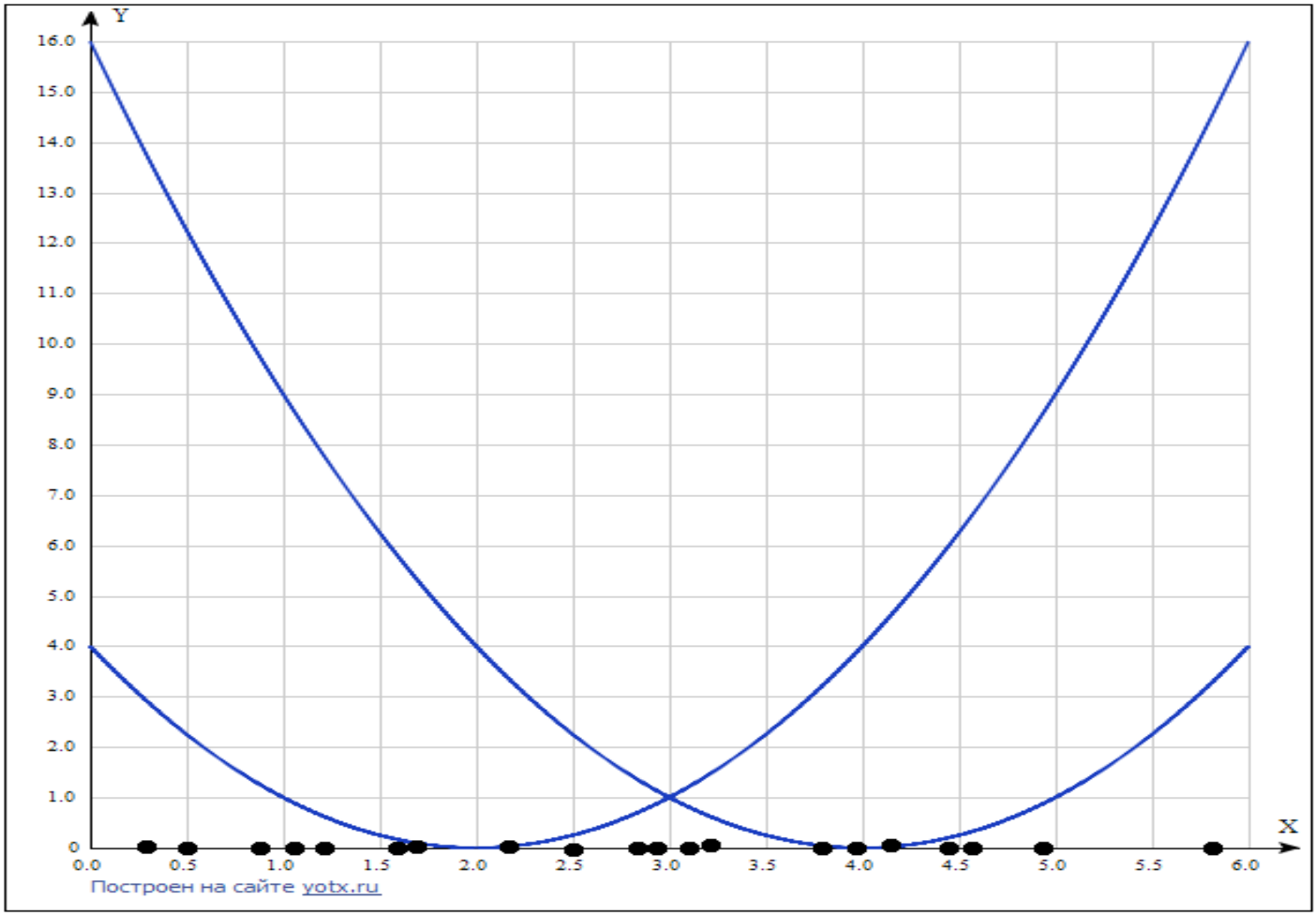
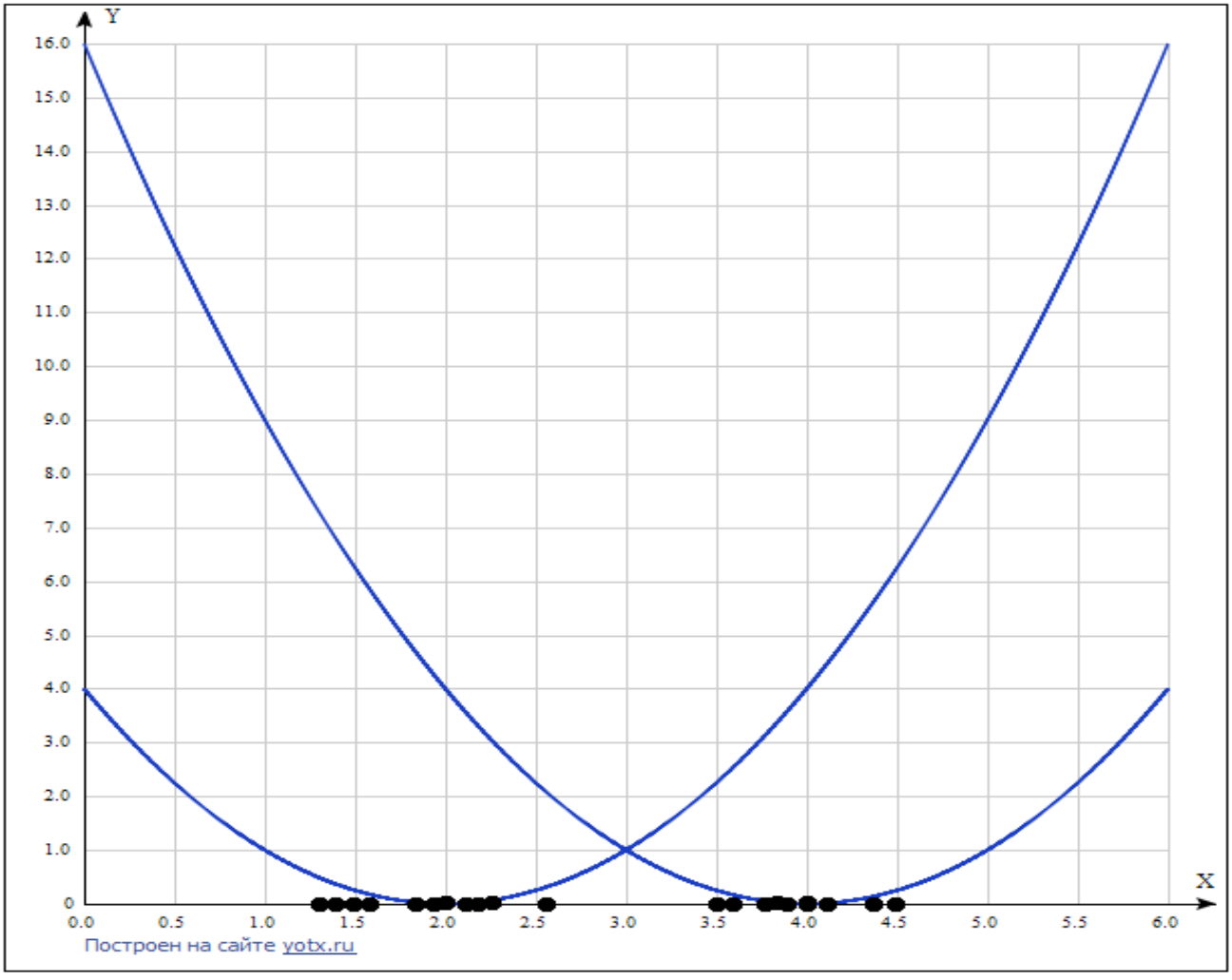
Yury Yuryn (II-10)
Parate Optimum Solution- Dominate & Rank
Iteration №1:

Binary										X	Y1	Y2	Rank
1	1	0	0	0	0	1	0	0	0	x=4.551319648093842	y1=6.509231946749686	y2=0.3039533543743178	Rank=2
1	0	1	1	1	1	1	0	0	0	x=4.457478005865102	y1=6.039198149310719	y2=0.2092861258503105	Rank=3
0	1	0	1	1	1	1	1	0	0	x=2.228739002932551	y1=0.052321531462577626	y2=3.137365519732373	Rank=7
0	0	1	1	1	0	0	1	0	0	x=1.3372434017595307	y1=0.4392463085112788	y2=7.0902727014731575	Rank=4
0	1	0	0	1	1	1	1	1	0	x=1.8651026392961876	y1=0.018197297924854475	y2=4.557786740740103	Rank=6
0	0	0	0	1	1	1	1	0	1	x=0.35777126099706746	y1=2.6969152312071616	y2=13.26583018721889	Rank=0
0	1	1	1	1	0	0	0	1	0	x=2.8269794721407626	y1=0.6838950473422144	y2=1.3759771587791638	Rank=5
1	0	0	1	1	0	0	1	1	1	x=3.6070381231671553	y1=2.582571529312613	y2=0.15441903664399179	Rank=5
1	0	1	1	0	1	1	1	0	0	x=4.293255131964809	y1=5.259019100282934	y2=0.08599857242369761	Rank=4
1	0	0	0	1	0	1	1	1	1	x=3.278592375366569	y1=1.6347984623455256	y2=0.5204289608792492	Rank=4
0	0	0	1	0	1	1	1	0	1	x=0.5454545454545454	y1=2.115702479338843	y2=11.933884297520661	Rank=1
1	1	1	1	0	1	1	1	1	1	x=5.812316715542522	y1=14.533758739604924	y2=3.284491877434835	Rank=0
1	0	0	1	1	1	1	1	0	0	x=3.730205278592375	y1=2.9936103060689185	y2=0.0727891916994179	Rank=5
1	1	0	1	0	0	1	1	1	0	x=4.961876832844575	y1=8.77271437294141	y2=0.9252070415631101	Rank=1
0	0	1	0	0	0	1	0	0	0	x=0.7976539589442815	y1=1.4456360024423596	y2=10.255020166665233	Rank=2
0	1	0	0	0	1	1	0	0	0	x=1.6422287390029326	y1=0.1280002751954317	y2=5.5590853191837	Rank=5
0	0	1	0	1	1	1	1	1	0	x=1.1143695014662756	y1=0.7843413799330933	y2=8.32686337406799	Rank=3
0	1	1	0	1	0	0	1	1	1	x=2.4809384164222874	y1=0.23130176039077752	y2=2.307548094701628	Rank=6
1	0	1	0	0	1	1	1	0	0	x=3.9178885630498534	y1=3.6782965402774317	y2=0.006742288078017893	Rank=5
1	0	0	0	1	0	0	1	1	1	x=3.231671554252199	y1=1.5170148175540281	y2=0.5903286005452313	Rank=4



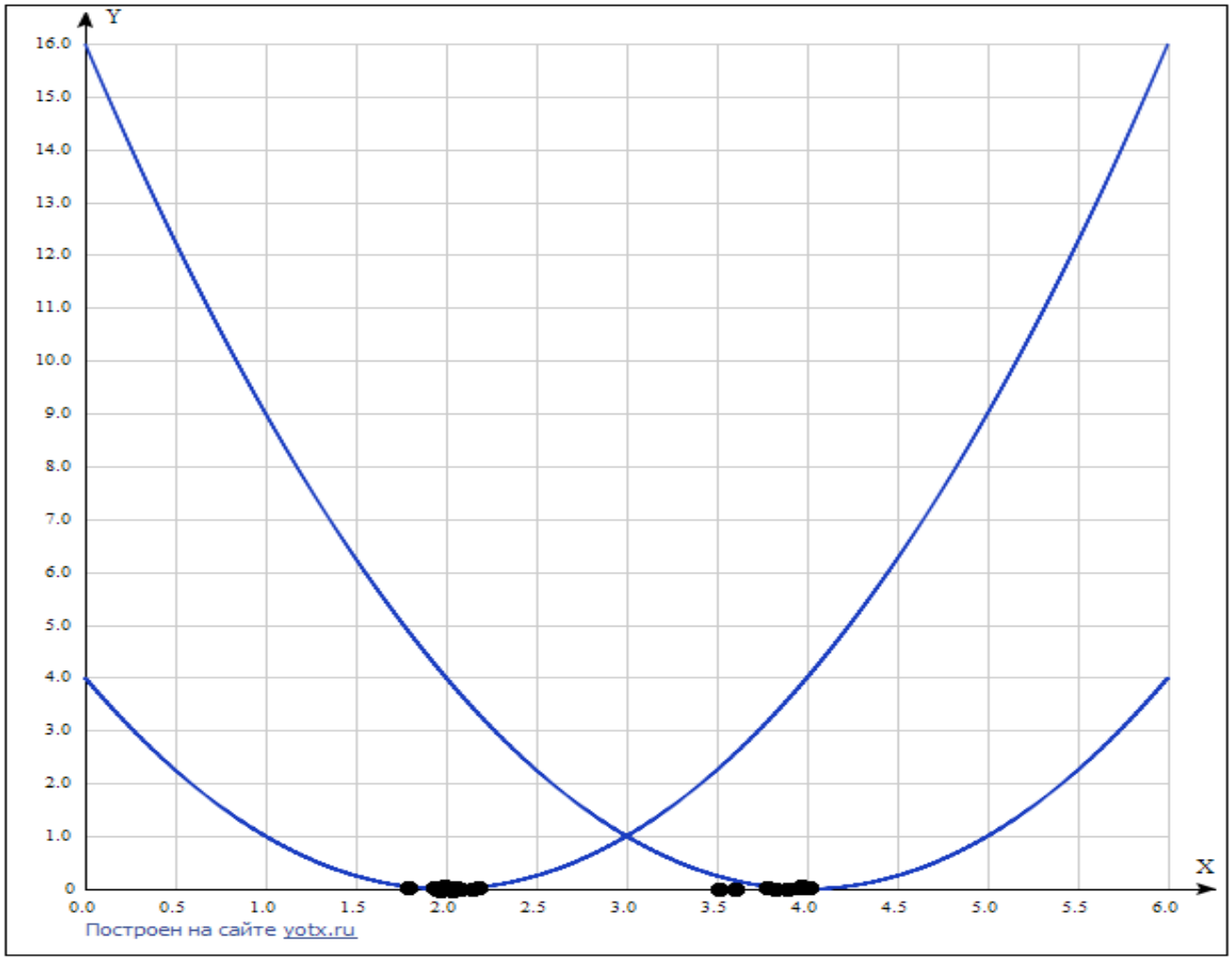
Iteration 350:

Binary										X	Y1	Y2	Rank
1	0	1	1	0	1	0	0	0	1	x=4.228739002932551	y1=4.9672775431927825	y2=0.052321531462577626	Rank=2
1	0	1	0	1	0	1	0	0	1	x=3.994134897360704	y1=3.9765739888717846	y2=3.4399428969479045E-5	Rank=3
0	1	0	1	1	0	1	1	1	1	x=2.1524926686217007	y1=0.023254013983367836	y2=3.413283339496565	Rank=5
0	1	0	0	1	0	1	0	1	1	x=1.7536656891495601	y1=0.06068059270216115	y2=5.046017836103919	Rank=4
0	1	0	1	0	1	1	0	1	0	x=2.029325513196481	y1=8.599857242369761E-4	y2=3.8835579329383134	Rank=6
0	0	1	1	1	0	1	0	0	0	x=1.3607038123167154	y1=0.4086996155863814	y2=6.9658843663195205	Rank=0
0	1	1	0	0	0	1	1	1	1	x=2.340175953079179	y1=0.1157196790533278	y2=2.7550158667366116	Rank=4
1	0	0	1	1	0	1	1	1	1	x=3.653958944281525	y1=2.735580189368856	y2=0.11974441224275686	Rank=2
1	0	1	0	0	1	1	0	0	1	x=3.900293255131965	y1=3.611114455500039	y2=0.009941434972179444	Rank=3
1	0	0	1	0	1	0	1	1	0	x=3.5073313782991202	y1=2.2720478840051257	y2=0.2427223708086446	Rank=0
0	0	1	1	1	1	1	0	1	0	x=1.466275659824047	y1=0.28486167129625645	y2=6.419759032000068	Rank=1
1	0	1	1	1	1	1	0	1	1	x=4.475073313782991	y1=6.125987908600715	y2=0.225694653468752	Rank=0
1	0	0	1	1	1	1	0	1	0	x=3.718475073313783	y1=2.953156577600811	y2=0.07925628434567997	Rank=2
1	0	1	1	1	1	1	0	0	0	x=4.457478005865102	y1=6.039198149310719	y2=0.2092861258503105	Rank=1
0	1	0	0	0	0	0	1	1	1	x=1.5425219941348973	y1=0.20928612585031092	y2=6.039198149310722	Rank=2
0	1	0	1	0	0	1	0	1	1	x=1.9413489736070382	y1=0.0034399428969479045	y2=4.238044048468796	Rank=5
0	1	0	0	0	0	1	1	1	0	x=1.5835777126099706	y1=0.17340752143514424	y2=5.839096670995262	Rank=3
0	1	0	1	1	1	1	0	1	0	x=2.2170087976539588	y1=0.047092818259216816	y2=3.179057627643382	Rank=5
1	0	1	0	0	0	1	0	1	1	x=3.8181818181818183	y1=3.3057851239669427	y2=0.03305785123966936	Rank=3
0	1	1	0	1	1	1	1	1	1	x=2.621700879765396	y1=0.3865119839010671	y2=1.899708464839484	Rank=1



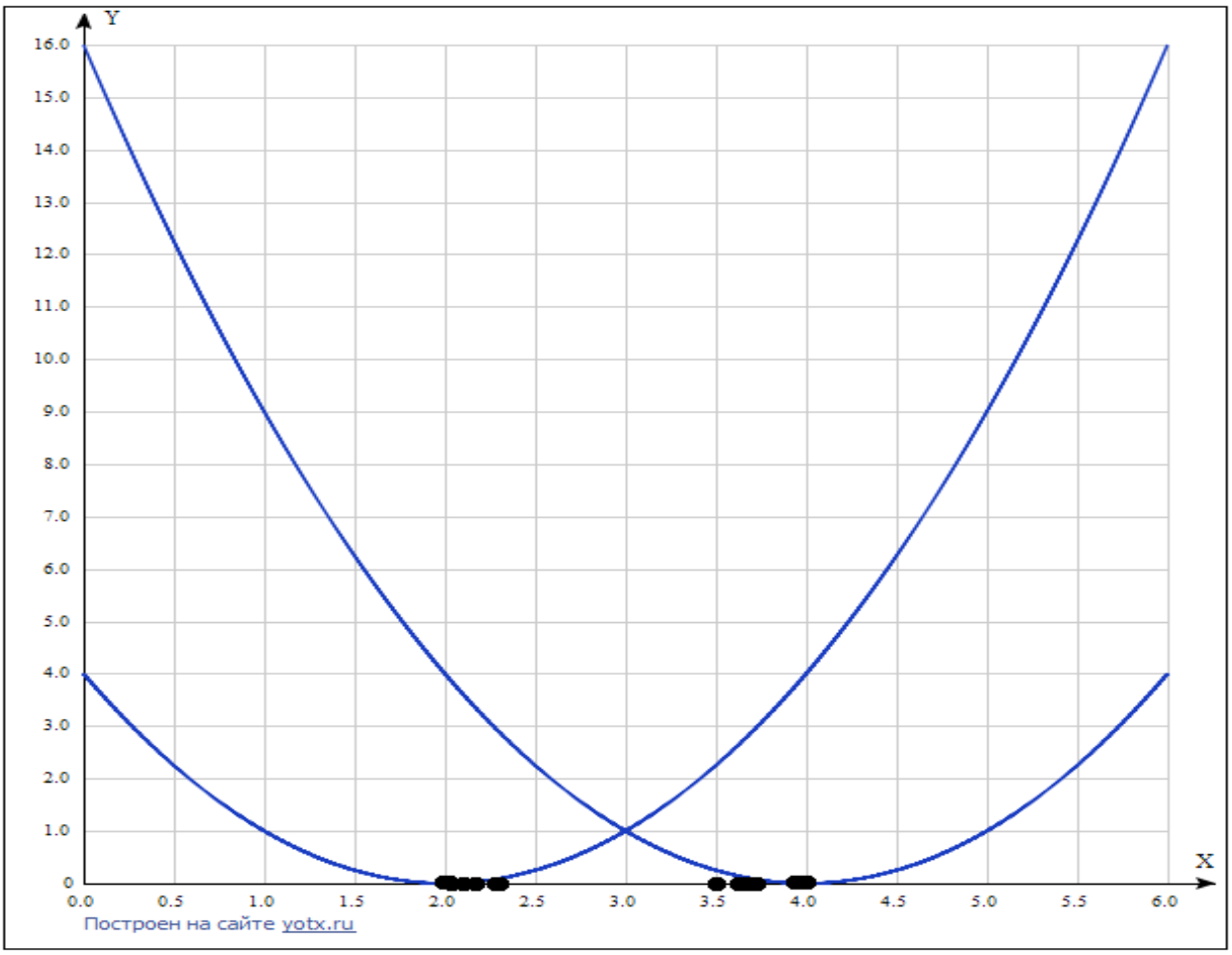
Iteration 700:

Binary										X	Y1	Y2	Rank
1	0	1	0	1	0	1	0	1	0	x=4.0	y1=4.0	y2=0.0	Rank=2
1	0	1	0	1	0	1	0	0	0	x=3.9882697947214076	y1=3.9532167766015083	y2=1.3759771587791618E-4	Rank=0
0	1	0	1	1	0	1	0	1	1	x=2.129032258064516	y1=0.016649323621227858	y2=3.5005202913631637	Rank=1
0	1	0	1	0	1	0	1	1	0	x=2.005865102639296	y1=3.4399428969479045E-5	y2=3.9765739888717846	Rank=3
0	1	0	1	1	0	0	0	1	0	x=2.0762463343108504	y1=0.005813503495841959	y2=3.7008281662524403	Rank=1
0	1	0	0	1	1	1	1	1	1	x=1.8709677419354838	y1=0.016649323621227917	y2=4.532778355879292	Rank=0
0	1	1	0	0	0	0	1	1	0	x=2.2873900293255134	y1=0.08259302895571945	y2=2.9330329116536658	Rank=0
1	0	0	1	1	0	1	0	1	1	x=3.63049853372434	y1=2.658525468477223	y2=0.13653133357986266	Rank=0
1	0	1	0	0	1	1	0	0	1	x=3.900293255131965	y1=3.611114455500039	y2=0.009941434972179444	Rank=0
1	0	0	1	0	1	0	1	1	0	x=3.5073313782991202	y1=2.2720478840051257	y2=0.2427223708086446	Rank=0
0	1	0	1	0	0	1	0	0	0	x=1.9237536656891496	y1=0.005813503495841959	y2=4.310798840739244	Rank=1
1	0	1	0	1	0	1	1	0	0	x=4.011730205278592	y1=4.047058418830248	y2=1.3759771587791618E-4	Rank=0
1	0	0	1	1	1	0	1	1	1	x=3.7008797653958942	y1=2.8929919763331924	y2=0.08947291474961526	Rank=0
1	0	1	0	1	0	1	0	1	1	x=4.005865102639296	y1=4.023494809986154	y2=3.4399428969479045E-5	Rank=1
0	1	0	1	0	1	0	0	0	0	x=1.970674486803519	y1=8.599857242369761E-4	y2=4.11816203851016	Rank=2
0	1	0	1	0	1	1	0	1	0	x=2.029325513196481	y1=8.599857242369761E-4	y2=3.8835579329383134	Rank=2
0	1	0	1	0	1	0	1	0	0	x=1.9941348973607038	y1=3.4399428969479045E-5	y2=4.023494809986154	Rank=3
0	1	0	1	1	1	0	1	0	1	x=2.187683284457478	y1=0.03522501526474654	y2=3.284491877434835	Rank=0
1	0	0	1	1	1	1	1	0	1	x=3.7360703812316713	y1=3.0139403685898807	y2=0.0696588436631953	Rank=0
0	1	1	0	0	1	0	1	0	0	x=2.36950146627566	y1=0.13653133357986266	y2=2.658525468477223	Rank=0

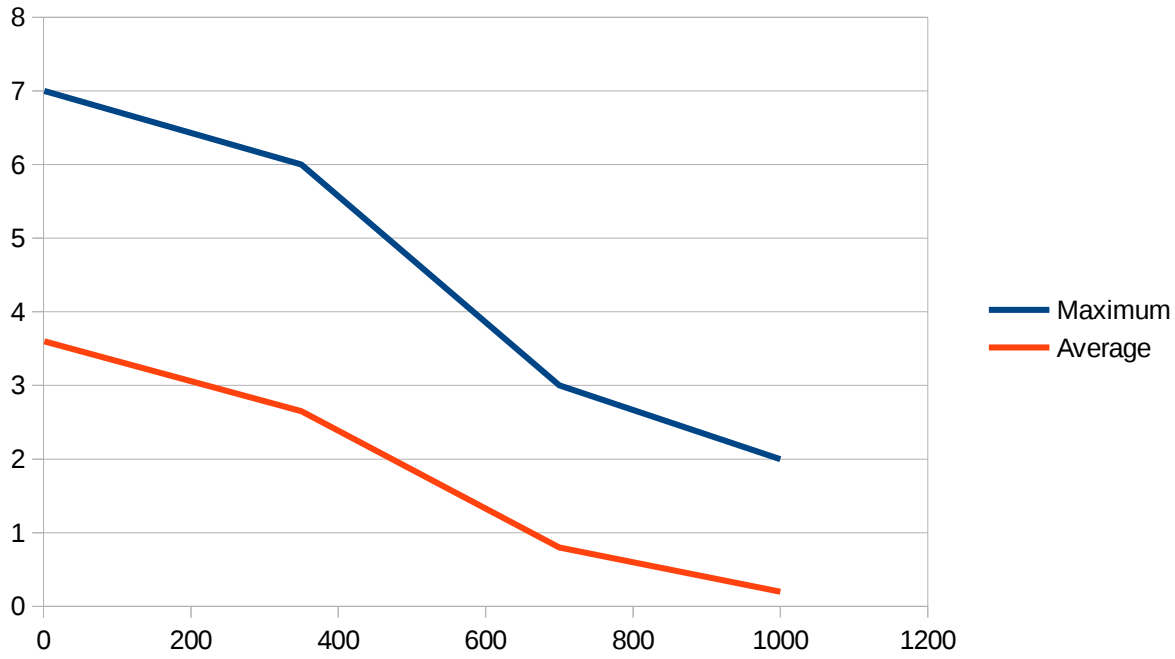


Last iteration:

Binary										X	Y1	Y2	Rank
1	0	1	0	1	0	1	0	1	0	x=4.0	y1=4.0	y2=0.0	Rank=2
1	0	1	0	1	0	1	0	0	0	x=3.9882697947214076	y1=3.9532167766015083	y2=1.3759771587791618E-4	Rank=0
0	1	0	1	1	0	1	0	1	1	x=2.129032258064516	y1=0.016649323621227858	y2=3.5005202913631637	Rank=0
0	1	0	1	0	1	1	0	1	0	x=2.029325513196481	y1=8.599857242369761E-4	y2=3.8835579329383134	Rank=0
0	1	0	1	0	1	1	1	0	1	x=2.0469208211143695	y1=0.002201563454046659	y2=3.814518278996569	Rank=0
0	1	0	1	0	1	0	1	0	0	x=1.9941348973607038	y1=3.4399428969479045E-5	y2=4.023494809986154	Rank=0
0	1	1	0	0	0	0	1	1	0	x=2.2873900293255134	y1=0.08259302895571945	y2=2.9330329116536658	Rank=0
1	0	0	1	1	0	1	0	1	1	x=3.63049853372434	y1=2.658525468477223	y2=0.13653133357986266	Rank=0
1	0	1	0	0	1	1	0	0	1	x=3.900293255131965	y1=3.611114455500039	y2=0.009941434972179444	Rank=0
1	0	0	1	0	1	0	1	1	0	x=3.5073313782991202	y1=2.2720478840051257	y2=0.2427223708086446	Rank=0
0	1	0	1	0	1	0	1	0	1	x=2.0	y1=0.0	y2=4.0	Rank=1
1	0	1	0	1	0	1	1	0	0	x=4.011730205278592	y1=4.047058418830248	y2=1.3759771587791618E-4	Rank=0
1	0	0	1	1	1	0	1	1	1	x=3.7008797653958942	y1=2.8929919763331924	y2=0.08947291474961526	Rank=0
1	0	1	0	1	0	1	0	1	1	x=4.005865102639296	y1=4.023494809986154	y2=3.4399428969479045E-5	Rank=1
0	1	0	1	0	1	0	1	1	0	x=2.005865102639296	y1=3.4399428969479045E-5	y2=3.9765739888717846	Rank=0
0	1	0	1	0	1	1	0	1	1	x=2.035190615835777	y1=0.0012383794429012456	y2=3.860475916099793	Rank=0
0	1	0	1	0	1	0	1	1	1	x=2.0117302052785924	y1=1.3759771587791618E-4	y2=3.9532167766015083	Rank=0
0	1	0	1	1	1	0	1	0	1	x=2.187683284457478	y1=0.03522501526474654	y2=3.284491877434835	Rank=0
1	0	0	1	1	1	1	1	0	1	x=3.7360703812316713	y1=3.0139403685898807	y2=0.0696588436631953	Rank=0
0	1	1	0	0	1	0	1	0	0	x=2.36950146627566	y1=0.13653133357	y2=2.658525468477223	Rank=0



Fitness(Rank) / generation:



Code of program:

```
public class Main {
    public static Double[][] params = new Double[20][2];
    public static Double[][] generation = new Double[20][10];
    public static Double[][] good = new Double[10][10];
    public static Double[][] childes = new Double[10][10];
    public static Double[] xs = new Double[20];
    public static Integer childCount = 0;
    public static Integer checkCount = 0;
    public static Double prevMax = 0.0;
    public static void main(String[] args) {
        Integer r = 0;
        createGenerate();
        showInfo();
        while(true) {
            Arrays.sort(generation, new Comparator<Double[]>() {
                @Override
                public int compare(Double[] o1, Double[] o2) {
                    Double int1 = fitness(o1);
                    Double int2 = fitness(o2);
                    return (int) (int2 - int1);
                }
            });
            Double max = fitness(generation[0]);
            if(prevMax.equals(max))
               checkCount ++;
            else
                checkCount = 0;
            prevMax = max;
            if(r.equals(5) || r.equals(3) || r.equals(7) || r.equals(0)) {
                System.out.println(returnMaxPrice(generation[0]));
                System.out.println(returnMaxVolume(generation[0]));
                showTable();
```

```
                }
                System.out.printf("%.3f %.3f", max,average());
                System.out.println();
                r++;
                if(checkCount > 30) {
                    System.out.println(returnMaxPrice(generation[0]));
                    System.out.println(returnMaxVolume(generation[0]));
                    showTable();
                    return;
                }
                setGood();
                Random random = new Random();
                childCount = 0;
                for (int i = 0; i < 5; i++) {
                    createChildes(good[random.nextInt(10)], good[random.nextInt(10)]);
                }
                createNewGeneration();
            }
        }
        public static void showTable() {
            System.out.println(" It. Pric. Vol.");
            for(int i=0;i<20;i++) {
                System.out.printf("%4d",generation[0][i].intValue());
                System.out.printf("%4d", params[generation[0][i].intValue()][0].intValue());
                System.out.printf("%4d", params[generation[0][i].intValue()][1].intValue());
                System.out.println();
            }
        }
        public static Double returnY1(Double x) {
            return Math.pow(x - 2,2);
        }
        public static Double returnY2(Double x) {
            return Math.pow(x - 4,2);
        }
        public static void createGenerate() {
            Random random = new Random();
            for (int j = 0; j < 10; j++) {
                generation[0][j] = 0.0;
            }
            for (int j = 0; j < 10; j++) {
                generation[19][j] = 1.0;
            }
            for(int i=1;i<19;i++) {
                for (int j = 0; j < 10; j++) {
                    generation[i][j] = random.nextInt(2) * 1.0;
                }
            }
        }
        public static void showInfo() {
            xs = new Double[20];
            for (int t=0;t<20;t++) {
                String str = "";
                for (int i = 0; i < 10; i++) {
                    str += String.valueOf(generation[t][i].intValue());
                }
                Double x = Integer.parseInt(str,2) / 170.5;
                xs[t] = x;
                System.out.printf("i=%s x=%.5f y1=%.5f y2=%.5f", str, x, returnY1(x),
returnY2(x));
```

```
            System.out.println();
        }
    }
    public static Double fitness(Double[] obj) {
        Double sumVolume = 0.0;
        Double sumPrice = 0.0;
        for (int i=0; i<obj.length;i++) {
            sumVolume += returnVolume(obj[i]);
        }
        for (int i=0; i<obj.length;i++) {
            sumPrice += returnPrice(obj[i]);
        }
        if(sumVolume < 100)
            return sumPrice * sumVolume;
        else return 0.0;
    }
    public static Double returnMaxPrice(Double obj[]) {
        Double sumPrice = 0.0;
        for (int i=0; i<obj.length;i++) {
            sumPrice += returnPrice(obj[i]);
        }
        return sumPrice;
    }
    public static Double returnMaxVolume(Double obj[]) {
        Double sumVolume = 0.0;
        for (int i=0; i<obj.length;i++) {
            sumVolume += returnVolume(obj[i]);
        }
        return sumVolume;
    }
    public static Double returnPrice(Double item) {
        return params[item.intValue()][0];
    }
    public static Double returnVolume(Double item) {
        return params[item.intValue()][1];
    }
    public static void setGood() {
        for(int i =0; i< 10; i++) {
            good[i] = generation[i];
        }
    }
    public static void createChildes(Double[] parent1, Double[] parent2) {
            Random random = new Random();
        Integer delimiter = random.nextInt(20);
        Double[] child1 = new Double[20];
        Double[] child2 = new Double[20];
        for(int i = 0;i<20;i++) {
            if(i<=delimiter)
                child1[i] = parent1[i];
            else
                child1[i] = parent2[i];
        }
        for(int i = 0;i<20;i++) {
            if(i<=delimiter)
                child2[i] = parent2[i];
            else
                child2[i] = parent1[i];
        }
        childes[childCount] = child1;
```

```
            childCount++;
            childes[childCount] = child2;
            childCount++;
        }
        public static void createNewGeneration() {
            for (int i=0;i<20;i++) {
                if(i<10)
                generation[i] = good[i];
                else
                    generation[i] = childes[i-10];
            }
        }
        public static Double average() {
            Double s = 0.0;
            for (int i=0;i<20;i++) {
                s+= fitness(generation[i]);
            }
            Double average = s / 20;
            return average;
        }
}
```