Homework I

- The deadline for on time submission is November 19, 2023. The grade is lowered in case of late submission. The deadline for late submission is 5 days before the oral exam. The solutions have to be uploaded on the course website under the name Homework1.
- The exercises may be solved by using numerical codes or by hand. You may implement your codes in any language, 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 and 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. You can work in group of up to 5 students and write a unique PDF report. However, every student has to submit her/his copy of the final report and code, and specify whom she/he has collaborated with and on what particular part of the work.

Exercise 1. Consider the network in Figure 1 with link capacities

$$c_1 = c_3 = c_5 = 3, \qquad c_6 = 1, \qquad c_2 = c_4 = 2.$$

(a) What is the minimum aggregate capacity that needs to be removed for no feasible flow from o to d to exist?

(b) What is the maximum aggregate capacity that can be removed from the links without affecting the maximum throughput from o to d?

(c) You are given $x > 0$ extra units of capacity ($x \in \mathbf{Z}$). How should you distribute them in order to maximize the throughput that can be sent from o to d? Plot the maximum throughput from o to d as a function of $x \geq 0$.

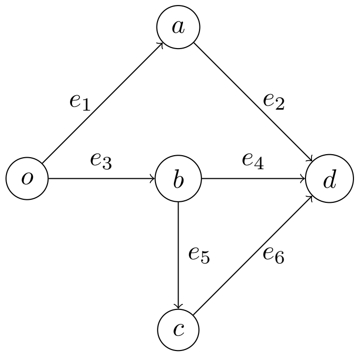

Figure 1

Exercise 2. 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) Exploit max-flow problems to find a perfect matching (if any).

(b) Assume now that there are multiple copies books, and the distribution of the number of copies is $(2, 3, 2, 2)$. Each person can take an arbitrary number of different books. Exploit the analogy with max-flow problems to establish how many books of interest can be assigned in total.

(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?

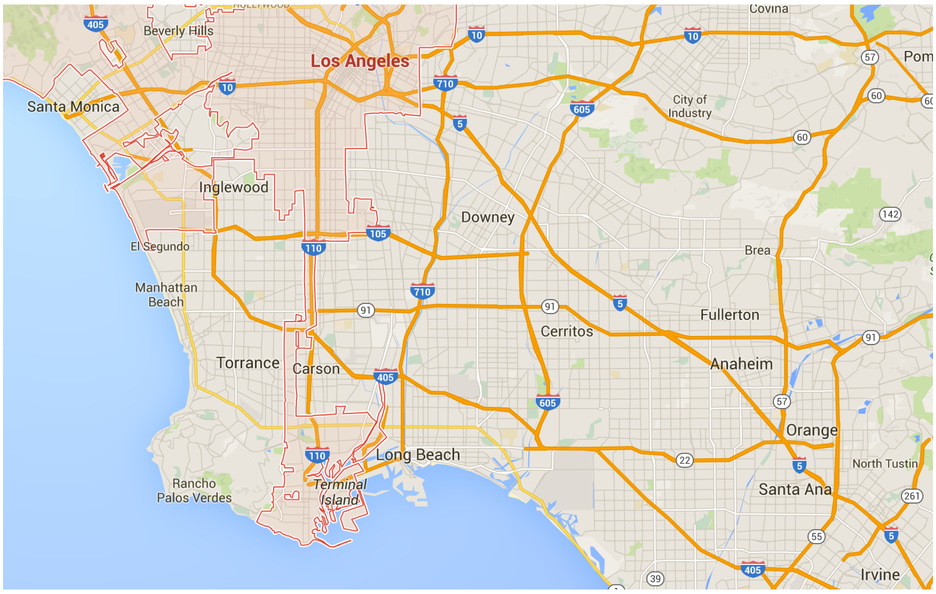

Figure 2: The highway network in Los Angeles.

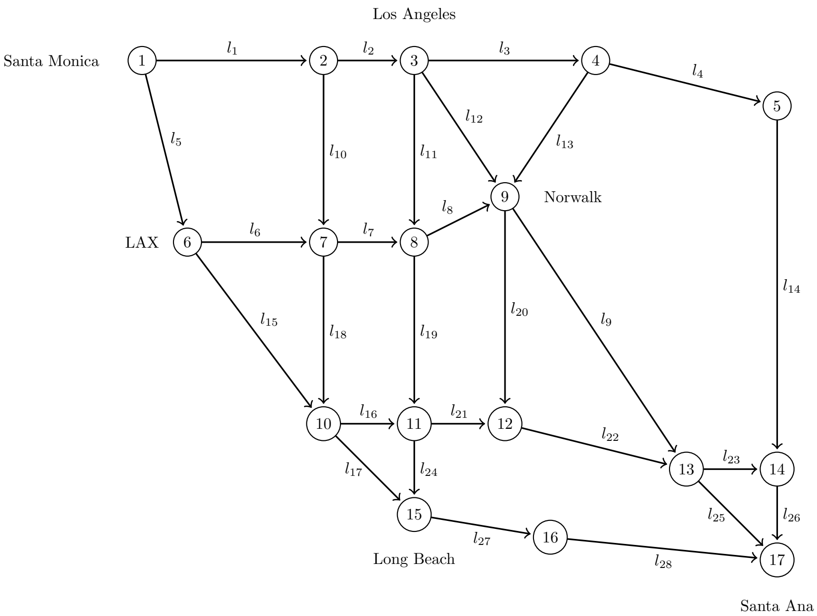

Figure 3: Some possible paths from Santa Monica (node 1) to Santa Ana (node 17).

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 $e_i \in \{e_1, \dots, e_{28}\}$, 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*. These values are simply retrieved by dividing the length of the highway segment with the assumed speed limit 60 miles/hour. For each link, we introduce the delay function

The formula describes how the delay (or travel time) on edge 'e' increases as the flow 'f(e)' approaches the edge's capacity 'c(e)'.

minimum travelling times

$$\tau_e(f_e) = \frac{l_e}{1 - f_e/c_e}, \qquad 0 \leq f_e < c_e.$$

flow

maximum flow capacity

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

If you use Python to solve the Exercise, you can load the .mat files by the following code:

```
f = scipy.io.loadmat('flow.mat')["flow"].reshape(28,)
C = scipy.io.loadmat('capacities.mat')["capacities"].reshape(28,)
B = scipy.io.loadmat('traffic.mat')["traffic"]
l = scipy.io.loadmat('traveltime.mat')["traveltime"].reshape(28,)
```

(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.

if +: node 'i' has a net inflow
if -: node 'i' has a net outflow

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

Incidence Matrix Flow Vector

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

External Inflow Vector

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 $\nu_{17} = -\nu_1$.

The social optimum refers to a flow distribution across the network's links that minimizes the total cost or delay for all users in the system.

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

*** Goal
Minimize the total delay across all edges in the network.

Total delay

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

account for the contribution of that edge to the overall network delay.

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

total cost for a flow distribution across all edges in the network to find the Wardrop equilibrium, you would minimize this cost

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

flow at the system optimum

derivative of the total cost function

Toll

The delay on link 'e' when the flow is at the system optimum.

*Goal: The goal is to align the Wardrop equilibrium with the system optimum.

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

The new Wardrop equilibrium flow

(g) Instead of the total travel time, let the cost for the system be the total additional travel time compared to the total travel time in free flow, given by

actual travel time

additional cost $\psi_e(f_e) = f_e(\tau_e(f_e) - l_e)$

The free-flow travel time

subject to the flow constraints. Compute the system optimum f^* for the costs above. Construct a toll vector ω^* such that the Wardrop equilibrium $f^{(\omega^*)}$ coincides with f^*. Compute the new Wardrop equilibrium with the constructed tolls $f^{(\omega^*)}$ to verify your result.

***Goal: Align user-optimal behavior (Wardrop equilibrium) with the socially optimal flow (system optimum) by imposing tolls.