*дата :*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **НУЛП, ІКНІ, САП** | | Тема | оцінка | підпис |
| КНC-13 | Лабораторна робота №4 | Знаходження екстремумів ф-ії з допомогою генетичного алгоритму |  |  |
| Тихун Юрій | |
| Методи нечіткої логіки та еволюційні алгоритми при автоматизованому проектуванні | | Викладач: | |
| Кривий Р. З. | |

**Мета:** Ознайомитися з основними теоретичними відомостями, вивчити еволюційні оператори схрещування та мутації, що використовуються при знаходженні значень цільової функції.

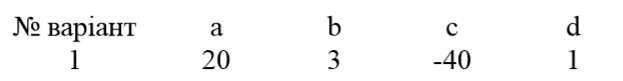
**Теоретичні відомості**

У загальному розумінні генетичні алгоритми (genetic algorithms) – це алгоритми, що використовують механізмами еволюції живої природи – природний відбір і генетичне наслідування. Генетичні алгоритми сьогодні застосовуються в різних галузях. Зокрема їх успішно використовують для розв’язування ряду важливих задач в економіці, бізнесі, техніці. З їх допомогою були розроблені промислові проектні рішення, що сприяли значній економії коштів і ресурсів. Фінансові компанії широко використовують ці засоби для прогнозування розвитку фінансових ринків для управління пакетами цінних паперів.

До основних характеристик ГА належать: розмір популяції (population size), оператор селекції (selection), оператор кросовера (crossover) і правила його використання, оператор мутації (mutation) і його параметри, оператор редукції (reduction), правило (критерій) зупинки процесу виконання генетичного алгоритму (stopping criteria). Оператори селекції, кросовера, мутації і редукції ще називають генетичними операторами.

**Індивідуальне завдання. Варіант 21(1)**

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



**Фрагменти коду програми для схрещування та мутації**

public class Genotype

{

public int[] genes;

public Genotype()

{

this.genes = new int[3];

for (int i = 0; i < genes.Length; i++)

{

this.genes[i] = (int)GetRandomNumber(-10, 53);

}

}

public void mutate()

{

for (int i = 0; i < genes.Length; i++)

{

if (GetRandomNumber(0.0, 100) < 5)

{

this.genes[i] = (int)GetRandomNumber(-10, 53);

}

}

}

}

public class Phenotype

{

double i\_x;

double i\_x2;

double i\_x3;

public Phenotype(Genotype g)

{

this.i\_x = g.genes[0];

this.i\_x2 = g.genes[1];

this.i\_x3 = g.genes[2];

}

public double evaluate(System.IO.StreamWriter file)

{

double fitness = 0;

if(countMax)

fitness -= 20 + 3 \* i\_x - 40 \* i\_x2 + i\_x3;

else

fitness += 20 + 3 \* i\_x - 40 \* i\_x2 + i\_x3;

file.WriteLine("x: " + i\_x + "; x2: "+i\_x2 + "; x3: "+i\_x3+"; func: "+ fitness);

return fitness;

}

}

public class Population

{

Individual[] pop;

public Population(System.IO.StreamWriter file, int populationNum)

{

this.pop = new Individual[populationNum];

for (int i = 0; i < populationNum; i++)

{

this.pop[i] = new Individual();

this.pop[i].evaluate(file);

}

Array.Sort(pop);

}

public void evolve(System.IO.StreamWriter file)

{

Individual a = select(100),

b = select(100),

x = breed(a, b);

if(countMax)

this.pop[0] = x;

else

this.pop[this.pop.Length - 1] = x;

x.evaluate(file);

Array.Sort(pop);

}

Individual select(int popNum)

{

int which = 0;

which = (int)Math.Floor(((float)popNum - 1E-6) \* (1.0 - Math.Pow(GetRandomNumber(0.0, 1.0), 2)));

return pop[which];

}

}

**Результат виконання**

При натисненні кнопки «Обчислити» програма генерує випадкові(рандомні) значення для генетичних операторів: мутації, схрещування та розмір популяції.

**№1.** Максимальне значення функції = 2095, при значеннях x=-9; x2=52; x3=-8

Мінімальне значення функції = -1972, при значеннях x=32; x2=52; x3=-8

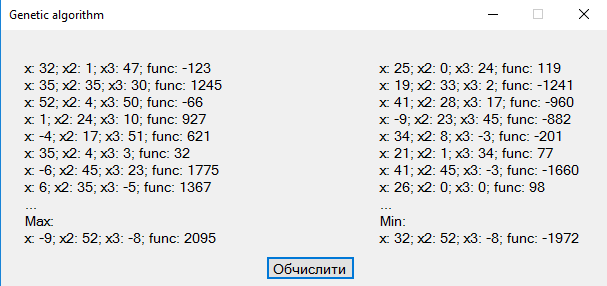


Рис.1. Результат виконання програми №1.

**№2.** Максимальне значення функції = 2053, при значеннях x=-8; x2=51; x3=-9

Мінімальне значення функції = -2004, при значеннях x=-2; x2=51; x3=10

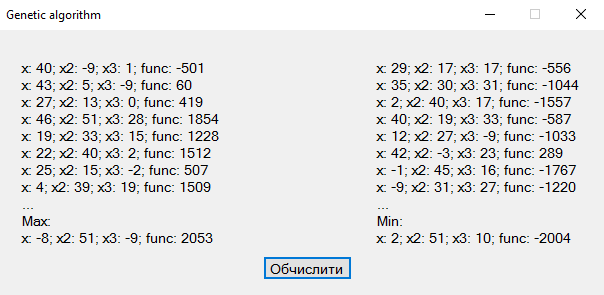


Рис.2. Результат виконання програми №2.

**№3.** Максимальне значення функції = 576, при значеннях x=-9; x2=14; x3=-9

Мінімальне значення функції = -2037, при значеннях x=4; x2=52; x3=11

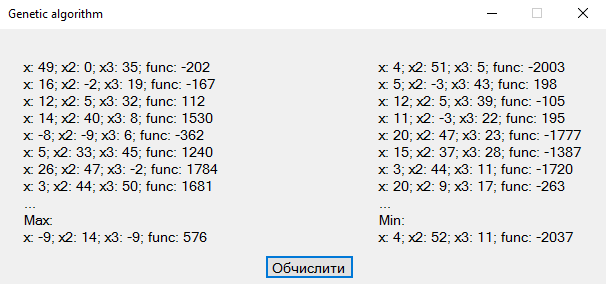


Рис.3. Результат виконання програми №3.

**№4.** Максимальне значення функції = 2092, при значеннях x=-9; x2=52; x3=-5

Мінімальне значення функції = -2021, при значеннях x=1; x2=51; x3=-4

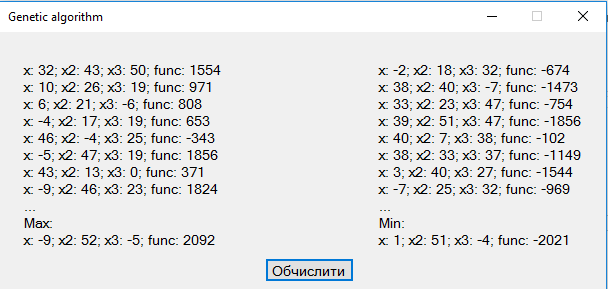


Рис.4. Результат виконання програми №4.

**Висновок**

Виконуючи лабораторну роботу я ознайомився з основними теоретичними відомостями, вивчив еволюційні оператори схрещування та мутації. Виконав індивідуальне завдання, яка полягає у знаходженні максимуму та мінімуму цільової функції за допомогою генетичного алгоритму.