Network Dynamics: Homework I

Andrea Silvi
Politecnico di Torino
andrea.silvi@studenti.polito.it

1. Exercise 1

In this first exercise we consider unitary *o-d* network flows on the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in Figure 1.

1.1. Exercise 1.a

1.a

We assume that each link l has integer capacity C_l. In order to find the infimum of the total capacity that needs to be removed for no feasible *o-d* unitary flows to exist, we can use an interpretation of the max-flow min-cut theorem, in terms of networks resilience: the min-cut capacity $C^*_{o,d}$ coincides with the total amount of capacity that needs to be removed in order to disconnect o and d, essentially making it impossible for a unitary flow to exist on a path between these two nodes. We can identify 4 cuts in the graph of Figure 1:

- $\mathcal{U}_1 = \{o\}$, with cut capacity $C_{\mathcal{U}_1} = C_1 + C_2$;
- $\mathcal{U}_2 = \{o, a\}$, with cut capacity $C_{\mathcal{U}_2} = C_2 + C_3 + C_4$;
- $\mathcal{U}_3 = \{o, b\}$, with cut capacity $C_{\mathcal{U}_3} = C_1 + C_5$;
- $\mathcal{U}_4 = \{o, a, b\}$, with cut capacity $C_{\mathcal{U}_4} = C_4 + C_5$.

Then, $C^*_{o,d} = \min\{C_{\mathcal{U}_1}, C_{\mathcal{U}_2}, C_{\mathcal{U}_3}, C_{\mathcal{U}_4}\}$ is the infimum of the total capacity that if removed will disconnect the origin and the destination of $\mathcal{G}$.

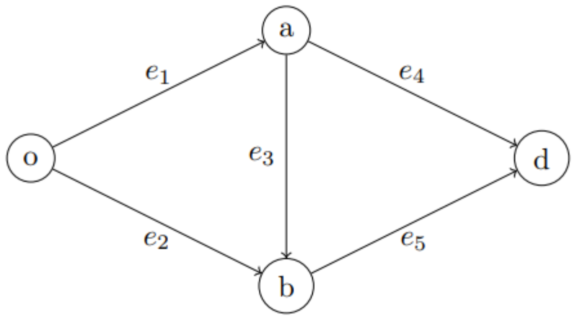

Figure 1: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

1.2. Exercise 1.b

We now assume for this point and the subsequent ones that $C_1 = C_4 = 3$, while $C_2 = C_3 = C_5 = 2$. If we compute the cut capacities using these values, we obtain that $C_{\mathcal{U}_1} = C_{\mathcal{U}_3} = C_{\mathcal{U}_4} = 5$, while $C_{\mathcal{U}_2} = 7$. For the max-flow min-cut theorem, this results in a maximum throughput of 5. If we then try to add an additional unit of capacity to one link in order to increase the maximum throughput, we can easily see from the cut capacities equations that no link is shared between $C_{\mathcal{U}_1}$, $C_{\mathcal{U}_3}$ and $C_{\mathcal{U}_4}$. This means that there is no way with just 1 unit of capacity to increase all three cut capacities to 6, so this does not change the maximum throughput, which is still $\tau = 5$.

1.3. Exercise 1.c

If we try to increase the throughput by adding 2 units of capacity to any 2 links, we can easily see from the cut capacities equations that we can reach a minimum cut capacity of 6 in three ways:

- by adding 1 capacity to links e_1 and e_5;
- by adding 1 capacity to links e_1 and e_4;
- by adding 1 capacity to links e_2 and e_5.

In these three cases we obtain respectively

- $C_{\mathcal{U}_1} = 6$, $C_{\mathcal{U}_2} = 7$, $C_{\mathcal{U}_3} = 7$ and $C_{\mathcal{U}_4} = 6$,
- $C_{\mathcal{U}_1} = 6$, $C_{\mathcal{U}_2} = 8$, $C_{\mathcal{U}_3} = 6$ and $C_{\mathcal{U}_4} = 6$,
- $C_{\mathcal{U}_1} = 6$, $C_{\mathcal{U}_2} = 8$, $C_{\mathcal{U}_3} = 6$ and $C_{\mathcal{U}_4} = 6$,

In all these three cases we finally obtain an optimal throughput $\tau = \min\{C_{\mathcal{U}_1}, C_{\mathcal{U}_2}, C_{\mathcal{U}_3}, C_{\mathcal{U}_4}\} = 6$.

1.4. Exercise 1.d

1c

Having now 4 units of capacity at our disposal that we can add to the network, we can divide the problem into two parts:

- adding the first 2 units in order to increase the throughput from 5 to 6, as done in Exercise 1.c (1.3);

- adding the last 2 units in order to increase the throughput from 6 to 7.

Then, if we consider that we have 3 different solutions to increase τ by 1, we have that all the combinations with repetitions of two of these 3 solutions add 2 to τ, thus obtaining $\frac{(3+2-1)!}{(3-1)!2!} = 6$ possible ways of reaching $\tau = 7$. These are:

- adding 2 units to e_1 and 2 to e_5, with $C_{\mathcal{U}_1} = 7$, $C_{\mathcal{U}_2} = 7$, $C_{\mathcal{U}_3} = 9$ and $C_{\mathcal{U}_4} = 7$;

- adding 2 units to e_1 and 2 to e_4, with $C_{\mathcal{U}_1} = 7$, $C_{\mathcal{U}_2} = 9$, $C_{\mathcal{U}_3} = 7$ and $C_{\mathcal{U}_4} = 7$;

- adding 2 units to e_2 and 2 to e_5, with $C_{\mathcal{U}_1} = 7$, $C_{\mathcal{U}_2} = 9$, $C_{\mathcal{U}_3} = 7$ and $C_{\mathcal{U}_4} = 7$;

- adding 2 units to e_1, 1 to e_4 and 1 to e_5, with $C_{\mathcal{U}_1} = 7$, $C_{\mathcal{U}_2} = 8$, $C_{\mathcal{U}_3} = 8$ and $C_{\mathcal{U}_4} = 7$;

- adding 1 unit to e_1, e_2, e_4 and e_5, with $C_{\mathcal{U}_1} = 7$, $C_{\mathcal{U}_2} = 9$, $C_{\mathcal{U}_3} = 7$ and $C_{\mathcal{U}_4} = 7$;

- adding 1 unit to e_1 and e_2 and 2 units to e_5, with $C_{\mathcal{U}_1} = 7$, $C_{\mathcal{U}_2} = 8$, $C_{\mathcal{U}_3} = 8$ and $C_{\mathcal{U}_4} = 7$.

There is no combination of assignments that can reach a throughput greater than 7 and additionally, it is interesting to note that no matter the different optimal allocations, all of them result in the sum of the cut capacities being $C_{\mathcal{U}_1} + C_{\mathcal{U}_2} + C_{\mathcal{U}_3} + C_{\mathcal{U}_4} = 30$.

2. Exercise 2

In this exercise we consider a matching problem where we have 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 has different interests in terms of the available books: p_1 is interested in b_1 and b_2, p_2 in b_2 and b_3, p_3 in b_1 and b_4, and finally p_4 in b_1, b_2 and b_4.

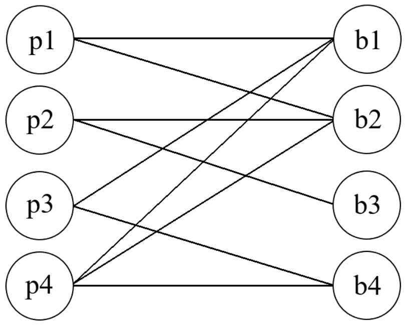

Figure 2: Graph $\mathcal{G} = (\mathcal{P} \cup \mathcal{B}, \mathcal{E})$.

2.1. Exercise 2.a

These relationships can be modelled using a simple bipartite graph, as in Figure 2. The set of vertices $\mathcal{V}$ is composed of $\mathcal{P} \cup \mathcal{B}$ and an edge $e = (p_i, b_j) \in \mathcal{E}$ represents the interest of person i in book j.

2.2. Exercise 2.b

2a

We now want to establish whether there exists a perfect matching that assigns to a person a book of interest. We can exploit an analogy of this problem to a max-flow problem, but firstly we need to remodel the graph:

- we make the graph directed, obtaining a graph $\tilde{\mathcal{G}} = (\mathcal{V}, \tilde{\mathcal{E}})$ where $(p_i, b_j) \in \tilde{\mathcal{E}}$ if $(p_i, b_j) \in \mathcal{E}$;

- we then add an origin o and a destination d to $\mathcal{V}$ and we add links $(o, p_i), \forall p_i \in \mathcal{P}$, and links $(b_i, d), \forall b_i \in \mathcal{B}$;

- we finally add capacities of 1 to every link $e \in \tilde{\mathcal{E}}$.

We can now use the max-flow algorithm in order to calculate the maximum throughput τ that can enter the network in o and exit in d: if τ is equal to $|\mathcal{P}| = 4$, then there exists a complete matching from $\mathcal{P}$ to $\mathcal{B}$. Additionally, since $|\mathcal{B}| = |\mathcal{P}| = 4$, this is also a perfect matching.

The algorithm results a maximum throughput $\tau = 4$, along with the following perfect matching: $(p_1 \to b_2)$, $(p_2 \to b_3)$, $(p_3 \to b_4)$, and $(p_4 \to b_1)$.

2.3. Exercise 2.c

2b

We now introduce a number of copies for each book: $copies = f(\mathcal{B}) = [2, 3, 2, 2]^T$. We can still use graph $\tilde{\mathcal{G}}$ to model this matching problem, but we need to change some of the links capacities:

- $\forall p_i \in \mathcal{P}$, the capacity of link (o, p_i) is now ∞, thus representing the fact that each person can get the maximum amount of books they can;

- $\forall b_i \in \mathcal{B}$, the capacity of link (b_i, d) is now $copies[b_i]$, thus representing the fact that each book can be taken at most a number of times equal to the number of its copies.

Now, when running the max-flow algorithm, in order to satisfy the flow balance constraint $Bf = \nu$, where B is the node-link incidence matrix of $\tilde{\mathcal{G}}$ and $\nu = \tau(\delta^{(o)} - \delta^{(d)})$ is the exogenous flow vector, from each person node a potentially infinite flow can exit, from a person to a book of interest only a flow of at most 1 can transition (since the capacities from $\mathcal{P}$ to $\mathcal{B}$ are kept to 1), and finally from each book node only a flow at most equal to its number of copies can go through and end up into d. The max-flow algorithm returns a throughput of $\tau = 8$, meaning that 8 total copies are taken, and the following matching:

- $(p_1 \to b_2)$;
- $(p_2 \to b_2)$, $(p_2 \to b_3)$;
- $(p_3 \to b_1)$, $(p_3 \to b_4)$;
- $(p_4 \to b_1)$, $(p_4 \to b_2)$, $(p_4 \to b_4)$.

2.4. Exercise 2.d 2c

We assume now that the library can sell 1 copy of a certain book and buy 1 copy of another one, in order to increase the total number of books that get assigned to people $\mathcal{P}$. If we compare the books each person is interested in with their number of copies and the matching we get in Exercise 2.c (2.3), we can easily see two problems:

- p_1 has only b_2 assigned to, but he is interested also in b_1;
- the library possesses 2 copies of b_3, but only p_1 is interested in it.

We can then conclude that the only way to increase the total number of books assigned is to sell a copy of book b_3 and buy another copy of book b_2, thus assigning all the copies of all books. By running the max-flow algorithm with this new amount of copies, we obtain a throughput of 9 and each person has 1 copy of every book they are interested to assigned to them.

3. Exercise 3

In this exercise we consider a simplified network of the highway map of Los Angeles, as represented in Figure 3. We are provided with the node-link incidence matrix $\mathcal{B} \in \{-1, 0, +1\}^{\mathcal{V}\times\mathcal{E}}$, which represents for each link which two nodes share it (where nodes represent intersections between highways). More specifically,

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

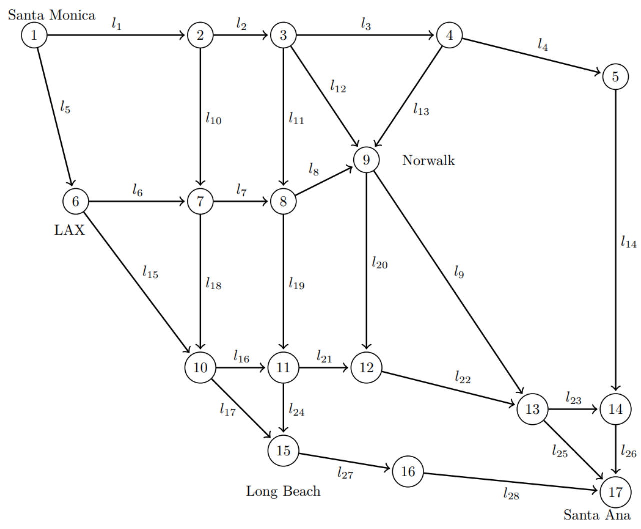

Figure 3: A simplified network of the Los Angeles highways network.

Each link e has also a maximum capacity C_e, and we are provided with the vector $C \in \mathbb{R}^{\mathcal{E}}$ that contains these values. Moreover, we are provided with a vector $l \in \mathbb{R}^{\mathcal{E}}$ which for each link e gives us the minimum travelling time l_e, calculated by dividing the length of the highway segment e with the assumed speed limit of 60 miles per hour.

We finally introduce for each link e the delay function

$$d_e(f_e) = \frac{l_e}{1 - f_e/C_e}, \quad 0 \le f_e < C_e, \tag{2}$$

while for $f_e \ge C_e$, $d_e(f_e) = \infty$.

3.1. Exercise 3.a

We are firstly interested in which is the shortest path between node 1 (Santa Monica) and node 17 (Santa Ana). This is equivalent to the path from 1 to 17 with shortest travelling time and can be found by modelling this problem as a network flow optimization problem of the form

$$\min_{\substack{0 \le f < C \\ Bf = \nu}} \sum_{e\in\mathcal{E}} \psi_e(f_e), \tag{3}$$

where $f \in \mathbb{R}^{\mathcal{E}}$ is a vector variable, the cost function assigned to each link e is $\psi_e(f_e) = l_e f_e$, while $\nu = (\delta^{(1)} - \delta^{(17)}) \in \mathbb{R}^{\mathcal{V}}$ is the exogenous flow vector with all 0's as values except in the origin node 1 and the destination node 17, where it is equal to 1. Basically we have 1 single car entering from node 1 and exiting in node 17 and by optimizing problem (3) we obtain which path that connects the origin and the destination node has the shortest travelling time.

The result of this problem is reported in the first column of Table 1. The shortest path that results from this is $(1) \to (2) \to (3) \to (9) \to (13) \to (17)$.

3.2. Exercise 3.b

We now want the maximum flow possible between node 1 and node 17. This time we can define a throughput optimization problem as

$$\begin{aligned} \max \quad & \tau \\ \text{s.t.} \quad & \tau \ge 0, \\ & 0 \le f < C, \\ & Bf = \tau(\delta^{(1)} - \delta^{(17)}) \end{aligned}$$

where $\tau \in \mathbb{R}_+$ and $f \in \mathbb{R}^{\mathcal{E}}$ are vector variables and the other conditions are similar of those of equation (3). As a result of this problem we obtain a maximum throughput $\tau^*_{(1,17)} = 22448$. By running the max-flow algorithm over the network, we obtain the same result.

3.3. Exercise 3.c

We now want to obtain the vector of exogenous flow ν through the use of the node-link incidence matrix $\mathcal{B}$ and a flow vector f which is given to us that represents a realistic situation of the highways network of Los Angeles. We obtain $\nu = \mathcal{B}f$ that we then simplify by keeping only $\nu[1] = 16806$ and $\nu[17] = -\nu[1] = -16806$, while we put to 0 all the other values of ν. This is done in order to obtain a single origin single destination problem, which is easier to solve.

3.4. Exercise 3.d

Through the use of this simplified vector ν we then want to find the social optimum f^* with respect to the delays on the different links $d_e(f_e)$. We do so by solving the optimization problem

$$\min_{\substack{0 \le f < C \\ Bf = \nu}} \sum_{e \in \mathcal{E}} f_e d_e(f_e), \tag{4}$$

where $d_e(f_e)$ is given by (2). This is also called the System-Optimum Traffic Assignment problem (SO-TAP). The social optimum is the optimal flow assignment that a central authority would do if they were able to route each of the drivers entering the network. This leads to the f^* reported in the second column of Table 1 and to a social optimum total cost

$$\sum_{e \in \mathcal{E}} f^*_e d_e(f^*_e) \approx 25943.61566. \tag{5}$$

3.5. Exercise 3.e

We then introduce the Wardrop equilibrium $f^{(0)}$, which is the optimal flow we obtain when solving the optimization problem

$$\min_{\substack{0 \le f < C \\ Bf = \nu}} \sum_{e \in \mathcal{E}} \int_0^{f_e} d_e(s) ds, \tag{6}$$

where

$$\int_0^{f_e} d_e(s) ds = C_e l_e \ln \frac{C_e}{C_e - f_e}, \tag{7}$$

where we remove the absolute values inside the logarithm since both C_e and $C_e - f_e$ are ≥ 0, $\forall e \in \mathcal{E}$. This situation actually represents the more realistic scenario where each driver selfishly selects their fastest route without considering what the best for also the others is, leading in general to worse results than the social optimum. This problem is instead called the User Optimum Traffic Assignment problem. The results we obtain are reported in the third column of Table 1. If we calculate the social optimum cost function but with the Wardrop equilibrium flow $f^{(0)}$ obtained from (6), we obtain

$$\sum_{e \in \mathcal{E}} f^{(0)}_e d_e(f^{(0)}_e) \approx 26292.96257, \tag{8}$$

which is slightly worse than what we obtain with f^*. If we calculate the Price of Anarchy, which represents how much the system is degraded because of selfish behaviour by the drivers, we obtain that

$$PoA(0) = \frac{\sum_{e \in \mathcal{E}} f^*_e d_e(f^*_e)}{\sum_{e \in \mathcal{E}} f^{(0)}_e d_e(f^{(0)}_e)} \approx 1.013465, \tag{9}$$

which is actually very close to 1. This possibly means that, despite the selfish behaviours of the users, the highways structure is well enough thought that it can mitigate this negative effect and return a performance closer to the one that we would have if a central authority decided for each driver what path they should take in order to reach node 17 from node 1.

3.6. Exercise 3.e'

We now introduce the concept of toll ω_e over each link e, which represents how much a driver has to pay in order to pass through the highway section represented by e. They are defined as

$$\omega_e = f^*_e d'_e(f^*_e), \tag{10}$$

where f^*_e is again the social optimum equilibrium on link e and $d'_e(f_e)$ is the derivative of the delay function defined in (2), and is equal to

$$d'_e(f_e) = \frac{l_e C_e}{(C_e - f_e)^2}. \tag{11}$$

If we then define again the Wardrop equilibrium problem but now with perceived cost $\tilde{d}_e(f_e) = d_e(f_e) + \omega_e$, we obtain

$$\min_{\substack{0 \le f < C \\ Bf = \nu}} \sum_{e \in \mathcal{E}} \int_0^{f_e} d_e(s) ds + \omega_e f_e. \tag{12}$$

By introducing the tolls with the formula reported in (10), we actually align the social optimum problem and the Wardrop equilibrium problem: the tolls completely compensate for selfish drivers behaviours and the two problems are actually the same one. We can check these results in the fourth column of Table 1. When we calculate the social optimum cost function with this new equilibrium f^ω, we obtain

$$\sum_{e \in \mathcal{E}} f^{(\omega)}_e d_e(f^{(\omega)}_e) \approx 25943.62282, \tag{13}$$

and finally if we calculate the Price of Anarchy we obtain

$$PoA(\omega) = \frac{\sum_{e \in \mathcal{E}} f^*_e d_e(f^*_e)}{\sum_{e \in \mathcal{E}} f^{(\omega)}_e d_e(f^{(\omega)}_e)} \approx 1.00000, \tag{14}$$

confirming what we were expecting.

	shortest path	**social optimum**	**Wardrop**	**Wardrop w. tolls**	**social optimum w. relative delay**	**Wardrop w. tolls w. relative delay**
e_1	1	6642	6716	6642	6653	6653
e_2	1	6059	6716	6059	5775	5775
e_3	0	3132	2367	3132	3420	3419
e_4	0	3132	2367	3132	3420	3419
e_5	0	10164	10090	10164	10153	10153
e_6	0	4638	4645	4638	4643	4642
e_7	0	3006	2804	3006	3106	3105
e_8	0	2543	2284	2543	2662	2662
e_9	1	3132	3418	3132	3009	3009
e_{10}	0	583	0	583	879	878
e_{11}	0	0	177	0	0	0
e_{12}	1	2927	4171	2927	2355	2356
e_{13}	0	0	0	0	0	0
e_{14}	0	3132	2367	3132	3420	3419
e_{15}	0	5525	5445	5526	5510	5510
e_{16}	0	2854	2353	2854	3044	3043
e_{17}	0	4886	4933	4886	4882	4882
e_{18}	0	2215	1842	2215	2416	2415
e_{19}	0	464	697	464	444	444
e_{20}	0	2338	3036	2338	2008	2009
e_{21}	0	3318	3050	3318	3487	3487
e_{22}	0	5656	6087	5656	5495	5496
e_{23}	0	2373	2587	2373	2204	2204
e_{24}	0	0	0	0	0	0
e_{25}	1	6414	6919	6414	6301	6301
e_{26}	0	5505	4954	5505	5623	5624
e_{27}	0	4886	4933	4886	4882	4882
e_{28}	0	4886	4933	4886	4882	4882

Table 1: The rounded results in terms of the flow vectors obtained when solving points (a) and (d) through (f) of Exercise 3. Note that we round the results to the nearest integer since the flow values are considered in terms of number of drivers.

3.7. Exercise 3.f

In this final point we try to recompute the social optimum equilibrium and the Wardrop equilibrium with tolls by leveraging for each link the cost function

$$c_e(f_e) = f_e(d_e(f_e) - l_e), \tag{15}$$

which considers as delay the difference between the actual delay in travel time on link e that we have when a certain flow f_e occurs minus the travel time it would take to traverse e when it is free of cars (l_e).

We firstly obtain the social optimum f^* by solving the problem

$$\min_{\substack{0 \le f < C \\ Bf=\nu}} \sum_{e \in \mathcal{E}} f_e(d_e(f_e) - l_e), \tag{16}$$

where $d_e(f_e)$ is given by (2). The result is reported in the fifth column of Table 1. We also obtain a total cost of

$$\sum_{e \in \mathcal{E}} f_e^*(d_e(f_e^*) - l_e) \approx 15095.50819 \tag{17}$$

We then calculate tolls with the vector f^* just found by using the same formula as (10), since when we consider our new delay $d_e(f_e) - l_e$, l_e is a constant vector that when derived goes to 0.

We then compute the new Wardrop equilibrium with tolls, by solving the problem

$$\min_{\substack{0 \le f < C \\ Bf=\nu}} \sum_{e \in \mathcal{E}} \int_0^{f_e} d_e(s)ds - l_e f_e + \omega_e f_e, \tag{18}$$

obtaining the result reported in the last column of Table 1. If we calculate the social optimum cost function with this new Wardrop equilibrium flow vector, we finally obtain

$$\sum_{e \in \mathcal{E}} f_e^{(\omega)}(d_e(f_e^{(\omega)}) - l_e) \approx 15095.51326, \tag{19}$$

and if we calculate the Price of Anarchy in this case we again obtain

$$PoA(\omega) = \frac{\sum_{e \in \mathcal{E}} f_e^*(d_e(f_e^*) - l_e)}{\sum_{e \in \mathcal{E}} f_e^{(\omega)}(d_e(f_e^{(\omega)}) - l_e)} \approx 1.00000. \tag{20}$$

Again adding the tolls to the User-Optimum Traffic Assignment problem makes the Wardrop equilibrium coincide with the social optimum equilibrium f^*.

It is also interesting to note how two different delay functions (cfr. (2) and (15)) led to different optimal assignments of flow that can be both considered valid if we consider the two definitions of delay that we have used both equally valid. Note though that we cannot compare the results we obtain as sum of optimal costs in the two cases, since as we said the two definitions of cost over a link e are different.