# Network Dynamics and Learning
Homework 1

November 13, 2022

Francesca Cossu, s305746@studenti.polito.it
Co-Author: Niccolò Dimonte, s303439@studenti.polito.it

## Exercise 1

Consider the network in fig with link capacities:

$$c_2 = c_4 = c_6 = 1$$

$$c_1 = c_3 = c_5 = 2$$

### 1.a

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

The minimum aggregate capacity that needs to be removed for no feasible flow to exist is equal to the maximum flow computed with min-cut max-flow theorem. Computing the cuts:

$$U : \{0\}; U^c : \{1,2,3,4\} : C_u = 4$$
$$U : \{0,1\}; U^c : \{2,3,4\} : C_u = 3$$
$$U : \{0,2\}; U^c : \{1,3,4\} : C_u = 5$$
$$U : \{0,2,3\}; U^c : \{1,4\} : C_u = 4$$
$$U : \{0,2,3,4\}; U^c : \{4\} : C_u = 3$$

The maximum flow is 3: the minimum capacity $C^*_{0,4}$ that needs to be removed is:

$$C^*_{0,4} = \underset{U \subseteq V}{minC_u} = 3$$

### 1.b

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

In order to computing the maximum aggregate capacity that can be removed without affecting the maximum throughput, we have to check not saturated links. Computing the difference between the total capacity and the maximum flow on each link, it's visible that the maximum capacity is 3.

### 1.c

*You are given $x > 0$ extra units of capacity. 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 $x \geq 0$.*

In this question we need to modify the capacity of the saturated links. For doing so first we have to consider the cuts with minimum capacity and add 1 from x surplus capacity on the saturated links.

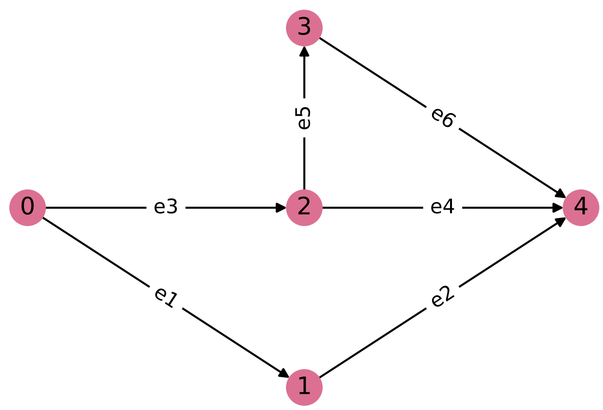


Figure 1: Network exercise 1

To increase throughput, we add capacity on most shared links from the min-cuts.

$$
\begin{aligned}
U &: \{0\}, \quad U^c : \{1,2,3,4\} \rightarrow C_u = 4 \\
U &: \{0,1\}, \quad U^c : \{2,3,4\} \rightarrow C_u = 3 \\
U &: \{0,2\}, \quad U^c : \{1,3,4\} \rightarrow C_u = 5 \\
U &: \{0,1,2\}, \quad U^c : \{3,4\} \rightarrow C_u = 4 \\
U &: \{0,1,2,3\}, \quad U^c : \{4\} \rightarrow C_u = 3
\end{aligned}
$$

Considerate the cuts with the respective links:

$$
\begin{aligned}
U : \{0,1\}, \quad U^c : \{2,3,4\} \rightarrow C_u = 3 \\
U : \{0,1,2,3\}, \quad U^c : \{4\} \rightarrow C_u = 3 \\
U : \{0,1\} \quad C_u = (0,2) + (1,4) \\
U : \{0,1,2,3\} \quad C_u = (1,4) + (2,4) + (3,4)
\end{aligned}
$$

$(a, d)$ is present in both cut, so we can put the additional capacity here.
Recomputing the cuts:

$$
\begin{aligned}
U &: \{0\}, \quad U^c : \{1,2,3,4\} \rightarrow C_u = 4 \\
U &: \{0,1\}, \quad U^c : \{2,3,4\} \rightarrow C_u = 4 \\
U &: \{0,2\}, \quad U^c : \{1,3,4\} \rightarrow C_u = 5 \\
U &: \{0,1,2\}, \quad U^c : \{3,4\} \rightarrow C_u = 5 \\
U &: \{0,1,2,3\}, \quad U^c : \{4\} \rightarrow C_u = 4
\end{aligned}
$$

Iterating more times we see **throughput grows with capacity x**.

# Exercise 2

The second exercise is about the problem of matching in a bipartite graph with two subsets of nodes, $V_0$ the set of people and $V_1$ the set of books. A matching or independent link set in an undirected graph is a subset of links that have no node in common.

## 2.a

*Exploit max-flow problems to find a perfect matching.*

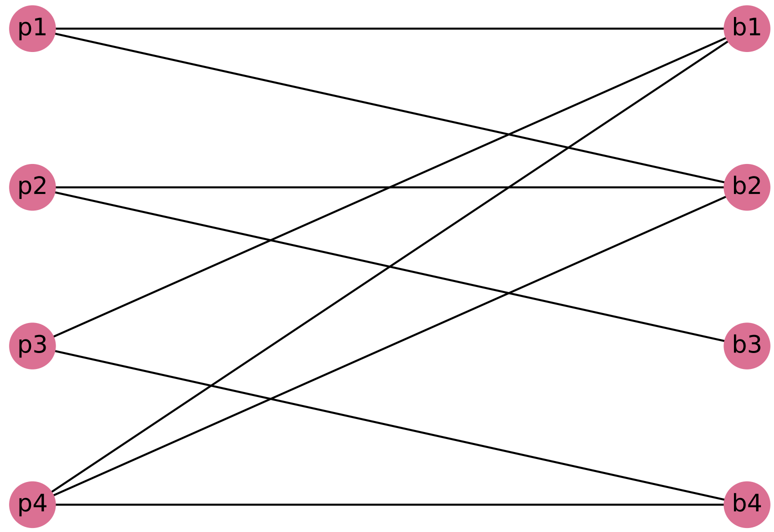


Figure 2: Bipartite network exercise 2.

A matching is perfect if all nodes are matched; in the particular case of bipartite graphs, with respect to the partition $V = V_0 \cup V_1$, a matching $M$ in $G$ is called $V_h - perfect$ if every node $i$ in $V_h$ is matched in $M$. This result is known as **Hall's Marriage Theorem**.
In this case, exploiting an analogy between maximal flows and perfect matching, I constructed an auxiliary graph $G_1$, represented in figure 3 with the following characteristics:

Given a simple bipartite graph $G = (V, E)$, I considered the directed capacitated graph $G1$, with node set $V \cup s \cup t$, and edge set constructed as follows:
- for every node $n \in V_0$, add an edge (s,n), with capacity 1;
- for every node $n \in V_1$, add an edge (n,t), with capacity 1;
- for every undirected edge $(i, j)$ in $G$, add a directed edge $(i, j)$ in $G1$ with capacity 1.

There is an analogy between perfect matching in G and maximal flow on the auxiliary network G1. In particular, a $V_0$-perfect matching on $G$ exists if and only if it there exists a flow with throughput $|V_0|$ on the network $G1$.

In fact, after calculating the maximum flow, it is equal to $V_0$, the number of people and the number of books, and this confirms that a perfect matching exist, as it visible in figure 4.

## 2.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.*

I assumed that the outgoing capacity going trough $d$ are respectively (2,3,2,2) and add infinite capacities on edges from the node $o$ because the number of books for each person is arbitrary.
With the updated capacities, it can be found that the maximum flow is 8, number that correspond to the number of possible matchings or more specifically the number of books that can be assigned in total.

## 2.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?*

To find a way to maximize the number of assigned I first calculated the minimum cut, finding that the book $b_3$ it's not included in the cut, meaning that there are still copies of that specific book. Computing the difference between edges' flow and edges' capacity I then found that the edge between $p_1$ and $b_1$ has one capacity left, so I can conclude that the sells could be optimized buying a book $b_1$ and selling a book $b_3$.

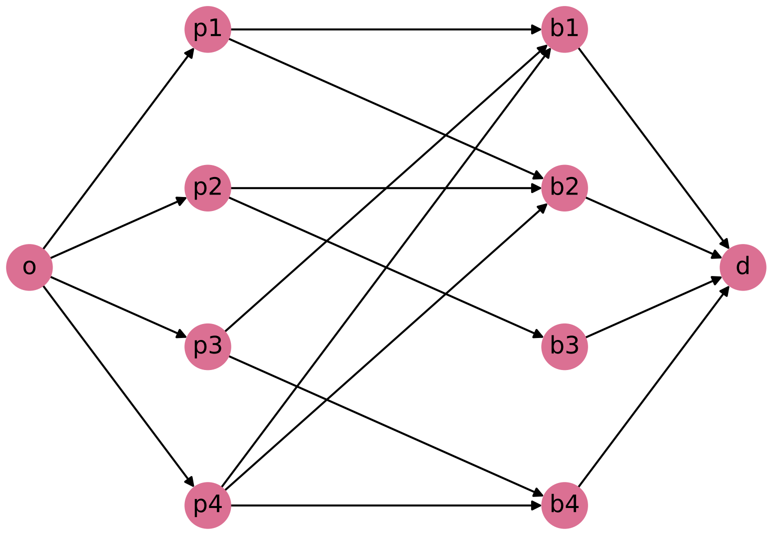


Figure 3: Auxiliary graph.

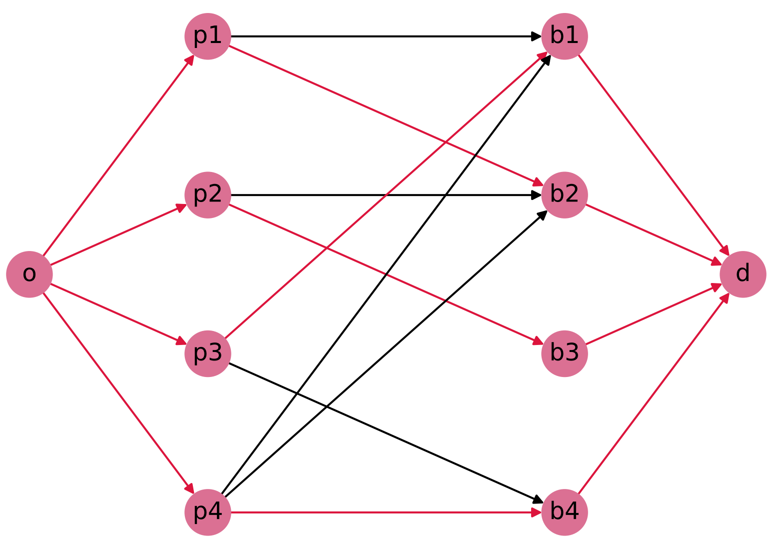


Figure 4: Perfect matching.

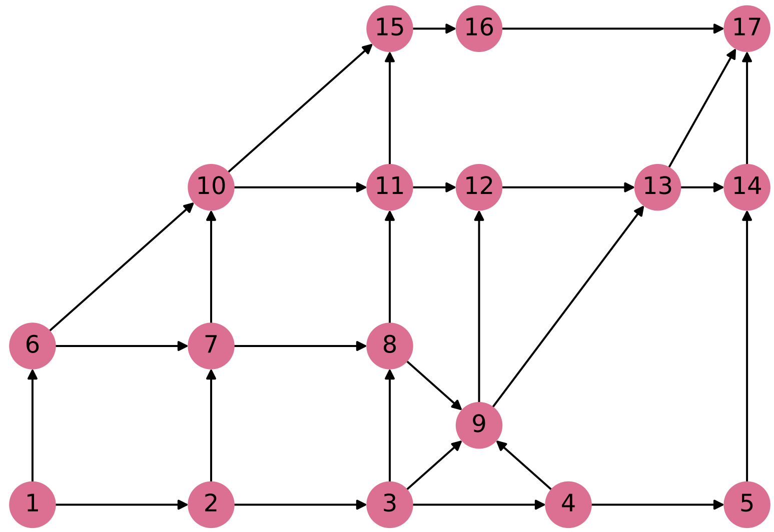


Figure 5: Los Angeles' highway network.

# Exercise 3

This exercise is about the highway network in Los Angeles, presented in figure 5

Given a multigraph $(V, E)$, an exogenous network flow is a vector $\nu \in \mathrm{R}^V$ such that

$$\sum_{i \in V} \nu_i = 0.$$

A network flow is a vector $f \in \mathrm{R}^E$ satisfying a positivity constraint and a mass conservation constraints, i.e.,

$$f \geq \mathbf{0}, \quad Bf = \nu.$$

Every edge is endowed with a separable non-decreasing convex cost function $\psi_e(f_e)$ such that $\psi_e(0) = 0$.
Given an exogenous flow $\nu$ and a network with node-edge matrix $B$, we study the following optimization problem:

$$f^* \in \underset{\substack{f \in \mathrm{R}^E_+ \\ Bf = \nu}}{\arg\min} \sum_{e \in E} \psi_e(f_e).$$

The ratio $\psi_e(f_e)/f_e$ may be interpreted as the cost per unit of flow sent along the edge $e$. The convexity of $\psi_e(f_e)$ is thus equivalent to requiring that the marginal cost for sending some flow on each edge is non-decreasing in the flow itself.

## 3.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.*

This in a linear problem, that is a special case of a convex problem.
Since the problem is linear, the solution will be to allocate the flow on a convex combination of the optimal paths. The optimal paths can be thus deduced from the non-zero components of the optimal flow.

I construct an algorithm (using the Python-embedded language cvxpy) that aims at minimize the cost, I found that the shortest path is the one composed by the edges 1,2,12,9 and 25 as it's noticeable in figure 6.

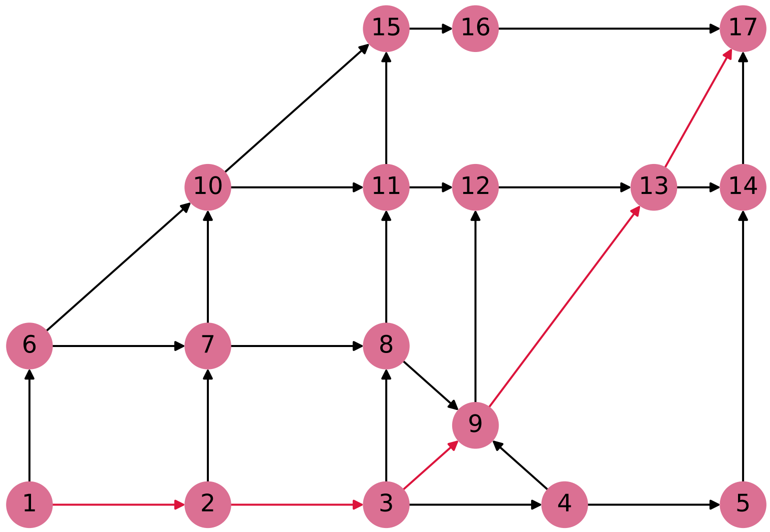


Figure 6: The shortest path.

### 3.b

*Find the maximum flow between node 1 and 17.*

The maximum flow is now obtained simply setting the capacities and computing the Networkx function. We observe that the maximum flow is equal to **18605**.

### 3.c

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

For this task is simply obtainable by doing the matrix multiplication between the node-link incidence matrix $B$ and the network flow vector $f$. I also computed the exogenous inflow requested from point d, where it is zero in all nodes except for node 1 and node 17 for whom $\nu_{17} = -\nu_1$.

### 3.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*

The cost function subject to the flow constrains:

$$\psi_e(f_e) = \sum_{e \epsilon \varepsilon} \frac{f_e l_e}{1 - f_e/c_e} = \sum_{e \epsilon \varepsilon} \frac{l_e c_e}{1 - f_e/c_e} - l_e c_e$$

A cost function describe the cost incurred when a flow $f_e 0$ passes trough link $e$ in $\varepsilon$.

The social optimum flow distribution is defined as the one minimizing the total delay $\sum_{e \in \mathcal{E}} \psi_e(f_e)$, where on every link $e \in \mathcal{E}$, the cost is $\psi_e(f_e) = f_e \tau_e(f_e)$ with $\tau_e(f_e)$ being the delay function

$$\tau_e(f_e) = \frac{l_e}{1 - f_e 7 c_e}$$

This is solvable with the Social Optimum Flow using cvxpy algorithm.

### 3.e1

*Find the Wardrop equilibrium $f^{(0)}$*

The Wardrop equilibrium is a path flow distribution such that if path p is used, then the cost of the path is minimal. We say that $f$ is a Wardrop equilibrium if it is induced by a Wardrop equilibrium $z$ via $f = Az$.

$f^{(0)}$ is a Wardrop equilibrium if and only if it is solution of a network flow optimization given that the cost functions $\psi_e(f_e)$ are chosen as

$$\psi_e(f_e) = \int_0^{f_e} \tau_e(s) \, \mathrm{d}s.$$

For our problem the objective function will be:

$$\sum_{e\epsilon\varepsilon} \int_0^{f_e} \tau_e(s)ds = \sum_{e\epsilon\varepsilon} \int_0^{f_e} \frac{l_e}{1 - f_e/c_e} = \sum_{e\epsilon\varepsilon} -l_e c_e log(1 - \frac{f_e}{c_e})$$

Then I compute the wardrop equilibrium using the formula. But this is not the cost of wardrop equilibrium, so we compute that using the formula: $\sum f_{war}\tau_e(f_{war})$

## 3.e2

*Introduce tolls, such that the toll on link e is $\omega^* = f_e^* {\tau_e}'(f_e^*)$ where $f_e^*$ is the flow as the system optimum. Now the delay on link e is given by $\tau_e(f_e) + \omega_e$.*

The objective function will be:

$$\sum_{e\epsilon\varepsilon} \int_0^{f_e} (\tau_e(s) + \omega_e)ds = \sum_{e\epsilon\varepsilon} \int_0^{f_e} \tau_e(s)ds + f_e\omega_e = \sum_{e\epsilon\varepsilon} \int_0^{f_e} \frac{l_e}{1 - f_e/c_e} = \sum_{e\epsilon\varepsilon}(-l_e c_e log(1 - \frac{f_e}{c_e}) + f_e\omega_e)$$

The Wardrop cost in this second case is lower than the Wardrop cost calculated in the question e.1 and the resulting flow vector $f^{(\omega)}$ is also the social optimum flow, as expected.

## 3.f

*Instead of the total travel time, let the cost for the system be the total additional delay compared to the total delay in free flow, given by:*

$$\psi(f_e) = f_e(\tau_e(f_e) - l_e)$$

*Compute the system optimum $f^*$ for the costs above. Construct tolls $\omega^*$ 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.*

The new System Optimum Flow is given by optimizing the following function:

$$f^* = \underset{f\epsilon R^E, Bf=\nu}{\text{argmin}} \sum_{e\epsilon\varepsilon} \psi_e(f_e) = \underset{f\epsilon R^E, Bf=\nu}{\text{argmin}} \sum_{e\epsilon\varepsilon} f_e(\tau_e(f_e) - l_e) = \underset{f\epsilon R^E, Bf=\nu}{\text{argmin}} \sum_{e\epsilon\varepsilon} (\tfrac{l_e c_e}{1 - f_e/c_e} - l_e c_e - l_e f_e)$$

As we know from theory, an optimal toll is given by

$$\omega_e^* = \psi_e'(f_e^*) - \tau_e(f_e^*)$$
$$\psi'(f_e^*) = \tau(f_e^*) + f_e^*\tau'(f_e^*) - l_e$$
$$\omega_e^* = \tau_e(f_e^*) + f_e^*\tau_e'(f_e^*) - l_e$$
$$\omega_e^* = \frac{f_e^* l_e c_e}{(c_e - f_e^*)^2} - l_e$$

The objective function for the Wardrop equilibrium with tolls will be:

$$\sum_{e\epsilon\varepsilon} \int_0^{f_e} \tau_e^{\omega_e^*}(s)ds = \sum_{e\epsilon\varepsilon} \int_0^{f_e} \tau_e(s)ds + \omega_e^* f_e = \sum_{e\epsilon\varepsilon}(-l_e c_e log(1 - \frac{f_e}{c_e}) + \omega_e^* f_e)$$