|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **НУЛП, ІКНІ, САП** | | Тема | оцінка | підпис |
| КНС-13 | 7 | Оптимізація за допомогою еволюційних методів |  |  |
| Дробко Володимир | |
|  | |
| Методи нечіткої логіки та еволюційні алгоритми при автоматизованому проектуванні | | Викладач: | |
| Кривий Р.З. | |

**1.Мета**

Ознайомитися з основними теоретичними відомостями, розробити програму, яка реалізовує генетичний алгоритм пошуку максимального і мінімального значення цільової функції.

1. **Індивідуальне завдання**

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

**Варіант № 7**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **a** | **b** | **c** | **d** |
| 7 | 23 | -80 | -64 | 5 |

1. **Розрахунок максимальних і мінімальних значень цільової функції за допомогою класичної теорії оптимізації**

Необхідно знайти максимальні та мінімальні значень функція в інтервалі x = [-10, 53].

f(x) = 23-80x-64x2+5x3

**Розв’язання:**  
Знаходимо першу похідну від заданої функції:  
 f'(х) = 15x2-128x-80  
Розв’язуємо рівняння:   
 15x2-128x - 80 = 0  
 Корені рівняння:

x1 = -0.585  
 x2 = 9.118

Обчислимо значення функції на границях діапазону, та в коренях рівняння:   
 f(-0.585) = 0.01  
 f(9.118) = -0.8  
 f(-10) = 40  
 f(53) = 35271  
 Отже fmin = -0.8 при х=9.118, fmax = 35271 при х=53.  
Графік функції відображено на рис. 1.:

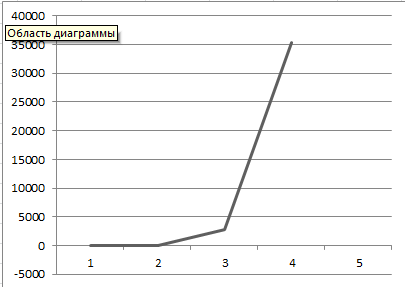


Рис. 1. Графік функції

1. **Програмна реалізація завдання**

**Код програми на C++ :**

Файл Lab4\_GA.cpp:

#include "stdafx.h"

#include <iostream>

#include <cstdlib>

#include <ctime>

#define FIRST -10

#define LAST 53

using namespace std;

const size\_t X = 10;

double fitness(double &n)

{

return 23 - 80 \* n - 64 \* pow(n,2) + 5 \* pow(n,3);

}

void selection\_max(double a[])

{

for (size\_t i = 0; i < X; ++i)

for (size\_t j = i + 1; j < X; ++j)

if (fitness(a[j]) < fitness(a[i]))

swap(a[i], a[j]);

copy(a + X / 2, a + X, a);

}

void selection\_min(double a[])

{

for (size\_t i = 0; i < X; ++i)

for (size\_t j = i + 1; j < X; ++j)

if (fitness(a[j]) > fitness(a[i]))

swap(a[i], a[j]);

copy(a + X / 2, a + X, a);

}

double mutation(double &t)

{

do { t += ((((double)(rand() % 2)) / 100 == 0.01) ? 0.1 : -0.01);

}

while (t == 0 || t<FIRST || t>LAST);

return t;

}

int main()

{

setlocale(LC\_ALL, "Russian");

srand(time(NULL));

double data\_base[X] = { 0 };

double data\_base2[X] = { 0 };

for (int rounds = 0; rounds < 500; rounds++)

{

for (size\_t i = 0; i < X; ++i)

data\_base[i] = mutation(data\_base[i]);

selection\_max(data\_base);

}

for (int rounds = 0; rounds < 500; rounds++)

{

for (size\_t i = 0; i < X; ++i)

data\_base2[i] = mutation(data\_base2[i]);

selection\_min(data\_base2);

}

double ymax = fitness(data\_base[4]);

double ymin = fitness(data\_base2[4]);

cout << "Пошук максимального значення функцiї" << endl;

cout << "Обчисленення [х,y] на всiх iтерацiях:" << endl;

for (size\_t i = 0; i < X; i++)

cout << "Iтерацiя №" << i+1 << ": x=" << data\_base[i] << ", y=" << fitness(data\_base[i]) << endl;

cout << "При x=" << data\_base[9] << endl;

cout << "y\_max=" << ymax;

cout << endl;

cout << endl;

cout << "Пошук мiнiмального значення функцiї" << endl;

cout << "Обчисленення [х,y] на всiх iтерацiях:" << endl;

for (size\_t i = 0; i < X; i++)

cout << "Iтерацiя №" << i+1 << ": x=" << data\_base2[i] << ", y=" << fitness(data\_base2[i]) << endl;

cout << "При x=" << data\_base2[9] << endl;

cout << "y\_min=" << ymin;

cin.get();

return 0;

}

**Результати виконання:**

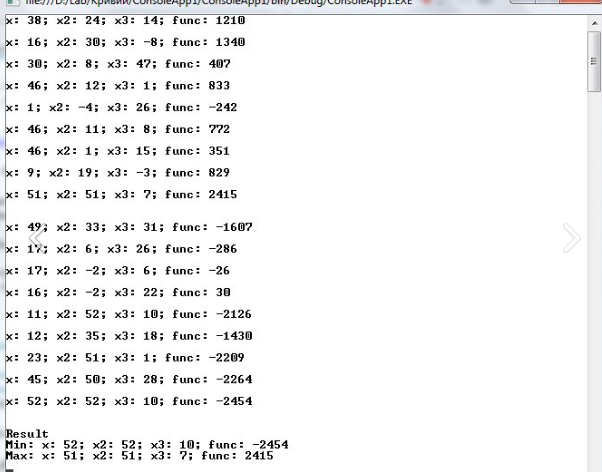


Рис. 2. Результат №1 запуску програми

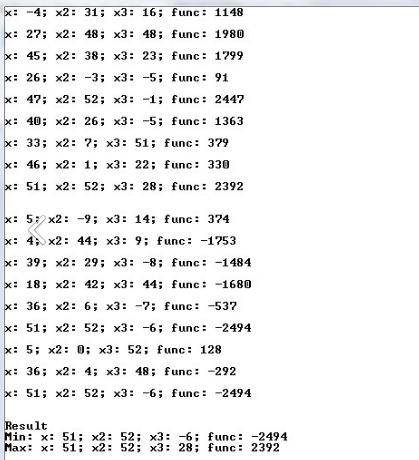


Рис. 3. Результат №2 запуску програми

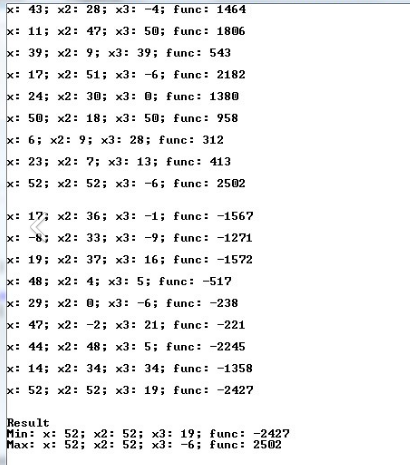


Рис. 4. Результат №3 запуску програми

**4. Висновок**

У результаті проведення лабораторної роботи я розробила програму, яка реалізовує генетичний алгоритм пошуку максимального і мінімального значення цільової функції з однією змінною.