Homework 1

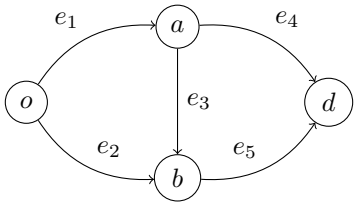
Andrea Rubeis, s290216

Abstract

Here are reported the solutions of the first homework of Network Dynamics and Learning course, solution discussed with Shannon Mc Mahon (s289958). Code here.

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 below and assume that each link l has integer capacity C_l.



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

The minimum number of capacities to be removed for no feasible unitary flows from o to d to exist is equal to the *min-cut* of the network. To find the min-cut I apply the *Menger's theorem* (which is a special case of min-cut theorem where all edges'capacities have value equal to 1) and I find out that the infimum of the total capacity that has to be removed is 2 for example by removing one capacity to e_4 and one capacity to e_5.

(b) **Assume that the link capacities are:**

$$C_1 = C_4 = 3;\ C_2 = C_3 = C_5 = 2$$

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?

First of all we generate all possible *o-d cuts* and their corresponding *cut capacities*, then we check what is the minimum among them. Denoting with $\mathcal{U}$ the first partition, with $\mathcal{V}$ the second one, with $\mathcal{E}_{\mathcal{U}}$ the edges going from $\mathcal{U}$ to $\mathcal{V}$ and with $\mathcal{C}_{\mathcal{U}}$ their corresponding capacities, we report the results we got in the following table:

$\mathcal{U}$	$\mathcal{V}$	$\mathcal{E}_{\mathcal{U}}$	$\mathcal{C}_{\mathcal{U}}$
o	a, b, d	e_1, e_2	5
o, a	b, d	e_2, e_3, e_4	7
o, b	a, d	e_1, e_5	5
o, a, b	d	e_4, e_5	5

Table 1: All *o-d* cut capacities

By the *min cut theorem* we know that the maximal throughput corresponds to the *min cut capacity*, looking at the table above this is equal to: $\min\left(\mathrm{C}_{\mathcal{U}_1}, \mathcal{C}_{\mathcal{U}_2}, \mathcal{C}_{\mathcal{U}_3}, \mathcal{C}_{\mathcal{U}_4}\right) = 5$. We notice that we have 3 cuts $\mathcal{C}_{\mathcal{U}_1}$, $\mathcal{C}_{\mathcal{U}_3}$, $\mathcal{C}_{\mathcal{U}_4}$ that share the same cut capacity equals to 5. If we want to try to maximize the feasible flow we have to add the additional capacity that we have to the cuts that share the same edges. However, whatever edge we choose we will always have at least one cut whose capacity is still equal to 5 so there is no way to maximize the throughput in this case.

(c) **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**.

For the same reasoning by looking the table above, the optimal assignments would be:

- 1 capacity to e_1 and 1 capacity to e_4.
- 1 capacity to e_1 and 1 capacity to e_5.
- 1 capacity to e_2 and 1 capacity to e_5

In all these cases the min cut capacity and max throughput are equal to 6.

(d) **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 and the optimal throughput.**

In this case, the optimal assignments would be:

- 1 capacity to e_1, 1 capacity to e_2, 1 capacity to e_4 and 1 capacity to e_5
- 2 capacities to e_1 and 2 capacities to e_4, .
- 2 capacities to e_1 and 2 capacities to e_5.
- 2 capacities to e_2 and 2 capacities to e_5
- 2 capacities to e_1, 1 capacity to e_4 and 1 capacity to e_5
- 1 capacity to e_1, 1 capacity to e_2, 2 capacities to e_5

In all the 6 cases the min cut capacity and max throughput are equal to 7, the sum of the cut capacities is 30.

Exercise 2

Consider the following problem. There are a set of people ($p1$; $p2$; $p3$; $p4$) and a set of books ($b1$; $b2$; $b3$; $b4$). 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.**

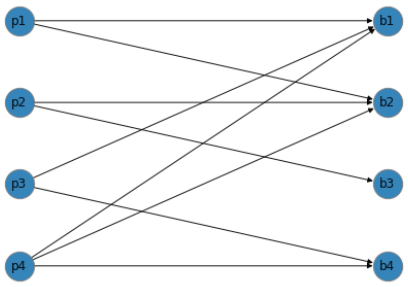


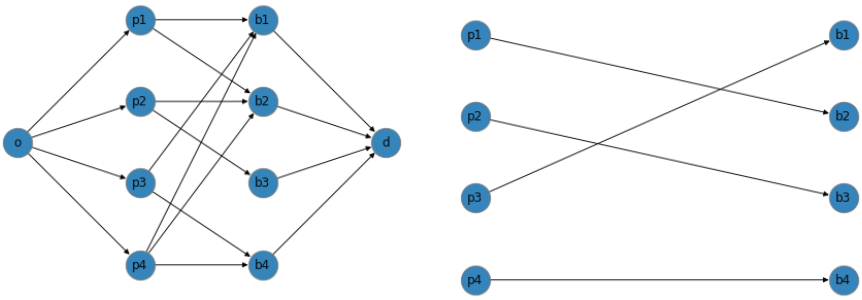
Figure 1: Interest pattern

The problem can be modeled as a bipartite graph made by two indipendent disjoint sets $\mathcal{P} = p_1, p_2, p_3, p_4$ and $\mathcal{B} = b_1, b_2, b_3, b_4$

(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.**

First of all, to see if we can have a perfect matching we need to check that both the partitions of the bipartite graph must have the same cardinality, otherwise we are sure that there is no way to have a perfect matching. Since both the partitions P and B have cardinality equals to 4 we try to see if there is also a perfect matching. To do so we need to construct the corresponding flow network of the bipartite graph (Figure 2.a) by adding one source o, one sink d, $|P| = |B| = 4$ edges from o to all nodes in P and 4 edges from all nodes in B to d. All edges have unitary capacity.

I know from the theory that, if there exists a maximum flow equals to $\frac{n}{2}$ where n is the number of nodes in the original bipartite graph, then there exists a perfect matching as well. We compute the maxflow for the network flow obtained which is 4 so it also exists a perfect matching (Figure 2.b).



(a) Corresponding flow network of the bipartite graph

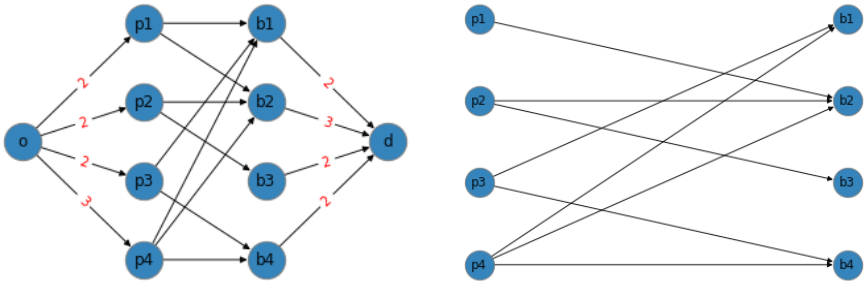
(b) Perfect matching found

Figure 2: Flow network and a perfect matching of the bipartite graph

(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.**

This can be solved by modifying the previous flow network's capacities (Figure 3.a).

Indeed for every person $p \in \mathcal{P}$ we can interpret the amount of books in which he/she is interested in as the (o, p) edge's capacity and the amount of copies avaiable for each book $b \in \mathcal{B}$ as (b, d) edge's capacity. To avoid that one person can take more than one copy of the same book we keep unitary capacity on all edges going from $\mathcal{P}$ to $\mathcal{B}$ (Figure 3.a). Then, we compute the max-flow of the network and we found that to the number of books of interest that can be assigned in total is equal to 8 (Figure 3.b).

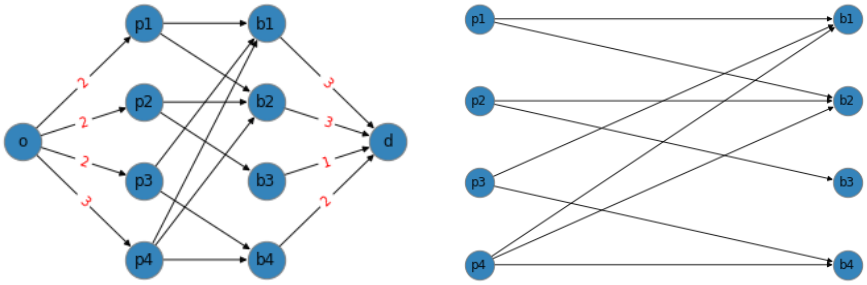


(a) Flow network (b) Matching

Figure 3: Flow network and a matching of the bipartite graph

(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?**

By looking at previous point (c) the demand for the books corresponds to the in-degree of each node b in $\mathcal{B}$ which is $(3, 3, 1, 2)$ and the copies are $(2, 3, 2, 2)$ so we can notice that to maximize the number of assigned books is sufficient to to buy one more copy of book b_1 and sell one copy of book b_3 so now to find the matching we apply the same reasoning of before with the only difference that we change the capacities of edge (b_1, d) and (b_3, d) with 3 and 1 respectively (Figure 4.a). This is proven by the max-flow which now is 9 instead of 8 so the number of assigned books has been maximized (Figure 4.b).



(a) Flow network (b) Matching

Figure 4: Flow network and matching of the bipartite graph

Exercise 3

We are given the highway network in Los Angeles. To simplify the problem, an approximate highway map is given in Figure 5, 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 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 $e_i \in e_1, e_2, ..., e_{28}$ 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:

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

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

(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.**

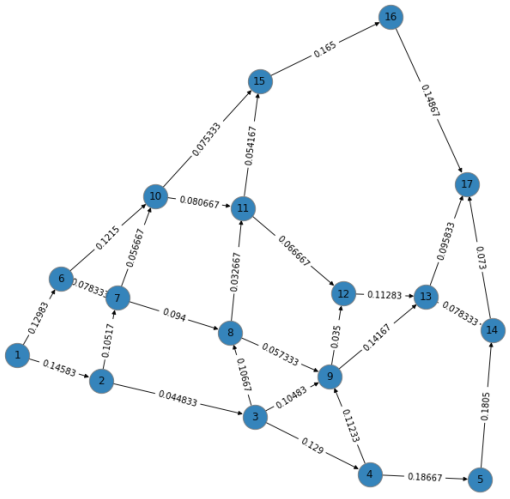


Figure 5: Los Angeles map with traveltime on edges

The shortest path problem in a network can be solved by minimizing the overall cost of the network where the cost on each link e is equal to $\psi_e = l_e f_e$. However, we think that this problem can also be solved by simply using the *shortest_path* function. We got the fastest path $[1, 2, 3, 9, 13, 17]$ with travel time 0.532996.

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

By applying *nx.algorithms.flow.maximum_flow* we got that the maximum flow between node 1 and 17 is equal to 22448.

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

By solving the equation Bf we can easily find the external inflow ν equals to $[16806, 8570, 19448, 4957, -746, 4768, 413, -2, -5671, 1169, -5, -7131, -380, -7412, -7810, -3430, -23544]$

In the following, we assume that the exogenous in 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 following cost function** $\psi_e(f_e)$**:**

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

subject to the flow constraints.

To find the social optimum f^* we need to solve the constrained optimization problem below:

$$\begin{aligned} \min \quad & \psi_e(f_e) \\ \text{s.t.} \quad & f \geq 0 \\ & Bf = \nu \end{aligned} \tag{1}$$

Where $Bf = \nu$ and $f \geq 0$ correspond to the mass conservation at nodes and non-negativity of the flow variables respectively. This problem can be solved with the aid of a Python-embedded modeling language for convex optimization problems called CVXPY.
We find a social optimum flow equals to [6642.3, 6058.9, 3132.4, 3132.4, 10163.7, 4638.4, 3006.36, 2542.59, 3131.52, 583.4, 0., 2926.5 0., ,3132.4, 5525.3, 2854.3, 4886.44, 2215.44, 463.78, 2337.56, 3318.08, 5655.64, 2373.04, 0., 6414.12, 5505.44, 4886.44, 4886.44] whose cost is 25943.62

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

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

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 $f^{(w)}$. What do you observe?

To find the Wardrop equilibrium $f^{(0)}$ we solve again (1) where $\psi_e(f_e)$ is now defined as:

$$\psi_e(f_e) = \sum_{e\in\mathcal{E}}\int_0^{f_e} d_e(s)\,ds = -\sum_{e\in\mathcal{E}} l_e C_e \log\left(1-\frac{f_e}{C_e}\right)$$

We got a Wardrop equilibrium $f^{(0)}$ equal to [6715.65, 6715.65, 2367.41, 2367.41, 10090.35, 4645.39, 2803.84, 2283.56, 3418.48, 0, 176.83, 4171.41, 0, 2367.41, 5444.96, 2353.17, 4933.34, 1841.55, 697.11, 3036.49, 3050.28, 6086.77, 2586.51, 0, 6918.74, 4953.92, 4933.34, 4933.34] whose cost is 26292.96

Now we modify the cost function by introducing the tolls w_e defined as above and so we have:

$$\psi_e(f_e) = \sum_{e\in\mathcal{E}}\int_0^{f_e} (d_e(s)+\omega_e(s))\,ds = \sum_{e\in\mathcal{E}} f_e^*\frac{l_e C_e}{(C_e-f_e^*)^2} f_e - l_e C_e \log\left(1-\frac{f_e}{C_e}\right)$$

The new Wardrop equilibrium $f^{(\omega)}$ that we got is [6642.3, 6059.07, 3132.3, 3132.3, 10163.7, 4638.01, 3006.25, 2542.45, 3131.54, 583.23, 0, 2926.77 0, 3132.3, 5525.68, 2854.25, 4886.43, 2215, 463.8, 2337.68, 3318.06, 5655.73, 2373.17 0, 6414.11, 5505.46, 4886.43, 4886.43] with cost equals to 72000.21. We recap in a table the solutions we got in order to visualize and explain them in a better way:

f^*	$f^{(0)}$	$f^{(w)}$
25943.62	26292.96	72000.21

Table 2: Social optimum, W. eq. without tolls and W. eq.with tolls

As we expect we have a social optimum $f^{(0)}$ whose value is smaller than both other Wardrop equilibriums, indeed the Wardrop equilibrium concides with the path that has the lowest delay with respect any other path in the network. For this reason, one user would like to take this path to experience the lowest possible delay, but if everyone would behave in this selfish way, we would have an higher cost higher the social optimum one. A solution to avoid this phenomenon is represented by the tolls, which penalize the selfish behaviour of users and try to lead the flow from user optimum to social optimum.

(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**

$$\psi_e(f_e) = f_e(d_e(f_e) - l_e)$$

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.

The social optimum we got after minimizing the new objective function $\psi_e(f_e)$ is equal to [6653.26, 5774.66, 3419.75, 3419.74, 10152.74, 4642.7, 3105.85, 2662.18, 3009.06, 878.6, 0.01, 2354.9, 0.01, 3419.74, 5510.04, 3043.69, 4881.8, 2415.46, 443.68, 2008.03, 3487.37, 5495.4, 2203.78, 0.0, 6300.68, 5623.52, 4881.8, 4881.8] whose cost is 15095.51.

To construct the tolls ω_e^* such that $f^(\omega^*)$ coincides with f^* we recall from the theory that given a graph $\mathcal{G}$ whose delay function d_e defined on each edge e is non decreasing, convex and differentiable, such that every cycle in $\mathcal{G}$ contains a link e with $d_e(0) > 0$ and tolls are choosen in such a way they are equal to $\omega_e = f_e^* d_e^{'}(f_e)$ then the Wardrop equilibrium is equal to the system optimum flow f^*. Since in our case all assumptions above are satisfied we just need to construct the tolls as $\omega_e = f_e^* d_e^{'}(f_e)$ and we got [1.95, 0.15, 0.06, 0.12, 1.43, 0.47, 0.12, 0.06, 0.25, 0.01, 0, 0.05, 0, 0.15, 0.48, 0.1, 0.07 0.02, 0, 0.01, 0.07, 0.24. 0.06, 0, 0.38, 0.31, 0.19, 0.53]
It is possible to verify that Wardrop equilibrium obtained with these tolls is equal to the system optimum by looking at the code.