# School of Computer Science and Engineering

## VIT-AP University

### Artificial Intelligence: CSE 3002

### Laboratory Assignment

## Topic: Genetic Algorithm for Roulette Wheel Selection

1. Write a program to evalaute the optimization problem where a fitness funtion is maximized. Based on the analysis of fitness values generate the roulette wheel with slot sized according to fitness. Futhermore, obtain the fitness value, pselect, expected count and actual count from Roulette Wheel for randomly generated initial population. Also, show the cross over site and mutation with obtained new population and updated fitness function. Print the avergage change in the fitness values and maximum fitness values of the selected polpuation after cross over and mutation.

   a) The fitness function if f(x)= $x^2$, on the integer interval [0, 31].
   b) The fitness function if f(x)= $x^2$, on the integer interval [0.63].
   c) The fitness function if f(x)= $x^3$, on the integer interval [0, 31].
   d) The fitness function if f(x)= $x^3$, on the integer interval [0, 64].

   Method :
   i. Encode the values pf input population
   ii. Selecet r number of population out of n polpulation.
   iii. Calculate the fitness value based on fitness function.
   iv. Based on fitness value of r seleceted enetitites generate Roulette Wheel.
   v. Based on these values find pselect, expected count, actual count from Roulette Wheel for crossover and mutation.

Sample result for fitness function f(x)= $x^2$, on the integer interval [0, 31] is shown below.

| No. | String | Fitness | % of Total |
|---|---|---|---|
| 1 | 01101 | 169 | 14.4 |
| 2 | 11000 | 576 | 49.2 |
| 3 | 01000 | 64 | 5.5 |
| 4 | 10011 | 361 | 30.9 |
| Total | | 1170 | 100.0 |

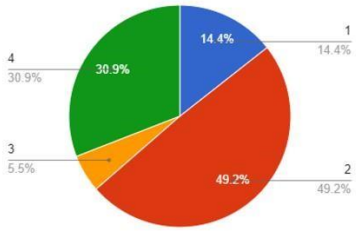


**Figure:** Simple reproduction allocates offspring using a roulette wheel with slots sized according to the fitness.

| String No | Initial Population (Randomly Generated) | x Value (Unsigned Integer) | f(x)= $x^2$ | pselect $\frac{f_i}{\sum f}$ | Expected Count $\frac{f_i}{\bar{f}}$ | Actual Count From Roulette |
|---|---|---|---|---|---|---|
| 1 | 01101 | 13 | 169 | 0.14 | 0.58 | 1 |
| 2 | 11000 | 24 | 576 | 0.49 | 1.97 | 2 |
| 3 | 01000 | 8 | 64 | 0.06 | 0.22 | 0 |
| 4 | 10011 | 19 | 361 | 0.31 | 1.23 | 1 |
| Sum | | | 1170 | 1.00 | 4.00 | 4.0 |
| Average | | | 293 | 0.2 | 1.00 | 1 |
| Max | | | 576 | 0.49 | 1.97 | 2.0 |

| Mating Pool after Reproduction (Cross Sit shown) | Mate (randomly Selected) | Crossover Site (Randomly Selecetd ) | New Population | X Value | F(x) $x^2$ |
|---|---|---|---|---|---|
| 0110 \| 1 | 2 | 4 | 01100 | 12 | 144 |
| 1100 \| 0 | 1 | 4 | 11001 | 25 | 625 |
| 11 \| 000 | 4 | 2 | 11011 | 27 | 729 |
| 10 \| 011 | 5 | 2 | 10000 | 16 | 256 |
| Sum | | | | | 1754 |
| Average | | | | | 439 |
| Max | | | | | 729 |

## Input:

```java
import java.util.Arrays;

public class GeneticAlgo {
    1 usage
    public static String generateRandom() {
        StringBuilder binaryNumber = new StringBuilder();
        for (int i = 0; i < 5; i++){
            int digit = (int) (2 * Math.random());
            binaryNumber.append(digit);
        }
        return String.valueOf(binaryNumber);
    }
    1 usage
    public static int binToDec(int num) {
        int len = String.valueOf(num).length();
        int temp = num;
        int dec = 0;
        for (int i = 0; i <= len; i++) {
            int dig = temp % 10;
            if (dig == 1) {
                dec += Math.pow(2,i);
            }
            temp /= 10;
        }
        return dec;
    }
```

```java
    public static void main(String[] args) {
        System.out.println("Testing 4 random Values");
        int num = 4;
        String[] binaryArr = new String[num];
        int[] binaryA = new int[num];
        for (int i = 0; i < num; i++) {
            String temp = generateRandom();
            binaryArr[i] = temp;
            int number = Integer.parseInt(temp);
            binaryA[i] = number;
        }
        int[] arr = new int[num];
        for (int i = 0; i < num; i++) {
            arr[i] = binToDec(binaryA[i]);
        }
        int[] fitVal = new int[num];
        for (int i = 0; i < num; i++)
        {
            int fitnessValue = arr[i] * arr[i];
            fitVal[i] = fitnessValue;
        }
        int total = 0;
        for (int i : fitVal) total+=i;
        int avg = total / num;
        float[] pSelect = new float[num];    // pSelect values
        for (int i = 0; i < num; i++) {
            float ps = (float) fitVal[i] / total;
            pSelect[i] = ps;
        }
        float[] expectedCount = new float[num];  // expected count
        for (int i = 0; i < num; i++) {
            float ec = (float) fitVal[i] / avg;
            expectedCount[i] = ec;
        }
        int [] actualCountRoulette = new int[num];
        for (int i = 0; i < num; i++) {
            int ac = Math.round(expectedCount[i]);
            actualCountRoulette[i] = ac;
        }

        System.out.println(Arrays.toString(binaryArr));
        System.out.println(Arrays.toString(arr));
        System.out.println(Arrays.toString(fitVal));
        System.out.println(Arrays.toString(pSelect));
        System.out.println(Arrays.toString(expectedCount));
        System.out.println(Arrays.toString(actualCountRoulette));

        float[] percentage = new float[num];

    }
}
```

## Output:

```
Testing 4 random Values
[10110, 00000, 10000, 00110]
[22, 0, 16, 6]
[484, 0, 256, 36]
[0.62371135, 0.0, 0.3298969, 0.04639175]
[2.4948454, 0.0, 1.3195876, 0.185567]
[2, 0, 1, 0]
```

**Submitted By:** Harshita Pasupuleti
**Registration number:** 21BCE8421