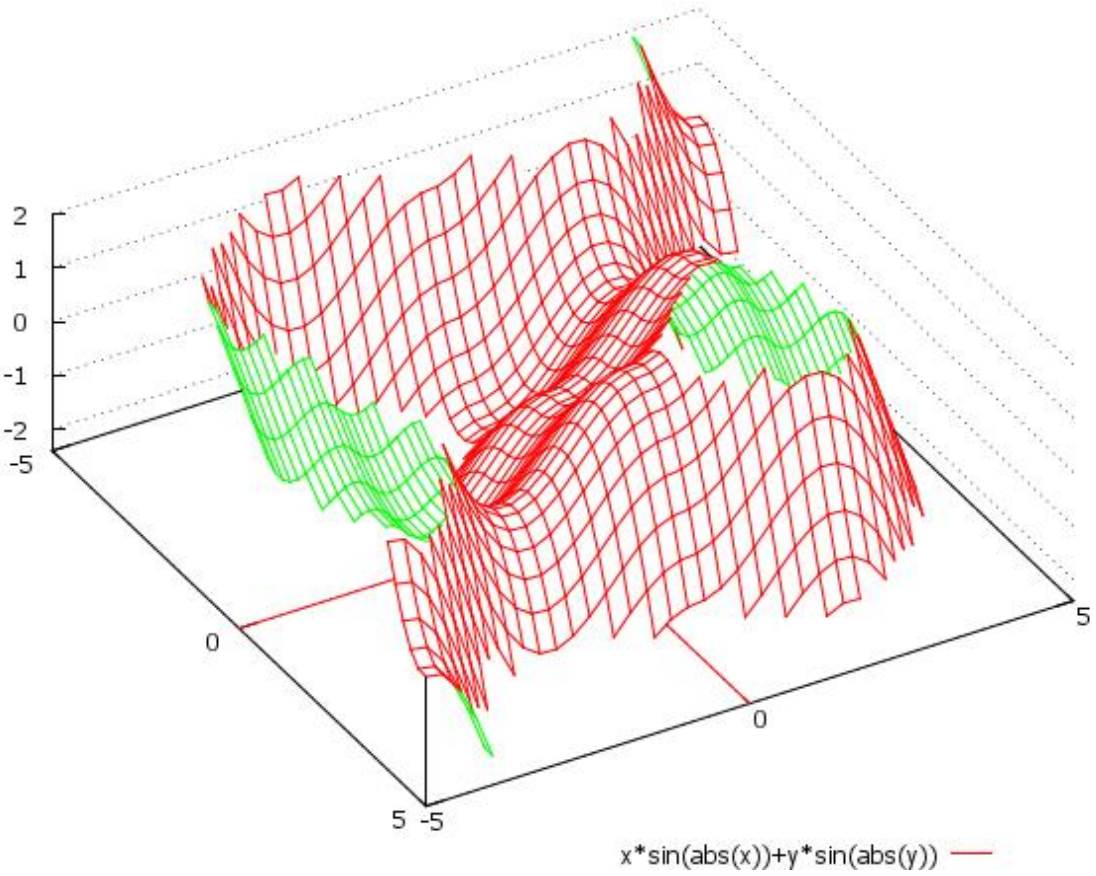
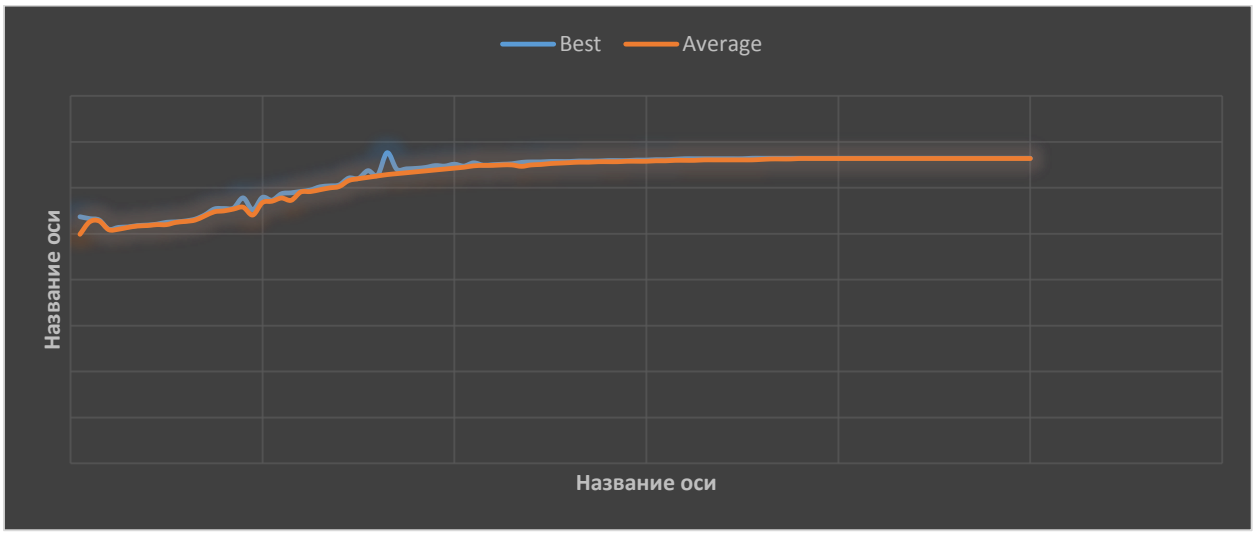
TASK – 3

Artyom Yudenkov

Maximization of 3D Schwefel function z=x*sin(|x|)+y*sin(|y|) -5<=x, y<=5 20 chrosomes with 22 binary genes by Fitness Sharing algorithm.





CODE :

```
import java.util.Random;
public class Main {

private static final int Ch
= 20;

   private static final int G = 22;


   public static void main(String... args){        int[][] dist = 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++){               dist[i][j] =
random.nextInt(2);

            avr += dist[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 += dist[max_fitness][_j];
countJ += dist[j][_j];

               }

               if(countJ > countI){

                  max_fitness = j;

               }

            }

            if(max_fitness != i){
for (int k = 0; k < G; k++){
```

```
                temp[k] = dist[i][k];

            }

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

                dist[i][k] = dist[max_fitness][k];

            }


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

                  dist[i][k] = firstChild[k];

                }

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

                  dist[i][k] = secondChild[k];

                }

            }

         }

      }while (true);

   }


   public static int isFitnessGood(int[][] dist){
```

```
      int count = 0;
for (int i = 0; i < Ch / 2;
i++){           count = 0;
for (int j = 0; j < G; j++){

            count += dist[i][j];

         }

         if(count ==
G){              return
i;

         }

      }

      return -1;



}

}
```