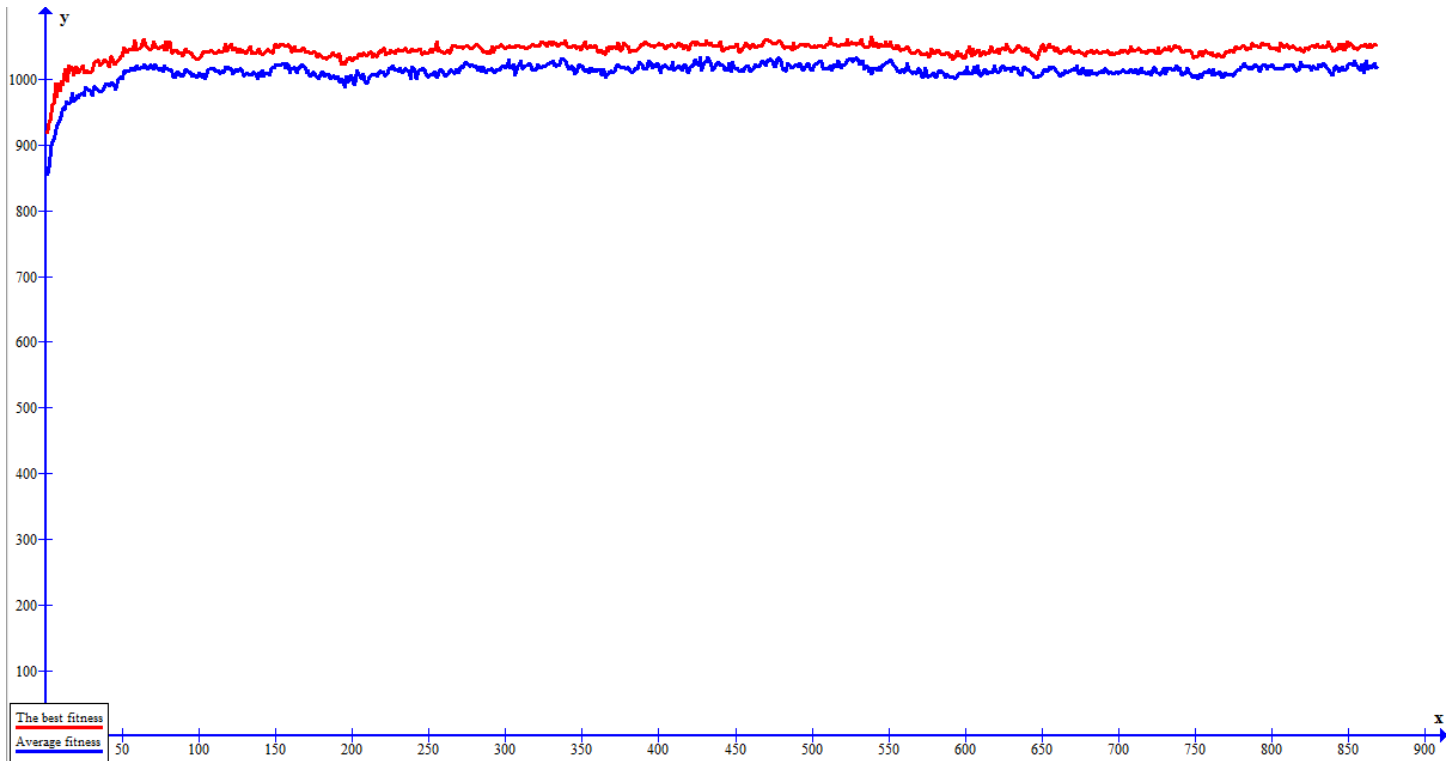


4 - Andrey Cheslow

Graphs:

Max fitness = 1066.

Generations = 1000.



Price = 1 – 10

Volume = 1 - 10

Items	1	2	3	4	5	6	7	8	9	10
Price	1	5	9	3	6	5	3	3	6	7
Volume	2	5	7	1	1	3	4	8	10	3

Items	11	12	13	14	15	16	17	18	19	20
Price	5	7	10	2	9	6	1	10	5	7
Volume	7	8	4	9	10	9	2	8	6	7

5 better chromosomes in generation:

Max size of a knapsack ≤ 1000 .

1st generation:

Chromosomes	1	2	3	4	5
Items count	12	1	12	5	12
	13	2	13	11	13
	10	12	1	3	10
	2	12	6	11	2
	7	4	9	8	7
	9	8	9	8	2
	1	9	12	8	4
	6	2	9	3	6
	8	3	10	7	8
	8	13	7	1	8
	9	1	7	9	9
	8	7	5	13	8
	7	9	13	13	7
	11	10	3	4	11
	10	10	6	3	10
	4	8	12	13	4
	5	12	9	2	5
	9	11	10	7	9
	12	12	11	13	12
	10	10	4	13	10
Total Volume	965	872	907	889	956
Total Price	917	915	899	897	891

500st generation:

Chromosomes	1	2	3	4	5
Items count	8	7	6	6	6
	8	8	8	8	8
	9	9	9	9	9
	8	8	8	8	8
	8	9	8	8	9
	9	9	9	9	9
	9	9	9	9	7
	3	3	3	3	4
	9	9	9	9	9
	9	9	9	9	9
	7	6	7	7	6
	13	13	13	13	13
	13	13	13	13	13
	1	1	1	1	1
	9	9	9	9	9
	13	13	13	13	13
	7	5	7	7	7
	7	7	7	7	7
	13	13	13	13	13
	13	13	13	13	13
Total Volume	999	987	995	995	989
Total Price	1049	1047	1047	1047	1045

1000st generation:

Chromosomes	1	2	3	4	5
Items count	7	7	4	6	7
	8	8	8	9	8
	9	9	9	9	9
	6	9	6	6	6
	9	8	8	8	8
	9	8	9	9	9
	6	6	8	7	6
	2	2	2	2	2
	7	7	7	6	7
	9	9	9	9	9
	8	8	8	8	8
	13	13	13	13	13
	13	13	13	13	13
	3	3	2	2	3
	9	9	9	9	9
	13	13	13	13	13
	7	7	8	8	7
	9	9	9	9	9
	13	13	13	13	13
	13	13	13	13	13
Total Volume	997	996	991	986	996
Total Price	1053	1051	1049	1047	1047

Source Code:

```
package siit_4;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Random;
/**
 *
 * @author Andrey
 */
public class Siit_4 {
    public static void main(String[] args) throws IOException {
        FileWriter writ = new FileWriter("C:\\Users\\Andrey\\Desktop\\Price&V.txt", false );
        writ.close();
        int[] price = new int [20];
        double[] V = new double [20];
        int [] fitness=new int [40];
        int[][] generation=new int[40][20];
        int[][] next_generation=new int[40][20];
        Random rnd=new Random();
        for(int j=0;j<20;j++){
            price [j]=(rnd.nextInt(10))+1;//max price 1-10
        }
        for(int j=0;j<20;j++){
            V[j]=(rnd.nextInt(10))+1;// max V 1-10
        }
        int inn=0;
        FileWriter writer = new FileWriter("C:\\Users\\Andrey\\Desktop\\Price&V.txt", true);
        String fff="Price\\tV";
        writer.write(fff);
        writer.write("\\r\\n");
        for(int j=0;j<20;j++){
            fff="";
            fff+=price[j]+"\\t"+V[j];
            writer.write(fff);
            writer.write("\\r\\n");
            writer.flush();
        }
        double sumka=0;

        for(int i=0;i<40;i++){
            sumka=100000;
            while(sumka>10000){
                for(int j=0;j<20;j++){
                    generation[i][j]=(rnd.nextInt(100));
                }
                sumka=0;
                for(int j=0;j<20;j++){
                    sumka+=V[j]*generation[i][j];
                }
                // System.out.println(sumka);
            }
            System.out.println("-----");
        }

        while(inn<300){
            for(int i=0;i<20;i++){
                fitness[i]=0;
            }
            for(int i=0;i<20;i++){
                fitness(price,generation,fitness,i,V);
            }
            sort(fitness,generation);
        }
    }
}
```

```

        inn++;
        System.out.println(inn+" "+fitness[0]);
        make_new_gen(generation,next_generation,V);
        generation=next_generation;
    }
}

public static void fitness(int[] price,int[][] generation,int[] fitness,int number,double[] V){
    double temp=0;
    for(int i=0;i<20;i++){
        fitness[number]+=price[i]*generation[number][i];
        temp+=V[i]*generation[number][i];
    }
    if(temp>10000)fitness[number]=0;
}

public static void sort(int[] fitness,int[][] generation){
    for(int i=0;i<39;i++){
        for(int j=i+1;j<40;j++){
            if (fitness[j]>fitness[i]){
                int 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,double[] V){
    Random rnd= new Random();
    for(int i=0;i<20;i++){
        int nump1=rnd.nextInt(20);
        int nump2=rnd.nextInt(20);
        for(int f=0;f<20;f++){
            int orrr=rnd.nextInt(2);
            if(orrr==0)
                generation[i+20][f]=generation[nump1][f];
            else generation[i+20][f]=generation[nump2][f];
        }
    }
    int randomnumber=11;
    for(int k=0;k<40;k++){
        for(int j=0;j<20;j++){
            if(randomnumber==rnd.nextInt(20)){
                next_generation[k][j]=(rnd.nextInt(100));
            }
        }
    }
}
}

```