Network Dynamics and Learning
Homework I

Alessandro Casella
Politecnico di Torino
Student ID: s306081
s306081@studenti.polito.it

1. Exercise 1

We consider the network in Figure 1, defined with the following link capacities:

- $c_{ad} = c_{bd} = c_{cd} = 1$
- $c_{oa} = c_{ob} = c_{bc} = 2$

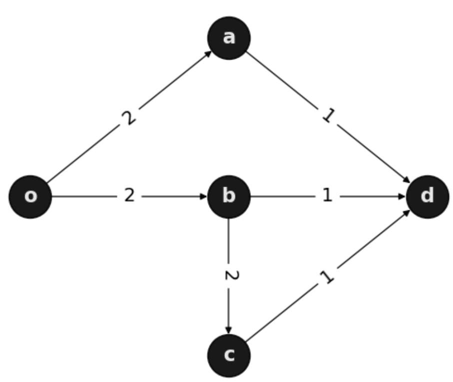

Figure 1: Network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with link capacities

1.1. Exercise 1.a

To find the minimum aggregate capacity that needs to be removed for no feasible flow from o to d to exist, we can exploit the network resilience interpretation of the max-flow min-cut theorem, which asserts that the minimum total capacity to be removed from the network to make d not reachable from o coincides with the min-cut capacity $c^*_{o,d}$. All the possible cuts are listed below:

- $\mathcal{U} = \{o\}$, $\mathcal{U}^c = \{a, b, c, d\}$ with cut capacity $= 4$
- $\mathcal{U} = \{o, a\}$, $\mathcal{U}^c = \{b, c, d\}$ with cut capacity $= 3$
- $\mathcal{U} = \{o, b\}$, $\mathcal{U}^c = \{a, c, d\}$ with cut capacity $= 5$
- $\mathcal{U} = \{o, a, b\}$, $\mathcal{U}^c = \{c, d\}$ with cut capacity $= 4$
- $\mathcal{U} = \{o, b, c\}$, $\mathcal{U}^c = \{a, d\}$ with cut capacity $= 4$
- $\mathcal{U} = \{o, a, b, c\}$, $\mathcal{U}^c = \{d\}$ with cut capacity $= 3$

Therefore, $c^*_{o,d} = 3$ is the min-cut capacity, which represents the minimum aggregate capacity to disconnect the origin from the destination of the network.

1.2. Exercise 1.b

We aim to maximize the removal of aggregate capacity from the links of the network without affecting the maximum throughput from o to d. Fixing the maximum throughput means that the min-cut capacity must not change. As shown in the previous exercise, there are two cuts in which the capacity is minimum (the second and the last cases of the list), so we cannot lower the capacities in the links that connect such cuts. In particular, the constraints are:

- $c_{ob} = 2$
- $c_{ad} = c_{bd} = c_{cd} = 1$

Thus, we determine the maximum quantity of capacity which can be removed in the remaining links, which are oa and bc. To do this, we proceed by considering all the remaining cuts where the cut capacity is > 3 to make it equal to 3. We see that:

- c_{oa} can be lowered of 1 unit
- c_{bc} can be lowered of 1 unit

without affecting the maximum throughput. Therefore, the maximum capacity we are looking for equals 2.

1.3. Exercise 1.c

We have to decide how to assign $x > 0$ extra units of capacity to the six links of the network, to maximize the throughput. For each x, we proceed by considering all the possible combinations of links where to assign x units of capacity and we compute all the cuts capacities to find the maximum one, corresponding to the maximum throughput in the network. By repeating this process as we increase $x \to \infty$, we observe that the maximum throughput's growth follows a constant trend (Figure 2).

2. Exercise 2

A matching problem consists of a set of people $\mathcal{P} = \{p_1, p_2, p_3, p_4\}$ and a set of books $\mathcal{B} = \{b_1, b_2, b_3, b_4\}$. Each person is interested in a subset of books as follows:

- $p_1 \to \{b_1, b_2\}$
- $p_2 \to \{b_2, b_3\}$

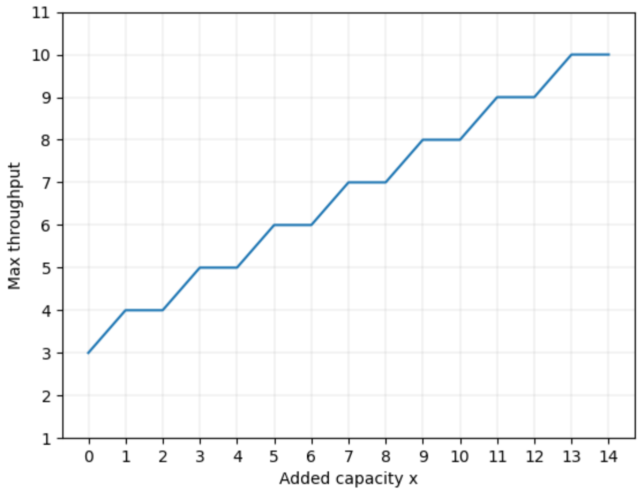

Figure 2: Max throughput from o to d as function of x

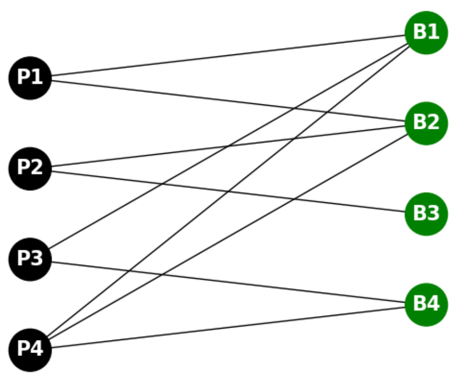

Figure 3: Graph $\mathcal{G} = (\mathcal{P} \cup \mathcal{B},\ \mathcal{E})$

- $p_3 \rightarrow \{b_1, b_4\}$
- $p_4 \rightarrow \{b_1, b_2, b_4\}$

These associations can be represented through a simple bipartite graph (Figure 3): the set of vertices is given by the union of people and books $\mathcal{V} = \mathcal{P} \cup \mathcal{B}$, while the edges $e = (p_i, b_j) \in \mathcal{E}$ represent the interest of person i in book j.

2.1. Exercise 2.a

Thanks to Hall's theorem, we prove the existence of perfect matching between people and books, since

$$|\mathcal{U}| \leq |\mathcal{N}_{\mathcal{U}}| \ ,\ \forall U \subseteq \mathcal{V}_0 \tag{1}$$

where $V_0 \in \mathcal{P} \cup \mathcal{B}$ and $\mathcal{N}_{\mathcal{U}} = \cup_{i \in \mathcal{U}} \mathcal{N}_i$ is the neighborhood of $\mathcal{U}$ in $\mathcal{G}$.
To identify the perfect matching people - books, we rely on the max flow problem. To apply it, we need to remodel the graph as here explained:

- we make it directed
- we create an origin and a destination node which are respectively connected to all the people and all the books
- we assign a unit of capacity to all the links

At this point, we can use the max flow algorithm which gives the maximum throughput τ that enters the network from o and exits in d. Since we expect a complete matching of cardinality equal to the number of people and books, then $\tau = 4$. Our expectations are met in the following perfect matching: $(p_1 \rightarrow b_2), (p_2 \rightarrow b_3), (p_3 \rightarrow b_4), (p_4 \rightarrow b_1)$.

2.2. Exercise 2.b

We introduce a different number of copies for each book: $copies = [2, 3, 2, 2]$. Our goal is to find the maximum number of books that can be assigned to the people.
To model multiple copies of books, we put the capacity of edge (b_j, d) equal to the number of copies of b_j. Since people can choose an arbitrary number of books of interest, we put the capacity of edge $(o, p_i) = +\infty$. We also suppose that each person can take at most one copy of the same book, so $c_{(p_i, b_j)} \leq 1$. Applying all these transformations to the graph, we obtain the network represented in Figure 4.

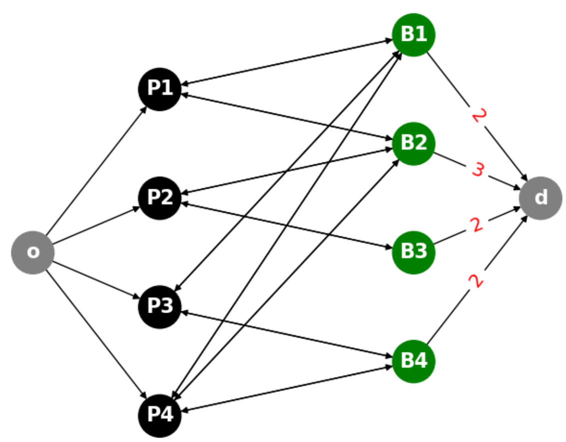

Figure 4: Graph $\widetilde{\mathcal{G}} = (\mathcal{P} \cup \mathcal{B} \cup o \cup d,\ \widetilde{\mathcal{E}})$

In terms of flow, the exogenous flow vector from o to p_i is potentially infinite, the one from p_i to b_j is at most 1 and lastly the one from b_j to d is at most equal to $copies_j$. We exploit the max flow problem to find the maximum throughput, which coincides with the total number of copies assigned to the people. By running the algorithm, we find $\tau = 8$, which means eight total copies can be assigned through the following matching, as shown in Figure 5:

- $(p_1 \rightarrow b_2)$
- $(p_2 \rightarrow b_2), (p_2 \rightarrow b_3)$
- $(p_3 \rightarrow b_1), (p_3 \rightarrow b_4)$
- $(p_4 \rightarrow b_1), (p_4 \rightarrow b_2), (p_4 \rightarrow b_4)$

2.3. Exercise 2.c

With the possibility of selling a copy of a book and buying a copy of another one, we can maximize the number of assigned books. In Figure 5, we see that all the people's requests for books are satisfied, except for p_1, who cannot receive b_1 because there are no more copies available. We also notice that for b_3 only one copy out of the available two is assigned because among all the people only p_2 is interested in this book.
Taking into account these observations, we have on one hand a surplus of b_3 copies and on the other hand an insufficient number of b_1 copies. Then, if we sell one copy of b_3 and buy one copy of b_1, the number of assigned books will be maximized. In fact, by running the max-flow algorithm with this new configuration, we obtain $\tau = 9$: all the people are satisfied with their books of interest and every copy is associated.

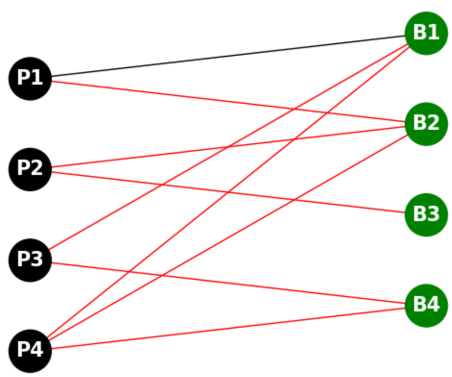

Figure 5: Matching (p_i, b_j) with max-flow problem

3. Exercise 3

A simplified network of a part of highway real-map in Los Angeles is represented in Figure 6. We are provided with the node-link incidence matrix $\mathcal{B} \in \{-1, 0, +1\}^{\mathcal{V}\times\mathcal{E}}$, which individuates the tail $(+1)$ and head (-1) nodes in the rows for each link $e_i = \{e_1, \dots, e_{28}\}$ in the columns. More specifically:

$$B_{i,j} = \begin{cases} +1 & \text{if } \theta(e_j) = i \\ -1 & \text{if } \kappa(e_j) = i \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Moreover, we are also provided with the vector c_e which contains the maximum capacities in each link e and with another vector l_e which gives the minimum travelling time for each link e. Eventually, the delay function is defined for each link e as follows:

$$\tau_e(f_e) = \begin{cases} \frac{l_e}{1 - f_e/c_e} & \text{if } 0 \leq f_e < c_e \\ \infty & \text{if } f_e \geq c_e \end{cases} \tag{3}$$

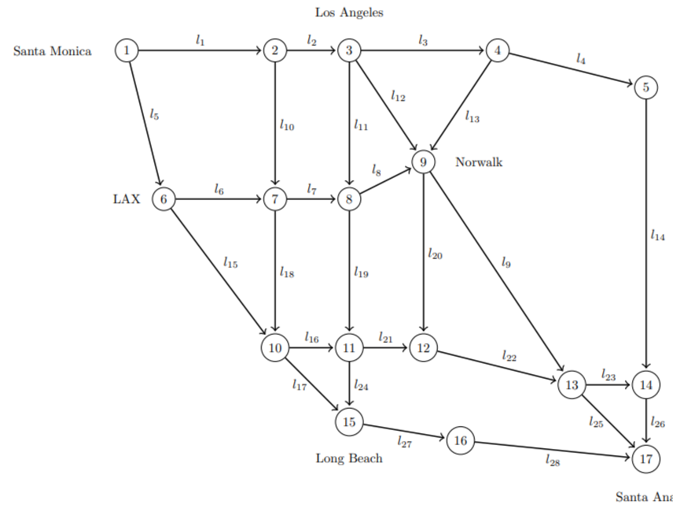

Figure 6: A simplified network of the Los Angeles highways

3.1. Exercise 3.a

Firstly, we want to determine the shortest path between node 1 and node 17, which is equivalent to the path with the shortest travelling time when the network is empty. To make it empty, we assume there is only one car entering from the first node (1, which is a source) and exiting in the last node (17, which is a sink). In the network of Figure 6, the inflow in node 1 is $\nu_1^+ = +1$, the outflow in node 17 is $\nu_{17}^- = +1$ and all the in-/out-flows in the middle nodes compensate each other. Hence, we can express the exogenous net flow ν as a vector of 17 components, equal to the total number of nodes in the network, where the first component equals +1, the last one is -1 and all the ones in the middle equal 0: $\nu = [+1, 0, \dots, 0, -1]$.
The problem of finding the shortest path can be modelled as a network flow optimization problem in the form:

$$\min_{\substack{f \geq 0 \\ Bf = \nu}} \sum_{e \in \mathcal{E}} \psi_e(f_e) \tag{4}$$

where f_e is the network flow on link e, $\psi_e(f_e) = l_e f_e$ are the link cost functions and $\nu = \delta^{(o)} - \delta^{(d)}$ is the exogenous flow vector with all 0s except in the nodes 1 and 17. Through this optimization, we obtain the path with the shortest travelling time from o to d: $[(1,2),(2,3),(3,9),(9,13),(13,17)]$.

3.2. Exercise 3.b

The maximum flow between the origin 1 and the destination 17 is given by the max-flow min-cut theorem: $\tau_{(1,17)} = 22448$. Alternatively, we could also proceed by solving the max flow problem to maximize the throughput, since it coincides with both the maximum flow and the min-cut capacity.

3.3. Exercise 3.c

The external inflow is given by $\nu = \mathcal{B}f$. Assuming that $v[17] = -v[1]$ and $v[i] = 0 \; if \; 1 < i < 17$, we obtain the following light version of the external inflow: $\nu = [16806, 0, \ldots, 0, -16806]$.

3.4. Exercise 3.d

We now compute the social optimum f^* by solving the System Optimum Traffic Assignment Problem (SO-TAP), with the following optimization:

$$\min_{\substack{0 \leq f < c \\ Bf = \nu}} \sum_{e \in \mathcal{E}} f_e \tau_e(f_e) \tag{5}$$

where

$$f_e \tau_e(f_e) = \psi_e(f_e) = \frac{l_e c_e}{1 - f_e/c_e} - l_e c_e \tag{6}$$

is the cost function on each link e.
Note that in the constraints we put that $f < c$ because otherwise $\tau_e(f_e = c_e) = \infty$ could lead us to a cost function whose value goes to infinity.
Our task consists in minimizing the total cost of the network, which is the summation of $\psi_e(f_e)$ over all the links. The obtained social optimum flow vector is reported in the second column of Table 1. Observe that we round the components of the flow vectors to the nearest integer since we are considering the flow in terms of the number of drivers/cars.

3.5. Exercise 3.e

We get the Wardrop equilibrium $f^{(0)}$ by solving the User Optimum Traffic Assignment Problem (UO-TAP):

$$\min_{\substack{f \geq 0 \\ Bf = \nu}} \sum_{e \in \mathcal{E}} \int_0^{f_e} \tau_e(s) \; ds \tag{7}$$

where $\tau_e(f_e) = \frac{l_e}{1 - f_e/c_e}$. By solving the integral, we obtain:

$$\int_0^{f_e} \tau_e(s) \; ds = l_e c_e \log \frac{c_e}{c_e - f_e} \tag{8}$$

As in the previous exercise, we proceed in minimizing the function obtained by the summation of this integral over all the links, so we obtain the flow vector stored in the third column of Table 1.
From a concrete point of view, this situation in which the Wardrop equilibrium holds is more realistic, since each driver chooses its fastest path without taking care of the other drivers. Quite the opposite, the social optimum of Exercise 3.d ideally represents the best traffic assignment for the overall network, which aims to minimize the total cost.

3.6. Exercise 3.e

We now want to find a new Wardrop equilibrium $f^{(w)}$ with the introduction of tolls ω_e over each link e, which represent the costs the drivers have to pay to travel on the highways. Each toll is defined as follows:

$$\omega_e = f_e^* \; \tau_e'(f_e^*) \tag{9}$$

where f_e^* is the flow on link e at system optimum.
We express the delay on each link by adding the toll value:

$$\widetilde{\tau}_e(f_e) = \tau_e(f_e) + \omega_e \tag{10}$$

Once having computed its derivative, we rewrite the equation (9):

$$\omega_e = f_e^* \frac{l_e c_e}{(c_e - f_e^*)^2} \tag{11}$$

through which we can compute all the tolls.
We also reformulate (8) to adapt it to the presence of tolls:

$$\int_0^{f_e} (\tau_e(s) + \omega_e) \; ds = l_e c_e \log \frac{c_e}{c_e - f_e} + \omega_e f_e \tag{12}$$

By solving the UO-TAP, we obtain the flow corresponding to Wardrop equilibrium with tolls in the fourth column of Table 1.
What is important to notice is that the flow in Wardrop equilibrium is analogous to the social optimum flow. This behaviour lets us conclude that the tolls override the consequences of drivers' individual choices (which are done to the exclusive advantage of personal interest) since they demonstrate to be completely successful at restoring the social optimum.

3.7. Exercise 3.f

Let the cost on each link $\psi_e(f_e)$ specified in (6) be compared to the total delay on free flow (3). The new cost function on each link is:

$$\psi_e(f_e) = f_e(\tau_e(f_e) - l_e) = \frac{l_e c_e}{1 - f_e/c_e} - l_e c_e - l_e f_e \tag{13}$$

The aim of this last point is to construct tolls ω^* such that the Wardrop equilibrium coincides with the social optimum:

$$f^{(w^*)} = f^* \tag{14}$$

Once having stated the new delay on each link:

$$\widetilde{\tau}_e(f_e) = \tau_e(f_e) - l_e = \frac{l_e}{1 - f_e/c_e} - l_e \tag{15}$$

we can proceed in solving the two optimization problems. For SO-TAP, it is sufficient to apply (13) to (5), in order to obtain the social optimum flow f^*.
Instead, the integral over all the links:

$$\int_0^{f_e} (\widetilde{\tau}_e(s) + \omega_e) \; ds = l_e c_e \, log(c_e - f_e) - l_e f_e + \omega_e f_e \tag{16}$$

allows to solve the UO-TAP with tolls, by applying it in (7). The values of the two rounded flow vectors are in Table 2.

TABLE 1: Rounded flow vectors (3.a, 3.d, 3.e)

Edge	Social optimum flow	Wardrop equilibrium	Wardrop equilibrium with tolls
1	6642	6716	6642
2	6059	6716	6059
3	3132	2367	3132
4	3132	2367	3132
5	10164	10090	10164
6	4638	4645	4638
7	3006	2804	3006
8	2543	2284	2542
9	3132	3418	3132
10	583	0	583
11	0	177	0
12	2926	4171	2927
13	0	0	0
14	3132	2367	3132
15	5525	5445	5526
16	2854	2353	2854
17	4886	4933	4886
18	2215	1842	2215
19	464	697	464
20	2338	3036	2338
21	3318	3050	3318
22	5656	6087	5656
23	2373	2587	2373
24	0	0	0
25	6414	6919	6414
26	5505	4954	5505
27	4886	4933	4886
28	4886	4933	4886

TABLE 2: Rounded flow vectors with edited delay (3.f)

Edge	Social optimum flow	Wardrop equilibrium with tolls
1	6653	6653
2	5775	5775
3	3420	3419
4	3420	3419
5	10153	10153
6	4643	4643
7	3106	3105
8	2662	2662
9	3009	3009
10	879	878
11	0	0
12	2355	2356
13	0	0
14	3420	3419
15	5510	5510
16	3044	3043
17	4882	4882
18	2415	2415
19	444	444
20	2008	2009
21	3487	3487
22	5495	5496
23	2204	2204
24	0	0
25	6301	6301
26	5624	5623
27	4882	4882
28	4882	4882