Network Dynamics Homework 1

Alessia Leclercq
Politecnico di Torino
s291871@studenti.polito.it

November 12, 2022

1 Exercise

I used the following functions: minimum_cut and maximum_flow from networkx.algorithms.flow.

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

As a consequence to the *Max-Flow Min-Cut Theorem* an hypothetical adversary should remove a total aggregate capacity equals to the min-cut capacity $c^*_{o,d}$ from the edges crossing the bottlenecks, in order to disconnect d from o. If so, the min-cut capacity between o and d becomes 0 and any flow from o to d is therefore not allowed. The cut capacities in the graph can be found in Table 1 INITIAL. The minimum aggregate capacity to be removed from the edges is 3 (Figure 1).

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

Consequently to *Max-Flow Min-Cut Theorem*, in order to keep the maximum throughput fixed, the bottlenecks capacities should not be reduced. I reduced the capacity of e_1 and e_5, since they are the only edges that do not cross the bottlenecks. In Table 1 the cut capacities resulting from the previous steps are reported. At this point, any other capacity reduction on any link can not be performed without affecting the bottlenecks capacities. In conclusion, the total amount of capacity that can be removed from the network is 2.

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 $x \geq 0$.

Starting from the initial cut capacities (see Table 1 INITIAL), one unit capacity is added at each iteration on an edge common to the bottlenecks (if available), otherwise on a random edge of a random bottleneck.

$\mathcal{U}$	$\mathcal{U}^C$	**INITIAL**	**ITERATION_1**	**ITERATION_2**
o	**abcd**	4	3	3
oa	**bcd**	3	3	3
ob	**acd**	5	4	3
oc	**abd**	5	4	4
oab	**cd**	4	4	3
oac	**bd**	4	4	4
obc	**ad**	4	3	3
oabc	**d**	3	3	3

Table 1: Cut capacities computed in exercise 1.b

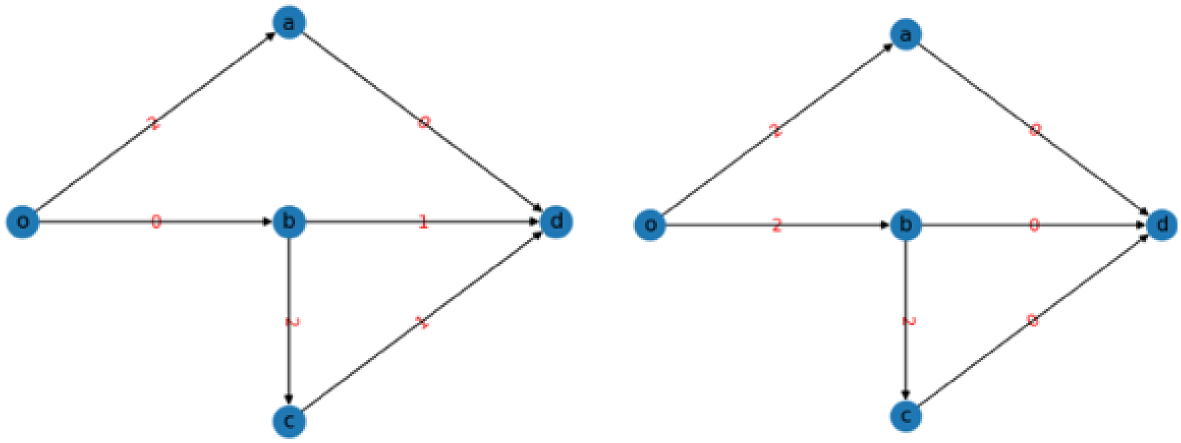

Figure 1: (left) capacity has been removed from e2 and e3; (right) capacity has been removed on e2, e4 and e6. In both cases the maximum allowed throughput is 0.

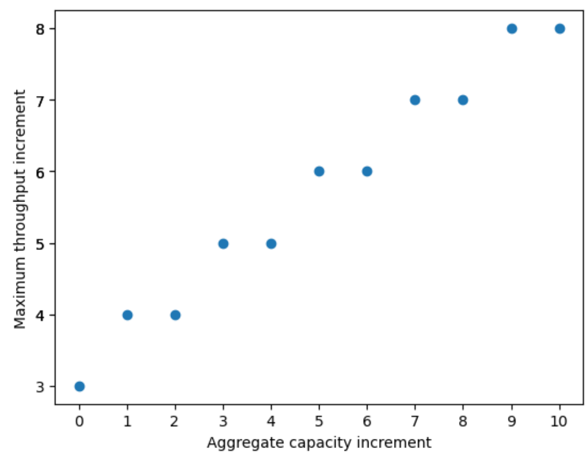

Figure 2: Maximum throughput behavior with respect to the capacity increase x

Then, all cut capacities in the graph are updated and the maximum throughput $\nu^*_{o,d}$ is computed. After 10 iterations, the throughput behaviour with respect to x is the following: at the very first step, increasing the capacity of a single unit increases the throughput of one unit; then, starting from the second iteration, a one unit increase of $\nu^*_{o,d}$ is obtained every time x is increased of two units (Figure 2).

2 Exercise

In Figure 4 (a) the problem graph is shown. To answer all questions, an associated graph $G1$ has been built and it is shown in Figure 3.
I used preflow_push from nx.algorithms.flow.

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

The edges capacities in G1 have been set this way: edges from p_i to b_j with potential infinite capacity, those from o to p_i with capacity 1, and those from b_j to d with capacity 1. Using max-flow problems, a perfect matching then exists i.f.f. a flow of throughput $|\mathcal{V}_0|$ can enter the network, and this is verified since $c^*_{o,d} = 4 = |\mathcal{V}_0|$. I applied the Ford Fulkerson's algorithm to find such a flow and the associated perfect matching. The result can be found in Figure 4 (b)

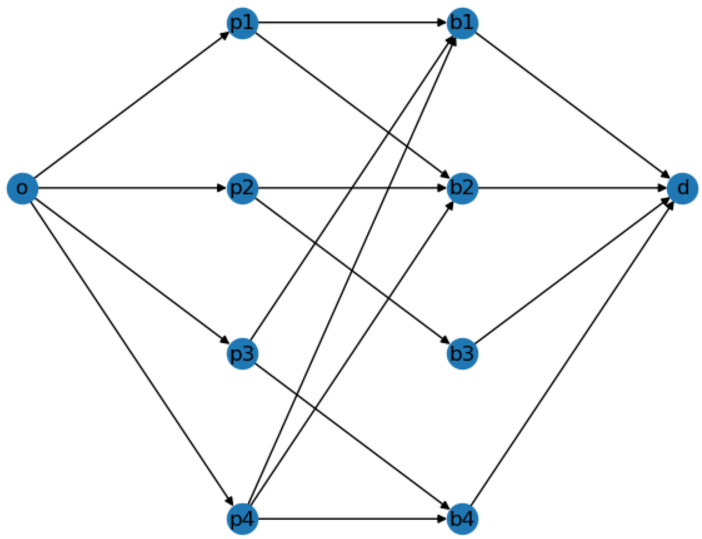

Figure 3: G1 structure associated graph of G in Exercise 2

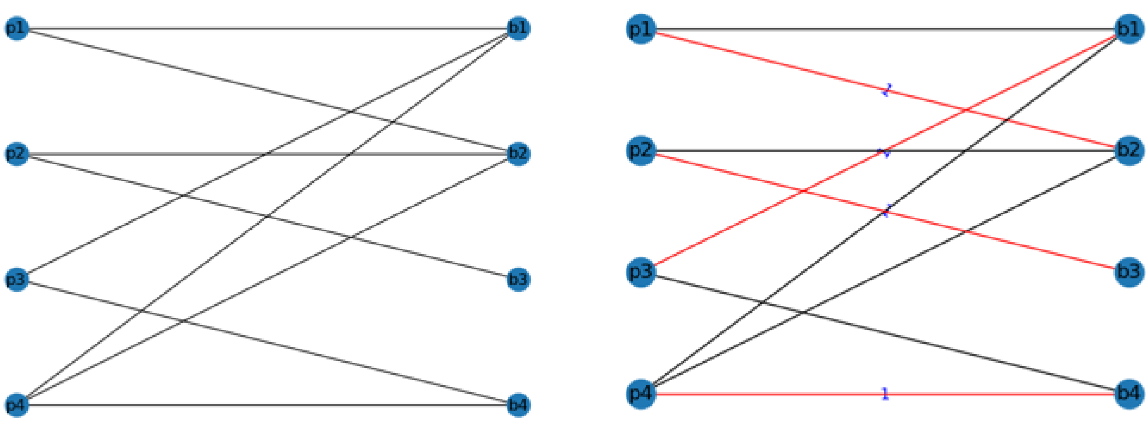

Figure 4: (left) the solution associated with exercise 2; (right) perfect matching result for part1

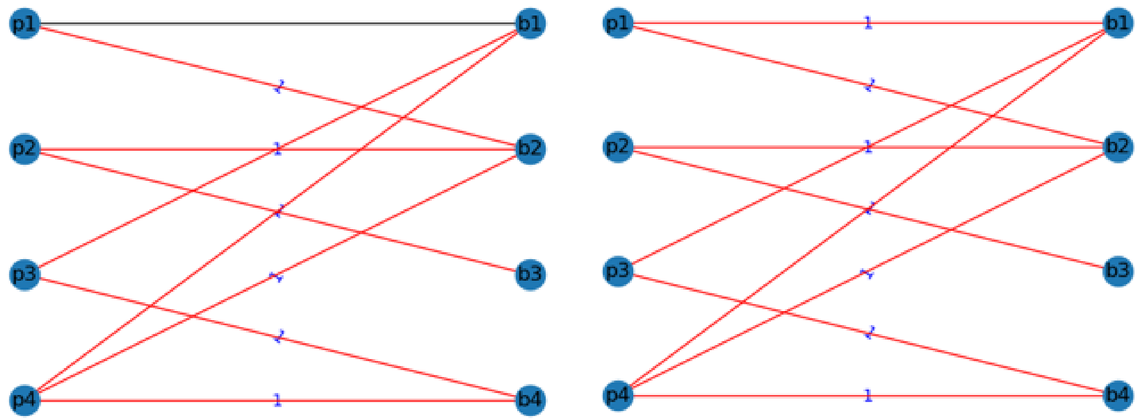

Figure 5: (a) the graph associated with exercise 2; (b) perfect matching result for part1

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.*

The second point is a multiple sources and destinations problem. Now, G1 capacities are set this way: each capacity from o to p_i is the number of books the i-th person can take, in our case infinite; then, the capacities from b_j to d corresponds to the number of copies available for the j-th book; finally, each edge from p_i to b_j identifies the interest of the i-th person in the j-th book and has capacity 1. Now, if a flow from o to d exists, then each person can be assigned the books of interests without violating any constraint. The result is shown in Figure 5(a).

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?

In point (B), there are two available copies for b_3, but only p_4 is interested in it. Conversely, b_1 has two copies and three persons interested (p_1, p_3 and p_4). If the library sells one copy of b_3 to acquire a copy of b_1 all the interests are matched, as well as all copies are sold. The result has been obtained by computing the flow on G1 and it is shown in Figure 5(a).

3 Exercise

I used cvxpy and max_flow from networkx.
All flows computed in the following points can be found in APPENDIX A.

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.

The shortest path problem searches for a flow f^* that minimizes the cost function $\psi_e(f_e) = l_e f_e$. Given an unitary exogenous flow vector ν from $node_1$ to $node_{17}$ to simulate an empty network (hence a single actor is present in the network), the shortest path γ is an o-d path whose sum of the edges lengths is the smallest; consequently, the optimal flow f^* is supported (i.e., has value 1) only on the edges along γ and it equals 0 on all the others. The shortest path problem is formulated as follows:

$$\min_f \sum_{e \in \varepsilon} l_e f_e \\ Bf = \nu \\ 0 \leq f \leq c \tag{1}$$

And it resulted in a path length of 0.53 supported on $\gamma = (e_1, e_2, e_9, e_{12}, e_{25})$.

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

To find the maximum flow between nodes 1 and 17 I computed the min-cut capacity of the graph and applied *Max-Flow Min-Cut Theorem* resulting in $\nu^*_{o,d} = c^*_{o,d} = 22448$.

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

Using a simple matrix multiplication, the resulting exogenous flow is $\nu =$ [16806, 8570, 19448, 4957, -746, 4768, 413, -2, -5671, 1169, -5, -7131, -380, -7412, -7810, -3430, -23544]

D. Assuming 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 $\nu_{17} = -\nu_1$. Find the social optimum f^* with respect to the delays on the different links $\tau_e(f_e)$.

The SO-TAP problem aims at minimizing the total travel time among all flows on the o-d paths from a social point of view, i.e., as it is performed by a single agent having control on the whole network. The problem is formulated as follows:

$$\min_f \sum_{e \in \varepsilon} \tau_e(f_e) f_e \\ Bf = \nu \\ 0 \leq f \leq c \tag{2}$$

Where the delay function is the following: $\tau_e(f_e) = \frac{l_e}{1 - f_e/c_e}$ for $0 \leq f_e < c_e$ and $+\infty$ for $f_e \geq c_e$. The cost resulting from the SO-TAP problem is 25943.62.

E. (a) Find the Wardrop equilibrium $f^{(0)}$

Now, every network user autonomously decides which path to follow, according to which path they consider to have the lowest delay. A Wardrop equilibrium is a network flow vector $f^{(0)}$ that is associated with an o-d path distribution z such that if $z_\gamma > 0$ (i.e., someone chooses path γ), then the total travel time associated with γ, $T_\gamma(z)$, cannot be worse than the total travel time associated with any other o-d path $\tilde{\gamma}$. An o-d flow $f^{(0)}$ is a Wardrop equilibrium if and only if:

$$f^{(0)} \in argmin_f \sum_{e \in \varepsilon} \int_0^{f_e} \tau_e(s) ds \\ 0 \leq f \leq c \\ Bf = \nu \tag{3}$$

Given the delay function $\tau_e(f_e)$, its primitive function is:

$$\int_0^{f_e} \tau_e(s) ds = -l_e c_e \log(1 - \frac{f_e}{c_e}) \tag{4}$$

Once the flow $f^{(0)}$ is found, then, one can compute the price of anarchy, which is the ratio between the UO-TAP and the SO-TAP costs. In our case, the total cost perceived by the user is 26292.96 and the price of anarchy with respect to the social optimum computed in point (D) is 1.013.

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

Consequently to the introduction of tolls, we say that an o-d flow $f^{(\omega)}$ is a Wardrop equilibrium if and only if:

$$\begin{aligned} f^{(\omega)} \in argmin_f \sum_{e \in \varepsilon} \int_0^{f_e} \tau_e(s)ds + \omega_e f_e \\ 0 \leq f \leq c \\ Bf = \nu \end{aligned} \tag{5}$$

The primitive of $\tau_e(f_e)$ is the same as the one identified in (4), whereas the derivative of $\tau_e(f_e)$ is the following:

$$\tau_e'(f_e) = \frac{l_e c_e}{(c_e - f_e)^2} \tag{6}$$

The total cost now is 25943.62, as in the system optimum case, and the resulting price of anarchy is 1.0, which means the the solution of the UO-TAP is the same of the SO-TAP problem.
This was expected because of the choice of the tolls, specifically: let f^* be the solution of the system optimum flow (SO-TAP) and let the tolls be chosen as $w_e^* = f_e^* \tau_e'(f_e^*)$ for $e \in \varepsilon$; then, the Wardrop equilibrium flow $f^{(\omega)}$ coincides with the system optimum flow f^*.

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 $\psi_e(f_e) = f_e(\tau_e(f_e) - l_e)$ subject to flow constraints. Compute the system optimum f^* for the costs above. Construct tolls w^* such that the Wardrop equilibrium $f^{(\omega*)}$ coincides with f^*. Compute the new Wardrop equilibrium to verify your results.

We only changed the cost function with respect to the previous point (E) to:

$$\tilde{\tau}_e(f_e) = \frac{l_e}{1 - \frac{f_e}{c_e}} - l_e \tag{7}$$

The new delay function has the following primitive function:

$$\int_0^{f_e} \tilde{\tau}_e(s)ds = -l_e c_e \log(1 - \frac{f_e}{c_e}) - l_e f_e \tag{8}$$

And the derivative is the one identified in (6).
In order to make the Wardrop equilibrium $f^{(\omega*)}$ coincide with the system optimum f^*, and following the discussion in point (E.b), I chose the following tolls: $w_e^* = f_e^* \tilde{\tau}_e{}'(f_e^*)$ for $e \in \varepsilon$. This resulted in a system optimum cost, as well as a cost associated with the Wardrop equilibium, equals to 15095.51 and a price of anarchy of 1.0.

4 APPENDIX A

Cooperation with s302294.

SO-TAP Flow point D

f^* : [6.64219910e+03, 6.05893789e+03, 3.13232779e+03, 3.13232589e+03, 1.01638009e+04, 4.63831664e+03, 3.00634073e+03, 2.54263460e+03, 3.13154448e+03, 5.83261212e+02, 1.45164550e-02, 2.92659559e+03, 1.89781986e-03, 3.13232589e+03, 5.52548426e+03, 2.85427264e+03, 4.88644874e+03, 2.21523712e+03, 4.63720641e+02, 2.33768761e+03, 3.31799129e+03, 5.65567890e+03, 2.37310712e+03, 1.99567283e-03, 6.41411626e+03, 5.50543301e+03, 4.88645073e+03, 4.88645073e+03]

Wardrop equilibrium point E.a

$f^{(0)}$: [6.71564895e+03, 6.71564803e+03, 2.36740801e+03, 2.36740792e+03, 1.00903510e+04, 4.64539489e+03, 2.80384316e+03, 2.28356194e+03, 3.41848003e+03, 9.22328268e-04, 1.76829408e+02, 4.17141061e+03, 8.92024178e-05, 2.36740792e+03, 5.44495611e+03, 2.35317044e+03, 4.93333832e+03, 1.84155266e+03, 6.97110629e+02, 3.03649261e+03, 3.05028094e+03, 6.08677356e+03, 2.58651143e+03, 1.24029072e-04, 6.91874216e+03, 4.95391934e+03, 4.93333845e+03, 4.93333845e+03]

Wardrop equilibrium with tolls point E.b

$f^{(w)}$: [6.64297472e+03, 6.05907677e+03, 3.13247186e+03, 3.13247182e+03, 1.01630252e+04, 4.63825869e+03, 3.00632629e+03, 2.54233540e+03, 3.13149028e+03, 5.83897949e+02, 1.90383646e-04, 2.92660472e+03, 4.66233160e-05, 3.13247182e+03, 5.52476655e+03, 2.85422624e+03, 4.88637066e+03, 2.21583035e+03, 4.63991080e+02, 2.33744989e+03, 3.31821726e+03, 5.65566715e+03, 2.37303587e+03, 5.97013278e-05, 6.41412156e+03, 5.50550769e+03, 4.88637072e+03, 4.88637072e+03]

SO-TAP flow point F

f^* : [6.65329658e+03, 5.77466230e+03, 3.41971657e+03, 3.41971062e+03, 1.01527034e+04, 4.64278036e+03, 3.10584008e+03, 2.66218478e+03, 3.00907935e+03, 8.78634280e+02, 7.42401749e-03, 2.35493830e+03, 5.94907576e-03, 3.41971062e+03, 5.50992306e+03, 3.04369256e+03, 4.88180506e+03, 2.41557456e+03, 4.43662730e+02, 2.00804968e+03, 3.48735309e+03, 5.49540277e+03, 2.20377848e+03, 2.20338871e-03, 6.30070364e+03, 5.62348910e+03, 4.88180726e+03, 4.88180726e+03]

Wardrop equilibrium point F

$f^{(w*)}$: [6.65329658e+03, 5.77466230e+03, 3.41971657e+03, 3.41971062e+03, 1.01527034e+04, 4.64278036e+03, 3.10584008e+03, 2.66218478e+03, 3.00907935e+03, 8.78634280e+02, 7.42401749e-03, 2.35493830e+03, 5.94907576e-03, 3.41971062e+03, 5.50992306e+03, 3.04369256e+03, 4.88180506e+03, 2.41557456e+03, 4.43662730e+02, 2.00804968e+03, 3.48735309e+03, 5.49540277e+03, 2.20377848e+03, 2.20338871e-03, 6.30070364e+03, 5.62348910e+03, 4.88180726e+03, 4.88180726e+03]