The most effective algorithm

(Demonstration of possibilities of
World's fastest algorithm)

Hamilton path
and
Hamilton cycle

Proof for the problems of class (P = NP)

The most effective algorithm

(Demonstration of possibilities of World's fastest algorithm

for the Hamilton problem)

Grzegorz Ileczko

e-mail: *ilex77@outlook.com*

phone: (+48) 518455706

skype: (Greg Ileczko, Poland - Koszalin)

1. Introduction.

This article is a short demonstration of computational possibilities of the extreme effective algorithm for the Hamilton problem. In fact, the algorithm can fast solve a few similar problems, well-known in literature as:

Hamilton path – The shortest possible path through all cities. Each single city can be visited only once. A correct Hamilton path should be a shortest possible line from city - A to city - B. That problem is one of the most important in the computer science.

Hamilton cycle – The problem is very similar to previous. The shortest possible path-line through all cities should be found. Only one difference exists, correct Hamilton cycle should start and stop at the same city – A. It's a shortest possible cycle /round through all cities.

Hamilton longest path – The problem is very similar to **Hamilton path**. Only one difference exists, correct path should be a longest possible line from city - A to city - B. That problem is opposite to Hamilton path.

Hamilton longest cycle – The problem is very similar to **Hamilton cycle**. Only one difference exists, the longest possible path-line through all cities should be found. That problem is opposite to Hamilton cycle.

The Hamilton problem seems be easy to solving. In practice, the calculations are highly not effective.

For specific 100 cities, exists many possible complete pathways: $\text{poss} = 9.333 \times 10^{157}$

It's quite impressive value, even for a strongest supercomputer. Solution time for that big problem can be very long – many centuries. I discovered more faster algorithm for Hamilton problem. My algorithm can solve that problem at a polynomial time. For 100 cities, the time of solution is about 1 second. It's almost a real time !!! I used old laptop to test the program and I demonstrate some examples. Video presentation should be interesting. You may see, that my algorithm is really extremely effective.

For the test, I designated a 100 cities. The number of chosen cities can be much larger, for example 10000 cities. It is no problem for algorithm.

This article is only demonstration of algorithm's possibilities. A structure of algorithm was hidden, but every time can be explained. If you are interested, you can buy the algorithm and the program. Then I will explain every detail of the algorithm with a great pleasure. I can also demonstrate this solution in real world, obviously.

2. Example 1. Hamilton path & Hamilton cycle.

One example of correct solution for **Hamilton path problem** and **Hamilton cycle problem** was presented below. That is the best possible solution for 100 random cities. Cities placement was generated automatically by program. Next, proper algorithm was executed for solving Hamilton problem. The results of calculations were presented at visual form and numerical form.

For specific 100 cities, exists many possible complete pathways: $\text{poss} = 9.333 \times 10^{157}$

It's quite impressive value, but my algorithm can solve that big problem at a polynomial time. For 100 cities, the time of solution is about 1 second. It's almost a real time !!! It's a very important fact.

Table 1 – Data for 100 cities.

Column 1 – city number

Column 2 – (x –axis) of proper city

Column 3 – (y –axis) of proper city

Table 2 – results of calculations for Hamilton problem. Hamilton path and Hamilton cycle.

Column 1 – city number. Hamilton path – appropriate sorted list of cities. Hamilton cycle – the same list of cities + one final distance.

Column 2 – (x –axis) of proper city

Column 3 – (y –axis) of proper city

Column 4 – The distance among two connected cities. That is the shortest possible line among two cities.

(Table 2, current row)

The distance among two connected cities

$$\text{Distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

(Table 2, next row)

Chart 1. View of 100 random cities.

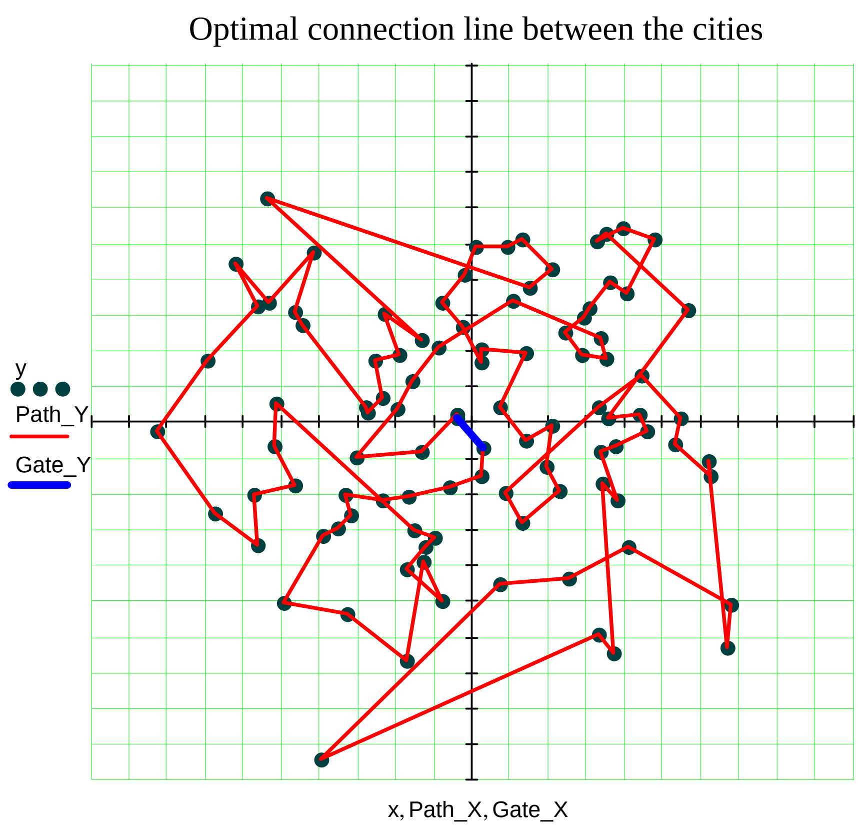


Chart 2. All of cities, Hamilton path (red line), Hamilton cycle (red + blue).

Tab. 1. Data of cities.

City number	x -axis	y -axis
1	0,7542	-2,0747
2	-8,9481	-7,2776
3	-23,242	-0,7995
4	-16,093	-5,692
5	-15,855	-9,6117
6	-9,4116	-5,7203
7	-7,7837	0,7014
8	-6,5067	8,4845
9	-19,535	4,824
10	3,5941	-7,9131
11	-2,2199	-13,982
12	-4,8043	-11,583
13	-4,8849	-18,693
14	-18,967	-7,1217
15	5,8648	-0,2856
16	-2,7482	-9,0429
17	-7,253	4,8272
18	-13,031	-4,9588
19	3,8765	-1,3977
20	-17,418	12,421
21	-13,866	-14,21
22	-14,61	-1,9659
23	-3,6534	-10,996
24	-2,3129	9,3193
25	-1,1515	0,571
26	-13,034	8,6075
27	5,7559	11,889
28	-15,878	9,1022
29	-12,588	7,6175
30	-0,5563	11,487
31	-4,7238	-5,8009
32	-8,5238	-2,7007
33	-4,4342	3,2658
34	-11,012	-8,9349
35	0,2379	13,649
36	-3,4729	-9,7386
37	-15,016	9,3347
38	-11,736	13,225
39	-5,5086	0,9511
40	10,044	10,946
41	0,5781	4,6827
42	-1,6701	-5,1668
43	-9,9856	-8,2763

Tab. 2. Results of calculations for Hamilton problem.

Hamilton path – appropriate list of cities (red line)	x -axis	y -axis	Length between the city and the city at next row
1	0,7542	-2,0747	2,1625
64	0,6237	-4,2332	2,4765
42	-1,6701	-5,1668	3,1189
31	-4,7238	-5,8009	1,9434
45	-6,649	-6,0659	2,7842
6	-9,4116	-5,7203	1,6248
2	-8,9481	-7,2776	1,4401
43	-9,9856	-8,2763	1,2192
34	-11,012	-8,9349	5,9975
21	-13,866	-14,21	4,7139
52	-9,224	-15,029	5,679
13	-4,8849	-18,693	7,7952
23	-3,6534	-10,996	3,3126
11	-2,2199	-13,982	3,5268
12	-4,8043	-11,583	2,2743
36	-3,4729	-9,7386	1,0046
16	-2,7482	-9,0429	1,7153
71	-4,3723	-8,4912	14,175
50	-14,406	1,5209	3,4927
22	-14,61	-1,9659	3,3837
18	-13,031	-4,9588	3,1481
4	-16,093	-5,692	3,927
5	-15,855	-9,6117	3,9858
14	-18,967	-7,1217	7,6318
3	-23,242	-0,7995	6,7354
9	-19,535	4,824	5,6279
28	-15,878	9,1022	3,6583
20	-17,418	12,421	3,9105
37	-15,016	9,3347	5,0883
38	-11,736	13,225	4,7964
26	-13,034	8,6075	1,0857
29	-12,588	7,6175	8,0126
56	-7,8462	1,1586	0,4614
7	-7,7837	0,7014	1,6426
46	-6,6519	1,8917	2,9963
17	-7,253	4,8272	1,9304
48	-5,3867	5,3208	3,356
8	-6,5067	8,4845	3,4358
98	-3,7731	6,403	15,875
53	-15,084	17,542	20,556
74	4,2437	10,542	2,0251
27	5,7559	11,889	3,1828
90	3,6891	14,31	1,2724

44	2,5681	13,708
45	-6,649	-6,0659
46	-6,6519	1,8917
47	0,6045	5,6194
48	-5,3867	5,3208
49	9,763	4,8972
50	-14,406	1,5209
51	-11,157	-26,37
52	-9,224	-15,029
53	-15,084	17,542
54	17,48	-4,2898
55	-1,182	0,26
56	-7,8462	1,1586
57	12,821	-0,6877
58	-3,8331	-2,3797
59	9,108	14,106
60	13,294	14,349
61	19,028	-14,345
62	10,65	-6,1167
63	10,535	-1,9577
64	0,6237	-4,2332
65	9,3198	-2,3827
66	1,964	1,2282
67	2,0159	-12,687
68	3,95	5,4267
69	11,239	10,049
70	-2,5115	5,8196
71	-4,3723	-8,4912
72	7,9637	5,2699
73	9,1886	-16,659
74	4,2437	10,542
75	6,8085	7,0489
76	5,4671	-3,4639
77	12,197	0,5354
78	9,7287	14,673
79	2,8792	9,4416
80	8,5218	8,9237
81	6,9963	-12,215
82	-0,7996	7,4003
83	12,405	3,7135
84	2,4047	-5,4826
85	6,398	-5,4312
86	11,011	15,135
87	14,804	-1,69
88	15,277	0,2928
89	8,0876	8,2384
90	3,6891	14,31
91	18,688	-17,71
92	9,5824	-4,7987
93	11,43	-9,7708
94	10,324	-18,07
95	9,1919	1,1164
96	15,878	8,714
97	9,3116	6,5145
98	-3,7731	6,403
99	17,297	-3,0605

44	2,5681	13,708	2,3309
35	0,2379	13,649	2,3033
30	-0,5563	11,487	2,7901
24	-2,3129	9,3193	2,444
82	-0,7996	7,4003	3,0469
41	0,5781	4,6827	0,9371
47	0,6045	5,6194	3,351
68	3,95	5,4267	4,6446
66	1,964	1,2282	3,2485
19	3,8765	-1,3977	2,2782
15	5,8648	-0,2856	3,2031
76	5,4671	-3,4639	2,1764
85	6,398	-5,4312	3,7445
10	3,5941	-7,9131	2,706
84	2,4047	-5,4826	9,4664
95	9,1919	1,1164	4,1314
83	12,405	3,7135	4,4663
88	15,277	0,2928	2,0383
87	14,804	-1,69	3,7306
54	17,48	-4,2898	1,2429
99	17,297	-3,0605	14,716
91	18,688	-17,71	3,3824
61	19,028	-14,345	8,8688
93	11,43	-9,7708	5,0624
81	6,9963	-12,215	5,0027
67	2,0159	-12,687	18,993
51	-11,157	-26,37	22,545
73	9,1886	-16,659	1,8112
94	10,324	-18,07	13,292
92	9,5824	-4,7987	1,6963
62	10,65	-6,1167	3,9639
65	9,3198	-2,3827	1,2872
63	10,535	-1,9577	2,6157
57	12,821	-0,6877	1,3734
77	12,197	0,5354	2,3457
100	9,8677	0,256	10,376
96	15,878	8,714	8,5633
78	9,7287	14,673	0,8408
59	9,108	14,106	2,163
86	11,011	15,135	2,4144
60	13,294	14,349	4,7658
69	11,239	10,049	1,4944
40	10,044	10,946	2,5312
80	8,5218	8,9237	0,8113
89	8,0876	8,2384	1,7467
75	6,8085	7,0489	2,1212
72	7,9637	5,2699	1,8375
49	9,763	4,8972	1,6791
97	9,3116	6,5145	7,0671
79	2,8792	9,4416	6,4945
70	-2,5115	5,8196	3,1966
33	-4,4342	3,2658	2,552
39	-5,5086	0,9511	4,7358
32	-8,5238	-2,7007	4,7017
58	-3,8331	-2,3797	3,9872
25	-1,1515	0,571	0,3125

100	9,8677	0,256

55	-1,182	0,26	3,0331

Gate = 3.033

Gate is a length of blue line for the Hamilton cycle problem

Red line – Hamilton path

Red line + Blue line – Hamilton cycle

Connection among two cities is definite as the shortest possible distance. That distance can be easily appointed by trigonometrical formula:

$$\text{Distance} = \sqrt{(\Delta x)^2 + (\Delta y)^2} \longrightarrow \text{Distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Hamilton path – That is a sum of all distances between the cities except the last one.

Table 2 – fourth column.

Length_Path = 433.809

Hamilton cycle – That is a sum of all distances between the cities including the last one.

Table 2 –fourth column.

Length_Cycle = 436.842

Length_Path = 433.809 ← **Hamilton path**

\+

Gate = 3.033 ← **last distance**

=

Length_Cycle = 436.842 ← **Hamilton cycle**

3. Example 2. Hamilton longest path & Hamilton longest cycle.

The example 2 is very similar to previous example 1. The difference inside the algorithm is minimal. We need to find the longest path and longest cycle for Hamilton problem. The chosen cities are identical as before for better comparison of calculations results. The data structure of table 3 & 4 is similar to previous data table 1 & 2.

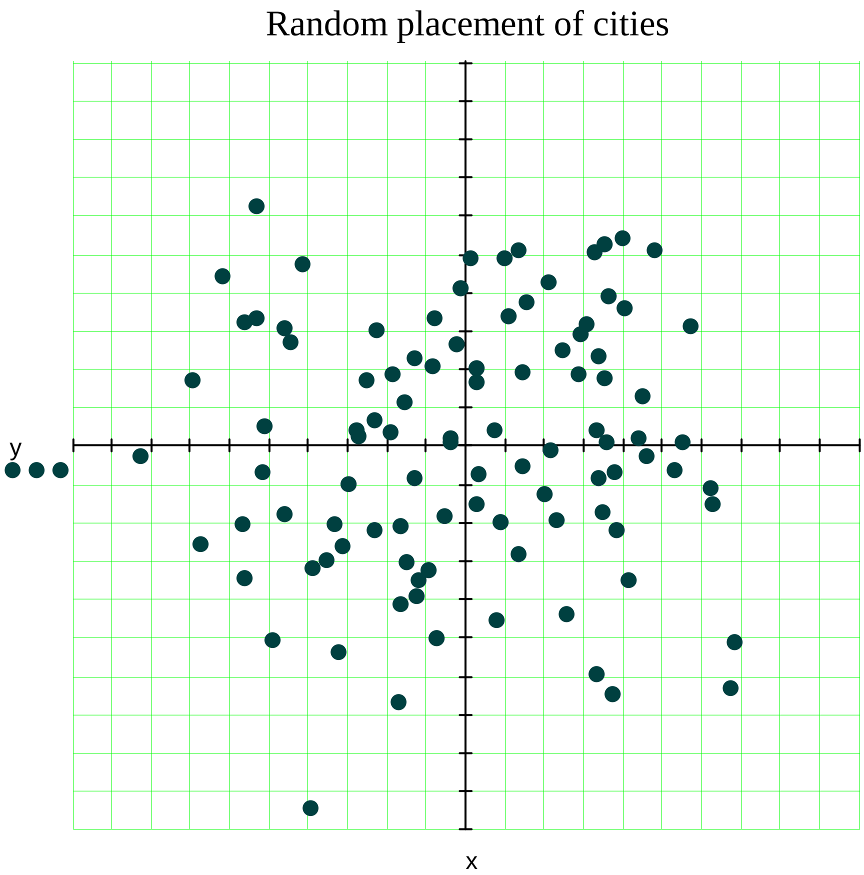


Chart 3. View of 100 random cities. The placement of cities is identical as before (Example 1 /Chart 1).

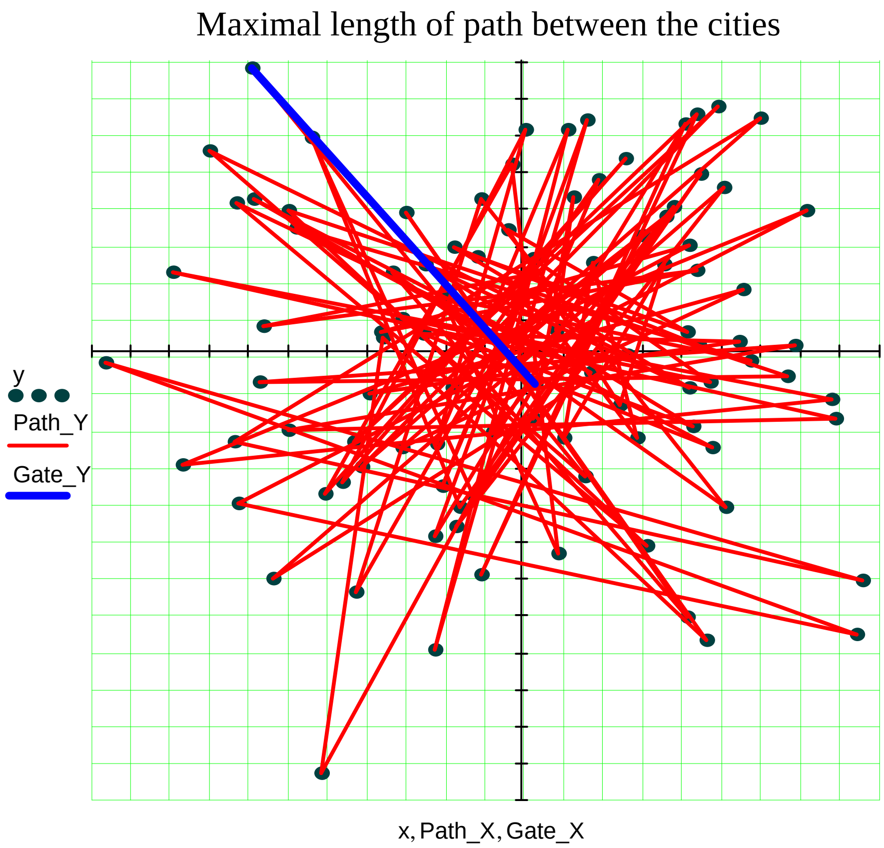


Chart 4. All of cities, Hamilton longest path (red line), Hamilton longest cycle (red + blue).

Tab. 3. Data of cities are identical as before.

City number	x -axis	y -axis
1	0,7542	-2,0747
2	-8,9481	-7,2776
3	-23,242	-0,7995
4	-16,093	-5,692
5	-15,855	-9,6117
6	-9,4116	-5,7203
7	-7,7837	0,7014
8	-6,5067	8,4845
9	-19,535	4,824
10	3,5941	-7,9131
11	-2,2199	-13,982
12	-4,8043	-11,583
13	-4,8849	-18,693
14	-18,967	-7,1217
15	5,8648	-0,2856
16	-2,7482	-9,0429

Tab. 4. Results of calculations for Hamilton longest problem.

Hamilton longest path – appropriate list of cities (red line)	x -axis	y -axis	Length between the city and the city at next row
1	0,7542	-2,0747	27,058
51	-11,157	-26,37	27,281
7	-7,7837	0,7014	26,082
94	10,324	-18,07	31,439
8	-6,5067	8,4845	29,64
73	9,1886	-16,659	27,055
17	-7,253	4,8272	23,709
93	11,43	-9,7708	23,522
24	-2,3129	9,3193	25,311
52	-9,224	-15,029	17,565
25	-1,1515	0,571	16,402
37	-15,016	9,3347	21,728
19	3,8765	-1,3977	22,372
28	-15,878	9,1022	31,267
81	6,9963	-12,215	28,892
26	-13,034	8,6075	29,682

17	-7,253	4,8272
18	-13,031	-4,9588
19	3,8765	-1,3977
20	-17,418	12,421
21	-13,866	-14,21
22	-14,61	-1,9659
23	-3,6534	-10,996
24	-2,3129	9,3193
25	-1,1515	0,571
26	-13,034	8,6075
27	5,7559	11,889
28	-15,878	9,1022
29	-12,588	7,6175
30	-0,5563	11,487
31	-4,7238	-5,8009
32	-8,5238	-2,7007
33	-4,4342	3,2658
34	-11,012	-8,9349
35	0,2379	13,649
36	-3,4729	-9,7386
37	-15,016	9,3347
38	-11,736	13,225
39	-5,5086	0,9511
40	10,044	10,946
41	0,5781	4,6827
42	-1,6701	-5,1668
43	-9,9856	-8,2763
44	2,5681	13,708
45	-6,649	-6,0659
46	-6,6519	1,8917
47	0,6045	5,6194
48	-5,3867	5,3208
49	9,763	4,8972
50	-14,406	1,5209
51	-11,157	-26,37
52	-9,224	-15,029
53	-15,084	17,542
54	17,48	-4,2898
55	-1,182	0,26
56	-7,8462	1,1586
57	12,821	-0,6877
58	-3,8331	-2,3797
59	9,108	14,106
60	13,294	14,349
61	19,028	-14,345
62	10,65	-6,1167
63	10,535	-1,9577
64	0,6237	-4,2332
65	9,3198	-2,3827
66	1,964	1,2282
67	2,0159	-12,687
68	3,95	5,4267
69	11,239	10,049
70	-2,5115	5,8196
71	-4,3723	-8,4912
72	7,9637	5,2699
73	9,1886	-16,659
74	4,2437	10,542
75	6,8085	7,0489

87	14,804	-1,69	29,415
22	-14,61	-1,9659	29,972
88	15,277	0,2928	28,791
18	-13,031	-4,9588	30,518
54	17,48	-4,2898	38,12
9	-19,535	4,824	37,666
99	17,297	-3,0605	36,491
14	-18,967	-7,1217	38,275
96	15,878	8,714	36,644
5	-15,855	-9,6117	35,479
91	18,688	-17,71	45,212
3	-23,242	-0,7995	44,387
61	19,028	-14,345	36,171
4	-16,093	-5,692	35,57
60	13,294	14,349	31,023
2	-8,9481	-7,2776	30,011
86	11,011	15,135	29,189
6	-9,4116	-5,7203	27,969
78	9,7287	14,673	31,047
11	-2,2199	-13,982	30,287
59	9,108	14,106	29,214
12	-4,8043	-11,583	27,25
90	3,6891	14,31	34,099
13	-4,8849	-18,693	33,247
44	2,5681	13,708	20,827
31	-4,7238	-5,8009	20,073
35	0,2379	13,649	24,192
43	-9,9856	-8,2763	25,582
27	5,7559	11,889	26,736
34	-11,012	-8,9349	22,943
30	-0,5563	11,487	24,311
67	2,0159	-12,687	17,208
33	-4,4342	3,2658	17,703
57	12,821	-0,6877	26,732
29	-12,588	7,6175	26,993
62	10,65	-6,1167	17,637
39	-5,5086	0,9511	17,71
77	12,197	0,5354	20,972
32	-8,5238	-2,7007	21,89
83	12,405	3,7135	21,417
45	-6,649	-6,0659	20,537
89	8,0876	8,2384	12,23
55	-1,182	0,26	12,216
97	9,3116	6,5145	17,974
56	-7,8462	1,1586	18,002
49	9,763	4,8972	24,404
50	-14,406	1,5209	15,314
41	0,5781	4,6827	14,073
71	-4,3723	-8,4912	14,963
47	0,6045	5,6194	12,491
63	10,535	-1,9577	17,613
46	-6,6519	1,8917	17,559
92	9,5824	-4,7987	18,069
48	-5,3867	5,3208	16,602
65	9,3198	-2,3827	15,767
98	-3,7731	6,403	14,962
100	9,8677	0,256	13,572
70	-2,5115	5,8196	12,613
95	9,1919	1,1164	11,803

76	5,4671	-3,4639
77	12,197	0,5354
78	9,7287	14,673
79	2,8792	9,4416
80	8,5218	8,9237
81	6,9963	-12,215
82	-0,7996	7,4003
83	12,405	3,7135
84	2,4047	-5,4826
85	6,398	-5,4312
86	11,011	15,135
87	14,804	-1,69
88	15,277	0,2928
89	8,0876	8,2384
90	3,6891	14,31
91	18,688	-17,71
92	9,5824	-4,7987
93	11,43	-9,7708
94	10,324	-18,07
95	9,1919	1,1164
96	15,878	8,714
97	9,3116	6,5145
98	-3,7731	6,403
99	17,297	-3,0605
100	9,8677	0,256

82	-0,7996	7,4003	12,542
76	5,4671	-3,4639	9,0837
72	7,9637	5,2699	12,104
84	2,4047	-5,4826	11,018
68	3,95	5,4267	11,131
85	6,398	-5,4312	8,0005
66	1,964	1,2282	8,2642
79	2,8792	9,4416	13,86
64	0,6237	-4,2332	12,866
75	6,8085	7,0489	14,218
58	-3,8331	-2,3797	15,239
74	4,2437	10,542	16,786
42	-1,6701	-5,1668	17,39
80	8,5218	8,9237	22,185
36	-3,4729	-9,7386	24,405
38	-11,736	13,225	24,013
16	-2,7482	-9,0429	23,732
40	10,044	10,946	25,866
23	-3,6534	-10,996	25,781
69	11,239	10,049	34,911
21	-13,866	-14,21	24,149
15	5,8648	-0,2856	26,524
20	-17,418	12,421	29,24
10	3,5941	-7,9131	31,572
53	-15,084	17,542	25,212

Gate = 25.212

Gate is a length of blue line for the Hamilton longest cycle

Red line **– Hamilton longest path**

Red line **+** Blue line **– Hamilton longest cycle**

Connection among two cities is definite as the shortest possible line. That distance can be easily appointed by trigonometrical formula:

$$\text{Distance} = \sqrt{(\Delta x)^2 + (\Delta y)^2} \longrightarrow \text{Distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Hamilton longest path – That is a sum of all distances between the cities except the last one.

Table 4 – fourth column.

$\text{Hamilton\_MAX\_Path} = 2.324 \times 10^3$

Hamilton longest cycle – That is a sum of all distances between the cities including the last one.
Table 4 – fourth column.

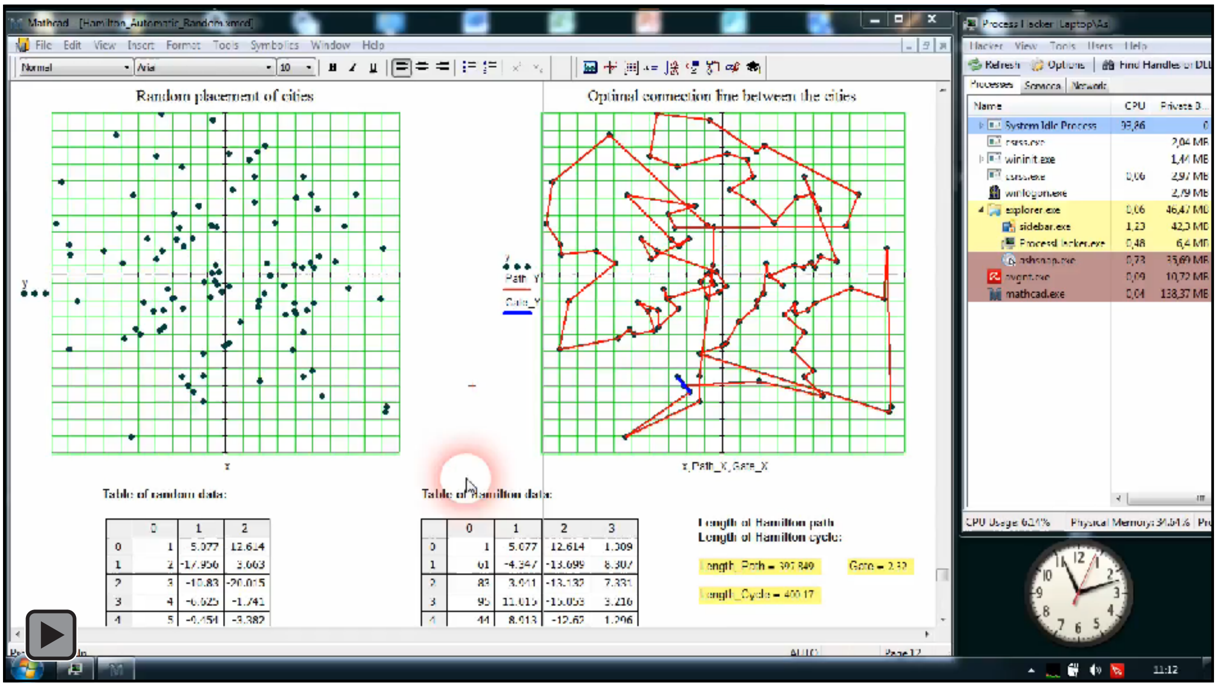
$\text{Hamilton\_MAX\_Cycle} = 2.349 \times 10^3$

4. Example 3. Video presentation (path/ cycle).

Presented algorithm is extremely effective and I can give proof for this. Video presentation is very helpful to understand the real speed of algorithm. Presentation was encoded by strong video coder - H.264. You should have correct PDF file reader to run presentation - **Adobe Reader XI** with **Adobe Flash Player**. Any alternative PDF file readers do not cope properly with animation playing. **The Adobe Reader XI** ensures faultless operation of the presented document.

Video presentation was prepared by use of special program - **Ashampoo Snap 7**. System pulpit was captured as video file. Presentation was prepared without any kind of external modifications. It's a 100% of view of Windows pulpit. We can see the **Mathcad** software with my algorithm for **Hamilton path** & **Hamilton cycle**. Only charts with data are visible. The main source of program was hidden. All calculations were executed for 100 cities and for total topology. It's mean, that every single city have complete connections with all other cities. Except the **Mathcad** software, only system time and system monitor are visible on presentation. System parameters were demonstrated for better situation understanding. You can see by yourself, that the Hamilton algorithm is really extremely effective.

You should click on the picture area for play.



5. Algorithm's most important parameters.

Problem type	Status	Solution time	Short description
Hamilton path	Yes	1 second for 100 random cities	There are four independent problems, but only one algorithm for all problems. Differences in source code are minimal. Algorithm can be run for total topology. When topology is not full, verification procedure should be run at first. It's independent algorithm, called as **Hamiltonian completion**.
Hamilton cycle	Yes	1 second for 100 random cities	
Hamilton longest path	Yes	1 second for 100 random cities	
Hamilton longest cycle	Yes	1 second for 100 random cities	
Hamiltonian completion	Ready to tests	2 seconds for automatic verification of topology structure	This is another type of algorithm – verification of topology structure. - Necessary, when topology is not full, - Unnecessary for total topology, Verification result can be positive or negative: - **positive verification result** – exists one or more Hamilton path. Hamilton algorithm can be used for solving that topology. - **negative verification result** – even one correct path not exists. Algorithm can't be used for that topology, data structure is wrong.

Data	Status	Solution time	Short description
Manual cities placement	Yes	A few minutes for 100 cities	The algorithm needs the input data matrix for proper working. It's possible to manually introduce the necessary data. We can also automatically generate the random data by program. Used method is not important for the algorithm, but automatic mode is faster than manual typing. Table of total topology is always generated automatically by program. The algorithm was tested only for 100 cities, For specific 100 cities, exists many possible complete pathways: $poss = 9.333\times 10^{157}$ It's quite impressive value, but we can run the algorithm even for 10000 cities. It isn't a too big challenge for PC-computer.
Automatic random cities placement	Yes	Less than 1 second for 100 cities	
Automatic generating of matrix of total topology	Yes	Less than 1 second for 100 cities (10000 connections)	
Manual preparation of topology	Yes	A few minutes / hours for 100 cities	
Semi-automatic preparation of topology	Yes	A few seconds / minutes for 100 cities	
Visualization of data	Yes	Less than 1 second for 100 cities	