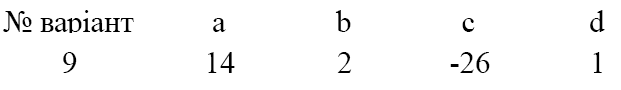
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **НУЛП, ІКНІ, САП** | | Тема | оцінка | підпис |
| СПКс-11 | 4 | КОМБІНАТОРНА  ОПТИМІЗАЦІЯ ЗА ДОПОМОГОЮ ЕВОЛЮЦІЙНИХ  МЕТОДІВ |  |  |
| Шкіндер А. В. | |
| Методи нечіткої логіки та еволюційні алгоритми при автоматизованому проектуванні | | Викладач: | |
| Кривий Р.З. | |

**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 - 14



14 48 -38 -71 5

**Максимальне і мінімальне значення цільової функції**

Графік функції згідно варіанту показаний на рис.1.

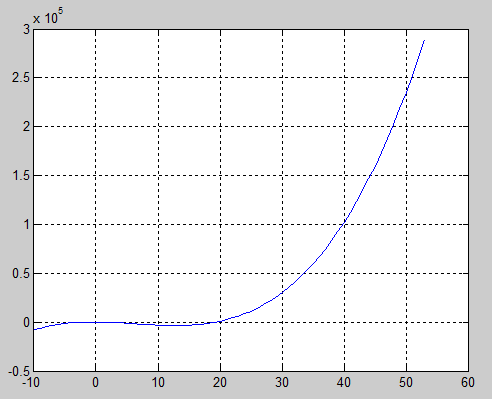


Рис.1. Графік функції на інтервалі [-10;53]

Для знаходження мінімуму функції було вибрано інтервал [0;53]. Пошук мінімуму цільової функції для перевірки реалізовано у Matlab за допомогою функції fminbnd() і рівний: -3.3309e+03 при x = 12.6610.

Для знаходження максимуму функції було вибрано інтервал [-10;10]. Пошук мінімуму цільової функції для перевірки реалізовано у Matlab за допомогою функції fminbnd() змінивши функцію на протилежну і рівний: 41.0965 при x = -0.4388. Графік функції на інтервалі [-10;10] показаний на рис.2.

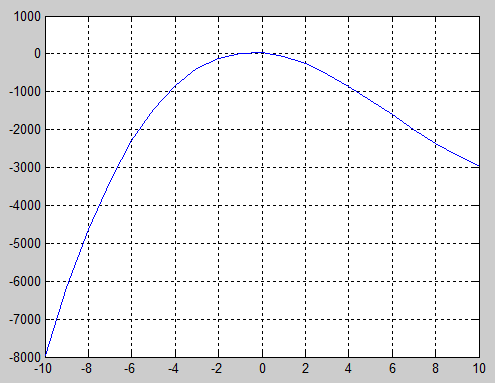


Рис.2. Графік функції на інтервалі[-10;10]

**Результати виконання програми**

Для виконання індивідуального завдання була вибрано використовувати турнірний відбір, одно точковий кросинговер і точкова мутація.

Результати:

|  |  |  |  |
| --- | --- | --- | --- |
| Кількість поколінь, розмір популяції | 10 | 100 | 300 |
| fmin | -2512.1626 | -2555.169 | -2555.2246 |
| xmin | 15.97067 | 17.248497 | 17.294273 |
| fmax | -12.889909 | 14.03714 | 14.038519 |
| xmax | 1.0796688 | 0.031248808 | 0.03855741 |

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

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

Початкова популяція  
0 [4.381027] fitness = -1060.7764   
1 [7.446253] fitness = -2107.3218   
2 [-7.0044956] fitness = -4887.6055   
3 [-6.994959] fitness = -4871.48   
4 [-1.882554] fitness = -165.44656   
5 [-7.0153008] fitness = -4905.9146   
6 [4.3090878] fitness = -1034.0294   
7 [-6.4649496] fitness = -4024.8486   
8 [-3.4060087] fitness = -843.799   
9 [5.768615] fitness = -1574.0598   
10 [-8.78878] fitness = -8496.6   
11 [1.2082243] fitness = -92.73988   
12 [-9.403229] fitness = -10029.748   
13 [7.2685604] fitness = -2059.2134   
14 [-5.126357] fitness = -2296.637   
15 [4.974456] fitness = -1282.4695   
16 [7.1513767] fitness = -2026.1624   
17 [8.5306] fitness = -2339.0056   
18 [-1.2772455] fitness = -29.709166   
19 [3.0864239] fitness = -598.6244   
Нова популяція  
0 [-1.2772455] fitness = -29.709166   
1 [1.2082243] fitness = -92.73988   
2 [-1.2772455] fitness = -29.709166   
3 [-1.882554] fitness = -165.44656   
4 [-1.2772455] fitness = -29.709166   
5 [-1.2772455] fitness = -29.709166   
6 [1.2082243] fitness = -92.73988   
7 [1.2082243] fitness = -92.73988   
8 [-1.882554] fitness = -165.44656   
9 [1.2082243] fitness = -92.73988   
10 [3.0864239] fitness = -598.6244   
11 [1.2082243] fitness = -92.73988   
12 [1.2082243] fitness = -92.73988   
13 [-1.882554] fitness = -165.44656   
14 [3.0864239] fitness = -598.6244   
15 [-3.4060087] fitness = -843.799   
16 [3.9174118] fitness = -889.8505   
17 [9.590469] fitness = -2436.295   
18 [2.4810371] fitness = -406.96246   
19 [-2.8352475] fitness = -528.9607   
Найкраще значення покоління -2436.295  
Нова популяція  
Найкраще значення покоління -10047.084  
Нова популяція  
0 [-1.2772455] fitness = -29.709166   
1 [-1.2772455] fitness = -29.709166   
2 [-1.2772455] fitness = -29.709166   
3 [-1.2772455] fitness = -29.709166   
4 [-1.2772455] fitness = -29.709166   
5 [-1.2772455] fitness = -29.709166   
6 [-1.2772455] fitness = -29.709166   
7 [-1.2772455] fitness = -29.709166   
8 [-1.2772455] fitness = -29.709166   
9 [-1.2772455] fitness = -29.709166   
10 [-1.2772455] fitness = -29.709166   
11 [-1.2772455] fitness = -29.709166   
12 [-1.2772455] fitness = -29.709166   
13 [-1.2772455] fitness = -29.709166   
14 [-1.2772455] fitness = -29.709166   
15 [-1.2772455] fitness = -29.709166   
16 [-3.747178] fitness = -1069.6198   
17 [1.2000761] fitness = -91.21422   
18 [9.815971] fitness = -2437.085   
19 [-2.7660584] fitness = -495.93304   
Найкраще значення покоління -2437.085  
Нова популяція  
0 [-1.2772455] fitness = -29.709166   
1 [-1.2772455] fitness = -29.709166   
2 [-1.2772455] fitness = -29.709166   
3 [-1.2772455] fitness = -29.709166   
4 [-1.2772455] fitness = -29.709166   
5 [-1.2772455] fitness = -29.709166   
6 [-1.2772455] fitness = -29.709166   
7 [-1.2772455] fitness = -29.709166   
8 [-1.2772455] fitness = -29.709166   
9 [-1.2772455] fitness = -29.709166   
10 [-1.2772455] fitness = -29.709166   
11 [-1.2772455] fitness = -29.709166   
12 [-1.2772455] fitness = -29.709166   
13 [-1.2772455] fitness = -29.709166   
14 [-1.2772455] fitness = -29.709166   
15 [-1.2772455] fitness = -29.709166   
16 [-8.079121] fitness = -6916.028   
17 [-2.763424] fitness = -494.69684   
18 [-6.824499] fitness = -4588.622   
19 [6.2868576] fitness = -1754.7188   
Найкраще значення покоління -6916.028  
Нова популяція  
0 [-1.2772455] fitness = -29.709166   
1 [-1.2772455] fitness = -29.709166   
2 [-1.2772455] fitness = -29.709166   
3 [-1.2772455] fitness = -29.709166   
4 [-1.2772455] fitness = -29.709166   
5 [-1.2772455] fitness = -29.709166   
6 [-1.2772455] fitness = -29.709166   
7 [-1.2772455] fitness = -29.709166   
8 [-1.2772455] fitness = -29.709166   
9 [-1.2772455] fitness = -29.709166   
10 [-1.2772455] fitness = -29.709166   
11 [-1.2772455] fitness = -29.709166   
12 [-1.2772455] fitness = -29.709166   
13 [-1.2772455] fitness = -29.709166   
14 [-1.2772455] fitness = -29.709166   
15 [-1.2772455] fitness = -29.709166   
16 [-0.065603256] fitness = 50.185944   
17 [-5.8393564] fitness = -3146.6226   
18 [5.5829296] fitness = -1507.0831   
19 [-1.6691818] fitness = -109.64213   
Найкраще значення покоління -3146.6226  
Нова популяція  
0 [-0.065603256] fitness = 50.185944   
1 [-1.2772455] fitness = -29.709166   
2 [-1.2772455] fitness = -29.709166   
3 [-0.065603256] fitness = 50.185944   
4 [-0.065603256] fitness = 50.185944   
5 [-0.065603256] fitness = 50.185944   
6 [-1.2772455] fitness = -29.709166   
7 [-0.065603256] fitness = 50.185944   
8 [-0.065603256] fitness = 50.185944   
9 [-1.2772455] fitness = -29.709166   
10 [-1.2772455] fitness = -29.709166   
11 [-1.2772455] fitness = -29.709166   
12 [-1.2772455] fitness = -29.709166   
13 [-0.065603256] fitness = 50.185944   
14 [-1.2772455] fitness = -29.709166   
15 [-1.2772455] fitness = -29.709166   
16 [3.323721] fitness = -679.0592   
17 [-2.1091232] fitness = -234.60088   
18 [-2.0851145] fitness = -226.7798   
19 [4.413843] fitness = -1072.9962   
Найкраще значення покоління -1072.9962  
Нова популяція  
0 [-0.065603256] fitness = 50.185944   
1 [-1.2772455] fitness = -29.709166   
2 [-0.065603256] fitness = 50.185944   
3 [-0.065603256] fitness = 50.185944   
4 [-0.065603256] fitness = 50.185944   
5 [-0.065603256] fitness = 50.185944   
6 [-0.065603256] fitness = 50.185944   
7 [-0.065603256] fitness = 50.185944   
8 [-0.065603256] fitness = 50.185944   
9 [-0.065603256] fitness = 50.185944   
10 [-0.065603256] fitness = 50.185944   
11 [-0.065603256] fitness = 50.185944   
12 [-0.065603256] fitness = 50.185944   
13 [-0.065603256] fitness = 50.185944   
14 [-0.065603256] fitness = 50.185944   
15 [-0.065603256] fitness = 50.185944   
16 [-1.9850273] fitness = -195.441   
17 [6.5158253] fitness = -1830.7972   
18 [0.001613617] fitness = 47.9385 19 [-2.1013832] fitness = -232.0666

Найкраще значення покоління -1830.7972

Нова популяція

0 [-0.065603256] fitness = 50.185944

1 [-0.065603256] fitness = 50.185944

2 [-0.065603256] fitness = 50.185944

3 [-0.065603256] fitness = 50.185944

4 [-0.065603256] fitness = 50.185944

5 [-0.065603256] fitness = 50.185944

6 [-0.065603256] fitness = 50.185944

7 [-0.065603256] fitness = 50.185944

8 [-0.065603256] fitness = 50.185944

9 [-0.065603256] fitness = 50.185944

10 [-0.065603256] fitness = 50.185944

11 [-0.065603256] fitness = 50.185944

12 [-0.065603256] fitness = 50.185944

13 [-0.065603256] fitness = 50.185944

14 [-0.065603256] fitness = 50.185944

15 [-0.065603256] fitness = 50.185944

16 [1.3181219] fitness = -113.99643

17 [-3.3857002] fitness = -831.26483

18 [7.1140823] fitness = -2015.4327

19 [3.9796047] fitness = -912.54

Найкраще значення покоління -2015.4327

Нова популяція

0 [-0.065603256] fitness = 50.185944

1 [-0.065603256] fitness = 50.185944

2 [-0.065603256] fitness = 50.185944

3 [-0.065603256] fitness = 50.185944

4 [-0.065603256] fitness = 50.185944

5 [-0.065603256] fitness = 50.185944

6 [-0.065603256] fitness = 50.185944

7 [-0.065603256] fitness = 50.185944

8 [-0.065603256] fitness = 50.185944

9 [-0.065603256] fitness = 50.185944

10 [-0.065603256] fitness = 50.185944

11 [-0.065603256] fitness = 50.185944

12 [-0.065603256] fitness = 50.185944

13 [-0.065603256] fitness = 50.185944

14 [-0.065603256] fitness = 50.185944

15 [-0.065603256] fitness = 50.185944

16 [2.9749956] fitness = -561.7899

17 [5.723032] fitness = -1557.7097

18 [-9.513462] fitness = -10321.537

19 [6.2893314] fitness = -1755.5546

Найкраще значення покоління -10321.537

Результат:

[-9.513462] fitness = -10321.537

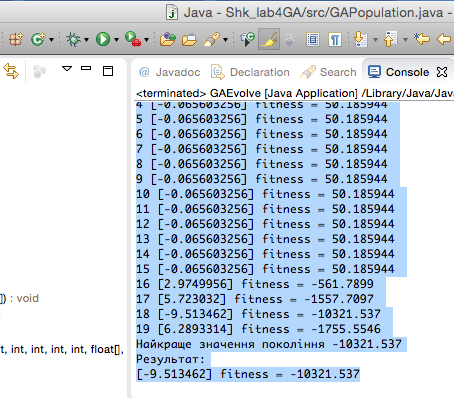


Рис.3. Результати виконання програми для знаходження мінімуму

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

Початкова популяція  
0 [-4.39018] fitness = -1576.6792   
1 [-3.7876177] fitness = -1098.3267   
2 [-7.177887] fitness = -5186.4043   
3 [-4.072888] fitness = -1312.8215   
4 [7.0204754] fitness = -1988.0668   
5 [-2.2807527] fitness = -293.982   
6 [6.0490017] fitness = -1673.1044   
7 [-0.100073814] fitness = 51.086742   
8 [-0.70973873] fitness = 37.41773   
9 [-0.3887787] fitness = 51.748207   
10 [6.1826305] fitness = -1719.2563   
11 [-0.30296326] fitness = 52.856705   
12 [0.22559357] fitness = 35.871483   
13 [-1.4982119] fitness = -71.25202   
14 [-0.4233551] fitness = 50.98281   
15 [8.143887] fitness = -2269.7622   
16 [-4.6144876] fitness = -1779.7806   
17 [-4.4404354] fitness = -1620.9741   
18 [-6.941766] fitness = -4782.1216   
19 [9.744926] fitness = -2437.6562

Нова популяція  
0 [-0.30296326] fitness = 52.856705   
1 [-0.100073814] fitness = 51.086742   
2 [-0.30296326] fitness = 52.856705   
3 [-0.30296326] fitness = 52.856705   
4 [-0.100073814] fitness = 51.086742   
5 [-0.30296326] fitness = 52.856705   
6 [-0.3887787] fitness = 51.748207   
7 [-0.3887787] fitness = 51.748207   
8 [-0.100073814] fitness = 51.086742   
9 [-0.30296326] fitness = 52.856705   
10 [-0.30296326] fitness = 52.856705   
11 [-0.100073814] fitness = 51.086742   
12 [-0.100073814] fitness = 51.086742   
13 [-0.3887787] fitness = 51.748207   
14 [-0.3887787] fitness = 51.748207   
15 [-0.30296326] fitness = 52.856705   
16 [-6.885372] fitness = -4688.4697   
17 [5.8352776] fitness = -1597.854   
18 [-9.391188] fitness = -9998.189   
19 [-1.4141331] fitness = -54.3865   
Найкраще значення покоління 52.856705  
Нова популяція  
0 [-0.30296326] fitness = 52.856705   
1 [-0.30296326] fitness = 52.856705   
2 [-0.30296326] fitness = 52.856705   
3 [-0.30296326] fitness = 52.856705   
4 [-0.30296326] fitness = 52.856705   
5 [-0.30296326] fitness = 52.856705   
6 [-0.30296326] fitness = 52.856705   
7 [-0.30296326] fitness = 52.856705   
8 [-0.30296326] fitness = 52.856705   
9 [-0.30296326] fitness = 52.856705   
10 [-0.30296326] fitness = 52.856705   
11 [-0.30296326] fitness = 52.856705   
12 [-0.30296326] fitness = 52.856705   
13 [-0.30296326] fitness = 52.856705   
14 [-0.30296326] fitness = 52.856705   
15 [-0.30296326] fitness = 52.856705   
16 [4.642008] fitness = -1158.1857   
17 [7.744589] fitness = -2182.2285   
18 [-7.369977] fitness = -5529.9756   
19 [5.97278] fitness = -1646.459   
Найкраще значення покоління 52.856705  
Нова популяція  
0 [-0.30296326] fitness = 52.856705   
1 [-0.30296326] fitness = 52.856705   
2 [-0.30296326] fitness = 52.856705   
3 [-0.30296326] fitness = 52.856705   
4 [-0.30296326] fitness = 52.856705   
5 [-0.30296326] fitness = 52.856705   
6 [-0.30296326] fitness = 52.856705   
7 [-0.30296326] fitness = 52.856705   
8 [-0.30296326] fitness = 52.856705   
9 [-0.30296326] fitness = 52.856705   
10 [-0.30296326] fitness = 52.856705   
11 [-0.30296326] fitness = 52.856705   
12 [-0.30296326] fitness = 52.856705   
13 [-0.30296326] fitness = 52.856705   
14 [-0.30296326] fitness = 52.856705   
15 [-0.30296326] fitness = 52.856705   
16 [0.8904381] fitness = -38.601074   
17 [-9.408634] fitness = -10043.936   
18 [-9.760468] fitness = -10994.278   
19 [-9.777014] fitness = -11040.287

Найкраще значення покоління 52.856705  
Нова популяція  
0 [-0.30296326] fitness = 52.856705   
1 [-0.30296326] fitness = 52.856705   
2 [-0.30296326] fitness = 52.856705   
3 [-0.30296326] fitness = 52.856705   
4 [-0.30296326] fitness = 52.856705   
5 [-0.30296326] fitness = 52.856705   
6 [-0.30296326] fitness = 52.856705   
7 [-0.30296326] fitness = 52.856705   
8 [-0.30296326] fitness = 52.856705   
9 [-0.30296326] fitness = 52.856705   
10 [-0.30296326] fitness = 52.856705   
11 [-0.30296326] fitness = 52.856705   
12 [-0.30296326] fitness = 52.856705   
13 [-0.30296326] fitness = 52.856705   
14 [-0.30296326] fitness = 52.856705   
15 [-0.30296326] fitness = 52.856705   
16 [0.59101963] fitness = 1.7728841   
17 [-4.010613] fitness = -1264.1866   
18 [0.16560745] fitness = 39.78239   
19 [0.1441555] fitness = 41.06163   
Найкраще значення покоління 52.856705  
Нова популяція  
0 [-0.30296326] fitness = 52.856705   
1 [-0.30296326] fitness = 52.856705   
2 [-0.30296326] fitness = 52.856705   
3 [-0.30296326] fitness = 52.856705   
4 [-0.30296326] fitness = 52.856705   
5 [-0.30296326] fitness = 52.856705   
6 [-0.30296326] fitness = 52.856705   
7 [-0.30296326] fitness = 52.856705   
8 [-0.30296326] fitness = 52.856705   
9 [-0.30296326] fitness = 52.856705   
10 [-0.30296326] fitness = 52.856705   
11 [-0.30296326] fitness = 52.856705   
12 [-0.30296326] fitness = 52.856705   
13 [-0.30296326] fitness = 52.856705   
14 [-0.30296326] fitness = 52.856705   
15 [-0.30296326] fitness = 52.856705   
16 [0.99979496] fitness = -55.96617   
17 [7.7719097] fitness = -2188.699   
18 [-4.06293] fitness = -1304.9763   
19 [3.0804968] fitness = -596.64935   
Найкраще значення покоління 52.856705  
Нова популяція  
0 [-0.30296326] fitness = 52.856705   
1 [-0.30296326] fitness = 52.856705   
2 [-0.30296326] fitness = 52.856705   
3 [-0.30296326] fitness = 52.856705   
4 [-0.30296326] fitness = 52.856705   
5 [-0.30296326] fitness = 52.856705   
6 [-0.30296326] fitness = 52.856705   
7 [-0.30296326] fitness = 52.856705   
8 [-0.30296326] fitness = 52.856705   
9 [-0.30296326] fitness = 52.856705   
10 [-0.30296326] fitness = 52.856705   
11 [-0.30296326] fitness = 52.856705   
12 [-0.30296326] fitness = 52.856705   
13 [-0.30296326] fitness = 52.856705   
14 [-0.30296326] fitness = 52.856705   
15 [-0.30296326] fitness = 52.856705   
16 [-7.32689] fitness = -5451.752   
17 [-7.4654937] fitness = -5705.7905   
18 [7.4343185] fitness = -2104.1692   
19 [4.9409] fitness = -1269.9426   
Найкраще значення покоління 52.856705  
Нова популяція  
0 [-0.30296326] fitness = 52.856705   
1 [-0.30296326] fitness = 52.856705   
2 [-0.30296326] fitness = 52.856705   
3 [-0.30296326] fitness = 52.856705   
4 [-0.30296326] fitness = 52.856705   
5 [-0.30296326] fitness = 52.856705   
6 [-0.30296326] fitness = 52.856705   
7 [-0.30296326] fitness = 52.856705   
8 [-0.30296326] fitness = 52.856705   
9 [-0.30296326] fitness = 52.856705   
10 [-0.30296326] fitness = 52.856705   
11 [-0.30296326] fitness = 52.856705   
12 [-0.30296326] fitness = 52.856705   
13 [-0.30296326] fitness = 52.856705   
14 [-0.30296326] fitness = 52.856705   
15 [-0.30296326] fitness = 52.856705   
16 [3.3961248] fitness = -704.09406   
17 [1.4657936] fitness = -144.50063   
18 [-4.4715176] fitness = -1648.7178   
19 [6.888693] fitness = -1948.5275   
Найкраще значення покоління 52.856705  
Нова популяція  
0 [-0.30296326] fitness = 52.856705   
1 [-0.30296326] fitness = 52.856705   
2 [-0.30296326] fitness = 52.856705   
3 [-0.30296326] fitness = 52.856705   
4 [-0.30296326] fitness = 52.856705   
5 [-0.30296326] fitness = 52.856705   
6 [-0.30296326] fitness = 52.856705   
7 [-0.30296326] fitness = 52.856705   
8 [-0.30296326] fitness = 52.856705   
9 [-0.30296326] fitness = 52.856705   
10 [-0.30296326] fitness = 52.856705   
11 [-0.30296326] fitness = 52.856705   
12 [-0.30296326] fitness = 52.856705   
13 [-0.30296326] fitness = 52.856705   
14 [-0.30296326] fitness = 52.856705   
15 [-0.30296326] fitness = 52.856705   
16 [8.806572] fitness = -2378.1055   
17 [-5.225179] fitness = -2405.223   
18 [1.3525543] fitness = -120.91285   
19 [-3.5361028] fitness = -926.4912

Найкраще значення покоління 52.856705

Нова популяція  
0 [-0.30296326] fitness = 52.856705   
1 [-0.30296326] fitness = 52.856705   
2 [-0.30296326] fitness = 52.856705   
3 [-0.30296326] fitness = 52.856705   
4 [-0.30296326] fitness = 52.856705   
5 [-0.30296326] fitness = 52.856705   
6 [-0.30296326] fitness = 52.856705   
7 [-0.30296326] fitness = 52.856705   
8 [-0.30296326] fitness = 52.856705   
9 [-0.30296326] fitness = 52.856705   
10 [-0.30296326] fitness = 52.856705   
11 [-0.30296326] fitness = 52.856705   
12 [-0.30296326] fitness = 52.856705   
13 [-0.30296326] fitness = 52.856705   
14 [-0.30296326] fitness = 52.856705   
15 [-0.30296326] fitness = 52.856705   
16 [4.157687] fitness = -977.9674   
17 [-1.9234285] fitness = -177.15909   
18 [-6.603446] fitness = -4236.7925   
19 [-8.51339] fitness = -7859.574   
Найкраще значення покоління 52.856705  
Результат:  
[-0.30296326] fitness = 52.856705

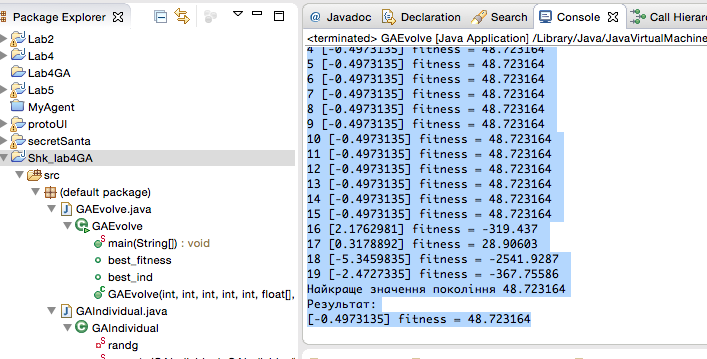


Рис.4. Результати виконання програми для знаходження максимуму

Код програми на мові програмування Java:

**class GAEvolve**

**public** **class** GAEvolve {

**public** **float**[] best\_fitness;// best\_fitness[i] є найкращим значенням для

// і-го покоління даної еволюції

**public** GAIndividual best\_ind; // найкращий індивідум(х) даної еволюції

**public** GAEvolve(**int** generations, **int** pop\_size, **int** genome\_size, **int** xrate,

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

// xrate: частота схрещування

// mrate: частота мутації

// xrate і mrate можуть змінюватися під час покоління генерації

best\_fitness = **new** **float**[generations];

GAPopulation gap = **new** GAPopulation(pop\_size, genome\_size, min\_range,

max\_range);

best\_fitness[0] = gap.ind[gap.best\_index].fitness;

**for** (**int** i = 1; i < generations; i++) {

gap = gap.generate(gap, xrate, mrate, min\_range, max\_range);

best\_fitness[i] = gap.ind[gap.best\_index].fitness;

// if(i<11)

System.***out***

.println("Найкраще значення покоління " + best\_fitness[i]);

}

best\_ind = gap.ind[gap.best\_index];

}

**public** **static** **void** main(String[] args) {

// вибір критерію пошуку мінімуму або максимуму

// для пошуку мінімуму

// float[] min = new float[] { 5f };

// float[] max = new float[] { 15f };

// для пошуку максимуму

**float**[] min = **new** **float**[] { -10f };

**float**[] max = **new** **float**[] { 10 };

GAEvolve gae = **new** GAEvolve(10, 20, 1, 80, 20, min, max);

System.***out***.println("Результат:");

System.***out***.println(gae.best\_ind);

}

}

**class GAIndividual**

**public** **class** GAIndividual {

**private** **static** Random *randg* = **new** Random(); // Рандомний генератор

**public** **int** genome\_size;

**public** **float**[] genome;

**public** **float** fitness;

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

// створити випадкового індивідума довжиною gsize

// і-й ген повине бути в діапазоні між min\_range[i] і max\_range[i]

genome\_size = gsize;

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

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

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

+ min\_range[i];

}

evalFitness();// оцінити придатність цього нового індивідума

}

**public** GAIndividual(**float** d[]) {

// Створити індивідума, що його ген є таки же як d []

genome\_size = d.length;

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

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

genome[i] = d[i];

}

evalFitness();// оцінити придатність цієї нового індивідума

}

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

// rate це шанс кожного гена мутувати

**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];

// застосування точкової мутації

**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 xover1p(GAIndividual f, GAIndividual m) {

// одноточковий кросинговер

Random rng = **new** Random();

**int** xpoint = 1 + rng.nextInt(1);

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

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

child[i] = f.genome[i];

}

**for** (**int** i = xpoint; i < f.genome\_size; i++) {

child[i] = m.genome[i];

}

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

}

**public** String toString() {

String s = "[";

s += genome[0] + "]";

s += " fitness = " + fitness;

**return** s;

}

**private** **void** evalFitness() {

**int** a = 30;

**int** b = -50;

**int** c = -55;

**int** d = 3;

// цільова функція

fitness = (a + b \* genome[0] + c \* genome[0] \* genome[0] + d

\* genome[0] \* genome[0] \* genome[0]);

}

}

**class GAPopulation**

**public** **class** GAPopulation {

**private** **static** Random *randg* = **new** Random();// Рандомний генератор

**public** **int** pop\_size;

**public** GAIndividual[] ind;

**public** **int** best\_index; // індекс найкращого індивідума в масиві ind

// best\_fitness = ind[best\_index].fitness

**public** GAPopulation(**int** psize, **int** gsize, **float**[] min\_range,

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

// створення рандомної популяції кількісью pop\_size

// psize: довжина популяції

// gsize: довжина геному

pop\_size = psize;

ind = **new** GAIndividual[pop\_size];

System.***out***.println("Початкова популяція");

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

{ind[i] = **new** GAIndividual(gsize, min\_range, max\_range);

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

//System.out.println();

evaluate();

}

**public** GAPopulation(GAIndividual[] p) {

// Створення популяції з такимиж індивідумами як в p

pop\_size = p.length;

ind = **new** GAIndividual[pop\_size];

System.***out***.println("Нова популяція");

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

ind[i] = p[i];

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

evaluate();

}

**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();

newg[newg\_index++] = GAIndividual.*xover1p*(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()];

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

}

**public** **int** tr\_select() {

//турнірна вибірка розміром pop\_size/10

//вона повертає індекс вибраного особи в ind []

**int** s\_index = *randg*.nextInt(pop\_size);

// індекс вибраного індивідума

**float** s\_fitness = ind[s\_index].fitness;

**int** tr\_size = Math.*min*(10, pop\_size);

**for** (**int** i = 1; i < tr\_size; i++) {

**int** tmp = *randg*.nextInt(pop\_size);

// вибір критерію пошуку мінімуму або максимуму

**if** (ind[tmp].fitness > s\_fitness) {//< для min//>для max

s\_index = tmp;

s\_fitness = ind[tmp].fitness;

}

}

**return** s\_index;

}

**private** **void** evaluate() {

//оцінювання

**int** best = 0;

// індекс найкрощого індивідума

**float** best\_fitness = ind[0].fitness;

// сума придатності особин даної популяції

**for** (**int** i = 1; i < pop\_size; i++) {

// вибір критерію пошуку мінімуму або максимуму

**if** (ind[i].fitness > best\_fitness) {//< для min//>для max

best = i;

best\_fitness = ind[i].fitness;

}

}

best\_index = best;

}

**public** String toString() {

String s = "best individual = " + ind[best\_index];

**return** s;

}

}

Код програми на мові програмування Matlab для знаходження екстремумів функції:

clc

a=48;

b=-38;

c=--71;

d=5;

x=-10:1:53

y=a+b\*x+c\*x.^2+d\*x.^3;

plot(x,y)

grid on

minX=fminbnd(@(x) a+b\*x+c\*x^2+d\*x^3,0,53)

minY=a+b\*minX+c\*minX^2+d\*minX^3

x=-10:1:10

y=a+b\*x+c\*x.^2+d\*x.^3;

figure

plot(x,y)

grid on

maxX=fminbnd(@(x) -(a+b\*x+c\*x^2+d\*x^3),-10,10)

maxY=a+b\*maxX+c\*maxX^2+d\*maxX^3

**Висновки:** виконавши лабораторну роботу я реалізував за допомогою програмної мови Java програмне забезпечення для пошуку оптимумів функції в якому використав турнірний відбір, одноточковий ію. 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.