

2-Andrey Cheslow

```
package java44444;
import java.io.FileWriter;
import java.io.IOException;
import static java.lang.Math.abs;
import static java.lang.Math.pow;
import static java.lang.Math.sin;
import java.util.Random;
/*
 * @author Andrey
 */
public class Java44444 {
    public static void main(String[] args) throws IOException {
        int generation[][]=new int[20][22];
        int next_generation[][]=new int[20][22];
        Random rnd=new Random();
        //make first generation
        for(int i=0;i<20;i++){
            for(int j=0;j<22;j++){
                generation[i][j]=(rnd.nextInt(2));
            }
        }
        double fitness[] = new double[20];
        int pop=0;
        int num=0;
        FileWriter writ = new FileWriter("C:\\Users\\Andrey\\Desktop\\andrey.txt", false );
        writ.close();
        while (num<3){
            System.out.println(pop+1+" population");
            pop++;
            if (fitness[0]>9)num ++;
            else num=0;
            for(int i=0;i<20;i++){
                fitness[i]=0;
            }
            for(int i=0;i<20;i++){
                int x1=0,x2=0;
                for(int k=0,f=10;k<10;k++,f--){
                    x1+=(int) (generation[i][f]*pow(2,k));

                    x2+=(int) (generation[i][f+11]*pow(2,k));
                }
                if(generation[i][0]==0) x1*=-1;
                if(generation[i][11]==0) x2*=-1;
                double x_1=((double)x1*5)/1023;
                double x_2=((double)x2*5)/1023;
                fitness[i]=x_1*sin(abs(x_1))+x_2*sin(abs(x_2));//71498314
                System.out.println(x_1+","+x_2+","+fitness[i]);
            }
        }
    }
}
```

```

    }
    double sum_fit=0;
    for(int i=0;i<20;i++){
        sum_fit+=fitness[i];
    }
    sort(fitness,generation);
    FileWriter writer = new FileWriter("C:\\Users\\Andrey\\Desktop\\andrey.txt", true);
    String fff="";
    fff+=fitness[0]+" "+sum_fit/20;
    writer.write(fff);
    writer.write("\r\n");
    writer.flush();
    make_new_gen(generation,next_generation);
    generation=next_generation;
}
}

public static void sort(double[] fitness,int[][] generation){
    for(int i=0;i<19;i++){
        for(int j=i+1;j<20;j++){
            if (fitness[j]>fitness[i]){
                double temp=fitness[i];
                fitness[i]=fitness[j];
                fitness[j]=temp;
                int temps[]=generation[i];
                generation[i] = generation[j];
                generation[j] = temps;
            }
        }
    }
}

}

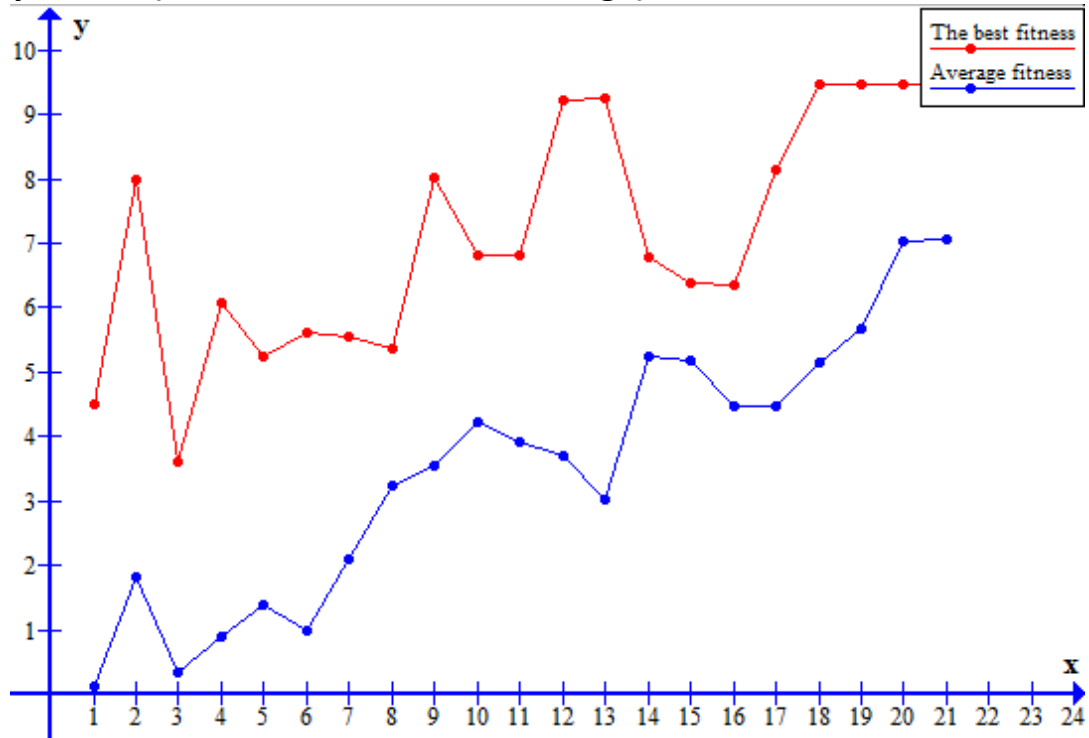
public static void make_new_gen(int[][] generation,int[][] next_generation){
    Random rnd= new Random();
    for(int i=0;i<20;i++){
        int nump1=rnd.nextInt(10);
        int nump2=rnd.nextInt(10);
        for(int f=0;f<22;f++){
            int orrr=rnd.nextInt(2);
            if(orrr==0)
                next_generation[i][f]=generation[nump1][f];
            else next_generation[i][f]=generation[nump2][f];
        }
    }
    int randomnumber=11;
    for(int k=0;k<20;k++){
        for(int j=0;j<22;j++){
            if(randomnumber==rnd.nextInt(21)){
                if (next_generation[k][j]==0) next_generation[k][j]=1;
                else next_generation[k][j]=0;
            } } }
}

```

Graph 2D:

x-generation

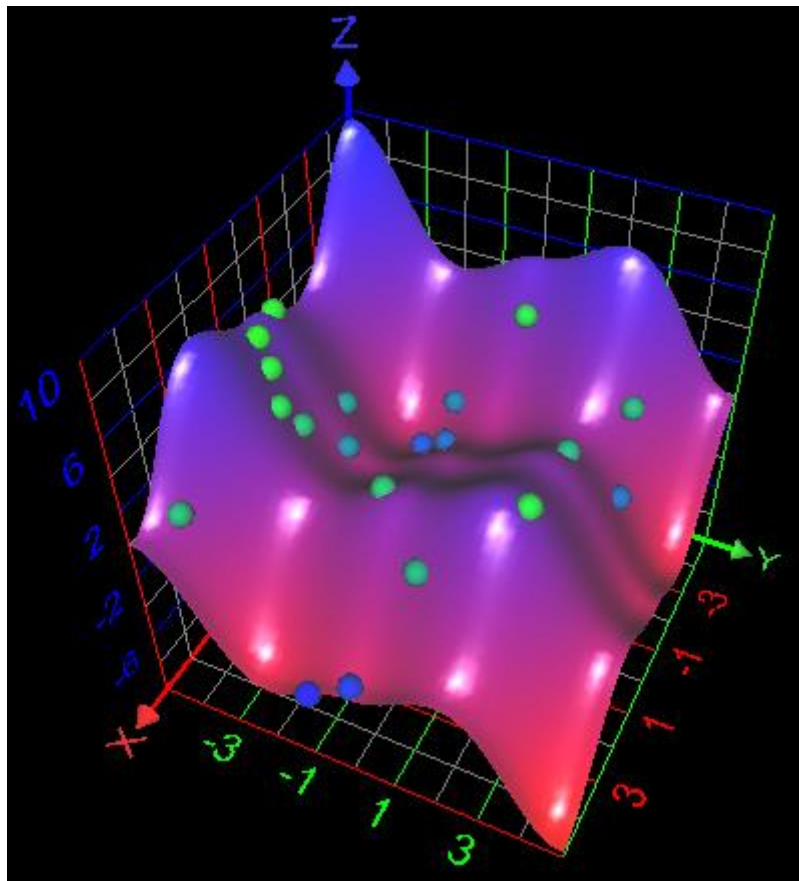
y-fitness (red— the best, blue— average)



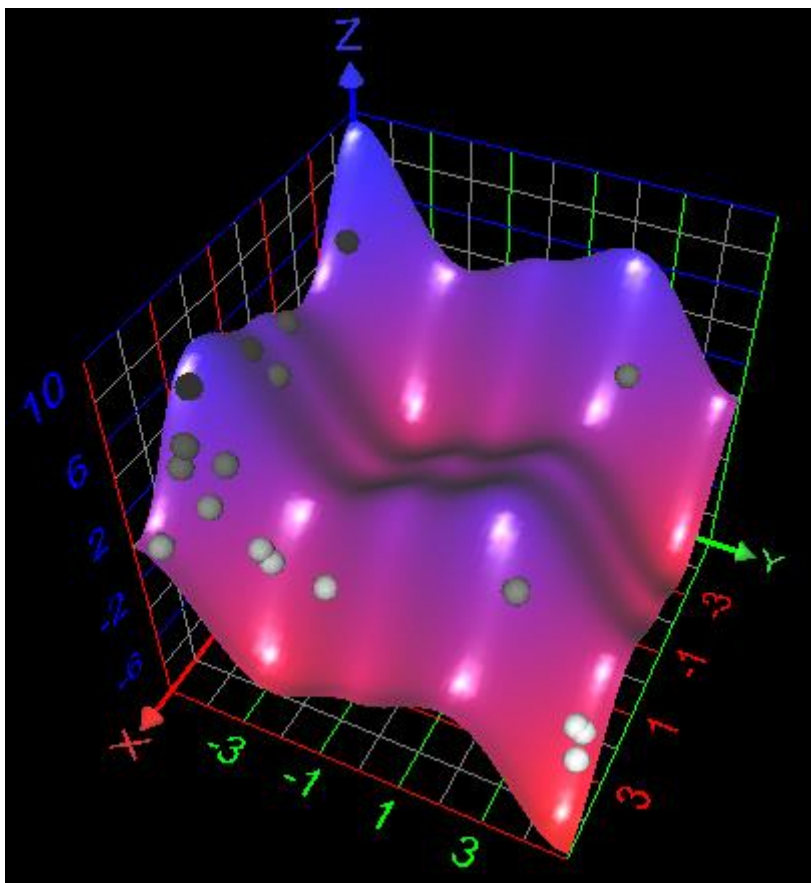
Graphs 3D:

x_1 -x, x_2 -y, y-z(fitness).

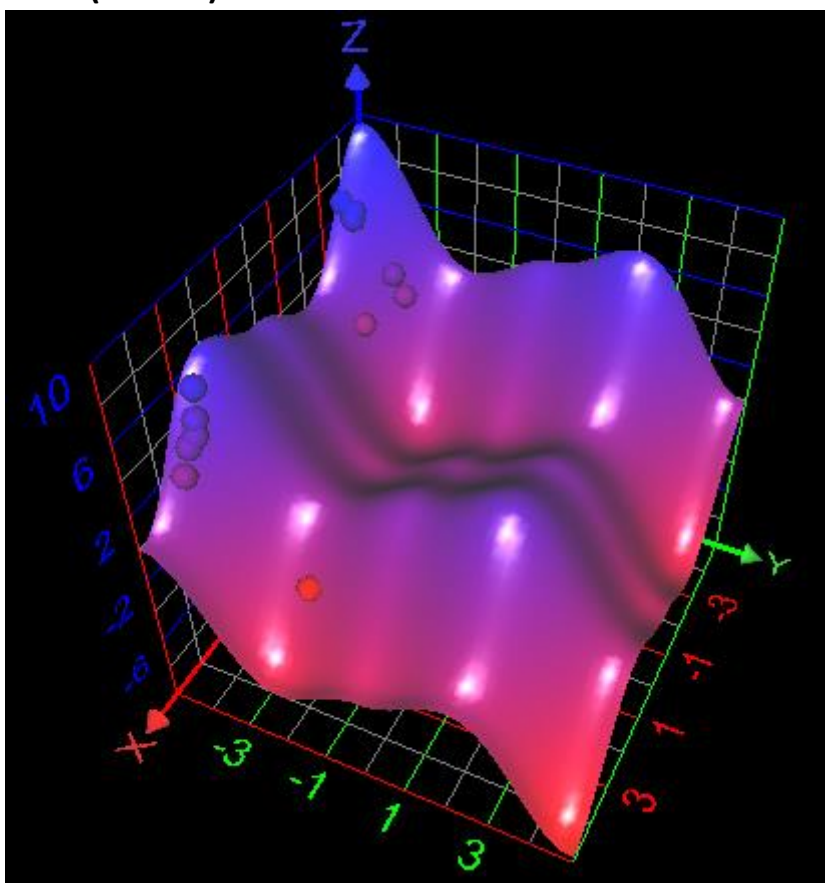
1-st



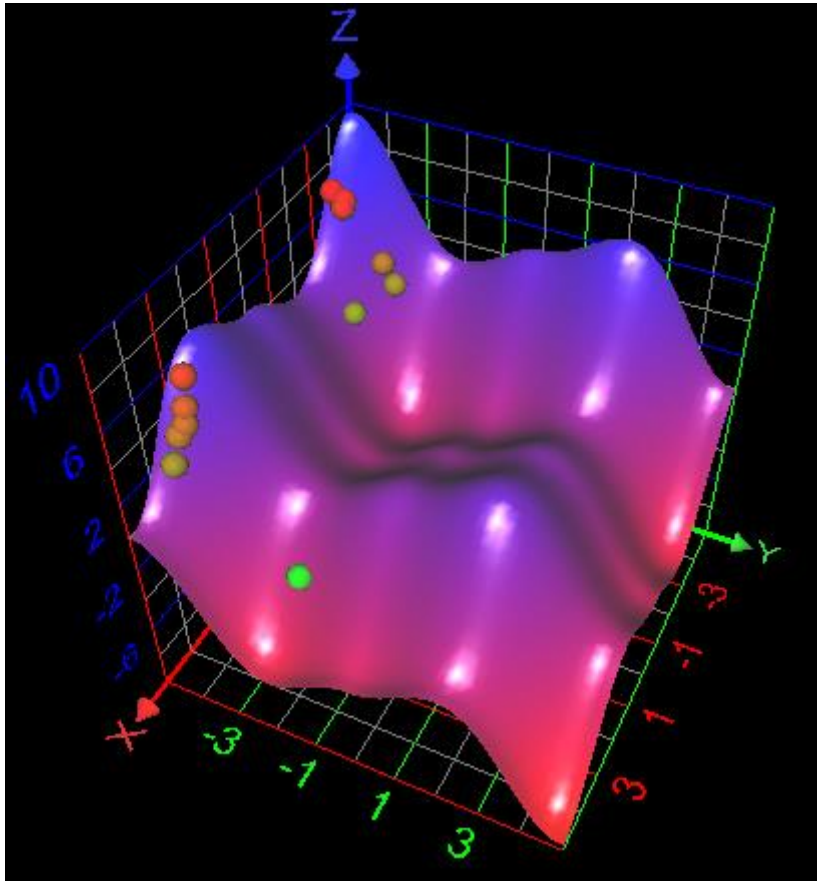
6-th



11-th(middle)



16-th



21-th(last)

