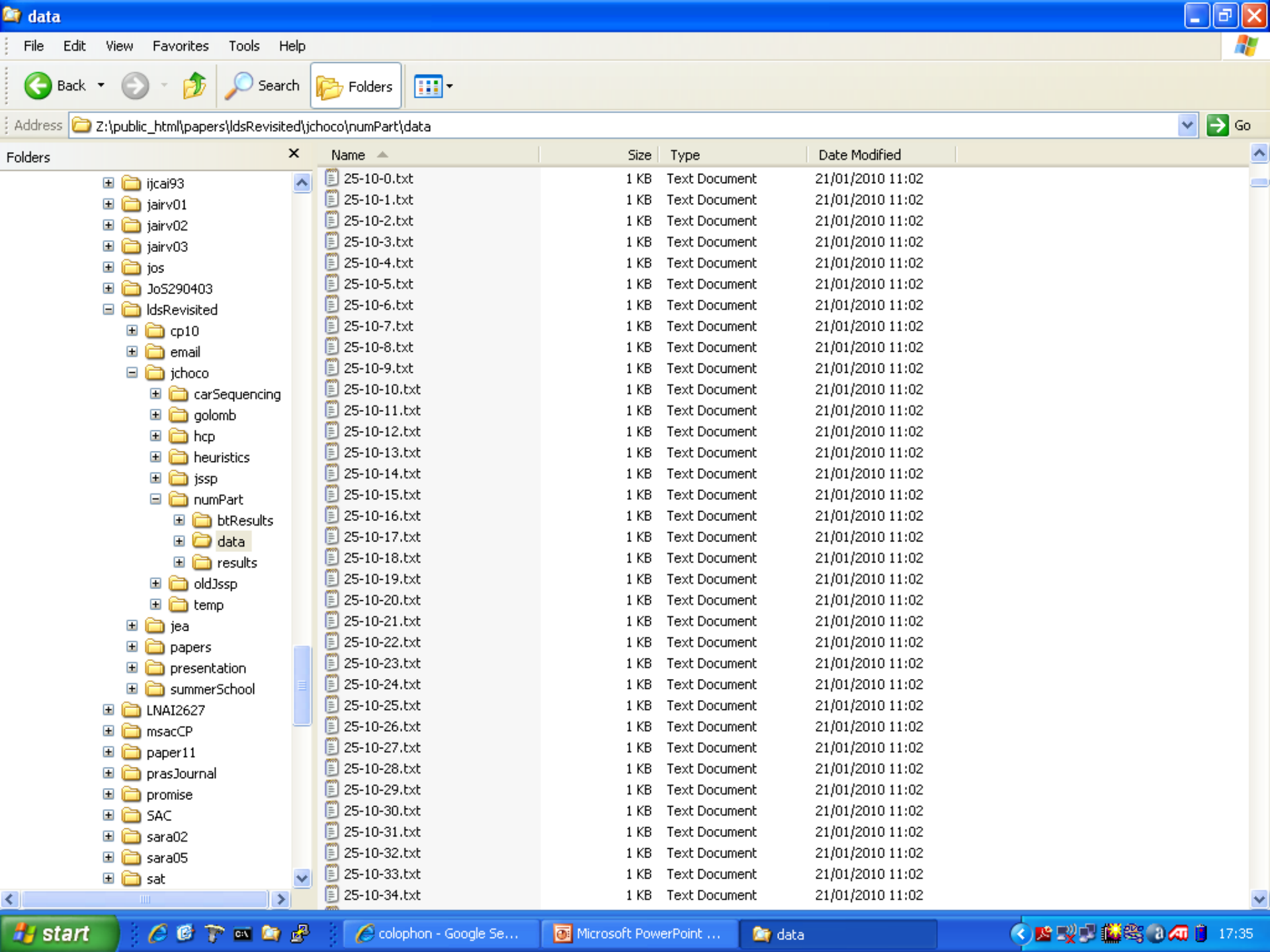


colophon

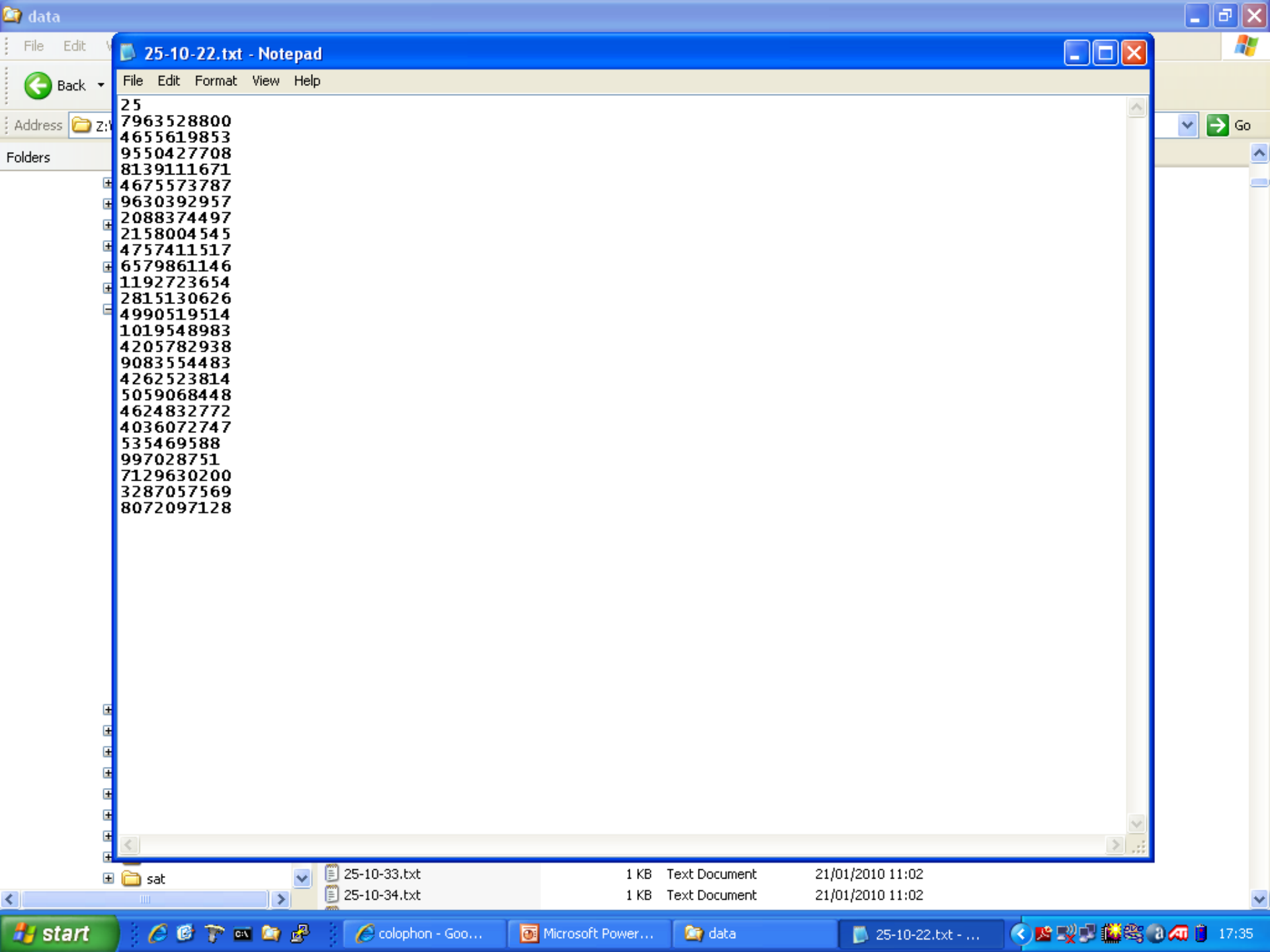
Or

"What I did early"

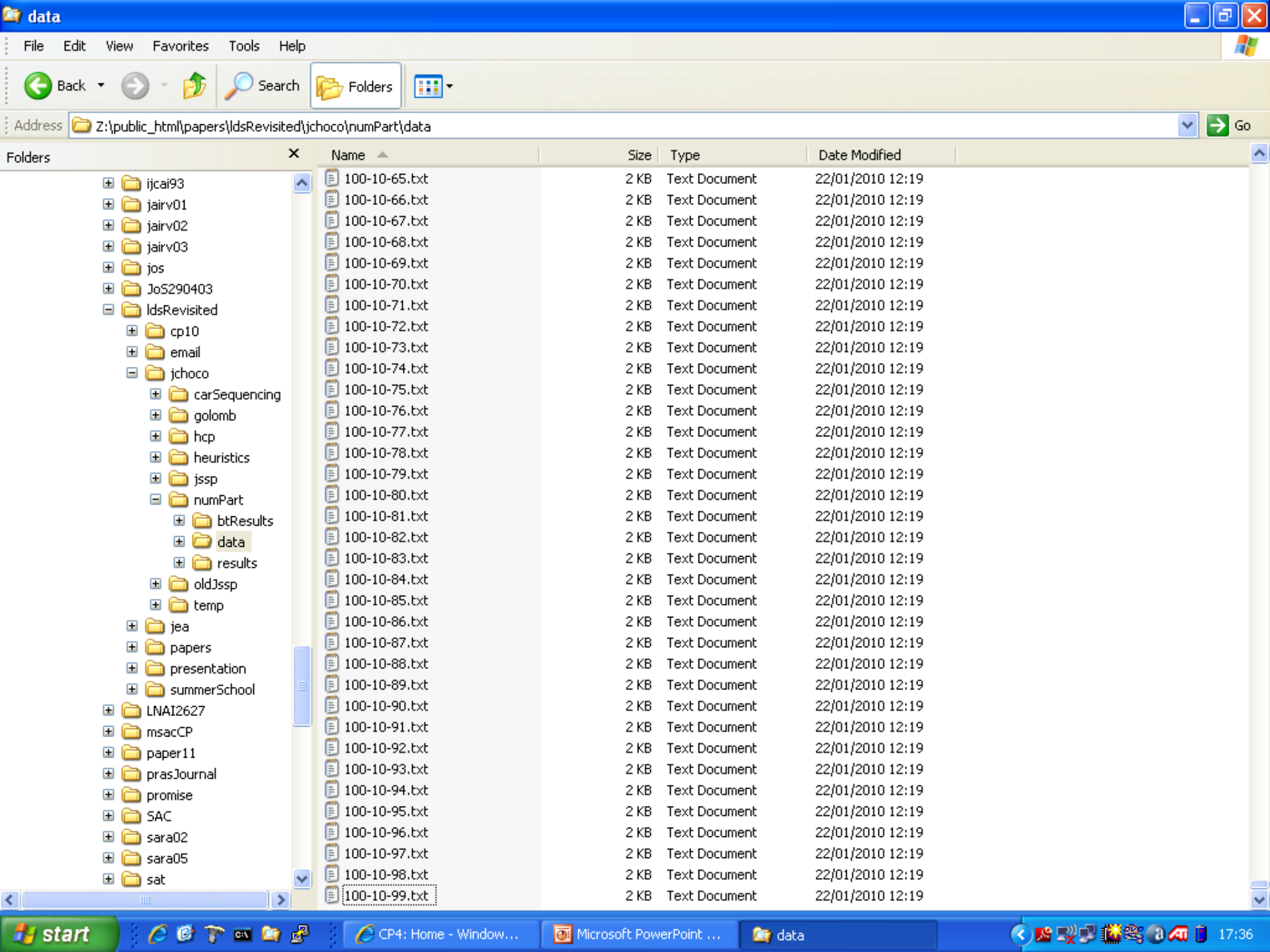
Experiments with  $25 \leq n \leq 100$ ,  $d=10$

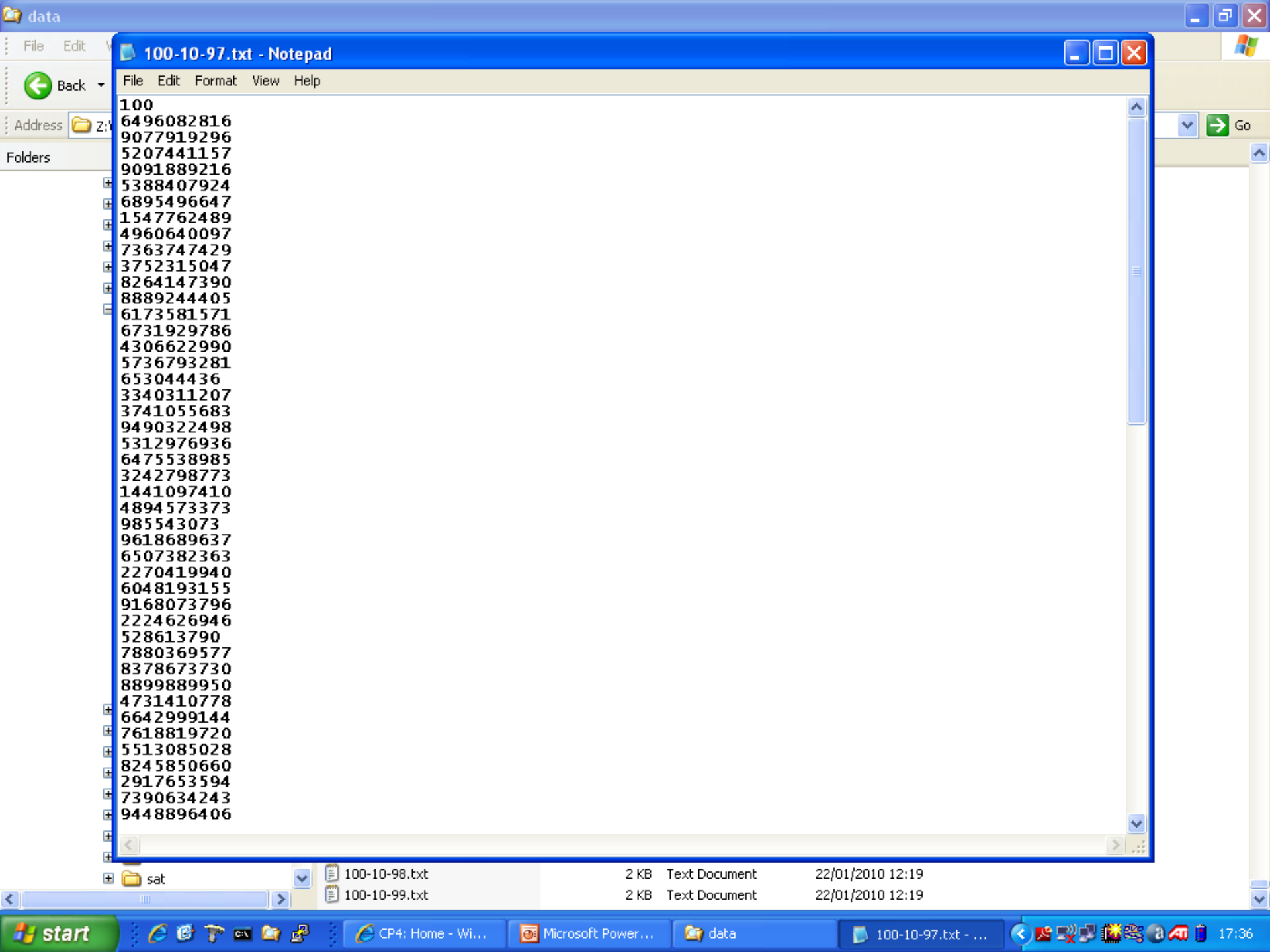


| data  |              |      |               |                  |  |
|---|--------------|------|---------------|------------------|--|
| File Edit View Favorites Tools Help                               |              |      |               |                  |  |
| Back Forward Up Search Folders                                    |              |      |               |                  |  |
| Address Z:\public_html\papers\ldsRevisited\jchoco\numPart\data Go |              |      |               |                  |  |
| Folders   | Name         | Size | Type          | Date Modified    |  |
| + jcai93  | 25-10-0.txt  | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + jairv01   | 25-10-1.txt  | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + jairv02   | 25-10-2.txt  | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + jairv03   | 25-10-3.txt  | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + jos   | 25-10-4.txt  | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + Jo5290403   | 25-10-5.txt  | 1 KB | Text Document | 21/01/2010 11:02 |  |
| - ldsRevisited  | 25-10-6.txt  | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + cp10  | 25-10-7.txt  | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + email   | 25-10-8.txt  | 1 KB | Text Document | 21/01/2010 11:02 |  |
| - jchoco  | 25-10-9.txt  | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + carSequencing   | 25-10-10.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + golomb  | 25-10-11.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + hcp   | 25-10-12.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + heuristics  | 25-10-13.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + jssp  | 25-10-14.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| - numPart   | 25-10-15.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + btResults   | 25-10-16.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + data  | 25-10-17.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + results   | 25-10-18.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + oldJssp   | 25-10-19.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + temp  | 25-10-20.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + jea   | 25-10-21.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + papers  | 25-10-22.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + presentation  | 25-10-23.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + summerSchool  | 25-10-24.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + LNAI2627  | 25-10-25.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + msacCP  | 25-10-26.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + paper11   | 25-10-27.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + prasJournal   | 25-10-28.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + promise   | 25-10-29.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + SAC   | 25-10-30.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + sara02  | 25-10-31.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + sara05  | 25-10-32.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
| + sat   | 25-10-33.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |
|   | 25-10-34.txt | 1 KB | Text Document | 21/01/2010 11:02 |  |



25  
7963528800  
4655619853  
9550427708  
8139111671  
4675573787  
9630392957  
2088374497  
2158004545  
4757411517  
6579861146  
1192723654  
2815130626  
4990519514  
1019548983  
4205782938  
9083554483  
4262523814  
5059068448  
4624832772  
4036072747  
535469588  
997028751  
7129630200  
3287057569  
8072097128





100

6496082816

9077919296

5207441157

9091889216

5388407924

6895496647

1547762489

4960640097

7363747429

3752315047

8264147390

8889244405

6173581571

6731929786

4306622990

5736793281

653044436

3340311207

3741055683

9490322498

5312976936

6475538985

3242798773

1441097410

4894573373

985543073

9618689637

6507382363

2270419940

6048193155

9168073796

2224626946

528613790

7880369577

8378673730

8899889950

4731410778

6642999144

7618819720

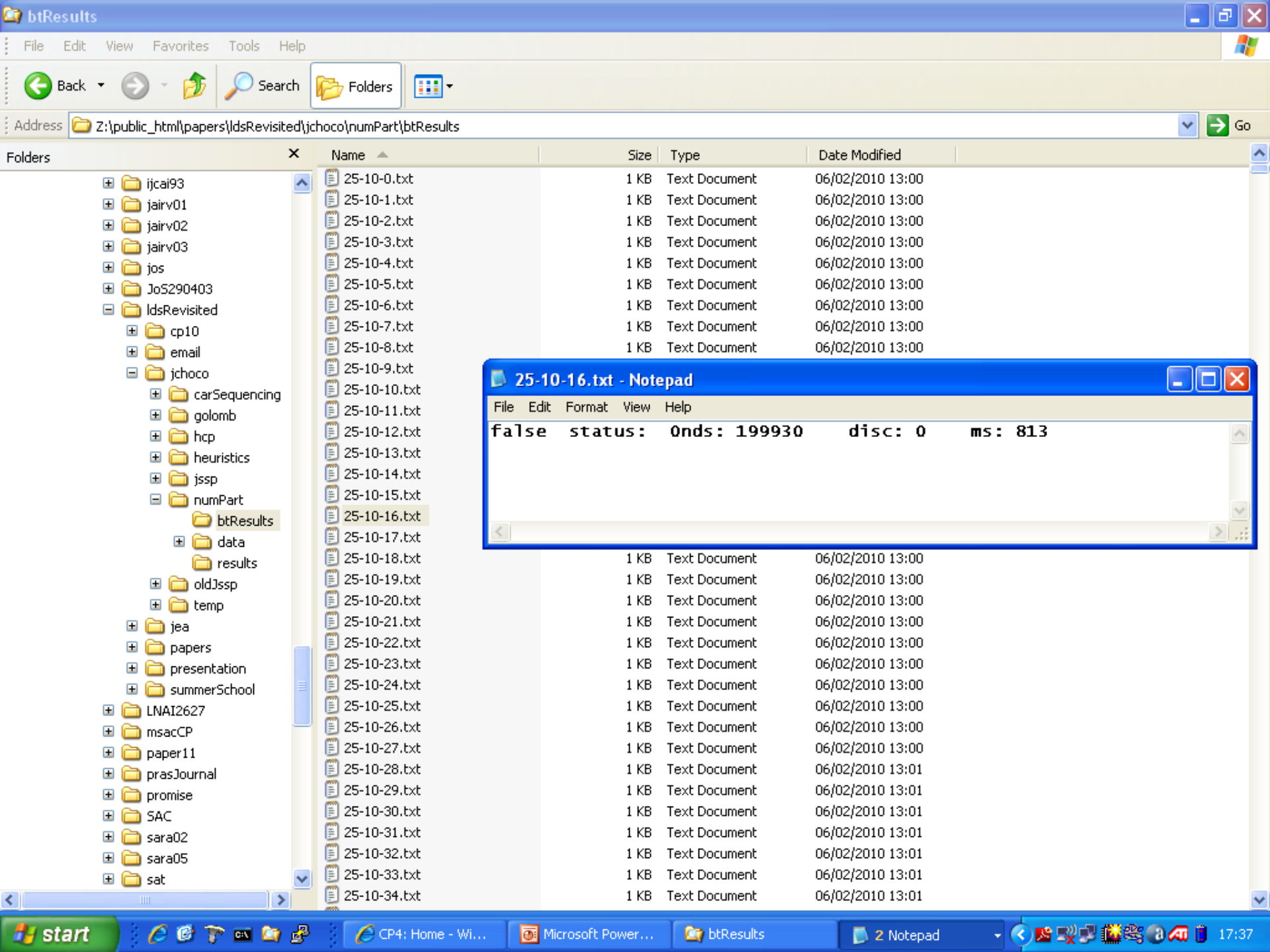
5513085028

8245850660

2917653594

7390634243

9448896406



btResults

File Edit View Favorites Tools Help

Back

Search

Folders

Address Z:\public\_html\papers\ldsRevisited\jchoco\numPart\btResults Go

Folders

+

ijcai93

+

jairv01

+

jairv02

+

jairv03

+

+

-

+

+

-

+

+

+

+

+

-

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

+

Name

Size

Type

Date Modified

25-10-0.txt

1 KB

Text Document

06/02/2010 13:00

25-10-1.txt

1 KB

Text Document

06/02/2010 13:00

25-10-2.txt

1 KB

Text Document

06/02/2010 13:00

25-10-3.txt

1 KB

Text Document

06/02/2010 13:00

25-10-4.txt

1 KB

Text Document

06/02/2010 13:00

25-10-5.txt

1 KB

Text Document

06/02/2010 13:00

25-10-6.txt

1 KB

Text Document

06/02/2010 13:00

25-10-7.txt

1 KB

Text Document

06/02/2010 13:00

25-10-8.txt

1 KB

Text Document

06/02/2010 13:00

25-10-9.txt

1 KB

Text Document

06/02/2010 13:00

25-10-10.txt

1 KB

Text Document

06/02/2010 13:00

25-10-11.txt

1 KB

Text Document

06/02/2010 13:00

25-10-12.txt

1 KB

Text Document

06/02/2010 13:00

25-10-13.txt

1 KB

Text Document

06/02/2010 13:00

25-10-14.txt

1 KB

Text Document

06/02/2010 13:00

25-10-15.txt

1 KB

Text Document

06/02/2010 13:00

25-10-16.txt

1 KB

Text Document

06/02/2010 13:00

25-10-17.txt

1 KB

Text Document

06/02/2010 13:00

25-10-18.txt

1 KB

Text Document

06/02/2010 13:00

25-10-19.txt

1 KB

Text Document

06/02/2010 13:00

25-10-20.txt

1 KB

Text Document

06/02/2010 13:00

25-10-21.txt

1 KB

Text Document

06/02/2010 13:00

25-10-22.txt

1 KB

Text Document

06/02/2010 13:00

25-10-23.txt

1 KB

Text Document

06/02/2010 13:00

25-10-24.txt

1 KB

Text Document

06/02/2010 13:00

25-10-25.txt

1 KB

Text Document

06/02/2010 13:00

25-10-26.txt

1 KB

Text Document

06/02/2010 13:00

25-10-27.txt

1 KB

Text Document

06/02/2010 13:00

25-10-28.txt

1 KB

Text Document

06/02/2010 13:01

25-10-29.txt

1 KB

Text Document

06/02/2010 13:01

25-10-30.txt

1 KB

Text Document

06/02/2010 13:01

25-10-31.txt

1 KB

Text Document

06/02/2010 13:01

25-10-32.txt

1 KB

Text Document

06/02/2010 13:01

25-10-33.txt

1 KB

Text Document

06/02/2010 13:01

25-10-34.txt

1 KB

Text Document

06/02/2010 13:01

25-10-16.txt - Notepad

File Edit Format View Help

false status: onds: 199930 disc: 0 ms: 813

start

CP4: Home - Wi...

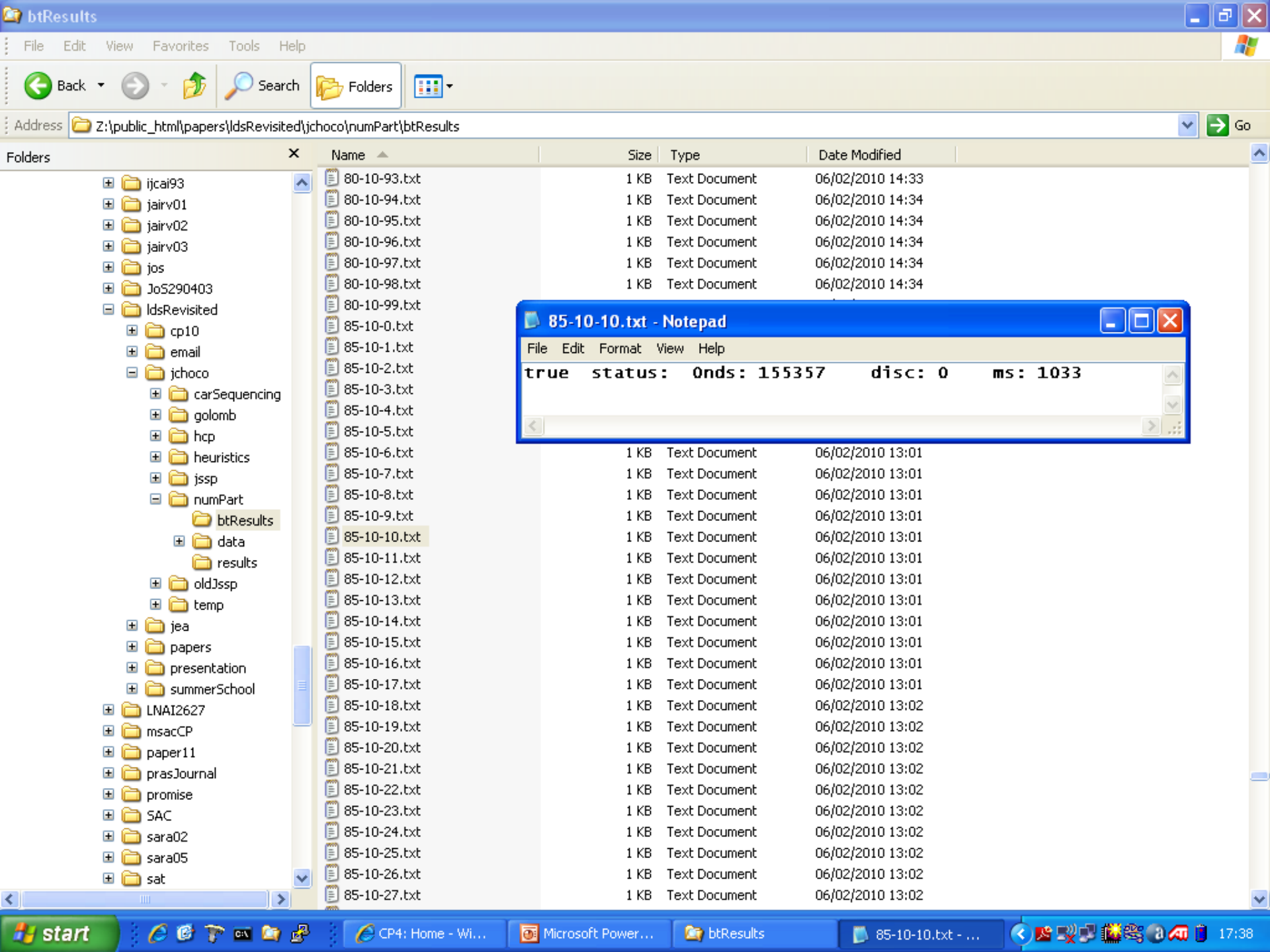
Microsoft Power...

btResults

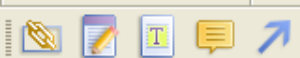
2 Notepad

17:37





The image is a screenshot of a Windows XP desktop. The primary window is a file explorer titled "btResults", showing the directory structure of a folder named "Z:\public\_html\papers\ldsRevisited\jchoco\numPart\btResults". The left pane displays a tree view of folders, including "ijcai93", "jairv01", "jairv02", "jairv03", "jos", "Jo5290403", "ldsRevisited", "cp10", "email", "jchoco", "carSequencing", "golomb", "hcp", "heuristics", "jssp", "numPart", "data", "results", "oldJssp", "temp", "jea", "papers", "presentation", "summerSchool", "LNAI2627", "msacCP", "paper11", "prasJournal", "promise", "SAC", "sara02", "sara05", and "sat". The right pane shows a list of files, all named "80-10-93.txt" through "85-10-27.txt", each with a size of 1 KB and a type of "Text Document". The "Date Modified" column shows various timestamps from 06/02/2010 14:33 to 13:02. A Notepad window titled "85-10-10.txt - Notepad" is open in the foreground, displaying the text: "true status: onds: 155357 disc: 0 ms: 1033". The taskbar at the bottom shows the Start button, several application icons, and the system clock indicating 17:38 on 06/02/2010.



removed from the front of the list. There are then two possible choices: (a) insert in order into  $L$  the difference  $X - Y$ , corresponding to placing the numbers in different bags, or (b) push the sum  $X + Y$  onto the front of  $L$ , corresponding to placing both numbers in the same bag. Of the two choices option (a) is preferred, i.e. it is the heuristic choice. There are then 3 possible outcomes resulting from a choice: (1)  $length(L) = 1$  and  $head(L) \in \{0, 1\}$ , or (2)  $head(L) - sum(tail(L)) \leq 1$ , or (3)  $head(L) - sum(tail(L)) > 1$ . In case (1) a perfect partition exists, in case (2) we can continue making choices, and in case (3) no perfect partition exists. In our model we use the Choco constraint programming toolkit [JCh], and have  $n$  0/1 constrained integer decision variables,  $v_0$  to  $v_{n-1}$ , and the list  $L$  as a reversible structure. The KK heuristic is encoded as a specialised constraint. If a variable  $v_i$  is assigned the value 0 we go with the heuristic, making choice (a), and if assigned the value 1 we go against the heuristic making choice (b). If this results in outcome (1) search terminates successfully, outcome (2) search proceeds, outcome (3) search fails and backtracking takes place.

Problem data sets were generated using the Java program segment given in Figure 5. Problem size  $n$  was varied from 25 to 100 in steps of 5, and for each value of  $n$  100 problem instances were produced containing  $n$  numbers drawn uniformly at random from the range 1 to  $10^d - 1$ . In replicating Korf's experiments  $d$  was set to 10 and of the numbers generated about 90% were 10 digits long and about 10% were 9 digits long or less, as expected.

<sup>1</sup>And, it is unclear if YIELDS incorporates Korf's improvement explicitly, although it might be incorporated implicitly into their value ordering heuristic, but they do not mention this.

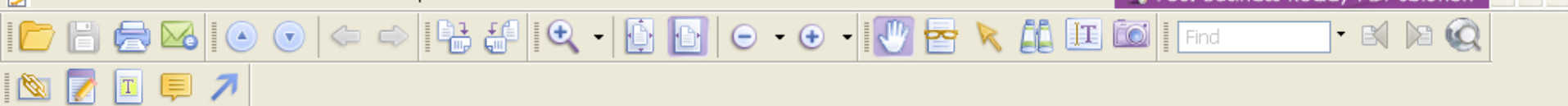
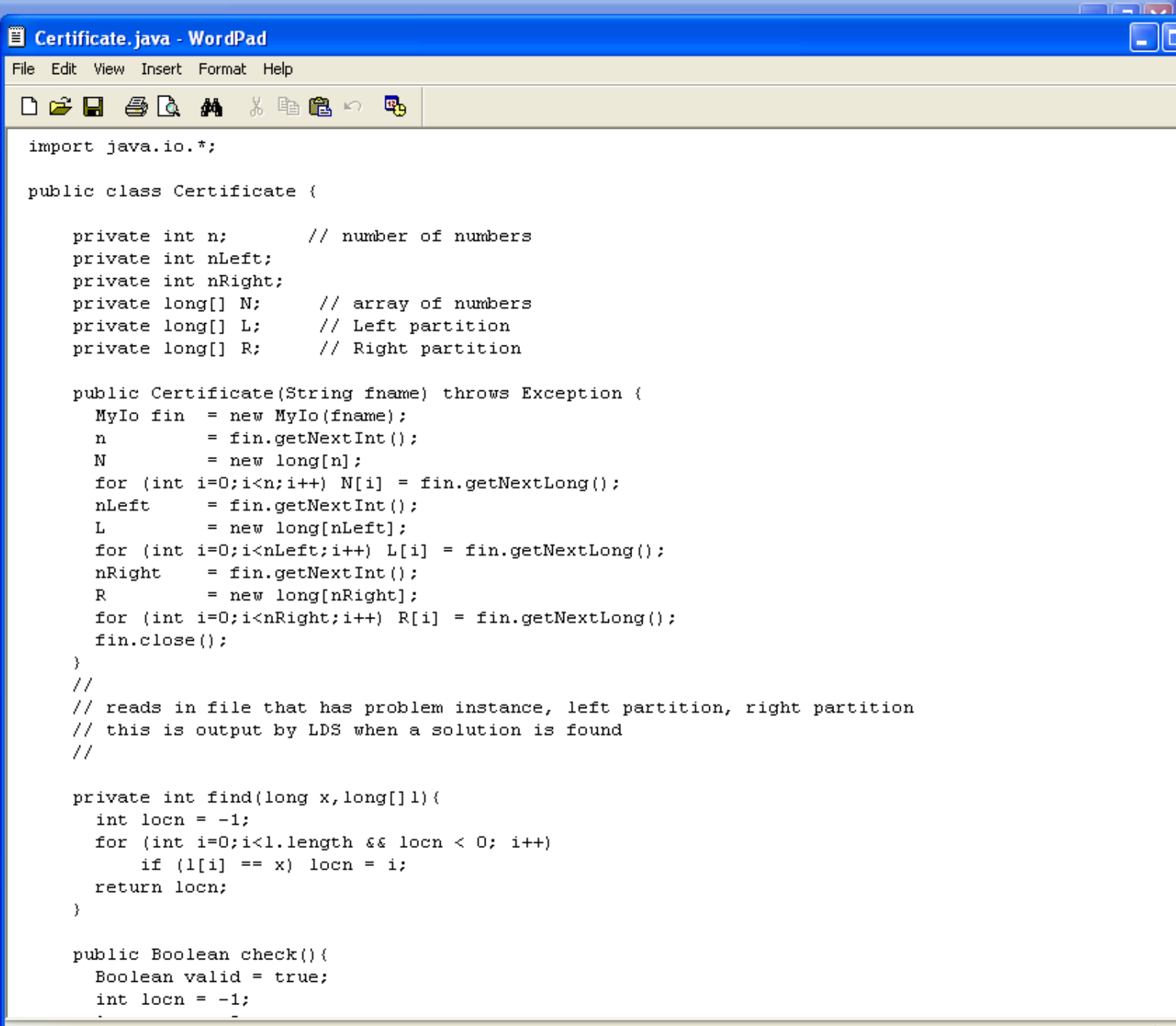
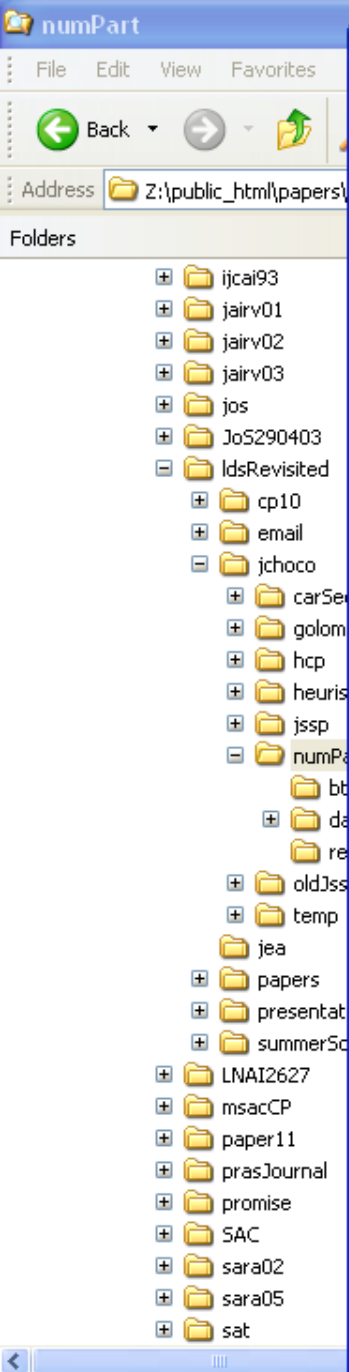


Fig. 6. On the left, log of average search effort (nodes visited) against problem size, solid contour for ILDS-early and broken contour for ILDS-late. On the right, the number of times ILDS-early beat ILDS-late (solid contour) and number of times ILDS-late beat ILDS-early (broken contour) against problem size.

Experiments were run as background jobs, farmed over 10 processors, taking 7 days elapsed time with most processor time spent on problems in the range  $30 \leq n \leq 45$ . Since a variety of processors were used we do not report run times. Figure 6, on the left, shows on a log scale the average number of search nodes<sup>2</sup> explored against problem size  $n$ , for ILDS taking discrepancies early (solid contour) and late (broken contour).

The contours generally agree with Korf's [Korf 1996]. Although not shown, search effort was plotted against Gent and Walsh's measure of constrainedness [Gent and Walsh 1998]  $\kappa = \log_2(l)/n$  where numbers are drawn uniformly and at random from  $(0,1]$ . The complexity peak occurred at  $\kappa = 0.95$  and problem satisfiability about 50%, as expected, i.e. where half of our problem instances had perfect partitions. Figure 6, on the right, shows how often ILDS-early beat ILDS-late, and vice versa. If on a problem instance ILDS-early took less nodes than ILDS-late then ILDS-early scores one point, if ILDS-late takes less nodes than ILDS-early then ILDS-late scores one point, and if they both take the same number of nodes there are no points.

<sup>2</sup>A node corresponds to a decision made by the search process, in this case assigning a value to a decision variable.



```
import java.io.*;
```

```
public class Certificate {
```

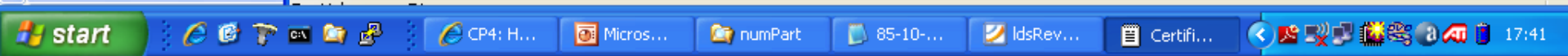
```
    private int n;          // number of numbers
    private int nLeft;
    private int nRight;
    private long[] N;       // array of numbers
    private long[] L;       // Left partition
    private long[] R;       // Right partition
```

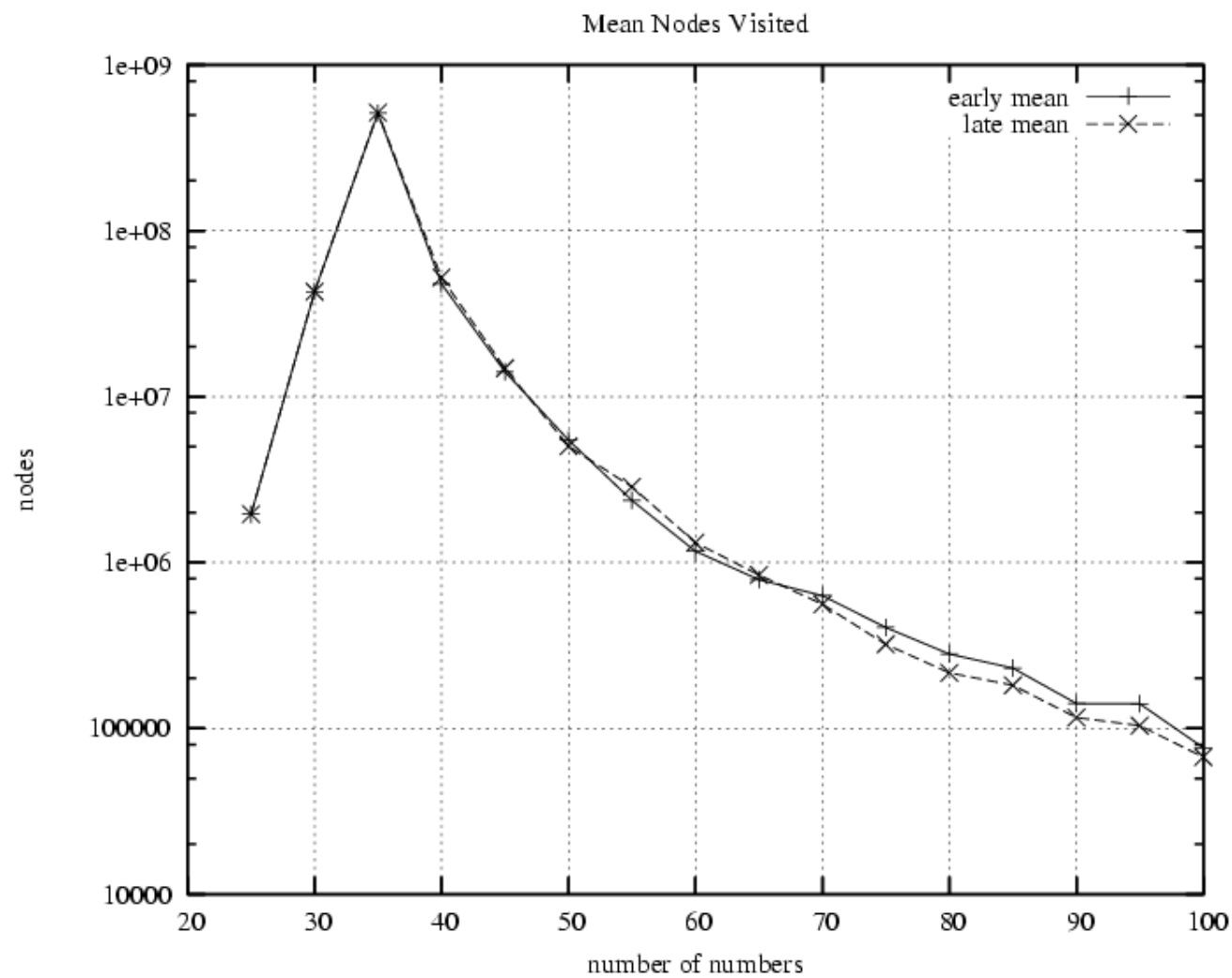
```
    public Certificate(String fname) throws Exception {
        MyIo fin = new MyIo(fname);
        n = fin.getNextInt();
        N = new long[n];
        for (int i=0;i<n;i++) N[i] = fin.getNextLong();
        nLeft = fin.getNextInt();
        L = new long[nLeft];
        for (int i=0;i<nLeft;i++) L[i] = fin.getNextLong();
        nRight = fin.getNextInt();
        R = new long[nRight];
        for (int i=0;i<nRight;i++) R[i] = fin.getNextLong();
        fin.close();
    }
```

```
    //
    // reads in file that has problem instance, left partition, right partition
    // this is output by LDS when a solution is found
    //
```

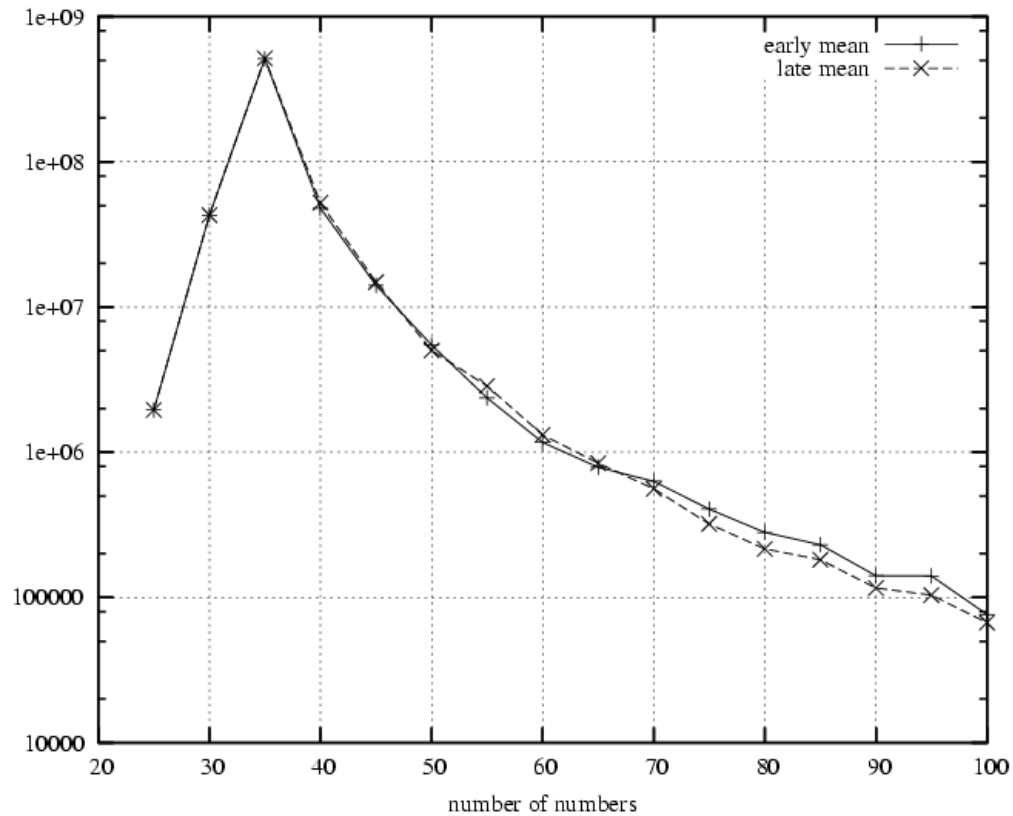
```
    private int find(long x, long[] l) {
        int locn = -1;
        for (int i=0;i<l.length && locn < 0; i++)
            if (l[i] == x) locn = i;
        return locn;
    }
```

```
    public Boolean check(){
        Boolean valid = true;
        int locn = -1;
```





Mean Nodes Visited



Reflect on this ...

You know that sorting is at best  $O(n \cdot \log(n))$

You know that when you increase the size of the data set ( $n$ ) run time increases

This is kind of what we assume: increase  $n$ , increase run time

Is it not strange that in numPart as we increase  $n$  runtime eventually falls?

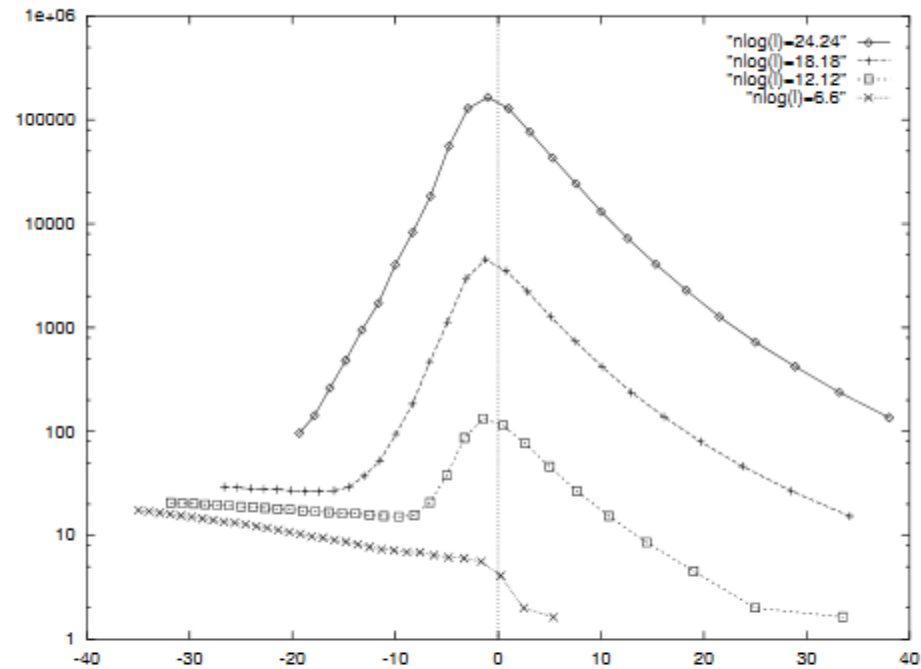


FIGURE 6. Average nodes searched by CKK to find the optimal partition (y-axis) against  $\gamma$  (x-axis) with  $\kappa_c = 0.96$  and  $\nu = 1$ , and the problem input size,  $n \log_2(l)$  fixed at  $6^2$ ,  $12^2$ ,  $18^2$ , and  $24^2$ . Each problem has  $n$  numbers drawn uniformly and at random from  $(0, l]$  with  $n$  varied from 1 to 36.

$$\kappa = \frac{\log_2(l)}{n}.$$

Recent results  
2019



```
9 public class Partition {
10
11     int n;
12     Model model;
13     Solver solver;
14     IntVar[] D;
15
16     Partition(int[] w){
17         n = w.length;
18         model = new Model("Partition");
19         solver = model.getSolver();
20
21         int total = 0;
22         for (int x : w) total = total + x;
23
24         D = model.intVarArray("D",n,0,1); // decision ... left or right?
25         model.scalar(D,w,"=",total/2).post();
26     }
27
28     boolean solve(){
29         solver.setSearch(Search.minDomUBSearch(D)); // take 1 then 0
30         return solver.solve();
31     }
32
33     long stats(){
34         return solver.getMeasures().getNodeCount();
35     }
36 }
```

```

3 public class Experiment {
4
5     private static int random(int d, Random gen) {
6         int x = 0;
7         for (int j=0; j<d; j++) x = x * 10 + gen.nextInt(10);
8         return x;
9     }
10    //
11    // make a random number with up to d digits
12    //
13
14    public static void main(String[] args) {
15        int nLow = Integer.parseInt(args[0]); // number of numbers
16        int nHigh = Integer.parseInt(args[1]); // number of numbers
17        int d = Integer.parseInt(args[2]); // number of digits
18        int m = Integer.parseInt(args[3]); // sample size
19
20        Random gen = new Random();
21
22        for (int n=nLow; n<=nHigh; n++)
23            for (int i=0; i<m; i++) {
24                int[] w = new int[n];
25                for (int j=0; j<n; j++) w[j] = random(d, gen);
26                Partition part = new Partition(w);
27                boolean solved = part.solve();
28                System.out.println(n + " " + solved + " " + part.stats());
29            }
30    }
31 }

```

nLwb-nUpb-digits.txt

Sample size of 50 (very small)

Z:\public\_html\cpM\weekByWeek\week9\numPart\10-100-8.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

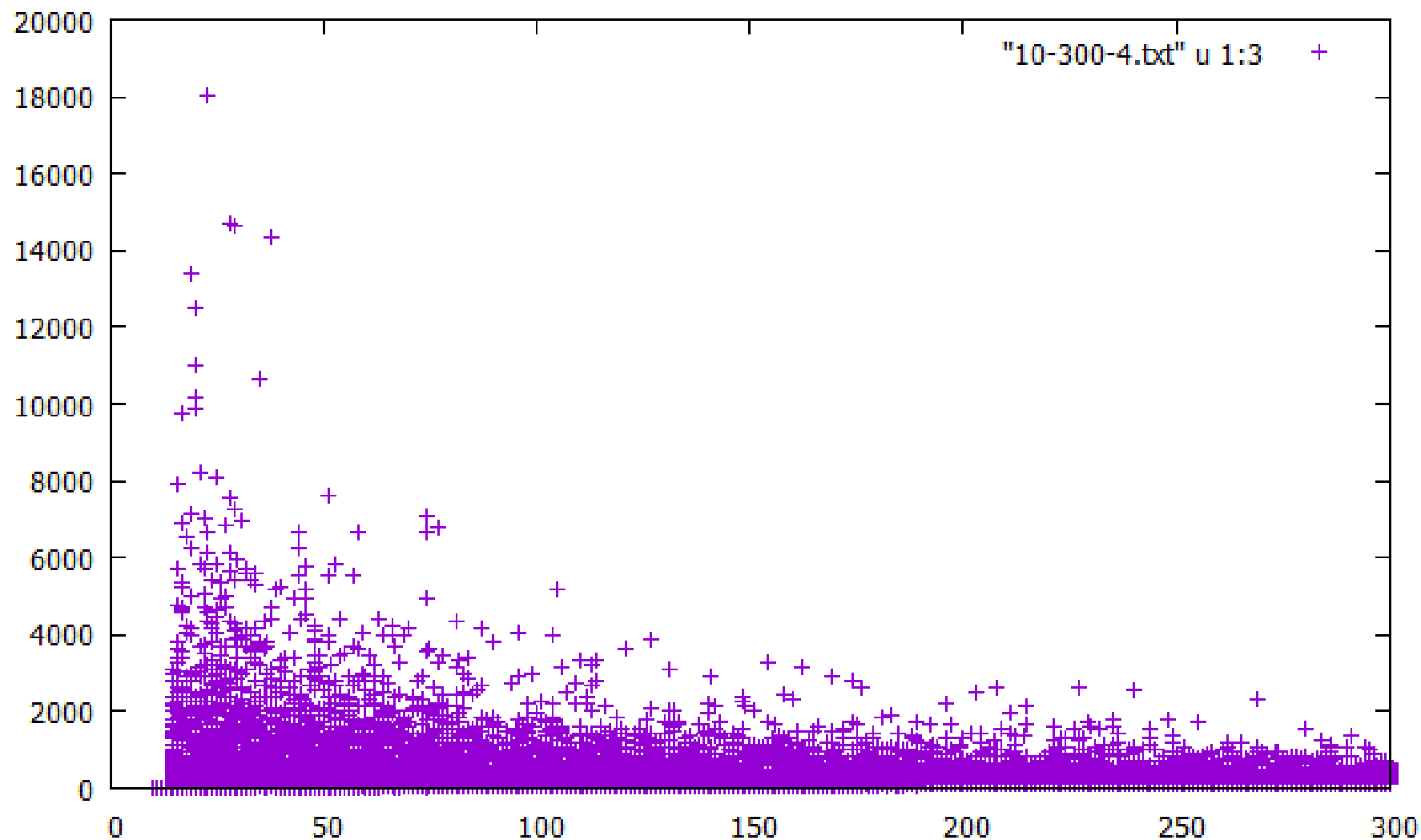
Decision.java x readme.txt x 10-100-8.txt x Partition.java x Experiment.java x

```
988 29 false 13310515
989 29 false 37590115
990 29 true 5461867
991 29 true 2377399
992 29 true 16008564
993 29 false 18111301
994 29 true 7858315
995 29 false 25098033
996 29 true 4440437
997 29 false 41311401
998 29 true 1835449
999 29 false 29818599
1000 29 false 23689739
1001 30 true 5159899
1002 30 true 7954270
1003 30 false 28544183
1004 30 true 4797590
1005 30 true 8607101
1006 30 true 5883780
1007 30 true 13177314
1008 30 true 3934553
1009 30 true 1681832
1010 30 false 31417721
1011 30 false 32281335
1012 30 true 18469792
1013 30 true 13989619
1014 30 false 28131625
1015 30 false 46543703
1016 30 true 15120016
1017 30 true 10279917
1018
```

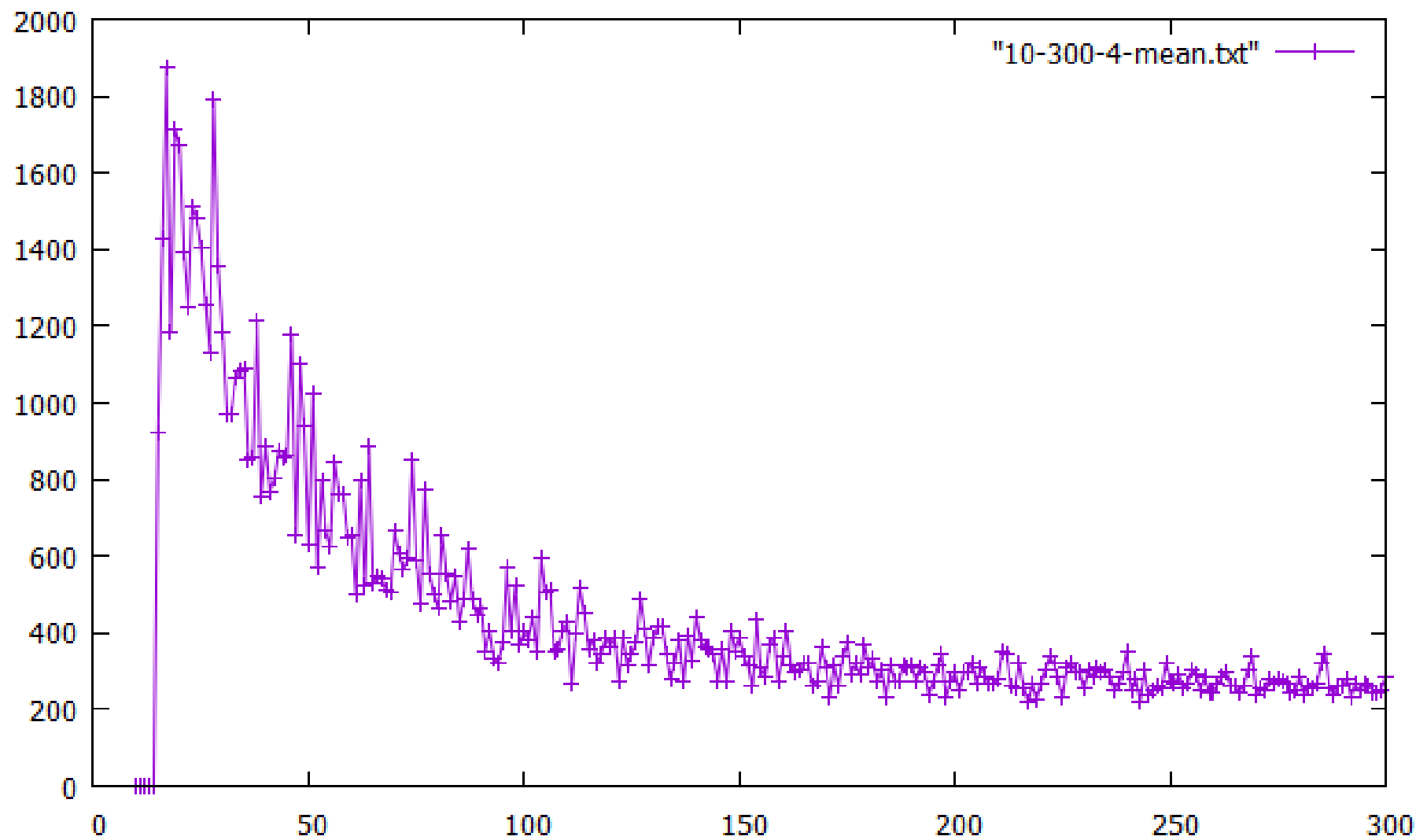
Normal text file length : 14,701 lines : 1,018 Ln : 1,017 Col : 17 Sel : 0 | 0 Unix (LF) UTF-8 INS



Gnuplot (window id : 0)

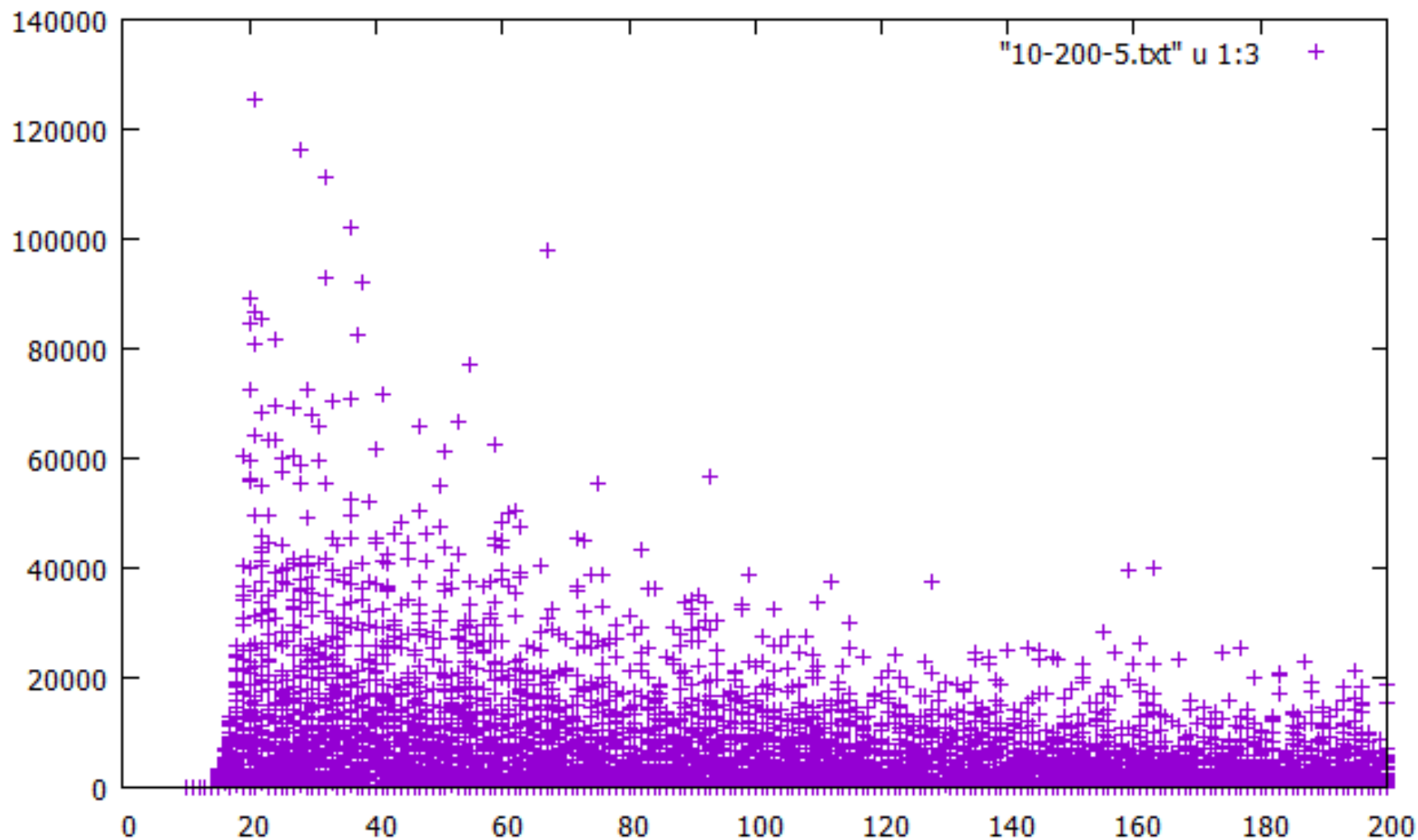


244.794, 20952.1





Gnuplot (window id : 0)



205.810, 120839.

