Homework I

- Solutions have to be uploaded on the course website by h 23:59 of Sunday November 14, 2021 under the name Homework1.
- You may implement your solutions in any language you see fit, but the teacher assistant can only guarantee you support with MATLAB and Python. 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. Upload a PDF lab-report together with your code.
- The PDF should read like a standard lab-report, including a description of what you are doing, a summary of the theory used, proper presentation of results (including readable figures with axis labels, if any). Writing the report in Latex is strongly encouraged. Clarity of the presentation (especially if the report is hand-written) and ability to synthesize are part of the evaluation of the homework.
- Comment your code well. Clarity is more important than efficiency.
- 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.

Exercise 1. Consider unitary *o*-*d* network flows (i.e., integer flows whose integer units cannot be split) on the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in Figure 1, and assume that each link l has integer capacity C_l.

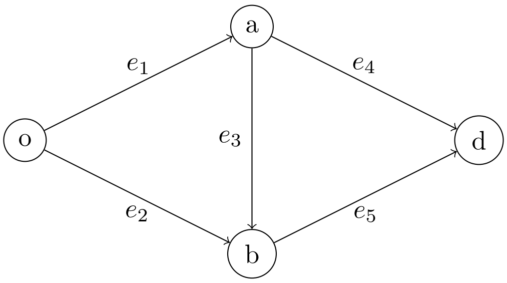

Figure 1

(a) What is infimum of the total capacity that needs to be removed for no feasible unitary flows from o to d to exist?

(b) Assume that the link capacities are

$$C_1 = C_4 = 3, \qquad C_2 = C_3 = C_5 = 2. \tag{1}$$

Where should 1 unit of additional capacity be allocated in order to maximize the feasible throughput from o to d? What is the maximal throughput?

(c) Consider link capacities (1). Where should 2 units of additional capacity be allocated in order to maximize the feasible throughput from o to d? Compute all the optimal capacity allocations for this case and the optimal throughput.

(d) Consider link capacities (1). Where should 4 units of additional capacity be allocated in order to maximize the feasible throughput from o to d? Compute all the optimal capacity allocations for this case. Among the optimal allocations, select the allocation that maximizes the sum of the cut capacities.

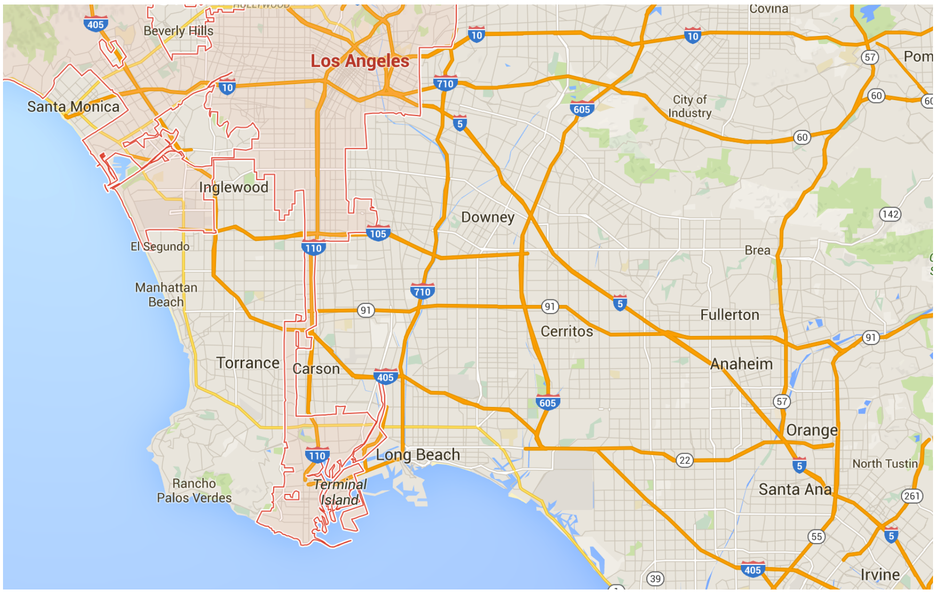

Figure 2: The highway network in Los Angeles.

Exercise 2. Consider the following problem. There are a set of people (p_1, p_2, p_3, p_4) and a set of books (b_1, b_2, b_3, b_4). Each person is interested in a subset of books, specifically

$$p_1 \to (b_1, b_2), \quad p_2 \to (b_2, b_3), \quad p_3 \to (b_1, b_4), \quad p_4 \to (b_1, b_2, b_4).$$

(a) Represent the interest pattern by using a simple bipartite graph.

(b) Exploit max-flow problems to establish whether there exists a perfect matching that assigns to every person a book of interest. If a perfect matching exists, find at least a perfect matching.

(c) Assume now that there are multiple copies of book, specifically the distribution of the number of copies is $(2, 3, 2, 2)$, and there is no constraint on the number of books that each person can take. The only constraint is that each person can not take more copies of the same book. Use the analogy with max-flow problems to establish how many books of interest can be assigned in total.

(d) Starting from point (c), suppose that the library can sell a copy of a book and buy a copy of another book. Which books should be sold and bought to maximize the number of assigned books?

Exercise 3. We are given the highway network in Los Angeles, see Figure 2. To simplify the problem, an approximate highway map is given in Figure 3, 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 i-th column of B has 1 in the row corresponding to the tail node of link e_i and (-1) in the row corresponding to the head node of link e_i. Each node represents an intersection between highways (and some of the area around).

Each link $e_i \in \{e_1, \ldots, e_{28}\}$, has a maximum flow capacity C_{e_i}. The capacities are given as a vector C_e in the file *capacities.mat*. Furthermore, each link has a minimum travelling time l_{e_i}, which the drivers experience when the road is empty. In the same manner as for the capacities, the minimum travelling times are given as a vector l_e in the file *traveltime.mat*. Theses values are simply retrieved by dividing the length of the highway segment with the assumed speed limit 60 miles/hour.

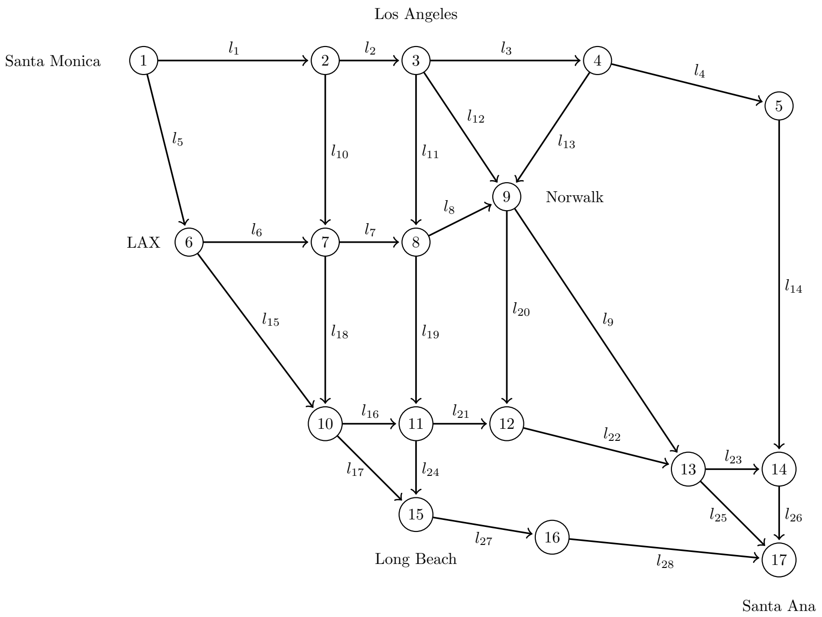

Figure 3: Some possible paths from Santa Monica (node 1) to Santa Ana (node 17).

For each link, we introduce the delay function

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

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

If you use Python to solve the Exercise, the files .mat be loaded by using the following code.

```
file = scipy.io.loadmat('capacities.mat')
capacities = file.get('capacities')
capacities = capacities.reshape(28,)

file = scipy.io.loadmat('traveltime.mat')
traveltime = file.get('traveltime')
traveltime = traveltime.reshape(28,)

file = scipy.io.loadmat('flow.mat')
flow = file.get('flow')
flow = flow.reshape(28,)

file = scipy.io.loadmat('traffic.mat')
traffic = file.get('traffic')
```

(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.

(b) Find the maximum flow between node 1 and 17.

(c) Given the flow vector in *flow.mat*, compute the external inflow ν satisfying $Bf = \nu$.

In the following, we assume that the exogenous inflow is zero in all the nodes except for node 1, for which ν_1 has the same value computed in the point **(c)**, and node 17, for which $\nu_{17} = -\nu_1$.

(d) Find the social optimum f^* with respect to the delays on the different links $d_e(f_e)$. For this, minimize the cost function

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

subject to the flow constraints.

Hint: use the Tutorial (https://www.cvxpy.org/tutorial/functions/index.html) to learn how to code functions in cvxpy.

(e) Find the Wardrop equilibrium $f^{(0)}$. For this, use the cost function

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

(e) 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 $d_e(f_e) + \omega_e$. compute the new Wardrop equilibrium $f^{(\omega)}$. What do you observe?

(f) Instead of the total delay, let the cost be the total additional delay compared to the total delay in free flow be given by

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

subject to the flow constraints. Compute the system optimum f^* for the costs above. Construct tolls ω_e^*, $e \in \mathcal{E}$ such that the new Wardrop equilibrium with the constructed tolls $f^{(\omega^*)}$ coincides with f^*. Compute the new Wardrop equilibrium with the constructed tolls $f^{(\omega^*)}$ to verify your result.