Homework I

- La soluzione degli esercizi *(ovvero un PDF contenente la risoluzione analitica degli esercizi ed il codice delle funzioni sviluppate nell'Esercizio 3)* deve essere caricata sul portale del corso entro le ore 23:59 di domenica 14 novembre 2021, sotto il nome di Homework1.
- Le qualità dell'esposizione, la capacità di sintesi e la chiarezza del documento finale rientrano nella valutazione dell'homework. La scrittura del documento finale in Latex o in qualsiasi altro formato elettronico è fortemente incoraggiata. Se il documento finale è scritto a mano deve essere facilmente leggibile.
- La collaborazione e lo scambio di idee sono incoraggiati. In ogni caso, ogni studente deve sottomettere il probprio documento (in formato PDF) e il proprio codice numerico, e specificare con chi ha collaborato e per quale specifica parte del lavoro.

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

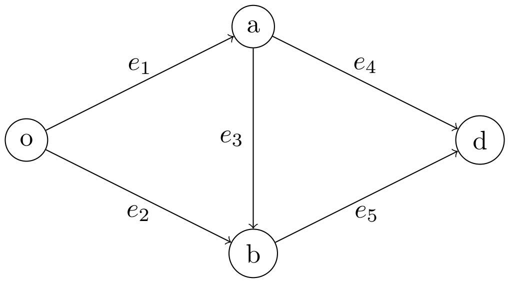

Figure 1

(a) What is the infimum of the total capacity that needs to be removed for no feasible unitary flows from o to d to exist?

(b) Assume that the link capacities are

$$C_1 = C_4 = 3, \qquad C_2 = C_3 = C_5 = 2. \tag{1}$$

Where should 1 unit of additional capacity be allocated in order to maximize the feasible throughput from o to d? What is the maximal throughput after the allocation?

(c) Consider link capacities (1). Where should 2 units of additional capacity be allocated in order to maximize the feasible throughput from o to d? Compute all the optimal capacity allocations for this case and the optimal throughput.

(d) Consider link capacities (1). Where should 4 units of additional capacity be allocated in order to maximize the feasible throughput from o to d? Compute all the optimal capacity allocations for this case. Among the optimal allocations, select the allocation that maximizes the sum of the cut capacities.

Esercizio 2. Consider n_1-n_6 network flows on the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in Figure 2. The links are endowed with delay functions given by

$$d_1(x) = d_6(x) = 3x\,, \qquad d_2(x) = d_3(x) = d_4(x) = d_5(x) = x + 1,$$

and the throughput is 2.

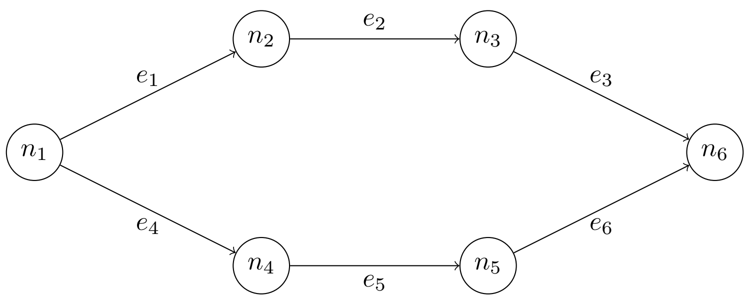

Figure 2

(a) Compute the social optimum flow vector, i.e., the flow vector that minimizes the average delay from n_1 to n_6.

(b) Compute the user optimum flow vector, i.e., the Wardrop equilibrium, and the price of anarchy.

(c) Consider a new link e_7 with delay function $d_7(x) = x$. Find a head and a tail of the link e_7 such that Braess' paradox arises, and compute the price of anarchy on the new graph.

(d) Compute an optimal toll distribution ω on the new graph, i.e., a non-negative toll distribution that reduces the price of anarchy to 1. If possible, compute a full-support optimal toll distribution, i.e., such that $\omega_e > 0$ for every link e. Construct an optimal toll distribution with the smallest possible support.

(e) Assume that the delay functions are given by

$$d_1(x) = d_6(x) = 3x\,, \qquad d_2(x) = d_3(x) = d_4(x) = d_5(x) = d_7(x) = x.$$

Can you find optimal tolls without computing the social optimum flows for this case? Does Braess' paradox still arise? *Hint*: focus on the optimization problem related to social optimum flows and Wardrop equilibria flows.

Esercizio 3. Consider the simple graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in Figure 3.

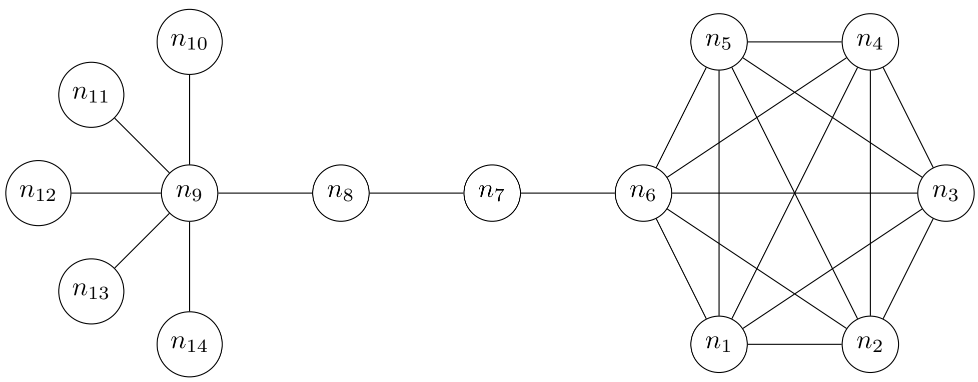

Figure 3

(a) Compute the degree centrality, the eigenvector centrality, the invariant distribution centrality, and comment the results. You can implement the computation in Matlab or Python.

(b) Write a Matlab or Python code for the distributed computation of the Katz centrality and Page-rank centrality, with $\beta = 0.15$ and uniform input.

(c) Analyse the result of point (b), focusing in particular on the centrality of nodes n_6 and n_9.