка застосовується аналогічний порядок дій.

* *Мутація*

У використаному операторі вибір точки мутації здійснюється випадково, а значення обраного біта міняється на протилежне.

**Індивідуальне завдання**

Варіант 18.Розробити програму, яка реалізовує генетичний алгоритм пошуку максимального і мінімального значення цільової функції - f(x) = a + bx + cx2 + dx3 в інтервалі x=[-10,53].



**Код програми**

Файл lab4max.m

clc;clear;close all;

nn=6; % кількість біт

ps=12; % чисельність популяції

ng=20; % Кількість поколінь

pm=0.05; % імовірність мутації

% Почати з випадкових x в межах х[-10,53]:

g=randi(64,1,ps)-11;

for gc=1:ng % цикл поколінь

%селекція турнірна 1111111111111

'нове покоління gc='

gc

gt=g

gs=0;%для кращого перегляду результату

gt(2,1:ps)=getY(g)

gt(3,:)=0;

for i=1:ps/2

test=1;

while test==1

ig=randi(ps);

test=gt(3,ig);

end

gt(3,ig)=1;

elem(1)=ig;

%2

test=1;

while test==1

ig=randi(ps);

test=gt(3,ig);

end

gt(3,ig)=1;

elem(2)=ig

%вибір кращого

if gt(2,elem(1))>gt(2,elem(2))

gs(i)=gt(1,elem(1));

else

gs(i)=gt(1,elem(2));

end

end

%схрещення 2222222222222222

gs

'одноточечне схрещення'

for psc=1:(ps/4)

ts=randi(nn-1)+1 %точка схрещення

i1=1+2\*(psc-1);

i2=2+2\*(psc-1);

gs(i1)+10;

gs(i2)+10;

p1=bitget(gs(i1)+10,6:-1:1) %перший батько в бітовому представленні

p2=bitget(gs(i2)+10,6:-1:1)%2 батько

ch10=gs(i1)+10;

for i=ts:6

i;

test\_p2i=p2(7-i);

ch10=bitset(ch10,i,p2(7-i)); % вставляємо біт

bitget(ch10,6:-1:1);

end

child=bitget(ch10,6:-1:1)%arhiv

gs(ps/2+i1)=ch10-10;

ts=randi(nn-1)+1

ch10=gs(i1)+10;

for i=ts:6

i;

test\_p2i=p2(7-i);

ch10=bitset(ch10,i,p2(7-i)); % вставляємо біт

bitget(ch10,6:-1:1);

end

child=bitget(ch10,6:-1:1)%arhiv

gs(ps/2+i2)=ch10-10;

end

g=gs

'батьки'

gs(1:ps/2)

'відповідні діти'

gs(ps/2+1:ps)

%мутація

for psc=1:ps

if rand<=pm

'мутація!!!'

'мутує елемент № psc'

psc

em=g(psc)+10

bitget(em,6:-1:1)

igen=randi(6)

gen=bitget(em,igen)

if gen==0

gen=1

else

gen=0

end

em=bitset(em,igen,gen)

g(psc)=em-10

end

end

%завершення

gt=g

gt(2,1:ps)=getY(g);

[maxFunc(gc) imax]=max(gt(2,:))

maxElem(gc)=gt(1,imax)

end

maxFunc

maxElem

x=-10:53;

y=getY(x);

maxY=max(y)

Файл lab4min.m

clc;clear;close all;

nn=6; % кількість біт

ps=12; % чисельність популяції

ng=30; % Кількість поколінь

pm=0.05; % імовірність мутації

% Почати з випадкових x в межах х[-10,53]:

g=randi(64,1,ps)-11;

for gc=1:ng % цикл поколінь

%селекція турнірна 1111111111111

'нове покоління gc='

gc

gt=g

gs=0;%для кращого перегляду результату

gt(2,1:ps)=getY(g)

gt(3,:)=0;

for i=1:ps/2

test=1;

while test==1

ig=randi(ps);

test=gt(3,ig);

end

gt(3,ig)=1;

elem(1)=ig;

%2

test=1;

while test==1

ig=randi(ps);

test=gt(3,ig);

end

gt(3,ig)=1;

elem(2)=ig

%вибір кращого

if gt(2,elem(1))<gt(2,elem(2))

gs(i)=gt(1,elem(1));

else

gs(i)=gt(1,elem(2));

end

end

%схрещення 2222222222222222

gs

'одноточечне схрещення'

for psc=1:(ps/4)

ts=randi(nn-1)+1 %точка схрещення

i1=1+2\*(psc-1);

i2=2+2\*(psc-1);

gs(i1)+10;

gs(i2)+10;

p1=bitget(gs(i1)+10,6:-1:1) %перший батько в бітовому представленні

p2=bitget(gs(i2)+10,6:-1:1)%2 батько

ch10=gs(i1)+10;

for i=ts:6

i;

test\_p2i=p2(7-i);

ch10=bitset(ch10,i,p2(7-i)); % вставляємо біт

bitget(ch10,6:-1:1);

end

child=bitget(ch10,6:-1:1)%arhiv

gs(ps/2+i1)=ch10-10;

ts=randi(nn-1)+1

ch10=gs(i1)+10;

for i=ts:6

i;

test\_p2i=p2(7-i);

ch10=bitset(ch10,i,p2(7-i)); % вставляємо біт

bitget(ch10,6:-1:1);

end

child=bitget(ch10,6:-1:1)%arhiv

gs(ps/2+i2)=ch10-10;

end

g=gs

'батьки'

gs(1:ps/2)

'відповідні діти'

gs(ps/2+1:ps)

%мутація

for psc=1:ps

if rand<=pm

'мутація!!!'

'мутує елемент № psc'

psc

em=g(psc)+10

bitget(em,6:-1:1)

igen=randi(6)

gen=bitget(em,igen)

if gen==0

gen=1

else

gen=0

end

em=bitset(em,igen,gen)

g(psc)=em-10

end

end

%завершення

gt=g

gt(2,1:ps)=getY(g);

[minFunc(gc) imin]=min(gt(2,:))

minElem(gc)=gt(1,imin)

end

minFunc

minElem

x=-10:53;

y=getY(x);

minY=min(y)

Файл getY.m

function y = getY( x )

y=26 + 6\*x - 93\*x.\*x + 2\*x.\*x.\*x;

end

**Результати виконання програми**

*Пошук мінімуму*

Мінімальне значення цільової функції виконане методом повного перебору:

minY =-29579

1)Розмір популяції=8, кількість поколінь=20, імовірність мутації=0.05:

minFunc = Columns 1 through 10

-29478 -29478 -29478 -29478 -29478 -29478 -29478 -29478 -29494 -29494

Columns 11 through 20

-29494 -29494 -29494 -29494 -29494 -29579 -29579 -29579 -29579 -29579

minElem = 32 32 32 32 32 32 32 32 30 30 30 30 30 30 30 31 31 31 31 31

2)Розмір популяції=20, кількість поколінь=20, імовірність мутації=0.05:

minFunc = Columns 1 through 10

-29494 -29494 -29494 -29494 -29494 -29579 -29579 -29579 -29579 -29579

Columns 11 through 20

-29579 -29579 -29579 -29579 -29579 -29579 -29579 -29579 -29579 -29579

minElem = 30 30 30 30 30 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31

3)Розмір популяції=12, кількість поколінь=20, імовірність мутації=0.05:

minFunc =

Columns 1 through 10

-27534 -29494 -27534 -27534 -27534 -29494 -29494 -29494 -29494 -29579

Columns 11 through 20

-29579 -29579 -29579 -29579 -29579 -29579 -29579 -29579 -29579 -29579

minElem = 26 30 26 26 26 30 30 30 30 31 31 31 31 31 31 31 31 31 31 31

Перші 2 цикли роботи програми

Розмір популяції=12, кількість поколінь=2, імовірність мутації=0.05

ans =

нове покоління gc=

gc = 1

gt = 0 23 1 -10 49 17 26 52 49 12 22 4

gt = Columns 1 through 10

0 23 1 -10 49 17 26 52 49 12

26 -24699 -59 -11334 12325 -16923 -27534 30082 12325 -9838

Columns 11 through 12

22 4

-23558 -1310

elem = 6 10

elem = 11 9

elem = 2 1

elem = 8 12

elem = 3 5

elem = 4 7

gs = 17 22 23 4 1 26

ans =одноточечне схрещення

ts = 3

p1 = 0 1 1 0 1 1

p2 = 1 0 0 0 0 0

child = 1 0 0 0 1 1

ts = 6

child = 1 1 1 0 1 1

ts = 4

p1 = 1 0 0 0 0 1

p2 = 0 0 1 1 1 0

child = 0 0 1 0 0 1

ts = 5

child = 0 0 0 0 0 1

ts = 4

p1 = 0 0 1 0 1 1

p2 = 1 0 0 1 0 0

child = 1 0 0 0 1 1

ts = 2

child = 1 0 0 1 0 1

g = 17 22 23 4 1 26 25 49 -1 -9 25 27

ans =батьки

ans = 17 22 23 4 1 26

ans =відповідні діти

ans = 25 49 -1 -9 25 27

gt = 17 22 23 4 1 26 25 49 -1 -9 25 27

minFunc = -28243

imin = 12

minElem = 27

ans =

нове покоління gc=

gc = 2

gt = 17 22 23 4 1 26 25 49 -1 -9 25 27

gt = Columns 1 through 10

17 22 23 4 1 26 25 49 -1 -9

-16923 -23558 -24699 -1310 -59 -27534 -26699 12325 -75 -9019

Columns 11 through 12

25 27

-26699 -28243

elem = 1 8

elem = 9 6

elem = 3 5

elem = 11 12

elem = 2 7

elem = 10 4

gs = 17 26 23 27 25 -9

ans =одноточечне схрещення

ts = 3

p1 = 0 1 1 0 1 1

p2 = 1 0 0 1 0 0

child = 1 0 0 1 1 1

ts = 2

child = 1 0 0 1 0 1

ts = 5

p1 = 1 0 0 0 0 1

p2 = 1 0 0 1 0 1

child = 1 0 0 0 0 1

ts = 4

child = 1 0 0 0 0 1

ts = 6

p1 = 1 0 0 0 1 1

p2 = 0 0 0 0 0 1

child = 0 0 0 0 1 1

ts = 5

child = 0 0 0 0 1 1

g = 17 26 23 27 25 -9 29 27 23 23 -7 -7

ans =батьки

ans = 17 26 23 27 25 -9

ans =відповідні діти

ans = 29 27 23 23 -7 -7

gt = 17 26 23 27 25 -9 29 27 23 23 -7 -7

minFunc = -28243 -29235

imin = 7

minElem = 27 29

minFunc = -28243 -29235

minElem = 27 29

ans =Мінімальне значення цільової функції виконане методом повного перебору

minY = -29579

*Пошук максимуму*

Максимальне значення цільової функції виконане методом повного перебору

maxY = 36861

1) Розмір популяції=8, кількість поколінь=20, імовірність мутації=0.05:

maxFunc = Columns 1 through 10

30082 30082 30082 30082 30082 30082 30082 30082 30082 30082

Columns 11 through 20

30082 36861 36861 36861 36861 36861 36861 36861 36861 36861

maxElem = 52 52 52 52 52 52 52 52 52 52 52 53 53 53 53 53 53 53 53 53

2) Розмір популяції=12, кількість поколінь=10, імовірність мутації=0.05:

maxFunc = 17826 17826 17826 17826 36861 36861 36861 36861 36861 36861

maxElem = 50 50 50 50 53 53 53 53 53 53

2) Розмір популяції=20, кількість поколінь=15, імовірність мутації=0.05:

maxFunc = Columns 1 through 10

23741 36861 36861 36861 36861 36861 36861 36861 36861 36861

Columns 11 through 15

36861 36861 36861 36861 36861

maxElem = 51 53 53 53 53 53 53 53 53 53 53 53 53 53 53

Перші 2 цикли роботи програми

Розмір популяції=8, кількість поколінь=2, імовірність мутації=0.05

ans =

нове покоління gc=

gc = 1

gt = 13 29 22 38 52 -6 37 39

gt =

13 29 22 38 52 -6 37 39

-11219 -29235 -23558 -24294 30082 -3790 -25763 -22555

elem = 2 3

elem = 7 5

elem = 1 4

elem = 6 8

gs = 22 52 13 -6

ans =одноточечне схрещення

ts = 2

p1 = 1 0 0 0 0 0

p2 = 1 1 1 1 1 0

child = 1 1 1 1 1 0

ts = 5

child = 1 1 0 0 0 0

ts = 5

p1 = 0 1 0 1 1 1

p2 = 0 0 0 1 0 0

child = 0 0 0 1 1 1

ts = 5

child = 0 0 0 1 1 1

g = 22 52 13 -6 52 38 -3 -3

ans =батьки

ans = 22 52 13 -6

ans =відповідні діти

ans = 52 38 -3 -3

ans =мутація!!!

ans =мутує елемент № psc

psc = 8

em = 7

ans = 0 0 0 1 1 1

igen = 3

gen = 1

gen = 0

em = 3

g = 22 52 13 -6 52 38 -3 -7

gt = 22 52 13 -6 52 38 -3 -7

maxFunc = 30082

imax = 2

maxElem = 52

ans =нове покоління gc=

gc = 2

gt = 22 52 13 -6 52 38 -3 -7

gt =

22 52 13 -6 52 38 -3 -7

-23558 30082 -11219 -3790 30082 -24294 -883 -5259

elem = 2 4

elem = 8 1

elem = 7 5

elem = 6 3

gs = 52 -7 52 13

ans =одноточечне схрещення

ts = 4

p1 = 1 1 1 1 1 0

p2 = 0 0 0 0 1 1

child = 0 0 0 1 1 0

ts = 4

child = 0 0 0 1 1 0

ts = 5

p1 = 1 1 1 1 1 0

p2 = 0 1 0 1 1 1

child = 0 1 1 1 1 0

ts = 2

child = 0 1 0 1 1 0

g = 52 -7 52 13 -4 -4 20 12

ans =батьки

ans = 52 -7 52 13

ans =відповідні діти

ans = -4 -4 20 12

ans =мутація!!!

ans =мутує елемент № psc

psc = 2

em = 3

ans = 0 0 0 0 1 1

igen = 1

gen = 1

gen = 0

em = 2

g = 52 -8 52 13 -4 -4 20 12

gt = 52 -8 52 13 -4 -4 20 12

maxFunc = 30082 30082

imax = 1

maxElem = 52 52

maxFunc = 30082 30082

maxElem = 52 52

ans =Максимальне значення цільової функції виконане методом повного перебору

maxY = 36861

**Висновок**

Виконуючи лабораторну роботу розроблено програму, яка реалізує генетичний алгоритм пошукумаксимального і мінімального значення цільової функції. В алгоритмі реалізовану одноточечне схрещування, турнірну селекцію, а також мутацію випадкового гена , при якому значення гена (в бітовому представленні) міняється на протилежне. За рахунок малого інтервалу для x, задавалася невелика початкова популяція, інакше мінімальне значення знаходилося на перших ітераціях і не мінялося.