Network Dynamics and Learning Homework 1

Giovanni Sciortino, Giuseppe Suriano

s302959@studenti.polito.it, s296605@studenti.polito.it

March 10, 2023

The code of the homework can be found in the following link here

1 Exercise 1

1.1 Point a

The graph represented in Figure 1 is the object of the exercise number 1. Building the graph we consider the following capacity constraints:

$$c_2 = c_4 = c_6 = 1 \qquad c_1 = c_3 = c_5 = 2$$

In the first point we have to compute the minimum aggregate capacity that has to be removed for no feasible flow from o to d. Considering the capacity of the minimal cut we can isolate the node o and d by removing capacities associated to the edges of this cut. This is the minimal amount that is needed to not have a feasible flow because we consider the minimal cut. The aggregate capacity that can be removed in this case is 3 with the following partition: $\{o, a, b, c\}, \{d\}$.

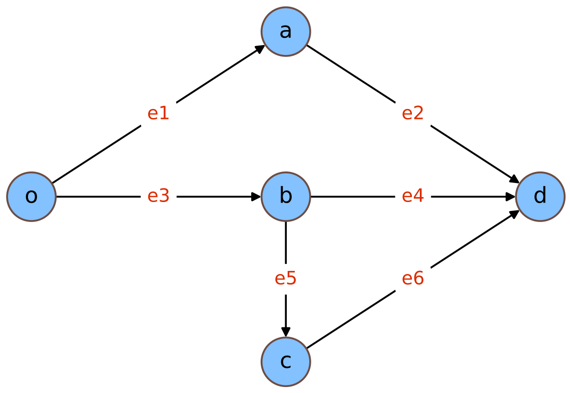

Figure 1: Graph Exercise 1.

1.2 Point b

The second question is about finding the maximal aggregate capacity to be removed to not affect the maximum throughput from o to d. Our solution considers the maximum flow problem applied on the graph, from this the residual capacities are computed as can be seen in the Figure 2. Since the residual capacities previously mentioned could be all removed without affecting the maximal throughout, the solution of this problem is the sum of them (in this case is 2).

1.3 Point c

In order to plot the increment of maximal throughput given a $x > 0$ extra units of capacity we consider the following algorithm:

Considering an arbitrary number of iterations, we implemented a solution that checks if adding a unit of capacity to each edge of the minimal cut (at that iteration) the throughput will increase. In this way if there is an edge

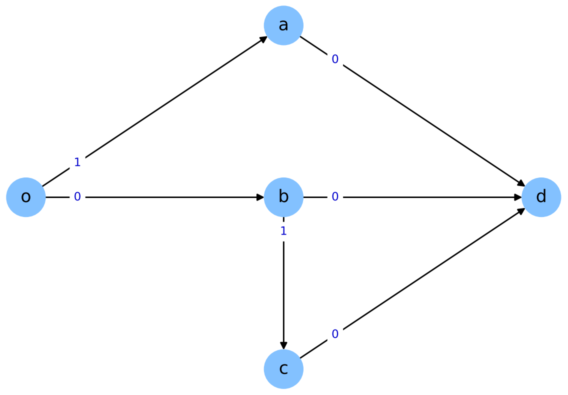

Figure 2: Residual capacities considering the maximal flow solution

Algorithm 1 (Pseudo Code) Additional throughput in relation to extra capacity

```
Require: $x > 0$
  for $iteration = 1, 2, \ldots$ do
      $min\_cut \leftarrow$ min cut
      $val \leftarrow min\_cut$ capacity
      $u \leftarrow min\_cut$ origin partition
      $v \leftarrow min\_cut$ destination partition
      $candidates \leftarrow min\_cut$ edges
      for each $c \in candidates$ do
          $c[capacity] \leftarrow c[capacity] + 1$
          $new\_min\_cut \leftarrow$ new min cut
          if $val < new\_min\_cut\ capacity$ then
              add capacity to edge $c$
          end if
      end for
  end for
```

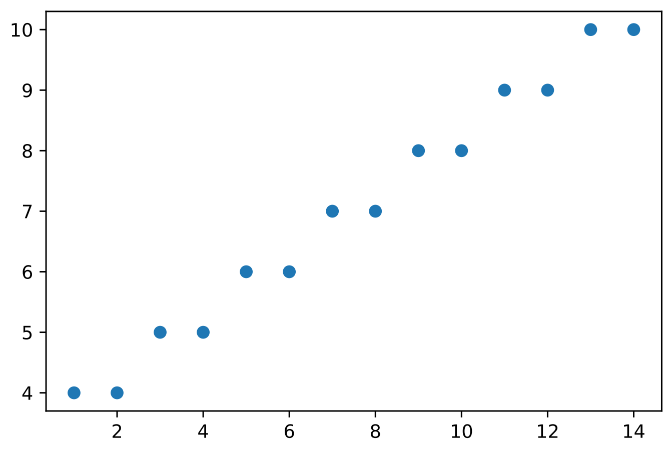

Figure 3: Plot representing how the throughput increase with respect to the extra units of capacity added in the graph (we took in consideration only discrete quantity of extra units of capacity).

that is in common with two different minimal cuts, the algorithm previously explained will choose this edge to add capacity. This new network will be the base for the next iteration.

As it's possible to see in Figure 3 the throughput increase of one unity every two extra units of capacity is added in the graph.

2 Exercise 2

2.1 Point a

To find a perfect matching between people and books, we constructed the graph and added a source node s and a destination node t. Imposing all the edges' capacity equals to 1 and an external inflow equals to 4 (the number of people and books), we can find a perfect matching setting a maximum flow problem. The resulting perfect matching is represented in the Figure 4.

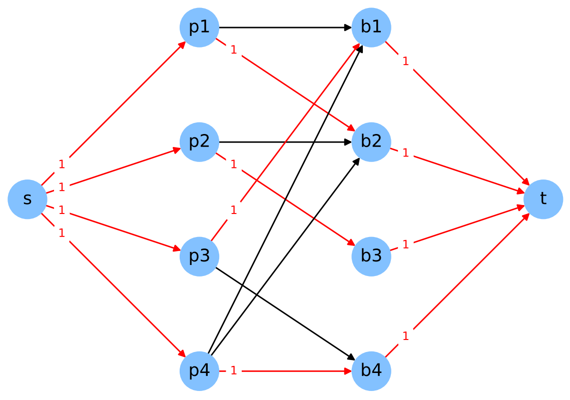

Figure 4: Maximum flow on the graph constructed with edges' capacities equal to 1. As can be seen the throughput of the maximum flow of the graph is 4 that is equal to the cardinality of the nodes representig the persons and the books. So there is a perfect matching on the initial bipartate graph.

2.2 Point b

To maximize the number of assigned books we constructed the edges‘capacity in a particular way that allows us to interpret this task as a maximum flow problem. Specifically, the edges that has head in the source have infinite capacity, the edges that link a person to a book have capacity equals to 1 and the edges that link the books to the destination have availability as capacity. The maximum flow solution is exactly the maximum number of assignments that can be done considering the number of copies available of the books. In this particular case the maximal number of assigned books is 8 (as we can see in Figure 5)

2.3 Point c

As shown in the Figure 6 we can say that the book to sell is the one that has a residual capacity on the edge that links the book to the destination (green edge on Figure 6), this means that no person could buy this book (in this case the book to sell is b3 because the edge that links b3 to the destination has flow 1 but capacity 2). On the other side the book that the library

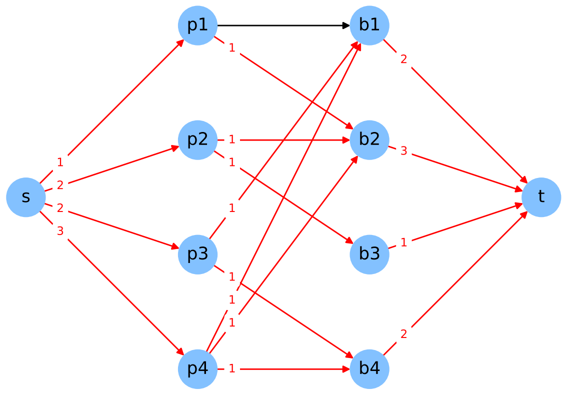

Figure 5: Representation of the max flow problem on the graph in Point b of exercise 2.

should buy is the one that a person can still buy, in other words there is an edge that links a person to a book that has a residual capacity (red edge on Figure 6).

3 Exercise 3

3.1 Point a

As can be seen in Figure 7 the shortest path from node 1 to node 17 is composed by the edges: $l_1,\ l_2,\ l_{12},\ l_9,\ l_{25}$.

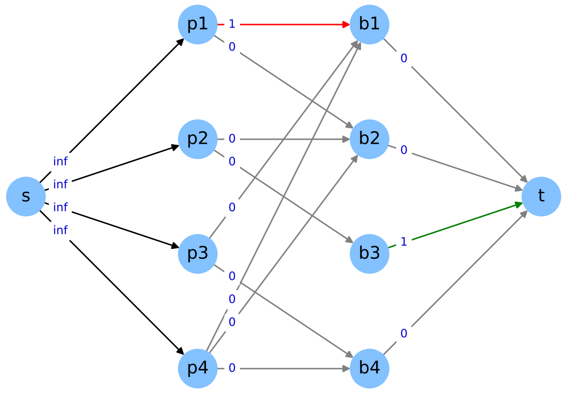

Figure 6: Graph that shows the residual capacities. In this case the book to sell is b3 since the edge that links it to the target node have one residual capacity left (green edge). The book to buy is b1 since the residual capacity of an the edge that have tail in b1 have residual capacity 1(in this case the edge from p1 to b1 highlighted in red).

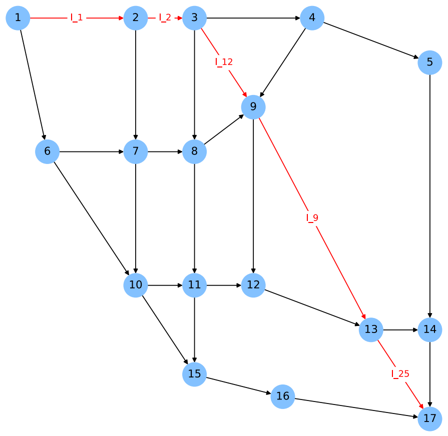

Figure 7: Shortest path from node 1 to 17.

3.2 Point b

In order to compute the maximum flow between nodes 1 and 17, we assign the given capacities from the file *capacities.mat* and use the function of networkx. The maximum flow computed is 22448.

3.3 Point c

The exogenous inflow v is computed simply as $v = Bf$; the result vector is:

$$\begin{bmatrix} 16806 & 8570 & 19448 & 4957 & -746 & 4768 & 413 & -2 & -5671 \\ 1169 & -5 & -7131 & -380 & -7412 & -7810 & -3430 & -23544 & \end{bmatrix}$$

But from now on, we will assume that the exogenous inflow is zero in all the nodes except for node 1, for which v_1 has the same value composed previously, and node 17, for which $v_{17} = v_1$.

3.4 Point d

In order to find the social optimum with respect to the delays on the different links, we define a minimization problem imposing as objective function:

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

respecting the constraints of mass conservation (considering v computed in point c), non-negativity of flow and capacity constraints.

The social optimum flow vector is:

$$\begin{matrix} [6.64219910\text{e}+03 & 6.05893789\text{e}+03 & 3.13232779\text{e}+03 & 3.13232589\text{e}+03 \\ 1.01638009\text{e}+04 & 4.63831664\text{e}+03 & 3.00634073\text{e}+03 & 2.54263460\text{e}+03 \\ 3.13154448\text{e}+03 & 5.83261212\text{e}+02 & 1.45164550\text{e}-02 & 2.92659559\text{e}+03 \\ 1.89781986\text{e}-03 & 3.13232589\text{e}+03 & 5.52548426\text{e}+03 & 2.85427264\text{e}+03 \\ 4.88644874\text{e}+03 & 2.21523712\text{e}+03 & 4.63720641\text{e}+02 & 2.33768761\text{e}+03 \\ 3.31799129\text{e}+03 & 5.65567890\text{e}+03 & 2.37310712\text{e}+03 & 1.99567283\text{e}-03 \\ 6.41411626\text{e}+03 & 5.50543301\text{e}+03 & 4.88645073\text{e}+03 & 4.88645073\text{e}+03] \end{matrix}$$

The cost of social optimum is equal to $25943, 62$.

3.5 Point e.1

To compute the Wardrop equilibrium we have to solve the integral of the delay function in the following way:

$$\begin{aligned}\sum_e \int_0^{f_e} d_e(s)ds &= \sum_e \int_0^{f_e} \frac{l_e}{1-s/C_e}ds \\ &= \sum_e l_e C_e \int_0^{f_e} \frac{1}{C_e - s}ds = \\ &= \sum_e l_e C_e \log(C_e) - l_e C_e \log(C_e - f_e)\end{aligned}$$

The Wardrop optimal flow vector is:

$$\begin{matrix}[6.71564887e+03 & 6.71564494e+03 & 2.36740693e+03 & 2.36740660e+03 \\ 1.00903510e+04 & 4.64539458e+03 & 2.80384448e+03 & 2.28356200e+03 \\ 3.41848060e+03 & 3.93236508e-03 & 1.76827141e+02 & 4.17141087e+03 \\ 3.27483440e-04 & 2.36740660e+03 & 5.44495644e+03 & 2.35317195e+03 \\ 4.93333851e+03 & 1.84155403e+03 & 6.97109625e+02 & 3.03649260e+03 \\ 3.05028112e+03 & 6.08677373e+03 & 2.58651209e+03 & 4.53191804e-04 \\ 6.91874224e+03 & 4.95391869e+03 & 4.93333897e+03 & 4.93333897e+03]\end{matrix}$$

The cost of Wardrop equilibrium is equal to $26292,96$.

As we know from the theory, we can see that the cost of Wardrop equilibrium computed in this case is greater than, or equal to, the cost of the social optimum. In fact the Price of Anarchy is equal to $1,0135$.

3.6 Point e.2

In order to compute the Wardrop equilibrium with tolls, we have to compute the tolls vector ω in the following manner:

$$\omega_e = f_e^* \frac{d}{df_e^*}\left(d_e\left(f_e^*\right)\right) = f_e^* \frac{d}{df_e^*}\left(\frac{C_e l_e}{C_e - f_e^*}\right) = \frac{f_e^* C_e l_e}{\left(C_e - f_e^*\right)^2}$$

where f_e^* is the social optimum flow solution from the definition we know from the theory.

To construct the problem as a Wardrop equilibrium with tolls, we have to construct a new cost function computing the following integral:

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

In conclusion the cost function in this problem results the same as in the point e.1 adding the product of the tolls ω_e and the flow (the variable of the problem).
The Wardrop equilibrium with tolls optimal flow is:

$$\begin{array}{llll}
[6.64297514\text{e}+03 & 6.05907639\text{e}+03 & 3.13247129\text{e}+03 & 3.13247120\text{e}+03 \\
1.01630248\text{e}+04 & 4.63825844\text{e}+03 & 3.00632588\text{e}+03 & 2.54233566\text{e}+03 \\
3.13148981\text{e}+03 & 5.83898752\text{e}+02 & 3.93932003\text{e}-04 & 2.92660470\text{e}+03 \\
9.82102660\text{e}-05 & 3.13247120\text{e}+03 & 5.52476638\text{e}+03 & 2.85422676\text{e}+03 \\
4.88637093\text{e}+03 & 2.21583131\text{e}+03 & 4.63990620\text{e}+02 & 2.33745064\text{e}+03 \\
3.31821725\text{e}+03 & 5.65566790\text{e}+03 & 2.37303631\text{e}+03 & 1.23274734\text{e}-04 \\
6.41412140\text{e}+03 & 5.50550751\text{e}+03 & 4.88637105\text{e}+03 & 4.88637105\text{e}+03]
\end{array}$$

The cost of Wardrop equilibrium with tolls is equal to 25943.62.

3.7 Point f

Supposing that the new cost fuction is the following:

$$\psi_e\left(f_e\right)=f_e\left(\tau_e\left(f_e\right)-l_e\right)$$

We have to redefine the system optimum objective function in following manner:

$$\sum_{e\in\mathcal{E}} f_e\left(d_e\left(f_e\right)-l_e\right)=\sum_{e\in\mathcal{E}}\left(f_e d_e\left(f_e\right)-l_e f_e\right)$$

where the first part was already computed in the point d.
The social optimum flow vector is:

$$\begin{array}{llll}
[6.65326049\text{e}+03 & 5.77465810\text{e}+03 & 3.41974720\text{e}+03 & 3.41974127\text{e}+03 \\
1.01527395\text{e}+04 & 4.64270104\text{e}+03 & 3.10584825\text{e}+03 & 2.66217967\text{e}+03 \\
3.00906114\text{e}+03 & 8.78602390\text{e}+02 & 7.47261013\text{e}-03 & 2.35490342\text{e}+03 \\
5.93671883\text{e}-03 & 3.41974127\text{e}+03 & 5.51003847\text{e}+03 & 3.04369304\text{e}+03 \\
4.88180062\text{e}+03 & 2.41545518\text{e}+03 & 4.43676052\text{e}+02 & 2.00802789\text{e}+03 \\
3.48736711\text{e}+03 & 5.49539501\text{e}+03 & 2.20377719\text{e}+03 & 1.97379951\text{e}-03 \\
6.30067895\text{e}+03 & 5.62351846\text{e}+03 & 4.88180259\text{e}+03 & 4.88180259\text{e}+03]
\end{array}$$

The cost of social optimum in this case is equal to $15095, 51$.

To compute the tolls in a Wardrop equilibrium problem also in this case we have to redefine the objective function. First the tolls are computed as the marginal cost tolls as we know from the theory, i.e. the product of the social optimum flow f_e^* and the derivative of the new delay function in f_e^*. In this case it is:

$$\omega_e^* = f_e^* \frac{d}{df_e^*} \left(d_e \left(f_e^* \right) \right) = \frac{f_e^* C_e l_e}{\left(C_e - f_e^* \right)^2}$$

Now we can recompute the integral inside the objective function in this way:

$$\sum_{e \in \mathcal{E}} \int_0^{f_e} d_e(s) - l_e + \omega_e^* ds$$

The Wardrop equilibrium with tolls optimal flow is:

```
[6.65336598e+03  5.77546849e+03  3.41941782e+03  3.41941652e+03
 1.01526339e+04  4.64273344e+03  3.10549439e+03  2.66172892e+03
 3.00921686e+03  8.77897488e+02  1.60401081e-03  2.35604907e+03
 1.30333772e-03  3.41941652e+03  5.50990049e+03  3.04332449e+03
 4.88171253e+03  2.41513654e+03  4.43767068e+02  2.00856243e+03
 3.48709099e+03  5.49565343e+03  2.20402891e+03  5.70695017e-04
 6.30084138e+03  5.62344543e+03  4.88171310e+03  4.88171310e+03]
```

Solving this problem, the cost of Wardrop equilibrium with tolls end up being equal to 15095,51, which is the same cost of the social optimum computed before.