

Code of program:

```
public class Main {
    public static Integer N = 20;
    public static Integer[][] generation = new Integer[20][N];
    public static Integer[][] good = new Integer[10][N];
    public static Integer[][] childes = new Integer[10][N];
    public static Integer childCount = 0;
    public static Integer checkCount = 0;
    public static Double prevMax = 0.0;
    public static Double prevAverage = 0.0;
    public static void main(String[] args) {
        createGenerate();
        while(true) {
            Arrays.sort(generation, new Comparator<Integer[]>() {
                @Override
                public int compare(Integer[] o1, Integer[] o2) {
                    Double int1 = fitness(o1);
                    Double int2 = fitness(o2);
                    return (int)(int1 - int2) * 100;
                }
            });
            Double max = fitness(generation[0]);
            Double average = average();
            if(prevMax.equals(max)) {
                checkCount++;
            }
            else
                checkCount = 0;
            prevMax = max;
            System.out.printf("%.3f %.3f", max, average());
            for(int i=0; i<20; i++) {
                System.out.printf("%10.3f", calcX(generation[i]));
            }
            System.out.println();
            if(checkCount > 10) {
                return;
            }
            setGood();
            Random random = new Random();
            childCount = 0;
            for (int i = 0; i < 5; i++) {
                createChildes(good[random.nextInt(10)], good[random.nextInt(10)]);
            }
            createNewGeneration();
        }
    }
    public static void createGenerate() {
        Random random = new Random();
        for (int i = 0; i < 20; i++) {
            for (int j = 0; j < N; j++) {
                generation[i][j] = random.nextInt(2);
            }
        }
    }
    public static void setGood() {
        for(int i=0; i<10; i++) {
            good[i] = generation[i];
        }
    }
    public static void createNewGeneration() {
        for (int i=0; i<20; i++) {
            if(i<10)
                generation[i] = good[i];
            else
                generation[i] = good[i];
        }
    }
}
```

```
        generation[i] = childes[i-10];  
    }  
}
```

Table of values:

Iteration	N
0	180
100	109
200	75
300	53
400	45
500	37
600	29
700	18
800	13
900	9
1000	5

Graph N/Iteration:

