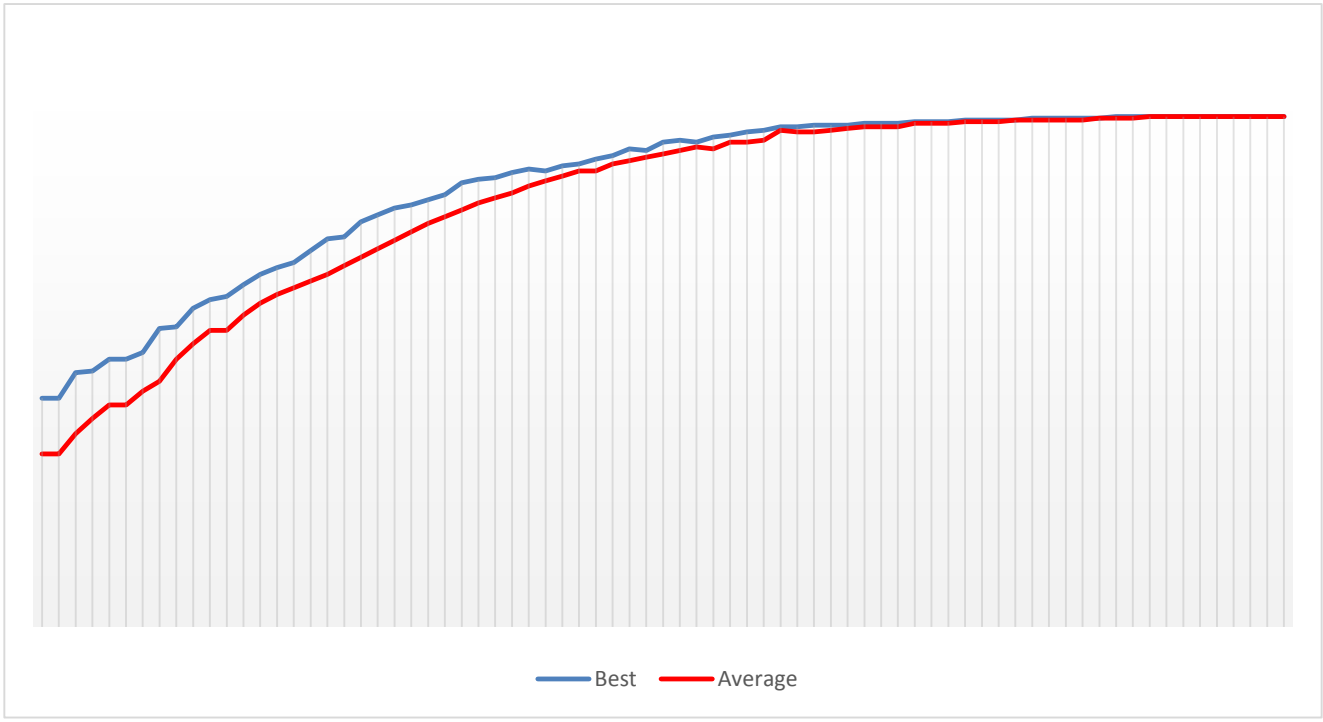
Task 3
3-Kravchuk Maxim

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.



The Code:

```
import java.util.Random;
public class Graf {
public static void main(String... args){ int[][] coord = new int[21][21]; Random random = new Random(); int iter = 0;

double avr = 0; double numb = 0; double z = 0; double x = 0; double y = 0;

z=x*Math.sin(Math.abs(x))+y*Math.sin(Math.abs(y));

for (int i = 0; i < 11; i++){ for (int j = 0; j < 22; j++){

coord[i][j] = random.nextInt(3);
}
}
```

```
for (int i = 0; i < 11; i++){ for (int j = 0; j < 22; j++){

coord[i][j]++; avr += coord[i][j];

}
}

System.out.println(avr / (11 * 22)); int[] temp = new int[22];
int[][] best = new int[5][111];
do{
for (int i = 0; i < 99; i++){ int min_fitness = i;
for(int j = i + 1; j < 11; j++) {
int fitMIN = pointToGr(coord[min_fitness]); int fitJ = pointToGr(coord[j]); if(fitJ<fitMIN)min_fitness = j;

}
if (min_fitness != i) { for(int k=0;k<22;k++)
temp[k]=coord[i][k];
for(int k=0;k<22;k++) coord[i][k]=coord[min_fitness][k];
for(int k=0;k<22;k++) coord[min_fitness][k]=temp[k];}
}
if(iter==0)
{
for(int s=0;s<5;s++)
{
coord c = new coord(); c.X= 50;

c.Y= 50;

for(int i=0;i<22;i++) { switch (coord[s][i]) {
case 0: c.Y--; break;
case 1: c.Y++; break;
case 2: c.X--; break;
case 3: c.X++; break;

default:
System.out.println("Error");
break;
}

System.out.println("X" + s + c.X); System.out.println("Y" + s + c.Y + "\n");

}
}

}
double AvrFit=0; int BestFit=0;

int numOfBest=0; coord c;

for(int i=0;i<5;i++) {
```

```
c = getCoord(coord[i]); if(c.X<0 || c.Y<0)continue; numOfBest++;

for(int j=0;j<22;j++) { best[i][j]=coord[i][j];

}
AvrFit += pointToGr(best[i]);

}
AvrFit = AvrFit/5;
System.out.println("AvrFit = " + AvrFit);
BestFit = pointToGr(best[0]);
System.out.println("BestFit = " + BestFit + "\n");
if(BestFit==0)break; int mom=0;

int dad=0; int cat=0;

int[] frstChild = new int[22]; int[] ScndChild = new int[22]; for(int i=0;i<11;i++) {

do
{
mom = random.nextInt(numOfBest); dad = random.nextInt(numOfBest);

}while(mom == dad); cat=random.nextInt(22); for(int k=0;k<cat;k++) {
frstChild[k]=best[mom][k];
ScndChild[k]=best[dad][k];
}
for(int k=cat;k<22;k++)
{
frstChild[k]=best[dad][k];
ScndChild[k]=best[mom][k];
}

int fitness1=0; int fitness2=0;
fitness1 = pointToGr(frstChild); fitness2 = pointToGr(ScndChild);

if(fitness1<fitness2)
{
for(int k=0;k<22;k++) coord[i][k]=frstChild[k];
}

else
{
for(int k=0;k<22;k++) coord[i][k]=ScndChild[k];
}
}

iter++;

}while (iter < 11);
```

```
}
static class coord{ int X;

int Y;
}

public static int pointToGr(int[] graf){ coord c = new coord();

c.X = 50; c.Y = 50;

for(int i=0;i<22;i++) { switch (graf[i]) {

case 0: c.Y--; break;

case 1: c.Y++; break;

case 2: c.X--; break;

case 3: c.X++; break;
default:

System.out.println("Error");
break;
}
}
int steps=0;

steps=Math.abs(c.X - 800) + Math.abs(c.Y - 800); return steps;
}
public static coord getCoord(int[] graf){ coord c = new coord();

c.X= 50; c.Y= 50;

for(int i=0;i<22;i++) { switch (graf[i]) {

case 0: c.Y--; break;

case 1: c.Y++; break;

case 2:
c.X--; break;

case 3: c.X++; break;

default:

System.out.println("Error");

break;
```

```
}

}
return c;
}
public static int isFitnessGood(int[][] coord){ int count = 0;

for (int i = 0; i < 5; i++){ count = 0;

for (int j = 0; j < 22; j++){ count += coord[i][j];

}

if(count == 22){ return i;

}
}
return -1;
}
}

}
```