**Мета роботи**: реалізувати генетичний алгоритм пошуку максимального і мінімального значення цільової функції згідно варіанту.

**Завдання:** реалізувати генетичний алгоритм пошуку максимального і мінімального значення цільової функції f(x) = a + bx + cx2 + dx3 на інтервалі

x = [-10, 53].

В - 1



**Максимальне і мінімальне значення цільової функції**

Графік функції згідно варіанту показаний на рис.1.

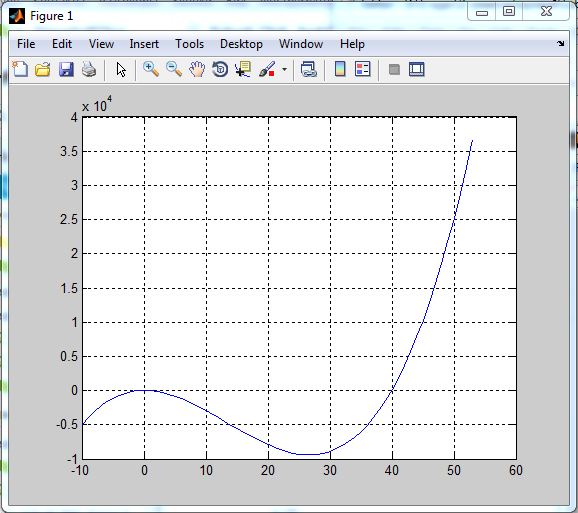
****

Рис.1. Графік функції на інтервалі [-10;53]

Для знаходження мінімуму функції було вибрано інтервал [20;30]. Пошук мінімуму цільової функції для перевірки реалізовано у Matlab за допомогою функції fminbnd() і xmin = 26.6291; ymin = -9.3815e+003.

Для знаходження максимуму функції було вибрано інтервал [5;15]. Пошук мінімуму цільової функції для перевірки реалізовано у Matlab за допомогою функції fminbnd() змінивши функцію на протилежну і xmax = 5.0001; ymax = -840.0192. Графік функції на інтервалі [5;15] показаний на рис.2.

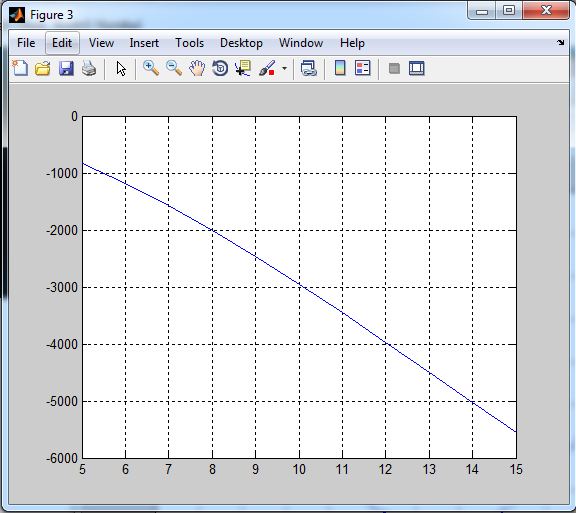


Рис.2. Графік функції на інтервалі[5;15]

**Результати виконання програми**

Для виконання індивідуального завдання була вибрано використовувати турнірний відбір, точкова мутація і рівномірне схрещування.

Результати:

|  |  |  |  |
| --- | --- | --- | --- |
| Кількість поколінь, розмір популяції | 10 | 100 | 300 |
| Minfmin | -9379.047 | -9381.537 | -9381.541 |
| xmin | 26.378517 | 26.624603 | 26.62791 |
| Maxfmax | -883.8748 | -855.7228 | -845.06506 |
| xmax | 5.134853 | 5.048645 | 5.015711 |

Отже похибка при знаходженні мінімуму і максимуму залежать від кількості поколінь і розміру популяції, при їхньому збільшенні похибки прямують до 0.

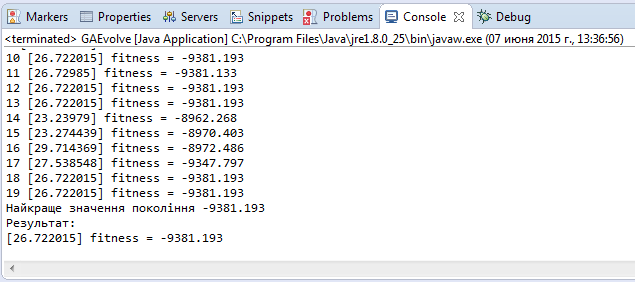


Рис.3. Результати виконання програми

Результат виконання програми для чотирьох поколінь з розміром популяції 20 для знаходження максимуму функції:

Початкова популяція

0 [7.709935] fitness = -1876.2915

1 [9.779672] fitness = -2840.9927

2 [10.158689] fitness = -3029.116

3 [5.781341] fitness = -1106.3772

4 [11.010742] fitness = -3461.522

5 [5.500584] fitness = -1007.3273

6 [12.781448] fitness = -4388.2256

7 [10.391974] fitness = -3146.288

8 [12.338593] fitness = -4154.181

9 [14.387513] fitness = -5238.6367

10 [10.7178135] fitness = -3311.536

11 [13.819708] fitness = -4938.5664

12 [12.160002] fitness = -4060.0996

13 [12.628331] fitness = -4307.205

14 [12.084074] fitness = -4020.1667

15 [10.257343] fitness = -3078.5447

16 [5.0446506] fitness = -854.42725

17 [14.268489] fitness = -5175.867

18 [14.10768] fitness = -5090.938

19 [7.914024] fitness = -1965.8591

Нова популяція

0 [5.7875667] fitness = -1108.6145

1 [5.040748] fitness = -853.16235

2 [5.5671124] fitness = -1030.4683

3 [5.0446506] fitness = -854.42725

4 [5.786369] fitness = -1108.1838

5 [5.0446506] fitness = -854.42725

6 [5.0016837] fitness = -840.54224

7 [5.0446506] fitness = -854.42725

8 [9.7970085] fitness = -2849.5337

9 [5.505467] fitness = -1009.0187

10 [5.036365] fitness = -851.7426

11 [5.669666] fitness = -1066.5433

12 [5.0446506] fitness = -854.42725

13 [5.6939435] fitness = -1075.1547

14 [12.420763] fitness = -4197.534

15 [11.959318] fitness = -3954.649

16 [13.380589] fitness = -4705.8013

17 [14.563093] fitness = -5331.0635

18 [7.914024] fitness = -1965.8591

19 [7.709935] fitness = -1876.2915

Найкраще значення покоління -840.54224

Нова популяція

0 [5.03684] fitness = -851.89636

1 [5.040748] fitness = -853.16235

2 [5.032457] fitness = -850.4774

3 [5.036365] fitness = -851.7426

4 [5.036838] fitness = -851.89575

5 [5.005592] fitness = -841.8014

6 [5.036836] fitness = -851.8951

7 [5.036365] fitness = -851.7426

8 [5.00168] fitness = -840.541

9 [5.0016856] fitness = -840.54285

10 [5.036842] fitness = -851.897

11 [5.0011916] fitness = -840.3837

12 [5.0446506] fitness = -854.42725

13 [5.0446506] fitness = -854.42725

14 [10.371208] fitness = -3135.8174

15 [9.401852] fitness = -2656.512

16 [9.31958] fitness = -2616.776

17 [7.5655756] fitness = -1813.7827

18 [5.040748] fitness = -853.16235

19 [5.0446506] fitness = -854.42725

Найкраще значення покоління -840.3837

Нова популяція

0 [5.005592] fitness = -841.8014

1 [5.0016856] fitness = -840.54285

2 [5.0011916] fitness = -840.3837

3 [5.0324416] fitness = -850.4725

4 [5.00168] fitness = -840.541

5 [5.0011916] fitness = -840.3837

6 [5.0051] fitness = -841.64276

7 [5.00168] fitness = -840.541

8 [5.00168] fitness = -840.541

9 [5.00168] fitness = -840.541

10 [5.032932] fitness = -850.6311

11 [5.0011916] fitness = -840.3837

12 [5.00168] fitness = -840.541

13 [5.0011916] fitness = -840.3837

14 [11.211855] fitness = -3565.1978

15 [14.758469] fitness = -5433.6436

16 [12.0464325] fitness = -4000.386

17 [14.07738] fitness = -5074.922

18 [5.0011916] fitness = -840.3837

19 [5.032457] fitness = -850.4774

Найкраще значення покоління -840.3837

Результат:

[5.0011916] fitness = -840.3837

Результат виконання програми для чотирьох поколінь з розміром популяції 20 для знаходження мінімуму функції:

Початкова популяція

0 [9.494883] fitness = -1454.9927

1 [23.060877] fitness = -1502.916

2 [23.116497] fitness = -1480.6318

3 [22.437338] fitness = -1734.6904

4 [9.517969] fitness = -1460.0999

5 [35.84837] fitness = 12741.891

6 [12.440836] fitness = -2059.7297

7 [27.703022] fitness = 1376.4043

8 [37.654533] fitness = 16613.852

9 [27.283651] fitness = 1024.1152

10 [39.607674] fitness = 21440.492

11 [16.69447] fitness = -2546.1118

12 [1.2049391] fitness = -19.589535

13 [29.379894] fitness = 2990.211

14 [6.9638867] fitness = -895.24225

15 [50.24516] fitness = 61323.234

16 [42.62348] fitness = 30300.14

17 [48.357048] fitness = 52390.53

18 [24.542166] fitness = -814.9971

19 [11.486011] fitness = -1877.8358

20 [33.019733] fitness = 7733.674

21 [2.1208382] fitness = -89.1657

22 [9.026643] fitness = -1350.9407

23 [50.247906] fitness = 61336.87

24 [51.608036] fitness = 68321.39

25 [19.261703] fitness = -2447.4736

26 [47.038513] fitness = 46658.35

27 [15.716618] fitness = -2494.686

28 [18.911469] fitness = -2483.3447

29 [28.85912] fitness = 2452.9316

Нова популяція

0 [18.942394] fitness = -2480.4854

1 [15.954351] fitness = -2511.124

2 [0.7105217] fitness = 2.653876

3 [17.449154] fitness = -2554.6035

4 [16.880257] fitness = -2550.8467

5 [18.973885] fitness = -2477.5117

6 [3.8279915] fitness = -303.24197

7 [20.708927] fitness = -2213.71

8 [30.692688] fitness = 4496.09

9 [16.69447] fitness = -2546.1118

10 [18.72382] fitness = -2499.4468

11 [16.69447] fitness = -2546.1118

12 [21.708935] fitness = -1964.8662

13 [7.9685097] fitness = -1115.0111

14 [12.444502] fitness = -2060.3916

15 [15.716618] fitness = -2494.686

16 [11.954886] fitness = -1969.4083

17 [4.9720764] fitness = -495.89856

18 [14.880707] fitness = -2418.444

19 [12.6945] fitness = -2104.7966

20 [24.690592] fitness = -734.86816

21 [28.85912] fitness = 2452.9316

22 [29.379894] fitness = 2990.211

23 [18.911469] fitness = -2483.3447

24 [28.85912] fitness = 2452.9316

25 [18.911469] fitness = -2483.3447

26 [9.026643] fitness = -1350.9407

27 [16.69447] fitness = -2546.1118

28 [15.716618] fitness = -2494.686

29 [15.716618] fitness = -2494.686

Найкраще значення покоління -2554.6035

Нова популяція

0 [17.383965] fitness = -2555.0176

1 [17.383965] fitness = -2555.0176

2 [17.383965] fitness = -2555.0176

3 [17.383965] fitness = -2555.0176

4 [17.383965] fitness = -2555.0176

5 [17.383965] fitness = -2555.0176

6 [17.383965] fitness = -2555.0176

7 [17.383965] fitness = -2555.0176

8 [17.383965] fitness = -2555.0176

9 [17.383965] fitness = -2555.0176

10 [17.383965] fitness = -2555.0176

11 [17.383965] fitness = -2555.0176

12 [17.383965] fitness = -2555.0176

13 [17.383965] fitness = -2555.0176

14 [17.383965] fitness = -2555.0176

15 [17.383965] fitness = -2555.0176

16 [17.383965] fitness = -2555.0176

17 [17.383965] fitness = -2555.0176

18 [17.383965] fitness = -2555.0176

19 [17.383965] fitness = -2555.0176

20 [17.383965] fitness = -2555.0176

21 [17.383965] fitness = -2555.0176

22 [9.384209] fitness = -1430.4744

23 [17.383965] fitness = -2555.0176

24 [53.0] fitness = 75963.0

25 [18.383965] fitness = -2523.2256

26 [17.383965] fitness = -2555.0176

27 [17.383965] fitness = -2555.0176

28 [17.383965] fitness = -2555.0176

29 [17.383965] fitness = -2555.0176

Найкраще значення покоління -2555.0176

Нова популяція

0 [17.383965] fitness = -2555.0176

1 [17.383965] fitness = -2555.0176

2 [17.383965] fitness = -2555.0176

3 [17.383965] fitness = -2555.0176

4 [17.383965] fitness = -2555.0176

5 [17.383965] fitness = -2555.0176

6 [17.383965] fitness = -2555.0176

7 [17.383965] fitness = -2555.0176

8 [17.383965] fitness = -2555.0176

9 [17.383965] fitness = -2555.0176

10 [17.383965] fitness = -2555.0176

11 [17.383965] fitness = -2555.0176

12 [17.383965] fitness = -2555.0176

13 [17.383965] fitness = -2555.0176

14 [17.383965] fitness = -2555.0176

15 [17.383965] fitness = -2555.0176

16 [17.383965] fitness = -2555.0176

17 [17.383965] fitness = -2555.0176

18 [17.383965] fitness = -2555.0176

19 [17.383965] fitness = -2555.0176

20 [17.383965] fitness = -2555.0176

21 [17.383965] fitness = -2555.0176

22 [17.383965] fitness = -2555.0176

23 [20.383965] fitness = -2278.7285

24 [17.383965] fitness = -2555.0176

25 [17.383965] fitness = -2555.0176

26 [33.383965] fitness = 8310.117

27 [17.383965] fitness = -2555.0176

28 [17.383965] fitness = -2555.0176

29 [17.383965] fitness = -2555.0176

Найкраще значення покоління -2555.0176

**Результат:**

**[17.383965] fitness = -2555.0176**

Код програми

**public** GAPopulation generate(GAPopulation p, **int** xrate, **int** mrate,

**float**[] min\_range, **float**[] max\_range) {

//Створення нової популяції з р, xrate відсотків індивідумів нового населення є

//схрещування, mrate відсотків з них створюються в результаті мутації, а інші по відтворення.

**if** (xrate < 0 || xrate > 100 || mrate < 0 || mrate > 100

|| xrate + mrate > 100)

System.***err***.println("error: xrate і/чи mrate неправилно встановлені");

GAIndividual[] newg = **new** GAIndividual[p.pop\_size];

**int** newg\_index = 0;

**int** xn = xrate \* p.pop\_size / 100;

//xn: Кількість нащадків, які будуть схрешення

**int** mn = mrate \* p.pop\_size / 100;

// mn: кількість нащадків які будуть створенні мутацією

// схрещування:

**for** (**int** i = 0; i < xn; i++) {

// select to parents for cross-over:

**int** p1 = p.tr\_select();

**int** p2 = p.tr\_select();

//System.out.println(p1+ " "+ p2);

newg[newg\_index++] = GAIndividual.*uniform*(p.ind[p1], p.ind[p2]);

}

// мутація:

**for** (**int** i = 0; i < mn; i++)

newg[newg\_index++] = p.ind[p.tr\_select()].mutate(min\_range,

max\_range);

// відтворення:

**for** (**int** i = newg\_index; i < p.pop\_size; i++)

newg[i] = p.ind[p.tr\_select()];

//for (int i = 0; i < p.pop\_size; i++) {

// System.out.println(i + " " + p.ind[i]);}

**return** **new** GAPopulation(newg);

}

**public** GAIndividual mutate(**float**[] min\_range, **float**[] max\_range) {

**float** rate = 1.0f / (**float**) genome\_size;

**float**[] result = **new** **float**[genome\_size];

**for** (**int** i = 0; i < genome\_size; i++)

result[i] = genome[i];

**for** (**int** i = 0; i < genome\_size; i++)

**if** ((**float**) Math.*random*() < rate)

result[i] = *randg*.nextFloat() \* (max\_range[i] - min\_range[i])

+ min\_range[i];

**return** **new** GAIndividual(result);

}

**public** **static** GAIndividual uniform(GAIndividual f, GAIndividual m) {

**float**[] child = **new** **float**[f.genome\_size];

Random random = **new** Random();

**for** (**int** k = 0; k < f.genome\_size; k++) {

String fs = *floatToBinary*(f.genome[k]);

String ms = *floatToBinary*(m.genome[k]);

**int**[] maska = **new** **int**[fs.length()];

**for**(**int** i = 0; i < fs.length(); i++){

maska[i] = random.nextInt(2);

}

String childs = "";

**for** (**int** i = 0; i < fs.length(); i++) {

**if**(maska[i] == 0) childs += fs.charAt(i);

**else** childs += ms.charAt(i);

}

child[0] = *binaryToFloat*(childs);

}

**return** **new** GAIndividual(child);

}

**Висновки:** виконавши лабораторну роботу я реалізував за допомогою програмної мови Java програмне забезпечення для пошуку оптимумів функції в якому використав використовувати турнірний відбір, точкова мутація і рівномірне схрещування. Програма показує результати за короткий період часу з невеликою похибкою навіть при невеликій кількості поколінь і розміру популяції, але при збільшенні цих параметрів похибка прямує до 0.