Network Dynamics and Learning: Homework 1

Fabio Tatti
Politecnico di Torino
s282383@studenti.polito.it

1. Exercise 1

We consider the unitary *o-d* network flows on the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ shown in Figure 1. We do assume that each link *l* has integer capacity C_l.

1.1. Question A

Given that each edge has relative capacity C_l we can assert that the infimum of capacity that we have to remove from the graph is the one corresponding to the edges present on the minimum capacity cut. From the theory we have that the capacity that will flow from *o* to *d*, corresponds to the sum of the capacities of all the edges included in the minimum cut: that is equal to amount of maximum flow that the network can provide to *d* from *o*. Given that a minimum cut by its definition is the cut with minimum flow in a graph given *o* and *d*, the minimum amount of capacity we have to eliminate from the graph in order to make *d* unreachable from *o* corresponds to the edges included in the min cut. Since in this case the capacities C_l still are not numerically defined, we can affirm it only from a theoretical perspective.

1.2. Question B

We assume to have

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

and we want to allocate one unit of additional capacity to maximize the throughput. We solved this task calculating all the different possibilities, where each configuration differ from the others in the updated capacity. Each configuration will add one unit of capacity to a different edge of the graph, then we will compute the maximum flow in order to check if any of them will result in an increased maximum throughput. What results from that computation is that adding only one unit of capacity will not result in a change in the maximum throughput of the graph. From a more theoretical point of view we can proceed as follows. We now consider the four possible *o-d* cuts:

$$\mathcal{U}_1 = (o), \mathcal{U}_2 = (o,a)$$

$$\mathcal{U}_3 = (o,b), \mathcal{U}_4 = (o,a,b)$$

with the relative capacities:

$$C_{\mathcal{U}1} = C_1 + C_2 = 5, C_{\mathcal{U}2} = C_2 + C_3 + C_4 = 7$$

$$C_{\mathcal{U}3} = C_1 + C_5 = 5, C_{\mathcal{U}4} = C_4 + C_5 = 5$$

We know from the theory that the maximum throughput is equal to the minimum cut capacity. In that configuration means that is equal to 5, given by three different equations. It's intuitive to say that if we increase of one unit the capacity of one of the edges C_1, C_2, C_4, C_5 contained in those equations, still we will have at least another one with total capacity equal to 5. That means that no matter which edge we choose, there is no configuration such that the maximum throughput increases with just one unit of capacity.

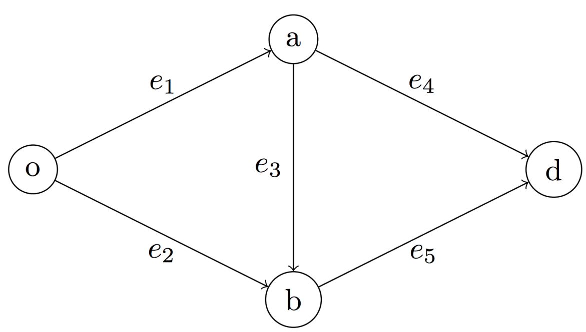

Figure 1: The graph considered for the first exercise

1.3. Question C

Since we increase by two units the capacity, we want to check if there exists a configuration such that we increase the throughput. By looking the previous capacities we can understand that we have to focus on the $C_{\mathcal{U}1}$, $C_{\mathcal{U}3}$ and $C_{\mathcal{U}4}$, since those are the one with the minimum capacity. We need to split the two units additional capacity among the edges e_1, e_2 from $\mathcal{U}_1$, e_1, e_5 from $\mathcal{U}_3$ and e_4, e_5 from $\mathcal{U}_4$. Any configuration that includes at least one edge from e_1, e_5 and maximum a different one from e_2, e_4 will result to have maximum throughput equal to 6, since

$C_{\mathcal{U}1} = C_{\mathcal{U}3} = C_{\mathcal{U}4} = 6$ that still will be the minimum capacity of the different cuts of the network. This because both taking the edges from those sets will ensure to increase all three equation of one unit of capacity. When we calculate the optimal capacity we can see that there is one configuration that distributes in the best way the flow among the available edges when we do increase the capacity of two units. That is reached by increasing C_1 and C_5 since the overall network will result in:

$$C_{\mathcal{U}1} = C_1 + C_2 = 6, C_{\mathcal{U}2} = C_2 + C_3 + C_4 = 7$$

$$C_{\mathcal{U}3} = C_1 + C_5 = 7, C_{\mathcal{U}4} = C_4 + C_5 = 6$$

1.4. Question D

Again we will increase the capacity of 4 units, so to do that we will compute all the different configurations that will result from the previously used equations. Since we noticed in the previous exercise that to matter in rising the maximum throughput are the combinations $[(C_1, C_4)$,(C_1, C_5),$(C_2, C_5)]$ where each couple are the edges whose capacity will be incremented, we consider to do two times that in order to reach 4 units. More specifically we will consider all the different configurations derived from

$$[(C_1, C_4), (C_1, C_5), (C_2, C_5)] \times [(C_1, C_4), (C_1, C_5), (C_2, C_5)]$$

resulting in

$$[(2C_1, C_4), (2C_1, 2C_5), (2C_2, 2C_5), (2C_1, C_4, C_5),$$

$$(C_1, C_2, C_4, C_5), (C_1, C_2, 2C_5)]$$

That will result in a maximum throughput of the network equal to 7, since all the various configurations will have $C_{\mathcal{U}1} = C_{\mathcal{U}4} = 7$. Moreover when we search for the optimal allocation we see that we can derive 2 equally optimal configurations :

$$Conf_A = [2C_1, C_4, C_5], Conf_B = [C_1, C_2, 2C_5]$$

both resulting in

$$C_{\mathcal{U}1} = 7, C_{\mathcal{U}2} = 8, C_{\mathcal{U}3} = 8, C_{\mathcal{U}4} = 7$$

2. Exercise 2

We have the set of people (p_1,p_2,p_3,p_4) and the set of books (b_1,b_2,b_3,b_4).

2.1. Question A

We did represent the interest pattern with a bipartite graph, as shown in Figure 2. A bipartite graph is defined to be as a graph whose vertices can be divided in two independent and disjointed sets, so it means that in the coloring problem has chromatic number equal to two. In this case we have a graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ where the set of vertices $\mathcal{V} = \mathcal{P} \cup \mathcal{B}$, having $\mathcal{P} = (p_1, p_2, p_3, p_4)$ and $\mathcal{B} = (b_1, b_2, b_3, b_4)$. From the theory we know that a simple graph is bipartite if and only if does not have any cycle of odd length. One of the two sets of the graph will be the one of people and the other will be the one of the books. As shown in the image, the relationship that links the two sets is:

$$p_1 \to (b_1, b_2), p_2 \to (b_2, b_3)$$

$$p_3 \to (b_1, b_4), p_4 \to (b_1, b_2, b_4)$$

2.2. Question B

We do exploit the max-flow paradigm in order to establish if there exists a perfect matching. To do that will be used the Ford-Fulkerson method that is based on the Min Cut Theorem, allowing us to calculate the maximum flow between a source o and a sink d. To do that we modify the original bipartite graph from Figure 2 adding the source and the sink, obtaining as result the graph shown in Figure 3. This will allow us to model the problem such that the flow incoming from $o \to \mathcal{P} = (p_1, p_2, p_3, p_4)$ will represent the number of books that each person can take. In that case, since we want to find a perfect match, we will model such that each person can only take one book, so each edge will from $o \to \mathcal{P}$ will have capacity one. Then we assign capacity one also to all the edges on $\mathcal{P} \to \mathcal{B} = (b_1, b_2, b_3, b_4)$, since each person can take maximum one copy of each

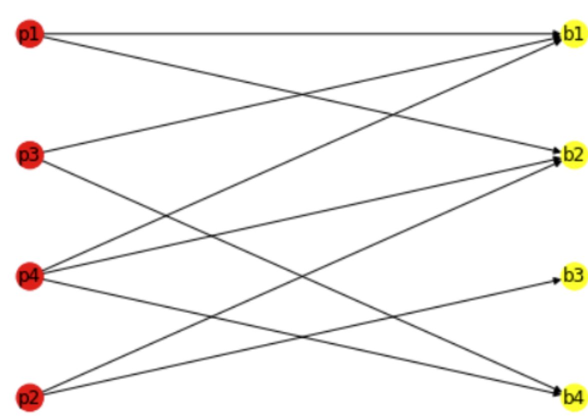

Figure 2: The bipartite graph built for the second exercise

book. Finally, given that we search for a perfect match we set all capacities from $\mathcal{B} \rightarrow d$ one, to allow just one person to take the only copy of each book. Then we use the Ford-Fulkerson method in order to find the minimum cut of the graph, that will correspond to the max flow that the network can transport on $o \rightarrow d$. Moreover, from the Hall's theorem we get that $\exists$ a complete matching from $\mathcal{P}$ to $\mathcal{B} \iff |\mathcal{N}_{\mathcal{P}}| \geq |\mathcal{S}|$. That means that every subset of $\mathcal{P}$ must have sufficiently many adjacent vertices in a subset of $\mathcal{B}$ We get that the maximum throughput is 4, with the cut that divides the graph such that we have two subsets of vertices

$$(o, p_1, p_2, p_3, p_4, b_1, b_2, b_3, b_4), (d) \tag{1}$$

That means, from Hall's theorem, that we can actually find a perfect match on the bipartite graph, given that the max flow is equal to the cardinality of $\mathcal{P}$. We find the best path to be:

$$(p_1 \rightarrow b_2), (p_2 \rightarrow b_3), (p_3 \rightarrow b_1), (p_4 \rightarrow b_4) \tag{2}$$

as shown in Figure 4.

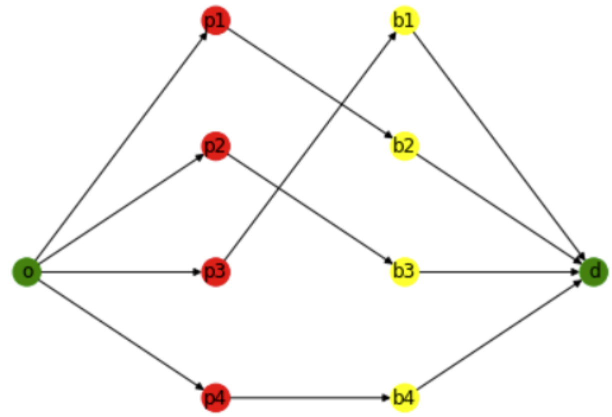

Figure 4: The bipartite graph in the perfect match configuration

2.3. Question C

Now we assume to have a given number of copies for each book, more specifically

$$b_1 = 2, b_2 = 3, b_3 = 2, b_4 = 2 \tag{3}$$

The people has not limit in the number of different books that can bring. To model this situation and understand which will be the final configuration given those constraints we will exploit again the max-flow paradigm, but changing the capacity of some edges respect to the previous point. Given that each person has not constraint in the number of books to take, each edge $o \rightarrow \mathcal{P}$ will have infinite capacity,in order to not be the bottleneck of our network. Again, given that each person can have just a copy of a given book, the capacities on $\mathcal{P} \rightarrow \mathcal{B}$ will have value one. Finally, we set the capacities of the edges on $\mathcal{B} \rightarrow d$ according to the number of copies respectively for each book. The resulting graph is shown in Figure 5.

In that case, when computing the maximum flow between o and d, we obtain 8, meaning that we can only assign 8 book of interest even if we have in total 9. This is also shown in the graph of Figure 6, since we only assign one book b_3 even if we have two available. We can see it looking at the minimum cut, because it cuts the edges incoming in b_3, in other words we have an incoming throughput that is smaller in value than the outgoing throughput in the case of this node.

2.4. Question D

Again, just looking at Figure 6 we can understand that we are selling less copies than the ones that are available in

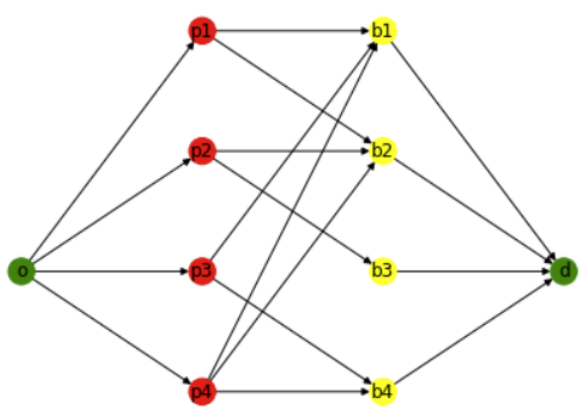

Figure 3: The graph built exploiting the max-flow paradigm to solve the perfect matching problem

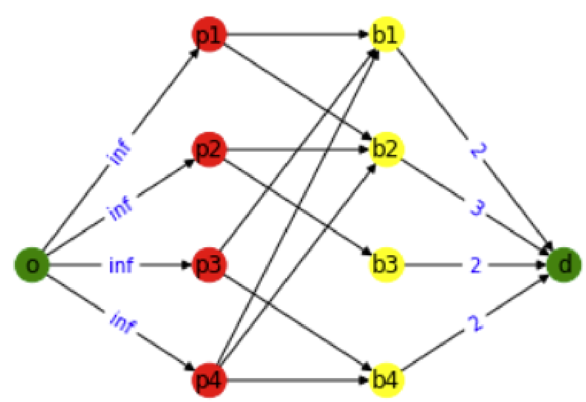

Figure 5: The graph built with given copies for each book

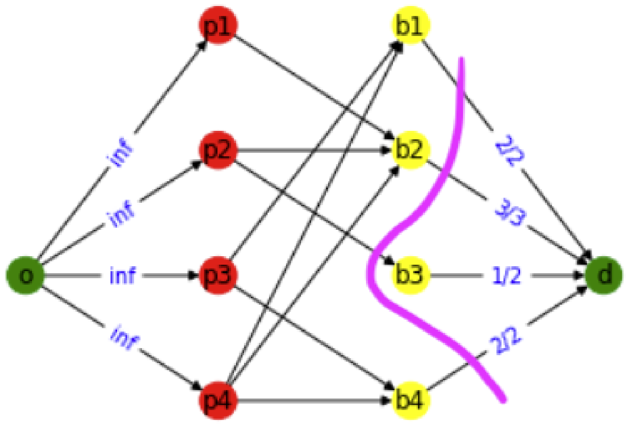

Figure 6: The graph with the sold copies respect to the available ones; in purple is drawn the minimum cut. Notice that b_3 sells only one copy given two, while b_1 sells all the copies without satisfying all the customers.

the case of b_3, that results in the minimum cut shape. Moreover we can also see from Figure 5 node p_1 was looking for book b_1, but it did not manage to get it as shown in Figure 6. What we can do is to decrease of one unit the number of copies of b_3 while increasing of one unit b_1. As shown in Figure 7 this will change the minimum cut with a maximum throughput of the network increased to 9, that is optimal in that case with those constraints, since each customer can have a copy of every book is interested in.

3. Exercise 3

3.1. Question A

We build the graph for the highway network, that is showed in Figure 8. We then compute the fastest path among node 1 and 17 by using the convex optimization. Using the networkx package for Python we simply exploit the built-in function *shortest_path* to double-check the result. We get as sequence of edges :

$$[1, 2, 3, 9, 12, 25]$$

as shown in Figure 8, with total length of 5.

3.2. Question B

We use the convex function minimization approach in order to find the maximum flow. We do maximize tau that will correspond to the maximum flow, given constraints

$$\{\tau \geq 0, f \geq 0, f \leq capacities, B \times f = \tau * \nu\}$$

We exploit again the min-cut theorem in order to double-check the maximum flow of the network, by using the function *minimum_cut*. The result is that the maximum throughput is 22448 with a minimum cut over the edges (l_1, l_5).

3.3. Question C

We now compute the external inflow ν that satisfies $Bf = \nu$ given the flow vector f. What we get is a vector:

$$\nu = [16806, 0, 0, ..., 0, 0, -16806]$$

of lenght 17, with all the entries of value 0 except for the first and the last.

3.4. Question D

We want to find the social optimum f^* with respect to the delays on the different links $d_e(f_e)$. The cost function

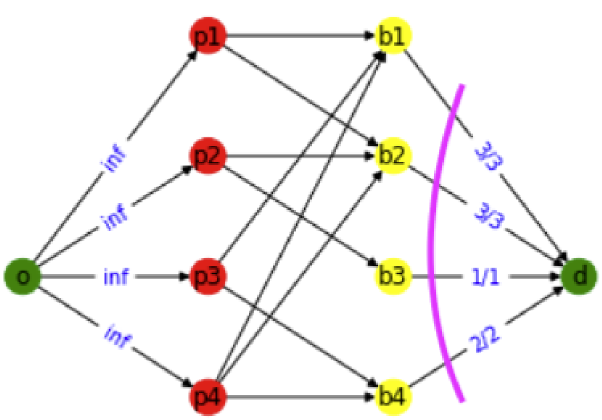

Figure 7: The optimized version of the graph with the sold copies respect to the available ones; in purple is drawn the minimum cut.

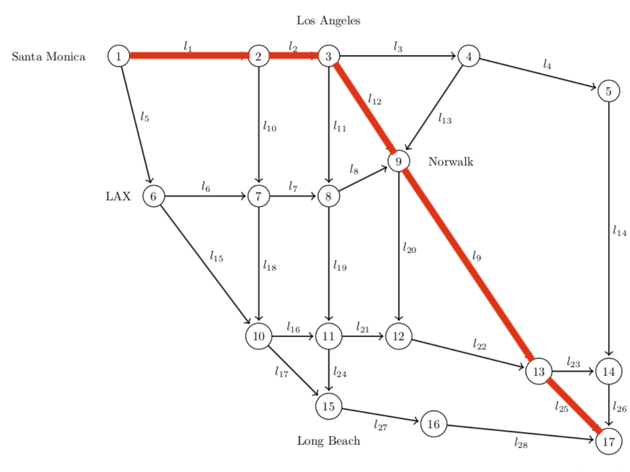

Figure 8: The graph of the highway network, in red the shortest path between node 1 and 17.

to minimize in order to obtain the social optimum is

$$\sum_{e\in\mathcal{E}}(\frac{l_e C_e}{1-f_e/C_e}-l_e C_e)$$

given the constraints

$$\{f^* \geq 0, f^* \leq C_e - \mathbb{1}, B \times f^* = \tau * \nu\}$$

where the ν is the external inflow from the previous question. We obtain a social optimum cost f^* equal to 25943.6157.

3.5. Question E

We want to find the Wardrop equilibrium $f^{(0)}$, by using the cost function

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

having

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

We obtain a cost at Wardrop equilibrium $f^{(0)}$ equal to 26292,9626 and a relative price of anarchy given by $\frac{f^{(0)}}{f^*}$ equal to 1.013465621509814.

3.6. Question F

We introduce the tolls and we compute the new system's Wardrop equilibrium $f^{(\omega)}$, reminding that we modify the delay function such that the cost on each node e is $d_e(f_e) + \omega_e$. We can observe that, by computing the price of anarchy $\frac{f^{(\omega)}}{f^*}$ that is equal to 1.000000275833883, introducing tolls we are able to decrease the cost on the network. That means that we are closing the gap between the real equilibrium and the social optimum, optimizing the flow on the network.

3.7. Question G

We modify the cost function such that

$$c_e(f_e) = f_e(d_e(f_e) - l_e)$$

So we compute from scratch the new social optimum f^* that has value equals to 15095.5082. Then we construct the new tolls, in order to have the Wardrop equilibrium to be equal to f^*. In fact to check if it works we compute the price of anarchy, that has value 1.000000335721437. That means that we managed to introduce effective tolls on the network.