Lund University
Faculty of Engineering (LTH)

FRTN 30

Network Dynamics

Hand-In 3

Author:
Marcel Attar, 941127-2173

The report was handed in on: May 20, 2019

Lunds universitet
Lunds Tekniska Högskola

1 Traffic tolls in Los Angeles

Task a)

Find the **shortest path** between node 1 and 17. This is equivalent to the fastest path (path with shortest traveling time) in an empty network. (Hint: Use the MATLAB function graphshortestpath.) Notice that, since quite a big area around an in- tersection is included in the node, the traveling time might be a bit shorter than one could expect.

See attached code for implementation. The shortest path is 1-2-3-9-13-17 with a time of 0.533 hours.

Task b)

Find the **maximum flow** between node 1 and 17. (Hint: Use the MATLAB function graphmaxflow.)

See attached code for implementation. The maximum flow was 22448.

Task c)

Given the flow vector in flow.mat, compute the **external inflow or outflow** at each node.

See attached code for implementation. The results can be seen in the table below.

Node:	**External inflow or outflow**	**Node:**	**External inflow or outflow**
1.	16806	10.	1169
2.	8570	11.	-5
3.	19448	12.	-7131
4.	4957	13.	-380
5.	-746	14.	-7412
6.	4768	15.	-7810
7.	413	16.	-3430
8.	-2	17.	-23544
9.	-5671		

A positive value is an out flow and a negative value is in flow.

Task d)

Using CVX, find the **social optimum** f^* with respect to the delays.

The cost function for this problem is

$$c_e(f_e) = \frac{l_e}{1 - f_e/C_e} \tag{1}$$

and we have the constraints

$$Bf = \lambda - \mu$$

$$0 \leq f \leq C_e$$

where λ is 17x1 with the first element = 16806 and μ has the same dimension but the last element is = 16806. Using the CVX we get

Link	**Flow**	**Link**	**Flow**
1.	6642	15.	5525
2.	6059	16.	2854
3.	3132	17.	4886
4.	3132	18.	2215
5.	10164	19.	464
6.	4638	20.	2338
7.	3006	21.	3318
8.	2543	22.	5656
9.	3132	23.	2373
10.	583	24.	0
11.	0	25.	6414
12.	2927	26.	5505
13.	0	27.	4886
14.	3132	28.	4886

With the optimal value: +25943.6.

Task e)

Using the CVX, find the **Wardrop equilibrium** $f^{(0)}$.

The cost function that we want to minimize is

$$\sum_{e \in \varepsilon} \int_0^{f_e} d_e(x)dx \tag{2}$$

with the same constraints as in the previous task. Calculating the integral we get

$$\int_0^{f_e} d_e(x)dx = \int_0^{f_e} \frac{l_e}{1 - \frac{x}{C_e}}dx = \int_0^{f_e} \frac{C_e l_e}{C_e - x}dx = -C_e l_e log(1 - \frac{f_e}{C_e}) \tag{3}$$

I then used this in the CVX. See attached code for implementation.

Link	**Flow**	**Link**	**Flow**
1.	6716	15.	5445
2.	6716	16.	2353
3.	2367	17.	4933
4.	2367	18.	1842
5.	10090	19.	697
6.	4645	20.	3036
7.	2804	21.	3050
8.	2284	22.	6087
9.	3418	23.	2587
10.	0	24.	0
11.	177	25.	6919
12.	4171	26.	4954
13.	0	27.	4933
14.	2367	28.	4933

With the optimal value: +15729.6

Task f)

Introduce tolls, such that the toll on link e is $w_e = f_e^* d_e'(f_e)$, where f_e is the flow at the system optimum. Now the delay on link e is given by $d_e(f_e) + w_e$. Use CVX to compute the **new Wardrop equilibrium** $f^{(w)}$. What do you observe?

We have now introduced tolls to the cost function, such that the toll on link e is $w_e = f_e^* d_e'(f_e)$ and thus the delay on link e is $d_e(f_e) + w_e$. First the toll vector is calculated and then a new Wardrop equilibria. The result was

w_e	**value**	w_e	**value**
w_1	1.9	w_{15}	0.48
w_2	0.19	w_{16}	0.08
w_3	0.05	w_{17}	0.07
w_4	0.11	w_{18}	0.02
w_5	1.44	w_{19}	0.002
w_6	0.47	w_{20}	0.01
w_7	0.11	w_{21}	0.07
w_8	0.06	w_{22}	0.26
w_9	0.28	w_{23}	0.07
w_{10}	0.01	w_{24}	0
w_{11}	0	w_{25}	0.41
w_{12}	0.08	w_{26}	0.28
w_{13}	0	w_{27}	0.19
w_{14}	0.13	w_{28}	0.53

Link	Flow	Link	Flow
1.	6642	15.	5525
2.	6059	16.	2854
3.	3132	17.	4886
4.	3132	18.	2215
5.	10164	19.	464
6.	4638	20.	2338
7.	3006	21.	3318
8.	2543	22.	5656
9.	3132	23.	2373
10.	583	24.	0
11.	0	25.	6414
12.	2927	26.	5505
13.	0	27.	4886
14.	3132	28.	4886

This is the same flow as the social optimum in d).

Task g)

We now want to calculate total additional delay, i.e.

$$c_e\left(f_e\right) = f_e\left(d_e\left(f_e\right) - l_e\right) \tag{4}$$

This gives the social optimum

Link	Flow	Link	Flow
1.	6653	15.	5510
2.	5774	16.	3044
3.	3420	17.	4882
4.	3420	18.	2415
5.	10153	19.	444
6.	4643	20.	2008
7.	3106	21.	3487
8.	2662	22.	5495
9.	3009	23.	2204
10.	878	24.	0
11.	0	25.	6301
12.	2355	26.	5623
13.	0	27.	4882
14.	3420	28.	4882

If we now introduce tolls so that the Wardrop equilibrium coincide with the social optimum, we can use theorem 5.2 which states that the weights of the tolls should be

$$\omega_e^* = c_e'\left(f_e^*\right) - d_e\left(f_e^*\right) \tag{5}$$

Using eq. (4) we get

$$\omega_e^* = f_e^* d_e'\left(f_e^*\right) - l_e \tag{6}$$

and using these tolls in the CVX yeilds the same link flow as in the table above (the social optimum). The resulting weights are

w_e	**value**	w_e	**value**
w_1	1.8	w_{15}	0.355
w_2	0.11	w_{16}	0.015
w_3	-0.069	w_{17}	-0.007
w_4	-0.063	w_{18}	-0.037
w_5	1.30	w_{19}	-0.031
w_6	0.393	w_{20}	-0.024
w_7	0.0223	w_{21}	0.007
w_8	0.0059	w_{22}	0.128
w_9	0.107	w_{23}	-0.021
w_{10}	-0.095	w_{24}	-0.054
w_{11}	-0.107	w_{25}	0.285
w_{12}	-0.054	w_{26}	0.241
w_{13}	-0.112	w_{27}	0.026
w_{14}	-0.029	w_{28}	0.377

2 Coloring

I cooperated with: Gustav Östgren, Ian Thorslund, Filip Östlund and John Felix Abrahamsson on the following tasks. In this part of the assignment we are asked to color a undirected graph, with the objective of not color two neighbors with the same color.

Task a)

First we have a linegraph, where all 10 nodes start with the same color (red). At every time instance t, one node $I(t)$ is chosen uniformly at random and update its color. The updated color is chosen from the following distribution

$$P\left(X_i(t+1) = a | X(t), I(t) = i\right) = \frac{e^{-\eta(t)\sum_j W_{ij} c\left(a, X_j(t)\right)}}{\sum_{s\in\mathcal{C}} e^{-\eta(t)\sum_j W_{ij} c\left(s, X_j(t)\right)}} \tag{7}$$

with the cost function

$$c\left(s, X_j(t)\right) = \begin{cases} 1 & \text{if } X_j(t) = s \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

We chose $\eta(t)$ as

$$\eta(t) = \frac{t}{100} \tag{9}$$

The potential function is given by

$$U(t) = \frac{1}{2} \sum_{i,j \in \mathcal{V}} W_{ij} c\Big(X_i(t), X_j(t)\Big) \tag{10}$$

where $\mathcal{V}$ is the set of nodes.
See attached code for implementation.

Figure 1: Figure of the colored linegraph after a 1000 time steps. As we can see there are no neighbooring nodes with the same color, we have reached a global minimum of the potential function.

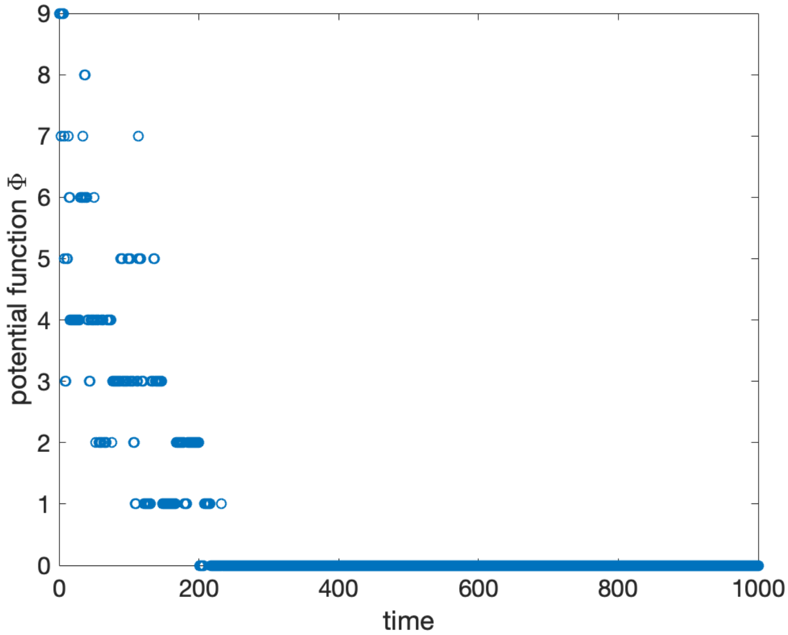

Figure 2: A graph of the potential function over time. When the time increases the closer we get to the global minimum and the noise decreases. After rougly 200 steps the potential function has reached 0 and for this particular run the noise never pushes it back up again.

Task b)

We now increased the complexity and looked at a network of routers where each color represented frequency bands. The cost function is now

$$c\Big(s, X_j(t)\Big) = \begin{cases} 2 & \text{if } X_j(t) = s \\ 1 & \text{if } \Big|X_j(t) - s\Big| = 1 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Now it's not only bad to have the same color as a neighbor but also to have a "similar" color, i.e. a frequency that is close to its neighbor's.

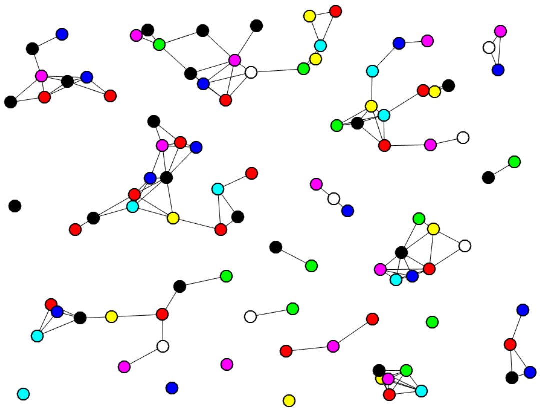

Figure 3: Figure of the colored linegraph after a 1000 time steps. Here $\eta = \frac{t}{100}$. The graph consist's of several connected components. This graph correspond to the smallest potential function.

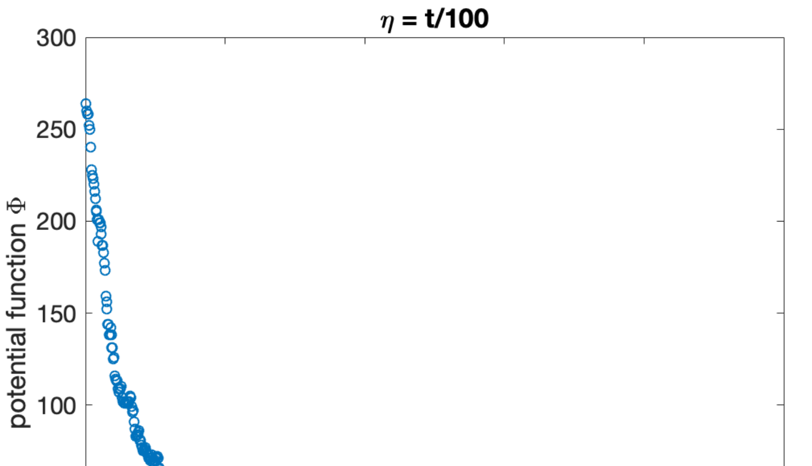

Figure 4: The graph converges to a potential of 4 as time increases. This seems to be the global minimum of the potential function.

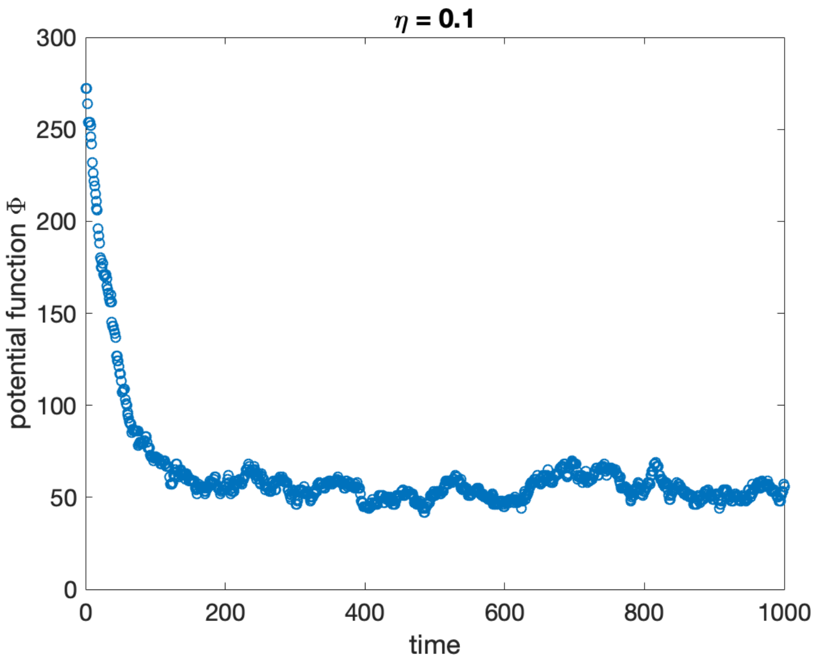

Figure 5: This time, when η is constant and small, the noise is big and therefore the function oscillates, it never converges to the global minimum.

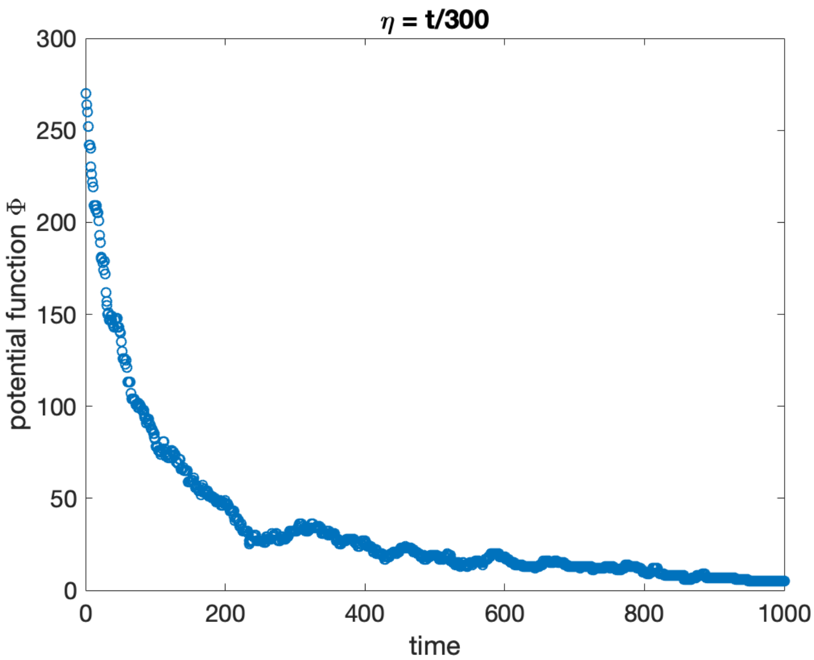

Figure 6: Now the noise doesn't decrease as quickly as in figure 4, thus the potential function doesn't converge as quickly.