**Міністерство освіти і науки України**

**Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформатики та програмної інженерії**

**Звіт**

з лабораторної роботи № 4 з дисципліни

«Проектування алгоритмів»

„**Проектування і аналіз алгоритмів для вирішення NP-складних задач ч.1**”

**Виконав(ла)**

(шифр, прізвище, ім'я, по батькові)

*ІП-15 Мочалов Дмитро Юрійович*



Київ 2022

Зміст

[**1**](#_30j0zll) 3

[**2**](#_1fob9te) 4

[**3**](#_3znysh7) 11

[3.1](#_2et92p0) 11

[*3.1.1*](#_tyjcwt) 11

[*3.1.2*](#_3dy6vkm) 13

[3.2](#_1t3h5sf) 19

[*3.2.1*](#_4d34og8) 20

[*3.2.2*](#_2s8eyo1) 21

[**Висновок 12**](#_17dp8vu)

[**Критерії оцінювання 13**](#_3rdcrjn)

# Мета лабораторної роботи

Мета роботи – вивчити основні підходи формалізації метаеврестичних алгоритмів і вирішення типових задач з їхньою допомогою.

# Завдання

Згідно варіанту, розробити алгоритм вирішення задачі і виконати його програмну реалізацію на будь-якій мові програмування.

Задача, алгоритм і його параметри наведені в таблиці 2.1.

Зафіксувати якість отриманого розв'язку (значення цільової функції) після кожних 20 ітерацій до 1000 і побудувати графік залежності якості розв'язку від числа ітерацій.

Зробити узагальнений висновок.

Таблиця 2.1 – Варіанти алгоритмів

|  |  |
| --- | --- |
| **№** | **Задача і алгоритм** |
| 1 | Задача про рюкзак (місткість P=250, 100 предметів, цінність предметів від 2 до 20 (випадкова), вага від 1 до 10 (випадкова)), генетичний алгоритм (початкова популяція 100 осіб кожна по 1 різному предмету, оператор схрещування одноточковий по 50 генів, мутація з ймовірністю 5% змінюємо тільки 1 випадковий ген). Розробити власний оператор локального покращення. |
| 2 | Задача комівояжера (100 вершин, відстань між вершинами випадкова від 5 до 50), мурашиний алгоритм (α = 2, β = 4, ρ = 0,4, Lmin знайти жадібним алгоритмом, кількість мурах М = 30, починають маршрут в різних випадкових вершинах). |
| 3 | Задача розфарбовування графу (200 вершин, степінь вершини не більше 20, але не менше 1), бджолиний алгоритм ABC (число бджіл 30 із них 2 розвідники). |
| 4 | Задача про рюкзак (місткість P=200, 100 предметів, цінність предметів від 2 до 20 (випадкова), вага від 1 до 10 (випадкова)), генетичний алгоритм (початкова популяція 100 осіб кожна по 1 різному предмету, оператор схрещування двоточковий порівну генів, мутація з ймовірністю 10% змінюємо тільки 1 випадковий ген). Розробити власний оператор локального покращення. |
| 5 | Задача комівояжера (150 вершин, відстань між вершинами випадкова від 5 до 50), мурашиний алгоритм (α = 2, β = 3, ρ = 0,4, Lmin знайти жадібним алгоритмом, кількість мурах М = 35, починають маршрут в різних випадкових вершинах). |
| 6 | Задача розфарбовування графу (250 вершин, степінь вершини не більше 25, але не менше 2), бджолиний алгоритм ABC (число бджіл 35 із них 3 розвідники). |
| 7 | Задача про рюкзак (місткість P=150, 100 предметів, цінність предметів від 2 до 10 (випадкова), вага від 1 до 5 (випадкова)), генетичний алгоритм (початкова популяція 100 осіб кожна по 1 різному предмету, оператор схрещування рівномірний, мутація з ймовірністю 5% два випадкові гени міняються місцями). Розробити власний оператор локального покращення. |
| 8 | Задача комівояжера (200 вершин, відстань між вершинами випадкова від 0(перехід заборонено) до 50), мурашиний алгоритм (α = 3, β = 2, ρ = 0,3, Lmin знайти жадібним алгоритмом, кількість мурах М = 45, починають маршрут в різних випадкових вершинах). |
| 9 | Задача розфарбовування графу (150 вершин, степінь вершини не більше 30, але не менше 1), бджолиний алгоритм ABC (число бджіл 25 із них 3 розвідники). |
| 10 | Задача про рюкзак (місткість P=150, 100 предметів, цінність предметів від 2 до 10 (випадкова), вага від 1 до 5 (випадкова)), генетичний алгоритм (початкова популяція 100 осіб кожна по 1 різному предмету, оператор схрещування рівномірний, мутація з ймовірністю 10% два випадкові гени міняються місцями). Розробити власний оператор локального покращення. |
| 11 | Задача комівояжера (250 вершин, відстань між вершинами випадкова від 0(перехід заборонено) до 50), мурашиний алгоритм (α = 2, β = 4, ρ = 0,6, Lmin знайти жадібним алгоритмом, кількість мурах М = 45, починають маршрут в різних випадкових вершинах). |
| 12 | Задача розфарбовування графу (300 вершин, степінь вершини не більше 30, але не менше 1), бджолиний алгоритм ABC (число бджіл 60 із них 5 розвідники). |
| 13 | Задача про рюкзак (місткість P=250, 100 предметів, цінність предметів від 2 до 30 (випадкова), вага від 1 до 25 (випадкова)), генетичний алгоритм (початкова популяція 100 осіб кожна по 1 різному предмету, оператор схрещування одноточковий 30% і 70%, мутація з ймовірністю 5% два випадкові гени міняються місцями). Розробити власний оператор локального покращення. |
| 14 | Задача комівояжера (250 вершин, відстань між вершинами випадкова від 1 до 40), мурашиний алгоритм (α = 4, β = 2, ρ = 0,3, Lmin знайти жадібним алгоритмом, кількість мурах М = 45 (10 з них дикі, обирають випадкові напрямки), починають маршрут в різних випадкових вершинах). |
| 15 | Задача розфарбовування графу (100 вершин, степінь вершини не більше 20, але не менше 1), класичний бджолиний алгоритм (число бджіл 30 із них 3 розвідники). |
| 16 | Задача про рюкзак (місткість P=250, 100 предметів, цінність предметів від 2 до 30 (випадкова), вага від 1 до 25 (випадкова)), генетичний алгоритм (початкова популяція 100 осіб кожна по 1 різному предмету, оператор схрещування двоточковий 30%, 40% і 30%, мутація з ймовірністю 10% два випадкові гени міняються місцями). Розробити власний оператор локального покращення. |
| 17 | Задача комівояжера (200 вершин, відстань між вершинами випадкова від 1 до 40), мурашиний алгоритм (α = 2, β = 4, ρ = 0,7, Lmin знайти жадібним алгоритмом, кількість мурах М = 45 (15 з них дикі, обирають випадкові напрямки), починають маршрут в різних випадкових вершинах). |
| 18 | Задача розфарбовування графу (300 вершин, степінь вершини не більше 50, але не менше 1), класичний бджолиний алгоритм (число бджіл 60 із них 5 розвідники). |
| 19 | Задача про рюкзак (місткість P=250, 100 предметів, цінність предметів від 2 до 30 (випадкова), вага від 1 до 25 (випадкова)), генетичний алгоритм (початкова популяція 100 осіб кожна по 1 різному предмету, оператор схрещування триточковий 25%, мутація з ймовірністю 5% два випадкові гени міняються місцями). Розробити власний оператор локального покращення. |
| 20 | Задача комівояжера (200 вершин, відстань між вершинами випадкова від 1 до 40), мурашиний алгоритм (α = 3, β = 2, ρ = 0,7, Lmin знайти жадібним алгоритмом, кількість мурах М = 45 (10 з них елітні, подвійний феромон), починають маршрут в різних випадкових вершинах). |
| 21 | Задача розфарбовування графу (200 вершин, степінь вершини не більше 30, але не менше 1), класичний бджолиний алгоритм (число бджіл 40 із них 2 розвідники). |
| 22 | Задача про рюкзак (місткість P=250, 100 предметів, цінність предметів від 2 до 30 (випадкова), вага від 1 до 25 (випадкова)), генетичний алгоритм (початкова популяція 100 осіб кожна по 1 різному предмету, оператор схрещування триточковий 25%, мутація з ймовірністю 5% змінюємо тільки 1 випадковий ген). Розробити власний оператор локального покращення. |
| 23 | Задача комівояжера (300 вершин, відстань між вершинами випадкова від 1 до 60), мурашиний алгоритм (α = 3, β = 2, ρ = 0,6, Lmin знайти жадібним алгоритмом, кількість мурах М = 45 (15 з них елітні, подвійний феромон), починають маршрут в різних випадкових вершинах). |
| 24 | Задача розфарбовування графу (400 вершин, степінь вершини не більше 50, але не менше 1), класичний бджолиний алгоритм (число бджіл 70 із них 10 розвідники). |
| 25 | Задача про рюкзак (місткість P=250, 100 предметів, цінність предметів від 2 до 20 (випадкова), вага від 1 до 10 (випадкова)), генетичний алгоритм (початкова популяція 100 осіб кожна по 1 різному предмету, оператор схрещування одноточковий по 50 генів, мутація з ймовірністю 5% змінюємо тільки 1 випадковий ген). Розробити власний оператор локального покращення. |
| 26 | Задача комівояжера (100 вершин, відстань між вершинами випадкова від 5 до 50), мурашиний алгоритм (α = 2, β = 4, ρ = 0,4, Lmin знайти жадібним алгоритмом, кількість мурах М = 30, починають маршрут в різних випадкових вершинах). |
| 27 | Задача розфарбовування графу (200 вершин, степінь вершини не більше 20, але не менше 1), бджолиний алгоритм ABC (число бджіл 30 із них 2 розвідники). |
| 28 | Задача про рюкзак (місткість P=200, 100 предметів, цінність предметів від 2 до 20 (випадкова), вага від 1 до 10 (випадкова)), генетичний алгоритм (початкова популяція 100 осіб кожна по 1 різному предмету, оператор схрещування двоточковий порівну генів, мутація з ймовірністю 10% змінюємо тільки 1 випадковий ген). Розробити власний оператор локального покращення. |
| 29 | Задача комівояжера (150 вершин, відстань між вершинами випадкова від 5 до 50), мурашиний алгоритм (α = 2, β = 3, ρ = 0,4, Lmin знайти жадібним алгоритмом, кількість мурах М = 35, починають маршрут в різних випадкових вершинах). |
| 30 | Задача розфарбовування графу (250 вершин, степінь вершини не більше 25, але не менше 2), бджолиний алгоритм ABC (число бджіл 35 із них 3 розвідники). |
| 31 | Задача про рюкзак (місткість P=250, 100 предметів, цінність предметів від 2 до 20 (випадкова), вага від 1 до 10 (випадкова)), генетичний алгоритм (початкова популяція 100 осіб кожна по 1 різному предмету, оператор схрещування одноточковий по 50 генів, мутація з ймовірністю 5% змінюємо тільки 1 випадковий ген). Розробити власний оператор локального покращення. |
| 32 | Задача комівояжера (100 вершин, відстань між вершинами випадкова від 5 до 50), мурашиний алгоритм (α = 2, β = 4, ρ = 0,4, Lmin знайти жадібним алгоритмом, кількість мурах М = 30, починають маршрут в різних випадкових вершинах). |
| 33 | Задача розфарбовування графу (200 вершин, степінь вершини не більше 20, але не менше 1), бджолиний алгоритм ABC (число бджіл 30 із них 2 розвідники). |
| 34 | Задача про рюкзак (місткість P=200, 100 предметів, цінність предметів від 2 до 20 (випадкова), вага від 1 до 10 (випадкова)), генетичний алгоритм (початкова популяція 100 осіб кожна по 1 різному предмету, оператор схрещування двоточковий порівну генів, мутація з ймовірністю 10% змінюємо тільки 1 випадковий ген). Розробити власний оператор локального покращення. |
| 35 | Задача комівояжера (150 вершин, відстань між вершинами випадкова від 5 до 50), мурашиний алгоритм (α = 2, β = 3, ρ = 0,4, Lmin знайти жадібним алгоритмом, кількість мурах М = 35, починають маршрут в різних випадкових вершинах). |

# Виконання

## Програмна реалізація алгоритму

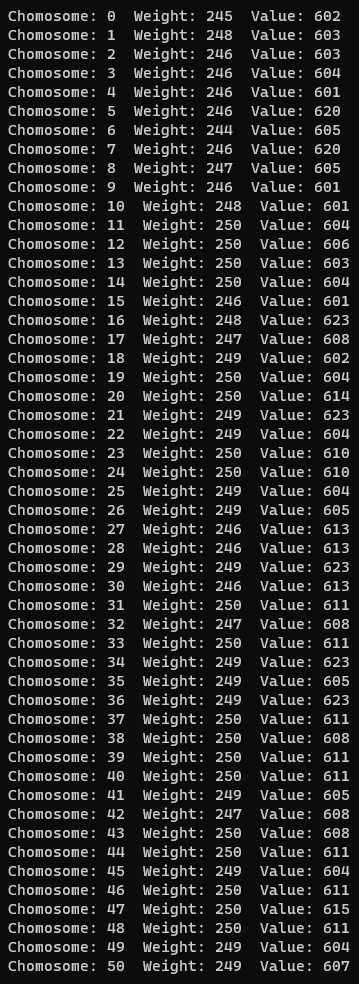
### Вихідний код

1. internal static class Solver
2. {
3. public static BackPack backPack = new BackPack();
4. static Population population = new Population(backPack);
5. static Chromosome Record;
6. public static void Solve()
7. {
8. population.GeneratePopulation();
9. Record = population.GetTheWorstChromosome;
10. int iterations = 0;
11. while (iterations <= 1000)
12. {
13. Chromosome FirstParent = population.GetBestChromosome;
14. Chromosome SecondParent;
15. while (true)
16. {
17. Random random = new Random();
18. int index = random.Next(0, 100);
19. if (population.Chromosomes[index] != FirstParent)
20. {
21. SecondParent = population.Chromosomes[index];
22. break;
23. }
24. }
25. Chromosome Offspring = CrossOver(FirstParent, SecondParent);
26. Chromosome mutation = Mutation(Offspring);
27. Chromosome improved = LocalImprovement(mutation);
28. if(improved.GetValue > Record.GetValue)
29. {
30. Record= improved;
31. }
32. Chromosome worstChromo = population.GetTheWorstChromosome;
33. if (improved.GetWeight <= backPack.Capacity)
34. {
36. population.Chromosomes.Remove(worstChromo);
37. population.Chromosomes.Add(improved);
38. }
39. else
40. {
41. if(Offspring.GetWeight <= backPack.Capacity)
42. {
43. population.Chromosomes.Remove(worstChromo);
44. population.Chromosomes.Add(Offspring);
45. }
46. else
47. {
48. continue;
49. }
50. }
51. if(iterations%20==0)
52. {
53. WriteRecordToFile(iterations, Record.GetValue);
54. }
55. iterations++;
56. }
57. for (int i = 0; i < population.Chromosomes.Count; i++)
58. {
59. Console.WriteLine("Chomosome: " + i + " Weight: " + population.Chromosomes[i].GetWeight + " Value: " + population.Chromosomes[i].GetValue);
60. }
61. }
62. private static Chromosome CrossOver(Chromosome FirstParent,Chromosome SecondParent)
63. {
64. List<int> mergedList = new List<int>(100);
65. int flag = 1;
66. for (int i = 0; i < 100; i++)
67. {
68. if (i % 25 == 0 && i!= 0) flag \*= -1;
69. if(flag > 0)
70. {
71. mergedList.Add(FirstParent.Gene[i]);
72. }
73. else
74. {
75. mergedList.Add(SecondParent.Gene[i]);
76. }
77. }
78. Chromosome Offspring = new Chromosome(mergedList);
79. return Offspring;
80. }
81. private static Chromosome Mutation(Chromosome chromosome)
82. {
83. Random random = new Random();
84. int chance = random.Next(1, 101);
85. if(chance <= 5)
86. {
87. int index\_1 = random.Next(0, chromosome.Gene.Count);
88. int index\_2;
89. while (true)
90. {
91. index\_2 = random.Next(0, chromosome.Gene.Count);
92. if (index\_2 != index\_1) break;
93. }
94. (chromosome.Gene[index\_1], chromosome.Gene[index\_2]) = (chromosome.Gene[index\_2], chromosome.Gene[index\_1]);
95. }
96. return chromosome;
97. }
98. private static Chromosome LocalImprovement(Chromosome chromosome)
99. {
100. Random random = new Random();
101. while (true)
102. {
103. int index = random.Next(0,chromosome.Gene.Count);
104. if (chromosome.Gene[index] == 0)
105. {
106. chromosome.Gene[index] = 1;
107. break;
108. }
109. }
110. return chromosome;
111. }
112. private static void WriteRecordToFile(int iteration,int value)
113. {
114. using(var stream = new StreamWriter("records.txt", true))
115. {
116. stream.WriteLine(value);
117. }
118. }

### Приклади роботи

На рисунках 3.1 і 3.2 показані приклади роботи програми.

Рисунок 3.1 –



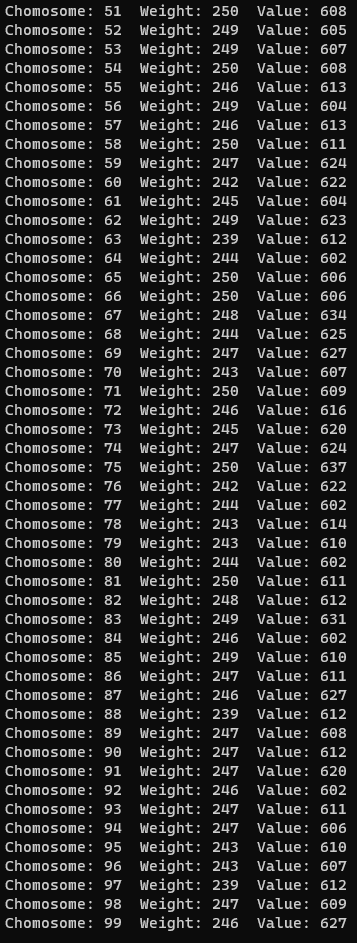
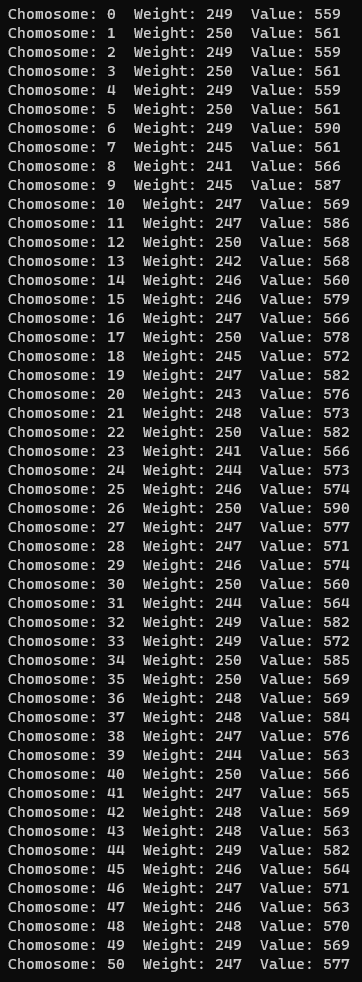
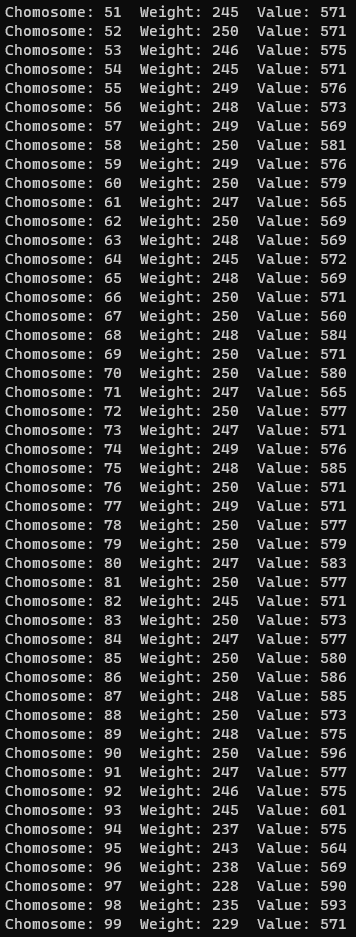


Рисунок 3.2 –



Тестування алгоритму

### Значення цільової функції зі збільшенням кількості ітерацій

У таблиці 3.1 наведено значення цільової функції зі збільшенням кількості ітерацій.

|  |  |
| --- | --- |
| Iterations | Record |
| 0 | 37 |
| 20 | 121 |
| 40 | 121 |
| 60 | 168 |
| 80 | 168 |
| 100 | 168 |
| 120 | 183 |
| 140 | 205 |
| 160 | 205 |
| 180 | 205 |
| 200 | 205 |
| 220 | 222 |
| 240 | 337 |
| 260 | 391 |
| 280 | 391 |
| 300 | 392 |
| 320 | 415 |
| 340 | 415 |
| 360 | 415 |
| 380 | 415 |
| 400 | 504 |
| 420 | 510 |
| 440 | 510 |
| 460 | 510 |
| 480 | 525 |
| 500 | 525 |
| 520 | 528 |
| 540 | 528 |
| 560 | 528 |
| 580 | 528 |
| 600 | 544 |
| 620 | 544 |
| 640 | 544 |
| 660 | 544 |
| 680 | 546 |
| 700 | 553 |
| 720 | 561 |
| 740 | 561 |
| 760 | 568 |
| 780 | 568 |
| 800 | 575 |
| 820 | 575 |
| 840 | 577 |
| 860 | 577 |
| 880 | 578 |
| 900 | 581 |
| 920 | 581 |
| 940 | 595 |
| 960 | 598 |
| 980 | 598 |
| 1000 | 608 |

Таблиця 3.1

### Графіки залежності розв'язку від числа ітерацій

На рисунку 3.3 наведений графік, який показує якість отриманого розв'язку.

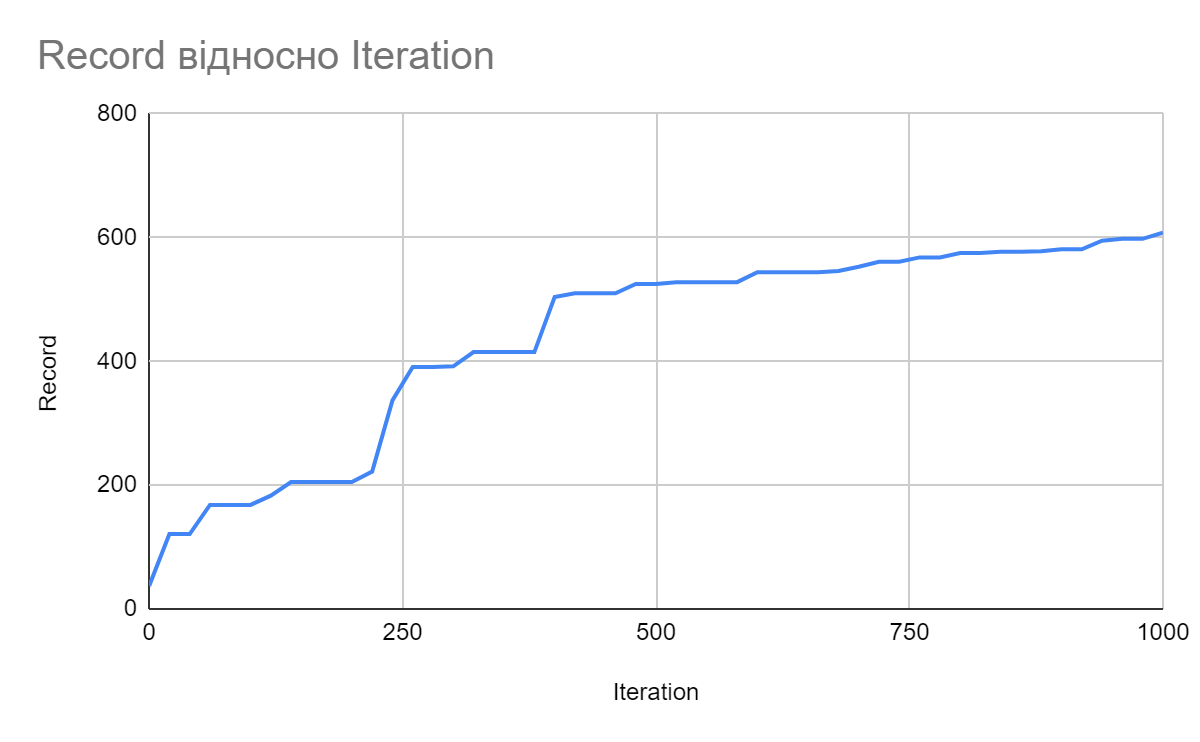


Рисунок 3.3 – Графік залежності розв'язку від числа ітерацій

Висновок

В рамках даної лабораторної роботи був використаний генетичний алгоритм, бля вирішення NP-складної задачі про рюкзак. В якості локального покращення, я змінював в новоотриманій хромосомі рандомний нульовий ген на 1, що збільшувало його значення та вагу. За допомогою такого покращення можемо швидше знайти рішення, оскільки при звичайному схрещуванні і мутації може бути таке, що найгірший ген просто не буде рости в значенні цільової функції. З проведеного дослідження отримали, що чим більше ітерації відбулось, тим повільніше зростає значення цільової функції, а наприкінці майже не змінюється.

Критерії оцінювання

При здачі лабораторної роботи до 27.11.2021 включно максимальний бал дорівнює – 5. Після 27.11.2021 максимальний бал дорівнює – 1.

Критерії оцінювання у відсотках від максимального балу:

* програмна реалізація алгоритму – 75%;
* тестування алгоритму– 20%;
* висновок – 5%.