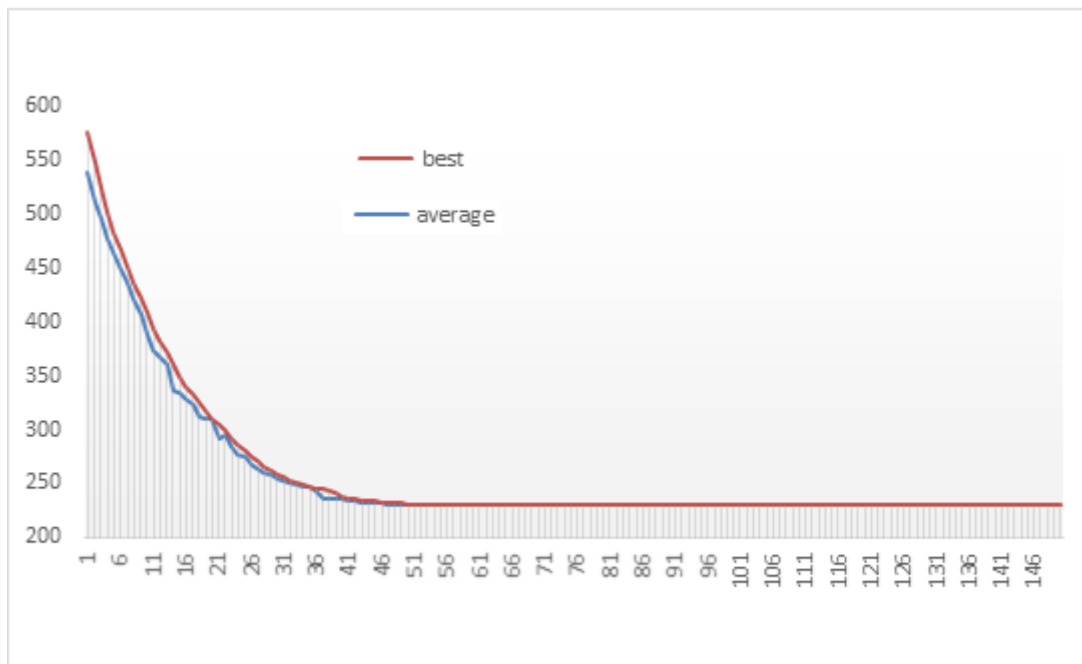
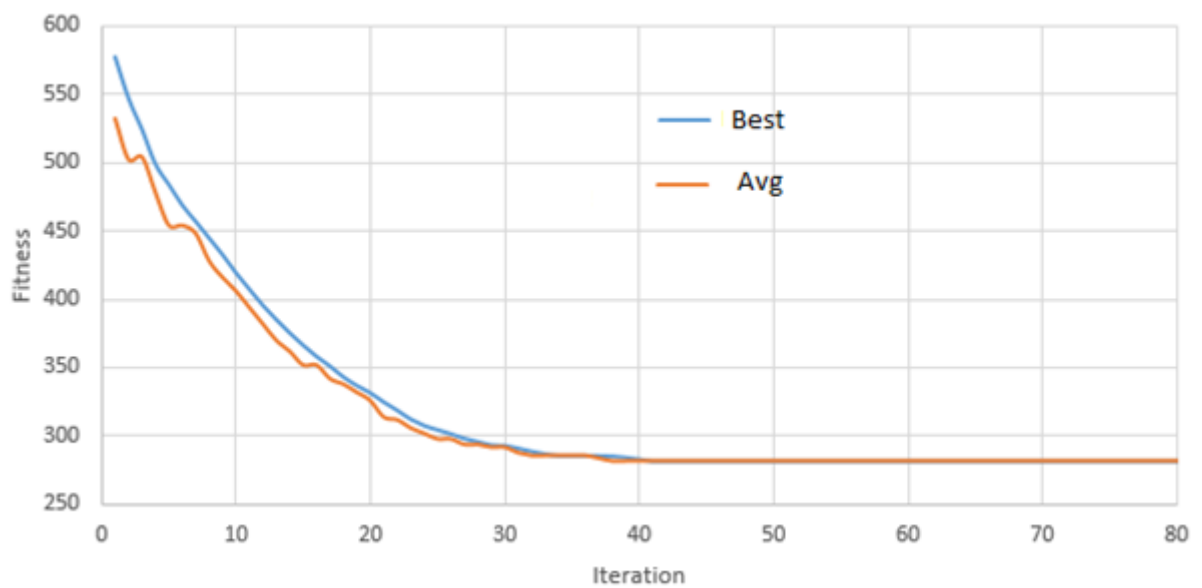


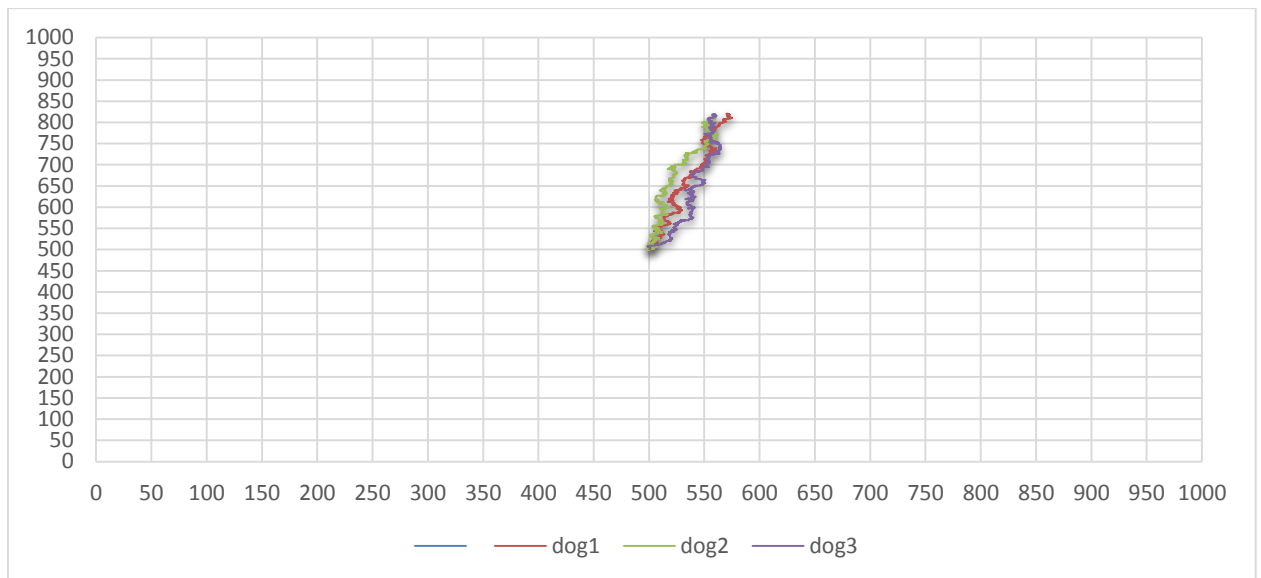
2-Artyom Yudenkov

This was realized by one-pointed crossover.



This one was realized by uniform crossover





-CODE-

```
import java.util.Random;

public class Main {

    private static final int Ch = 100;

    private static final int G = 1000;

    public static void main(String... args){

        int[][] osob = new int[Ch][G];

        double avr = 0;

        double numb = 0;

        Random random = new Random();

        for (int i = 0; i < Ch; i++){

            for (int j = 0; j < G; j++){

                osob[i][j] = random.nextInt(2);

                avr += osob[i][j];

            }

        }

    }

}
```

```

}

System.out.println(avr / (Ch * G));

int[] temp = new int[G];

int[][] best = new int[Ch / 2][G];

int it = 1;

do{

    System.out.println(it);

    it++;

    for (int i = 0; i < Ch - 1; i++){

        int max_fitness = i;

        for (int j = i + 1; j < Ch; j++){

            int countI = 0;

            int countJ = 0;

            for (int _j = 0; _j < G; _j++){

                countI += osob[max_fitness][_j];

                countJ += osob[j][_j];

            }

            if(countJ > countI){

                max_fitness = j;

            }

        }

        if(max_fitness != i){

            for (int k = 0; k < G; k++){

                temp[k] = osob[i][k];

            }

            for(int k=0;k<G;k++) {

                osob[i][k] = osob[max_fitness][k];

            }

            for(int k=0;k<G;k++) {

                osob[max_fitness][k] = temp[k];

            }

        }

    }

}

```

```

    }
}
double AvrFit=0;
int BestFit=0;
for(int i=0;i < Ch/2;i++){
    int fit=0;
    for(int j=0; j < G;j++) {
        best[i][j]=osob[i][j];
        fit += best[i][j];
    }
    AvrFit += fit;
}
AvrFit = AvrFit/50;
System.out.println("AvrFit=" + AvrFit);
for(int j = 0;j<G;j++) {
    BestFit += best[0][j];
}
System.out.println("BestFit" + BestFit + "\n");
if(isFitnessGood(best)!=-1)break;
int mom=0;
int dad=0;
int cat=0;
int[] firstChild = new int[G];
int[] secondChild = new int[G];
for(int i=0;i<Ch;i++) {
    do {
        mom = random.nextInt(Ch / 2);
        dad = random.nextInt(Ch / 2);
    }while(mom == dad);
    cat = random.nextInt(G);
    for(int k=0; k < cat;k++){
        firstChild[k] = best[mom][k];
    }
}

```

```

        secondChild[k] = best[dad][k];
    }
    for(int k=cat;k<G;k++) {
        firstChild[k] = best[dad][k];
        secondChild[k] = best[mom][k];
    }
    int fitness1=0;
    int fitness2=0;
    for(int k=0;k<G;k++) {
        fitness1 += firstChild[k];
        fitness2 += secondChild[k];
    }
    if(fitness1>fitness2) {
        for(int k=0; k < G; k++) {
            osob[i][k] = firstChild[k];
        }
    } else {
        for(int k=0;k<G;k++) {
            osob[i][k] = secondChild[k];
        }
    }
}
}while (true);
}

public static int isFitnessGood(int[][] osob){
    int count = 0;
    for (int i = 0; i < Ch / 2; i++){
        count = 0;
        for (int j = 0; j < G; j++){
            count += osob[i][j];
        }
        if(count == G){

```

```
        return i;
    }
}
return -1;
}
```

```
}
```