FRTN30 Network Dynamics
Hand-in 3

Due: 2019-05-20

Instructions:

- You may implement your solutions in any language you see fit, but the TAs can only guarantee you support with Matlab/Octave. Your code should be written in a quite general manner, i.e., if a question is slightly modified, it should only require slight modifications in your code as well. Send a PDF-file with your answers together with your code to `frtn30@control.lth.se`. As a subject for the mail, you enter `handin3` and your full name.
- Comment your code well. Clarity is more important than efficiency.
- Late submission is discouraged: you get 1 point in your exam (out of 25) for each on-time submission.
- Collaboration policy: Collaboration such as exchange of ideas among students is encouraged, however, every student has to submit her/his own manuscript (in pdf format) and code, and specify whom she/he has collaborated with and on what particular part of the work.
- Up to five best hand-ins may be rewarded with an extra point.
- If anything is unclear with the hand-in, do not hesitate to ask the TAs.

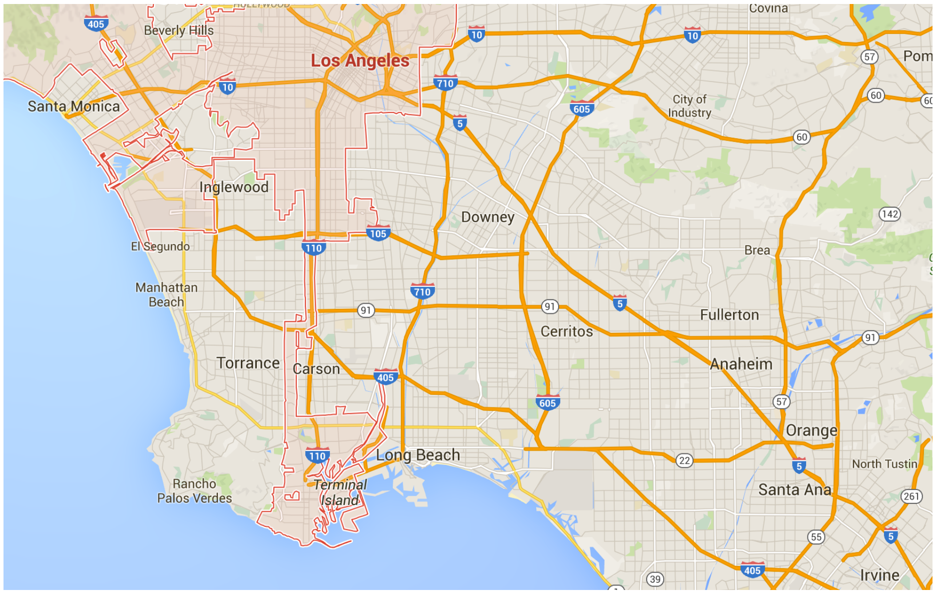

Figure 1: The highway network in Los Angeles.

Preparation: In order to be prepared for part 1 of the assignment, it is recommended to read **Chapter 5** (*Network flow optimization*) in the lecture notes. To be prepared for part 2, it is recommended to read **Chapter 6** (*Games, networks, and learning*). Furthermore, it can be helpful to look at the exercises from exercise session 9, since they are focused on practicing the implementational aspects of the assignment.

1 Traffic tolls in Los Angeles

In the first part of this hand-in, we will study traffic flows on the highway network in Los Angeles, see Figure 1. To simplify the problem, an approximative highway map is given in Figure 2, covering part of the real highway network. The node-link incidence matrix, B, for this traffic network is given in the file `traffic.mat`. The rows of B are associated with the nodes of the network and the columns of B with the links. The ith column of B has a 1 in the row corresponding to the **tail** node of link e_i and a -1 in the row that corresponds to the **head** node of link e_i. Each node represents an intersection between highways (and some of the area around).

Each link e, where $e = e_1, \ldots, e_{28}$, has a maximum flow capacity C_e. The capacities are given as a vector in the file `capacities.mat`, where C_{e_i} is given by entry i. The flow capacities are retrieved from measured data, as described in Appendix A. Further, each link has a minimum traveling time l_e, which the drivers experience when the road is empty. In the same manner as for the capacities, the minimum traveling times are given as a vector in the file `traveltime.mat`. Theses values are simply retrieved by dividing the length of the

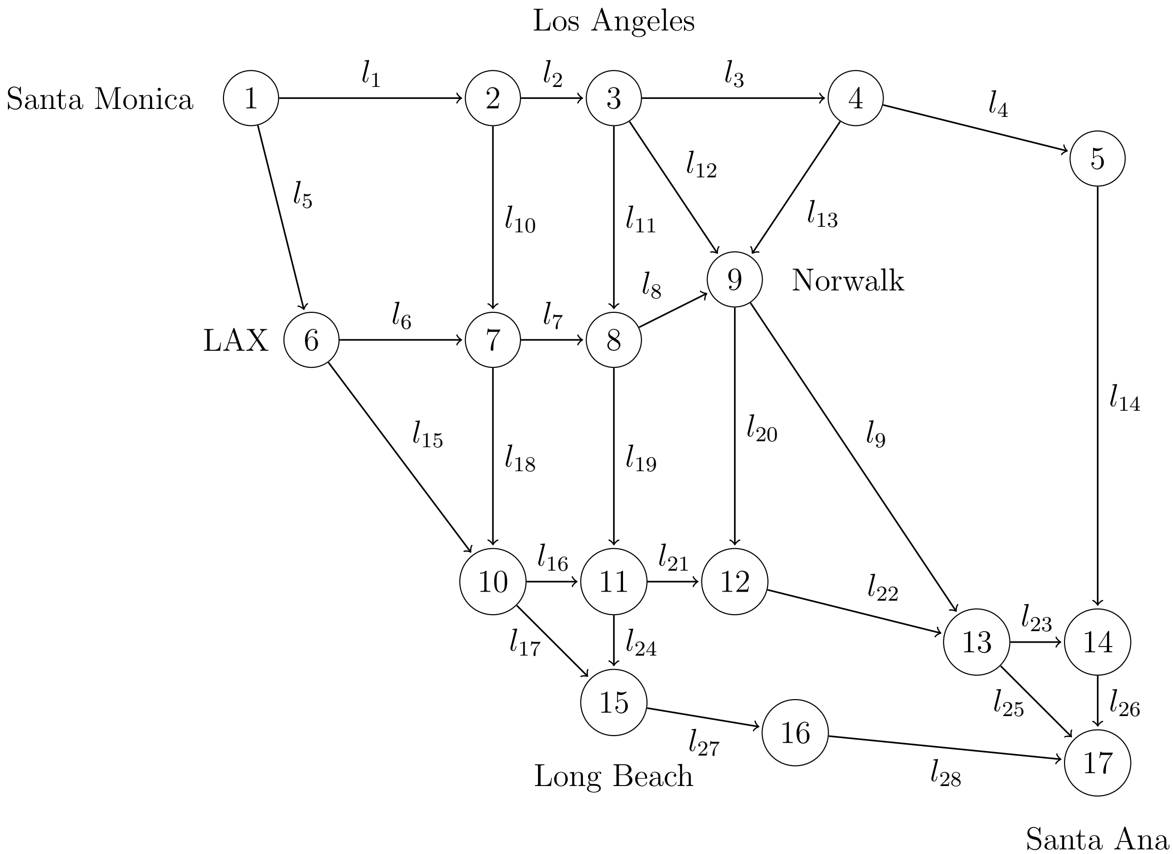

Figure 2: Some possible paths from Santa Monica (node 1) to Santa Ana (node 17).

highway segment with the assumed speed limit 60 miles/hour. For each link, we introduce the delay function

$$d_e(f_e) = \frac{l_e}{1 - f_e/C_e}, \quad 0 \le f_e < C_e.$$

For $f_e \ge C_e$, the value of $d_e(f_e)$ is considered as $+\infty$.

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 intersection is included in the node, the traveling time might be a bit shorter than one could expect.

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

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

For the following subproblems you will use CVX which is a MATLAB-based modeling system for convex optimization. To download CVX, go to `cvxr.com/cvx/download`. You will use CVX to solve optimization problems concerning flows on the approximative traffic network of Los Angeles. As an example, the following flow optimization problem

$$\begin{aligned} \text{minimize} \ & \sum_{e=1}^{M} f_e^2 \\ \text{subject to} \ & Bf = \lambda - \mu \\ & 0 \leq f \leq C \end{aligned}$$

can be written for CVX in Matlab as

```
cvx_begin
    variable f(M)
    minimize sum(f.*f)
    subject to
            B*f == lambda - mu
            0 <= f <= c
cvx_end
```

Installation instructions can be found in CVX's users' guide `http://cvxr.com/cvx/doc/install.html`. If anything is unclear with how to download or use CVX, you can ask the TAs.

For the following points, we assume that all net inflows are zero except for the one at node 1, where we keep the computed one from part c). We also assume that all of the net inflow at node 1 leaves the network at node 17.

d) Using CVX, find the **social optimum** f^* with respect to the delays, i.e., minimize

$$\sum_{e\in\mathcal{E}} f_e d_e(f_e) = \sum_{e\in\mathcal{E}} \frac{f_e l_e}{1 - f_e/C_e} = \sum_{e\in\mathcal{E}} \left(\frac{l_e C_e}{1 - f_e/C_e} - l_e C_e \right)$$

subject to the constraints on the flows. (Hint: Use the CVX-function `inv_pos`.)

e) Using CVX, find the **Wardrop equilibrium** $f^{(0)}$. Hint: Use the cost function

$$\sum_{e\in\mathcal{E}} \int_0^{f_e} d_e(x)dx.$$

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

g) Instead of the total delay, let the **cost** be the **total additional delay** compared to the total delay in free flow, i.e.,

$$c_e(f_e) = f_e(d_e(f_e) - l_e).$$

Compute the system optimum f^* for the costs above (the CVX function `quad_over_lin` can be useful for this) and construct tolls such that the Wardrop equilibrium $f^{(\omega)}$ coincides with f^*. Verify your result with CVX. (Hint: Use Theorem 5.2 in the lecture notes.)

2 Coloring

In this part, we will study graph coloring as an application of distributed learning in potential games. The aim of graph coloring is to assign a color to each node in a given undirected graph, such that none of the neighbors of a node have the same color as that node. We will begin with a simple line graph to illustrate the distributed learning algorithm, and then look at a more general example, which can be seen as a distributed solution approach to assign non-interfering channels to wifi access points.

a) In this example, study a line graph with 10 nodes. Denote each node's state $X_i(t)$ and the set of possible states $\mathcal{C} = \{\text{red}, \text{green}\}$. At initialization, each node is red, i.e, $X_i(t) = \text{red}$ for all $i = 1, \dots, 10$. Every time instance t, one node $I(t)$, chosen uniformly at random, wakes up and updates its color. The color which the node updates to is chosen from a probability distribution given by

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

where the cost is given by

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

In the above expression, $\eta(t)$ is the inverse of the noise. To decide upon a good choice of $\eta(t)$, some heuristics are required, but it is preferable to have it increasing in time so that the noise is decreasing. For this exercise you can start with

$$\eta(t) = \frac{t}{100}.$$

To see how close to a solution the learning algorithm is, one can study the potential function, which is given by

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

where $\mathcal{V}$ is the set of nodes. If the potential is zero, there are no conflicting nodes and a solution is found.

Your task is to simulate the learning dynamics described above in MATLAB (or similar). To animate the dynamics in MATLAB, the functions `gplot` and `scatter` can be useful in order to draw the graph, together with the `pause` function to slow down the animation. Include plots of the potential function in your report.

b) Now, you will use the coloring algorithm for the problem of assigning wifi-channels to routers. The adjacency matrix of a network of 100 routers is given in `wifi.mat` and the routers' coordinates are given in `coord.mat`. Here, a link between two nodes means that the two routers are able to interfere with each other. The set of possible states is $\mathcal{C} = \{1 : \text{red}, 2 : \text{green}, 3 : \text{blue}, 4 : \text{yellow}, 5 : \text{magenta}, 6 : \text{cyan}, 7 : \text{white}, 8 : \text{black}\}$, where colors represent frequency bands, and the cost function is

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

This cost function symbolizes that routers that are close by should not use channels with the same frequency band or a frequency band right next to each other. Use $\eta(t) = t/100$ and verify that a near-zero potential solution is found after a sufficient number of iterations. Try with some other function $\eta(t)$ and observe what happens. Include plots of the potential functions for some different tested cases and an illustration of the node coloring corresponding to the smallest potential function (that is, the best obtained solution) in the report.

c) *Optional:* Construct your own graph and and evaluate what happens for different choices of $\eta(t)$, i.e., small and constant, large and constant, more or less slope on increasing η etc.. Comment on what you observe.

A PeMS data for the traffic network

Link	**Freeway**	**PM Start**	**PM End**	f_{max} [veh/h]	f [veh/h]
l_1	I10-E	3.46	12.21	8741	7524
l_2	I10-E	12.76	15.45	9864	6537
l_3	SR60-E	3.97	11.71	13350	11139
l_4	SR60-E	12.24	23.44	10926	9282
l_5	I405-S	52.93	45.14	13707	9282
l_6	I105-E	2.50	7.20	6960	6398
l_7	I105-E	7.56	13.20	7422	6728
l_8	I105-E	13.86	17.30	6678	5988
l_9	I5-S	123.21	114.71	6297	5951
l_{10}	I110-S	20.53	14.22	11102	9557
l_{11}	I710-S	17.54	11.14	8899	7423
l_{12}	I5-S	129.92	123.63	8970	7423
l_{13}	I605-S	19.15	12.41	9753	6814
l_{14}	SR57-S	15.88	5.05	9719	8536
l_{15}	I405-S	44.37	37.08	9083	7652
l_{16}	SR91-E	0.56	5.40	7416	6537
l_{17}	I405-S	36.34	31.82	13353	11924
l_{18}	I110-S	13.33	9.93	11216	9640
l_{19}	I710-S	10.29	8.33	10947	8161
l_{20}	I605-S	9.35	7.25	10019	8603
l_{21}	SR91-E	6.20	10.22	8732	7974
l_{22}	SR91-E	11.37	18.14	10763	9446
l_{23}	SR91-E	19.17	23.87	6677	5562
l_{24}	I710-S	7.54	4.29	9403	6719
l_{25}	I5-S	113.99	107.25	10355	9455
l_{26}	SR57-S	4.75	0.37	9067	6686
l_{27}	I405-S	30.55	20.65	11990	10833
l_{28}	SR22-E	2.54	11.46	8258	7403

Comments:

- All the flow data is acquired from PeMS `http://pems.dot.ca.gov/`.
- The f_{max} is the maximum flow detected on one of the segments between the nodes, during the time period 12-01-2016 to 12-02-2016.
- The flow vector is the maximum flow measured on each segment at 08-02-2014 between 17.00 and 18.00.
- The free flow traveling time was computed by assuming $v_{ff} = 60$ miles/h for all links.