



**Politecnico  
di Torino**

# Dinamiche su Network

## HOMEWORK 1

*Studente: Innocenti Athos*

*Matr: s307748*

A.A. 2022-23

# Indice

<b>1</b>	<b>Esercizio 1</b>	<b>2</b>
1.1	Punto A . . . . .	3
1.2	Punto B . . . . .	3
1.3	Punto C . . . . .	4
<b>2</b>	<b>Esercizio 2</b>	<b>6</b>
2.1	Punto A . . . . .	7
2.2	Punto B . . . . .	8
2.3	Punto C . . . . .	9
2.4	Punto D . . . . .	12
2.5	Punto E . . . . .	15
<b>3</b>	<b>Esercizio 3</b>	<b>18</b>
3.1	Punto A . . . . .	19
3.2	Punto B . . . . .	21
3.3	Punto C . . . . .	22
3.4	Punto D . . . . .	23
3.5	Note . . . . .	23

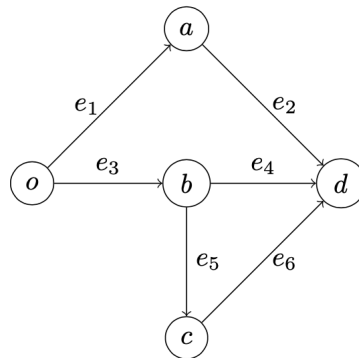
# Capitolo 1

## Esercizio 1

Consider the network in the figure with link capacities

$$c_2 = c_4 = c_6 = 1, \quad c_1 = c_3 = c_5 = 2$$

- (a) What is the minimum aggregate capacity that needs to be removed for no feasible flow from  $o$  to  $d$  to exist?
- (b) What is the maximum aggregate capacity that can be removed from the links without affecting the maximum throughput from  $o$  to  $d$ ?
- (c) You are given  $x > 0$  extra units of capacity. How should you distribute them in order to maximize the throughput that can be sent from  $o$  to  $d$ ? Plot the maximum throughput from  $o$  to  $d$  as a function of  $x \geq 0$ .



## 1.1 Punto A

Per poter calcolare la minimum aggregate capacity che deve essere rimossa affinché non esista nessun feasible flow dal nodo  $o$  al nodo  $d$  devo considerare tutti i possibili  $o$ - $d$  cuts sul grafo e calcolarne la loro capacità. La minima capacità che deve essere rimossa affinché non esista nessun feasible flow da  $o$  a  $d$  è quella del min-cut  $c_{o,d}^*$ .

Ricordo che un  $o$ - $d$  cut è una partizione dell'insieme dei nodi  $\mathcal{V}$  del grafo in due sottoinsiemi  $\mathcal{U}$  e  $\mathcal{V} \setminus \mathcal{U}$  tale che  $o \in \mathcal{U}$  e  $d \in \mathcal{V} \setminus \mathcal{U}$ . La capacità di un taglio  $\mathcal{U}$  viene definita come la capacità aggregata dei links da  $\mathcal{U}$  a  $\mathcal{V} \setminus \mathcal{U}$ :

$$c_{\mathcal{U}} := \sum_{e: \theta(e) \in \mathcal{U}, \kappa(e) \in \mathcal{V} \setminus \mathcal{U}} c_e$$

La capacità di taglio minimo è quindi definita come la capacità minima tra tutti gli  $o$ - $d$  cuts:

$$c_{o,d}^* = \min_{\substack{\mathcal{U} \subseteq \mathcal{V} \\ o \in \mathcal{U}, d \in \mathcal{V} \setminus \mathcal{U}}} c_{\mathcal{U}}$$

a cui corrisponde il taglio minimo  $\mathcal{U}$  tale che  $c_{\mathcal{U}} = c_{o,d}^*$ .

Considerando quindi il grafo, i suoi possibili  $o$ - $d$  cuts sono:

1.  $\mathcal{U}_1 = \{o\}$  e  $\mathcal{U}_1^C = \{a, b, c, d\}$  con capacità  $c_{\mathcal{U}_1} = c_{e_1} + c_{e_3} = 4$
2.  $\mathcal{U}_2 = \{o, a\}$  e  $\mathcal{U}_2^C = \{b, c, d\}$  con capacità  $c_{\mathcal{U}_2} = c_{e_2} + c_{e_3} = 3$
3.  $\mathcal{U}_3 = \{o, b\}$  e  $\mathcal{U}_3^C = \{a, c, d\}$  con capacità  $c_{\mathcal{U}_3} = c_{e_1} + c_{e_4} + c_{e_5} = 5$
4.  $\mathcal{U}_4 = \{o, c\}$  e  $\mathcal{U}_4^C = \{a, b, d\}$  con capacità  $c_{\mathcal{U}_4} = c_{e_1} + c_{e_3} + c_{e_6} = 5$
5.  $\mathcal{U}_5 = \{o, a, b\}$  e  $\mathcal{U}_5^C = \{c, d\}$  con capacità  $c_{\mathcal{U}_5} = c_{e_2} + c_{e_4} + c_{e_5} = 4$
6.  $\mathcal{U}_6 = \{o, a, c\}$  e  $\mathcal{U}_6^C = \{b, d\}$  con capacità  $c_{\mathcal{U}_6} = c_{e_2} + c_{e_3} + c_{e_6} = 4$
7.  $\mathcal{U}_7 = \{o, b, c\}$  e  $\mathcal{U}_7^C = \{a, d\}$  con capacità  $c_{\mathcal{U}_7} = c_{e_1} + c_{e_4} + c_{e_6} = 4$
8.  $\mathcal{U}_8 = \{o, a, b, c\}$  e  $\mathcal{U}_8^C = \{d\}$  con capacità  $c_{\mathcal{U}_8} = c_{e_2} + c_{e_4} + c_{e_6} = 3$

Posso quindi affermare che la minima capacità aggregata che deve essere rimossa affinché non esista nessun feasible flow da  $o$  a  $d$  è pari a  $\mathbf{c}_{o,d}^* = \mathbf{3}$ , la capacità di taglio minima corrispondente ai tagli  $\mathcal{U}_2$  e  $\mathcal{U}_8$ .

## 1.2 Punto B

Per il teorema del massimo flusso - minimo taglio so che il massimo flusso  $\tau_{o,d}^*$  dal nodo  $o$  al nodo  $d$  coincide con la capacità di taglio minima  $c_{o,d}^*$  tra tutti gli  $o$ - $d$  cuts:  $\tau_{o,d}^* = c_{o,d}^*$ .

Affinché il massimo throughput  $\tau_{o,d}^*$  da  $o$  a  $d$  rimanga invariato, dato che questo è pari alla capacità di taglio minima  $c_{o,d}^*$  per il teorema appena enunciato, significa che quest'ultima deve rimanere invariata. Poiché si ha il taglio minimo in  $\mathcal{U}_2$  e  $\mathcal{U}_8$ , le capacità dei links soggetti a questi due tagli non possono essere modificate. Posso dunque escludere i links  $e_2$ ,  $e_3$ ,  $e_4$  ed  $e_6$ . Restano i soli links  $e_1$  ed  $e_5$ , entrambi di capacità 2, che compaiono nei tagli  $\mathcal{U}_1$ ,  $\mathcal{U}_3$ ,  $\mathcal{U}_4$ ,  $\mathcal{U}_5$  e  $\mathcal{U}_7$  di capacità aggregata 4 e 5.

Posso dunque concludere che la capacità massima aggregata che posso rimuovere dai links senza ledere al massimo throughput  $\tau_{o,d}^*$  è **2**.

Nella rimozione della capacità aggregata devo però rispettare la definizione di capacità di un link per cui deve valere che  $c_e > 0 \forall e \in \mathcal{E}$ . Questo mi impedisce, per esempio, di rimuovere due unità di capacità dal link  $e_5$  che farebbe scendere la capacità di  $\mathcal{U}_5$  a 2 ottenendo un nuovo taglio minimo. Stesso discorso può essere fatto per il link  $e_1$  nei tagli  $\mathcal{U}_1$  e  $\mathcal{U}_7$ .

### 1.3 Punto C

Nell'ottica di massimizzare il throughput  $\tau_{o,d}$  che può essere inviato dal nodo  $o$  al nodo  $d$  sfruttando  $x > 0$  nuove unità di capacità posso fare due osservazioni:

1. I due tagli  $\mathcal{U}_2$  e  $\mathcal{U}_8$  per cui si ha la minima capacità di taglio  $c_{o,d}^* = 3$  hanno in comune il link  $e_2$
2. Tutti gli otto tagli contengono il link  $e_1$  o il link  $e_2$  e analogamente contengono il link  $e_3$  o il link  $e_4$ . Cioè per ciascuna delle due coppie di links  $(e_1, e_2)$  ed  $(e_3, e_4)$  posso partizionare l'insieme dei tagli in due sottoinsiemi: l'insieme dei tagli che contengono  $e_1$  e quello dei tagli che contengono  $e_2$ , e allo stesso modo, l'insieme dei tagli che contengono  $e_3$  e l'insieme di quelli che contengono  $e_4$

Sfruttando queste due osservazioni e partendo da una condizione iniziale in cui il throughput è  $\tau_{o,d}^* = 3$ , se  $x = 1$  sfrutto la prima osservazione e aggiungo la nuova unità di capacità nel link  $e_2$  così che le capacità dei tagli  $\mathcal{U}_2$  e  $\mathcal{U}_8$  passino da 3 a 4 il che mi permette di massimizzare il throughput a  $\tau_{o,d}^* = c_{o,d}^* = 4$ , infatti:

1.  $\mathcal{U}_1 = \{o\}$  e  $\mathcal{U}_1^C = \{a, b, c, d\}$  con capacità  $c_{\mathcal{U}_1} = c_{e_1} + c_{e_3} = 4$
2.  $\mathcal{U}_2 = \{o, a\}$  e  $\mathcal{U}_2^C = \{b, c, d\}$  con capacità  $c_{\mathcal{U}_2} = (c_{e_2} + 1) + c_{e_3} = 4$
3.  $\mathcal{U}_3 = \{o, b\}$  e  $\mathcal{U}_3^C = \{a, c, d\}$  con capacità  $c_{\mathcal{U}_3} = c_{e_1} + c_{e_4} + c_{e_5} = 5$
4.  $\mathcal{U}_4 = \{o, c\}$  e  $\mathcal{U}_4^C = \{a, b, d\}$  con capacità  $c_{\mathcal{U}_4} = c_{e_1} + c_{e_3} + c_{e_6} = 5$
5.  $\mathcal{U}_5 = \{o, a, b\}$  e  $\mathcal{U}_5^C = \{c, d\}$  con capacità  $c_{\mathcal{U}_5} = (c_{e_2} + 1) + c_{e_4} + c_{e_5} = 5$
6.  $\mathcal{U}_6 = \{o, a, c\}$  e  $\mathcal{U}_6^C = \{b, d\}$  con capacità  $c_{\mathcal{U}_6} = (c_{e_2} + 1) + c_{e_3} + c_{e_6} = 5$
7.  $\mathcal{U}_7 = \{o, b, c\}$  e  $\mathcal{U}_7^C = \{a, d\}$  con capacità  $c_{\mathcal{U}_7} = c_{e_1} + c_{e_4} + c_{e_6} = 4$
8.  $\mathcal{U}_8 = \{o, a, b, c\}$  e  $\mathcal{U}_8^C = \{d\}$  con capacità  $c_{\mathcal{U}_8} = (c_{e_2} + 1) + c_{e_4} + c_{e_6} = 4$

Generalizzando:

- Se  $x$  è *dispari* aggiungo dapprima una unità di capacità ad  $e_2$  (come appena visto) e le rimanenti  $(x - 1)$  unità, un numero pari, le divido a metà tra  $(e_1, e_2)$  o tra  $(e_3, e_4)$  per quanto detto nella seconda osservazione
- Se  $x$  è *pari* sfrutto il fatto che i due tagli minimi  $\mathcal{U}_2$  e  $\mathcal{U}_8$  presentano il primo il link  $e_3$  e il secondo il link  $e_4$  che godono delle proprietà di cui sopra. La soluzione più semplice è dunque quella di dividere le  $x$  nuove unità di capacità a metà tra i due links  $e_3$  ed  $e_4$

Per  $x > 0$ , supponendo che  $x \in \mathbb{N}$ , la crescita del massimo throughput che può essere inviato dal nodo  $o$  al nodo  $d$  può essere modellata in funzione di  $x$  come

$$f(x) = 3 + \left\lceil \frac{x}{2} \right\rceil$$

intendendo con  $\lceil \cdot \rceil$  la parte intera superiore della divisione  $x/2$ . Ne riporto di seguito un grafico qualitativo

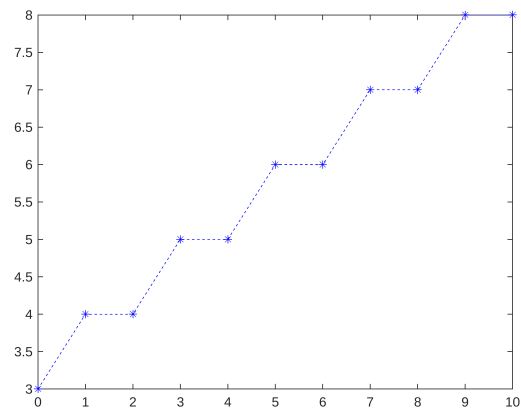


Figura 1.1: Maximum throughput in funzione di  $x$

# Capitolo 2

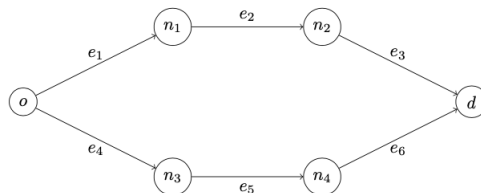
## Esercizio 2

Consider  $o$ - $d$  network flows on the graph in the figure. The links are endowed with delay functions

$$\tau_1(x) = \tau_6(x) = 3x, \tau_2(x) = \tau_3(x) = x + 1, \tau_4(x) = 2x, \tau_5(x) = 2$$

and the throughput is 2.

- (a) Compute the social optimum flow vector, i.e., the flow vector that minimizes the average delay from  $o$  to  $d$ .
- (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  $\tau_7(x) = x$ . 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 vector  $\omega$  on the new graph, i.e., a non-negative toll vector that reduces the price of anarchy to 1. If possible, compute a full-support optimal toll vector, i.e., such that  $\omega_e > 0$  for every link  $e$ . Construct an optimal toll vector with the smallest possible support.
- (e) Consider the original graph, and add an additional link  $e_7 = (n_1, n_4)$  with delay function  $\tau_7(x) = \alpha x + 2$ ,  $\alpha \geq 0$ . Consider  $o$ - $d$  network flows on the new graph with throughput  $\chi > 0$ . Find an optimal toll vector independent of  $\chi$  and  $\alpha$ . Hint: focus on the optimization problem related to social optimum flows and Wardrop equilibria flows.



## 2.1 Punto A

Considero i due possibili *o-d paths*:

- $p^{(1)}$  è il cammino  $o \rightarrow n_1 \rightarrow n_2 \rightarrow d$ , ad esso è associato il flusso  $z_1$  e il ritardo  $\Delta_1 = 3z_1 + z_1 + 1 + z_1 + 1 = 5z_1 + 2$
- $p^{(2)}$  è il cammino  $o \rightarrow n_3 \rightarrow n_4 \rightarrow d$ , ad esso è associato il flusso  $z_2$  e il ritardo  $\Delta_2 = 2z_2 + 2 + 3z_2 = 5z_2 + 2$

So che il throughput è 2, ovvero vale che  $z_1 + z_2 = 2$ . Le matrici associate al grafo sono:

$$A = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}$$

e

$$B = \begin{pmatrix} +1 & 0 & 0 & +1 & 0 & 0 \\ -1 & +1 & 0 & 0 & 0 & 0 \\ 0 & -1 & +1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & +1 & 0 \\ 0 & 0 & 0 & 0 & -1 & +1 \\ 0 & 0 & -1 & 0 & 0 & -1 \end{pmatrix}$$

Per trovare il *social optimum flow vector* devo risolvere il problema di minimo:

$$\begin{aligned} & \text{minimize } z_1(5z_1 + 2) + z_2(5z_2 + 2) \\ & \text{s.t } z_1 + z_2 = 2 \end{aligned}$$

che può essere riscritto in funzione del solo flusso  $z_1$  dato che deve valere il vincolo  $z_1 + z_2 = 2$ . Ottengo dunque:

$$\begin{aligned} & \text{minimize } z_1(5z_1 + 2) + (2 - z_1)[5(2 - z_1) + 2] \\ & \text{s.t } 0 \leq z_1 \leq 2 \end{aligned}$$

la funzione obiettivo è

$$\begin{aligned} f(z_1) &= z_1(5z_1 + 2) + (2 - z_1)(10 - 5z_1 + 2) \\ &= 5z_1^2 + 2z_1 + (2 - z_1)(12 - 5z_1) \\ &= 5z_1^2 + 2z_1 + 24 - 10z_1 - 12z_1 + 5z_1^2 \\ &= 10z_1^2 - 20z_1 + 24 \end{aligned}$$

che ha derivata prima

$$f'(z_1) = 20z_1 - 20$$

e derivata seconda

$$f''(z_1) = 20$$

poiché  $f''(z_1) > 0$  la funzione obiettivo è convessa e il suo minimo lo trovo ponendo  $f'(z_1) = 0$  da cui ottengo

$$20z_1 - 20 = 0 \rightarrow 20z_1 = 20 \rightarrow z_1 = 1$$



ma poiché  $z_2 = 2 - z_1$  ottengo  $\mathbf{z}_1 = \mathbf{z}_2 = \mathbf{1}$  e quindi il vettore:

$$z = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

a cui è associato il *social optimum flow vector*:

$$f = Az = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Verifico che il flusso trovato è valido. Affinché sia valido deve valere che  $f \geq 0$  e che  $Bf = v(\delta^{(o)} - \delta^{(d)})$  dove  $v = 2$  è il throughput:

$$Bf = \begin{pmatrix} +1 & 0 & 0 & +1 & 0 & 0 \\ -1 & +1 & 0 & 0 & 0 & 0 \\ 0 & -1 & +1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & +1 & 0 \\ 0 & 0 & 0 & 0 & -1 & +1 \\ 0 & 0 & -1 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ -2 \end{pmatrix}$$

posso concludere che il flusso trovato è corretto. Il costo associato a questo flusso è

$$\sum_{e \in \mathcal{E}} d_e(f_e) f_e = 3 + 2 + 2 + 2 + 2 + 3 = \mathbf{14}$$

## 2.2 Punto B

Lo *user optimum flow vector* o *equilibrio di Wardrop*, per un dato throughput  $v > 0$ , è un *network flow*

$$f^{(o)} = A^{(o,d)} z$$

dove  $z \in \mathbb{R}^{\Gamma_{o,d}}$  è tale che  $z \geq 0$ ,  $\mathbf{1}'z = v$  e per ogni *path*  $\gamma$  in  $\Gamma_{o,d}$

$$z_\gamma > 0 \rightarrow T_\gamma(z) \leq T_{\bar{\gamma}}(z) \quad \forall \bar{\gamma} \in \Gamma_{o,d}$$

Ovvero: una configurazione di flussi è un equilibrio di Wardrop se per ogni *o-d path*, il suo flusso  $z_i > 0$  solo se  $\Delta_i \leq \Delta_j$  per ogni altro *o-d path*  $p^{(j)}$ . In questo caso:

- $z_1 > 0$  se  $\Delta_1 \leq \Delta_2$  questo si verifica se

$$\begin{aligned} 5z_1 + 2 &\leq 5z_2 + 2 \\ 5z_1 &\leq 5z_2 \\ z_1 &\leq z_2 \end{aligned}$$

- $z_2 > 0$  se  $\Delta_2 \leq \Delta_1$  questo si verifica se

$$\begin{aligned} 5z_2 + 2 &\leq 5z_1 + 2 \\ 5z_2 &\leq 5z_1 \\ z_2 &\leq z_1 \end{aligned}$$

Queste condizioni sono soddisfatte se  $z_1 = z_2$ , ma so che  $z_1 + z_2 = 2$  e dunque  $\mathbf{z}_1 = \mathbf{z}_2 = \mathbf{1}$  (a cui sono associati i ritardi  $\Delta_1 = \Delta_2 = 7$ ) che corrisponde all' *user optimum flow vector*:

$$f = Az = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

il cui costo associato è

$$\sum_{e \in \mathcal{E}} d_e(f_e) f_e = 3 + 2 + 2 + 2 + 2 + 3 = \mathbf{14}$$

e dunque il *price of anarchy* è

$$POA^{(o)} = \frac{\sum_{e \in \mathcal{E}} f_e^{(o)} \tau_e(f_e^{(o)})}{\min_{\substack{f \geq 0 \\ Bf = v(\delta^{(o)} - \delta^{(d)})}} \sum_{e \in \mathcal{E}} f_e \tau_e(f_e)} = \frac{14}{14} = \mathbf{1}$$

## 2.3 Punto C

Considero il nuovo link  $e_7$  che ha ritardo  $\tau_7(x) = x$ . Il paradosso di Braess consiste nel fatto che, rimuovendo un link nel grafo, le performance del sistema all'equilibrio di Wardrop migliorano (ottenendo quindi un costo minore).

Per arrivare al paradosso bisogna inserire il link  $e_7$  in modo che colleghi i nodi  $n_1$  e  $n_4$  così da ottenere il nuovo *o-d path*  $p^{(3)} = o \rightarrow n_1 \rightarrow n_4 \rightarrow d$  a cui è associato il ritardo  $\Delta_3 = 3(z_1 + z_3) + z_3 + 3(z_2 + z_3)$ . Con questo nuovo percorso, i due percorsi preesistenti  $p^{(1)}$  e  $p^{(2)}$  hanno nuovi ritardi:

- $\Delta_1 = 3(z_1 + z_3) + 2z_1 + 2 = 5z_1 + 3z_3 + 2$
- $\Delta_2 = 2z_2 + 2 + 3(z_2 + z_3) = 5z_2 + 3z_3 + 2$

e inoltre vale che  $z_1 + z_2 + z_3 = 2$ .

Calcolo dunque il nuovo equilibrio di Wardrop:

- $z_1 > 0$  se

–  $\Delta_1 \leq \Delta_2$  questo si verifica se

$$5z_1 + 3z_3 + 2 \leq 5z_2 + 3z_3 + 2$$

$$5z_1 + 3z_3 + 2 \leq 2z_2 + 2 + 6 - 3z_1$$

$$8z_1 + 3z_3 \leq 2z_2 + 6$$

$$5z_1 + 6 - 3z_2 \leq 2z_2 + 6$$

$$z_1 \leq z_2$$

–  $\Delta_1 \leq \Delta_3$  questo si verifica se

$$5z_1 + 3z_3 + 2 \leq 3(z_1 + z_3) + z_3 + 3(z_2 + z_3)$$

$$2z_1 + 2 \leq z_3 + 6 - 3z_1$$

$$5z_1 \leq z_3 + 4$$

- $z_2 > 0$  se

–  $\Delta_2 \leq \Delta_1$  questo si verifica se

$$5z_2 + 3z_3 + 2 \leq 5z_1 + 3z_3 + 2$$

$$5z_2 \leq 5z_1$$

$$z_1 \geq z_2$$

–  $\Delta_2 \leq \Delta_3$  questo si verifica se

$$5z_2 + 3z_3 + 2 \leq 3(z_1 + z_3) + z_3 + 3(z_2 + z_3)$$

$$5z_2 + 3z_3 + 2 \leq 6 - 3z_2 + z_3 + 3z_2 + 3z_3$$

$$5z_2 \leq z_3 + 4$$

- $z_3 > 0$  se

–  $\Delta_3 \leq \Delta_1$  questo si verifica se

$$3(z_1 + z_3) + z_3 + 3(z_2 + z_3) \leq 5z_1 + 3z_3 + 2$$

$$z_3 + 6 - 3z_1 \leq 2z_1 + 2$$

$$z_3 + 4 \leq 5z_1$$

$$5z_1 \geq z_3 + 4$$

–  $\Delta_3 \leq \Delta_2$  questo si verifica se

$$3(z_1 + z_3) + z_3 + 3(z_2 + z_3) \leq 5z_2 + 3z_3 + 2$$

$$6 - 3z_2 + z_3 + 3z_2 + 3z_3 \leq 5z_2 + 3z_3 + 2$$

$$4 + z_3 \leq 5z_2$$

$$5z_2 \geq z_3 + 4$$

Mettendo tutto insieme,  $z_1 > 0$ ,  $z_2 > 0$  e  $z_3 > 0$  se e solo se

$$\begin{cases} z_1 = z_2 \\ 5z_1 = z_3 + 4 \\ 5z_2 = z_3 + 4 \\ z_1 + z_2 + z_3 = 2 \end{cases}$$

da cui ottengo  $\mathbf{z}_1 = \mathbf{z}_2 = \mathbf{6/7}$  e  $\mathbf{z}_3 = \mathbf{2} - \mathbf{2z}_1 = \mathbf{2/7}$  e quindi il vettore

$$z = \begin{pmatrix} 6/7 \\ 6/7 \\ 2/7 \end{pmatrix}$$

a cui è associato il flusso

$$f = Az = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 6/7 \\ 6/7 \\ 2/7 \end{pmatrix} = \begin{pmatrix} 8/7 \\ 6/7 \\ 6/7 \\ 6/7 \\ 6/7 \\ 2/7 \end{pmatrix}$$

che verifica l'equazione  $Bf = v(\delta^{(o)} - \delta^{(d)})$ , infatti

$$Bf = \begin{pmatrix} +1 & 0 & 0 & +1 & 0 & 0 & 0 \\ -1 & +1 & 0 & 0 & 0 & 0 & +1 \\ 0 & -1 & +1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & +1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & +1 & -1 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} 8/7 \\ 6/7 \\ 6/7 \\ 6/7 \\ 6/7 \\ 8/7 \\ 2/7 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -2 \end{pmatrix}$$

Il costo associato a questo equilibrio di Wardrop è

$$\begin{aligned} \sum_{e \in \mathcal{E}} d_e(f_e) f_e &= \frac{8}{7} 3 \frac{8}{7} + \frac{12}{7} \left( \frac{6}{7} + 1 \right) + \frac{6}{7} 2 \frac{6}{7} + 2 \frac{6}{7} + \frac{8}{7} 3 \frac{8}{7} + \frac{2}{7} \frac{2}{7} \\ &= \frac{192}{49} + \frac{156}{49} + \frac{72}{49} + \frac{1}{7} + \frac{192}{49} + \frac{4}{49} = \frac{100}{7} \approx \mathbf{14.28} \end{aligned}$$

Ma il costo dell'equilibrio di Wardrop senza  $e_7$  era 14 e dunque si ha il paradosso di Braess richiesto.

Per poter calcolare il *price of anarchy* devo prima trovare l'ottimo sociale del nuovo sistema. Per farlo, devo risolvere:

$$\begin{aligned} &\text{minimize } z_1(5z_1 + 3z_3 + 2) + z_2(5z_2 + 3z_3 + 2) + z_3(3z_1 + 3z_2 + 7z_3) \\ &\text{s.t } z_1 + z_2 + z_3 = 2 \end{aligned}$$

posso scrivere  $z_3$  in funzione di  $z_1$  e  $z_2$  come  $z_3 = 2 - z_1 - z_2$ . Così facendo ottengo il nuovo problema di ottimizzazione

$$\begin{aligned} &\text{minimize } 6z_1^2 + 6z_2^2 - 14z_1 - 14z_2 + 2z_1z_2 + 28 \\ &\text{s.t } 0 \leq z_1 + z_2 \leq 2 \end{aligned}$$

Considero il gradiente della funzione obiettivo:

$$\nabla f(z_1, z_2) = \begin{pmatrix} \frac{\partial f}{\partial z_1} \\ \frac{\partial f}{\partial z_2} \end{pmatrix} = \begin{pmatrix} 12z_1 - 14 + 2z_2 \\ 12z_2 - 14 + 2z_1 \end{pmatrix}$$

e la sua matrice hessiana

$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial z_1^2} & \frac{\partial^2 f}{\partial z_1 \partial z_2} \\ \frac{\partial^2 f}{\partial z_2 \partial z_1} & \frac{\partial^2 f}{\partial z_2^2} \end{pmatrix} = \begin{pmatrix} 12 & 2 \\ 2 & 12 \end{pmatrix}$$

analizzando la matrice hessiana posso dire che  $f(z_1, z_2)$  è convessa e posso dunque trovare il suo minimo uguagliando a zero il gradiente ottenendo il sistema

$$\begin{cases} 12z_1 - 14 + 2z_2 = 0 \\ 12z_2 - 14 + 2z_1 = 0 \end{cases}$$

che ha soluzione  $\mathbf{z}_1 = \mathbf{z}_2 = \mathbf{1}$  e dunque  $\mathbf{z}_3 = \mathbf{2} - \mathbf{2z}_1 = \mathbf{0}$  e quindi il vettore

$$\mathbf{z} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

a cui corrisponde il flusso

$$f = Az = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

che è un flusso ammissibile perché soddisfa l'equazione  $Bf = v(\delta^{(o)} - \delta^{(d)})$ , infatti

$$Bf = \begin{pmatrix} +1 & 0 & 0 & +1 & 0 & 0 & 0 \\ -1 & +1 & 0 & 0 & 0 & 0 & +1 \\ 0 & -1 & +1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & +1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & +1 & -1 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -2 \end{pmatrix}$$

Il costo associato è

$$\sum_{e \in \mathcal{E}} d_e(f_e) f_e = 3 + 1 + 1 + 1 + 1 + 2 + 2 + 3 + 0 = \mathbf{14}$$

Dunque, il *price of anarchy* con il nuovo link  $e_7$  è

$$PoA^{(o)} = \frac{\sum_{e \in \mathcal{E}} f_e^{(o)} \tau_e(f_e^{(o)})}{\min_{\substack{f \geq 0 \\ Bf = v(\delta^{(o)} - \delta^{(d)})}} \sum_{e \in \mathcal{E}} f_e \tau_e(f_e)} = \frac{100/7}{14} = \frac{50}{40} = \mathbf{1,0204}$$

## 2.4 Punto D

Otengo l'*optimal toll vector*  $\omega$  definendolo come

$$\omega_e^* = f_e^* d'_e(f_e^*)$$

dove  $f^*$  è il flusso corrispondente all'ottimo sociale. Per cui

$$\omega^* = \begin{pmatrix} 1 \times 3 \\ 1 \times 1 \\ 1 \times 1 \\ 1 \times 2 \\ 1 \times 0 \\ 1 \times 3 \\ 0 \times 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ 1 \\ 2 \\ 0 \\ 3 \\ 0 \end{pmatrix}$$

Con questo vettore di *tolls* i ritardi dei tre percorsi diventano

- $\Delta_1 = 5z_1 + 3z_3 + 7$
- $\Delta_2 = 5z_2 + 3z_3 + 7$
- $\Delta_3 = 3z_1 + 3z_2 + 7z_3 + 6$

Verifico che, con questi pedaggi,  $PoA^{(o)} = 1$  sapendo che il costo all'ottimo sociale è 14 (calcolato nel punto precedente). Per farlo, devo trovare il nuovo equilibrio di Wardrop:

- $z_1 > 0$  se

- $\Delta_1 \leq \Delta_2$  questo si verifica se

$$\begin{aligned} 5z_1 + 3z_3 + 7 &\leq 5z_2 + 3z_3 + 7 \\ z_1 &\leq z_2 \end{aligned}$$

- $\Delta_1 \leq \Delta_3$  questo si verifica se

$$\begin{aligned} 5z_1 + 3z_3 + 7 &\leq 3z_1 + 3z_2 + 7z_3 + 6 \\ 2z_1 + 1 &\leq 3z_2 + 4z_3 \\ 2z_1 + 1 &\leq z_3 + 6 - 3z_1 \\ 5z_1 &\leq z_3 + 5 \end{aligned}$$

- $z_2 > 0$  se

- $\Delta_2 \leq \Delta_1$  questo si verifica se

$$\begin{aligned} 5z_2 + 3z_3 + 7 &\leq 5z_1 + 3z_3 + 7 \\ z_1 &\geq z_2 \end{aligned}$$

- $\Delta_2 \leq \Delta_3$  questo si verifica se

$$\begin{aligned} 5z_2 + 3z_3 + 7 &\leq 3z_1 + 3z_2 + 7z_3 + 6 \\ 5z_2 + 3z_3 + 7 &\leq 12 + 4z_3 \\ 5z_2 &\leq 5 + z_3 \end{aligned}$$

- $z_3 > 0$  se

- $\Delta_3 \leq \Delta_1$  questo si verifica se

$$\begin{aligned} 3z_1 + 3z_2 + 7z_3 + 6 &\leq 5z_1 + 3z_3 + 7 \\ 3z_2 + 4z_3 &\leq 2z_1 + 1 \\ z_3 + 6 - 3z_1 &\leq 2z_1 + 1 \\ 5z_1 &\geq z_3 + 5 \end{aligned}$$

- $\Delta_3 \leq \Delta_2$  questo si verifica se

$$\begin{aligned} 3z_1 + 3z_2 + 7z_3 + 6 &\leq 5z_2 + 3z_3 + 7 \\ 12 + 4z_3 &\leq 5z_2 + 3z_3 + 7 \\ 5z_2 &\geq 5 + z_3 \end{aligned}$$

Mettendo tutto insieme,  $z_1 > 0$ ,  $z_2 > 0$  e  $z_3 > 0$  se e solo se

$$\begin{cases} z_1 = z_2 \\ 5z_1 = z_3 + 5 \\ 5z_2 = z_3 + 5 \\ z_1 + z_2 + z_3 = 2 \end{cases}$$

da cui ottengo  $\mathbf{z}_1 = \mathbf{z}_2 = \mathbf{1}$  e  $\mathbf{z}_3 = \mathbf{2} - 2\mathbf{z}_1 = \mathbf{0}$  e quindi il vettore

$$z = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

a cui corrisponde il flusso

$$f = Az = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

che è un flusso ammissibile perché soddisfa l'equazione  $Bf = v(\delta^{(o)} - \delta^{(d)})$ , infatti

$$Bf = \begin{pmatrix} +1 & 0 & 0 & +1 & 0 & 0 & 0 \\ -1 & +1 & 0 & 0 & 0 & 0 & +1 \\ 0 & -1 & +1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & +1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & +1 & -1 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -2 \end{pmatrix}$$

Il costo associato è

$$\sum_{e \in \mathcal{E}} d_e(f_e) f_e = 3 + 1 + 1 + 1 + 1 + 2 + 2 + 3 + 0 = \mathbf{14}$$

che corrisponde al costo associato all'ottimo sociale e dunque

$$PoA^{(o)} = \frac{\sum_{e \in \mathcal{E}} f_e^{(o)} \tau_e(f_e^{(o)})}{\min_{\substack{f \geq 0 \\ Bf = v(\delta^{(o)} - \delta^{(d)})}} \sum_{e \in \mathcal{E}} f_e \tau_e(f_e)} = \frac{14}{14} = \mathbf{1}$$

come richiesto dalla definizione di *optimal toll vector*.

Non è possibile costruire un *full - support optimal toll vector*, per cui cioè  $\omega_e > 0 \forall e \in \mathcal{E}$ , poiché il link  $e_5$  ha delay costante  $d_{e_5} = 2$ , dunque derivata nulla e quindi avrò sempre  $\omega_5^* = 0$ . Per  $e_7$  invece, l'ottimo sociale ha  $f_{e_7}^* = 0$  e dunque il prodotto  $f_{e_7}^* d'_{e_7}(f_{e_7}^*)$  sarà sempre nullo. **[Per l'osservazione che segue e per la costruzione di un *optimal toll vector* con supporto minimo ho collaborato con Rampazzo Bianca - s314677]** È possibile porre  $\omega_7 \neq 0$  così da impedire la scelta del percorso  $p^{(3)}$  come accade nell'ottimo sociale. Per farlo considero il sistema di condizioni

$$\begin{cases} A_{p^{(1)}}(\Delta(f^{\omega^*}) + \omega^*) = A_{p^{(2)}}(\Delta(f^{\omega^*}) + \omega^*) \\ A_{p^{(3)}}(\Delta(f^{\omega^*}) + \omega^*) \geq A_{p^{(1)}}(\Delta(f^{\omega^*}) + \omega^*) \\ A_{p^{(3)}}(\Delta(f^{\omega^*}) + \omega^*) \geq A_{p^{(2)}}(\Delta(f^{\omega^*}) + \omega^*) \end{cases}$$

così facendo, il flusso si distribuisce tra i percorsi  $p^{(1)}$  e  $p^{(2)}$  aventi il medesimo ritardo che è ora minore di quello del percorso  $p^{(3)}$ . Sostituendo i flussi all'ottimo ottengo il sistema di pedaggi

$$\begin{cases} \omega_1 + \omega_2 + \omega_3 = \omega_4 + \omega_5 + \omega_6 \\ \omega_7 \geq \omega_4 + \omega_5 - \omega_6 + 1 \end{cases}$$

da cui segue che posso imporre qualsiasi pedaggio  $\omega_7 > 0$  a patto di mantenere i medesimi primi sei pedaggi che compaiono nel vettore  $\omega^*$  calcolato all'inizio del punto D.

Per costruire un *optimal toll vector* con il più piccolo supporto possibile è sufficiente porre  $\omega_i = 0 \ \forall i = 1, \dots, 6$  da cui, sostituendo nel sistema, ottengo la condizione  $\omega_7 \geq 1$ . Il supporto minimo si raggiunge dunque considerando il vettore dei pedaggi  $\omega_{min}^* = (0, 0, 0, 0, 0, 0, 1)$ .

## 2.5 Punto E

Modifico il grafo inserendo il nuovo link  $e_7 = (n_1, n_4)$  avente delay  $\tau_7(x) = \alpha x + 2$  con  $\alpha \geq 0$  e considero un generico throughput  $\chi = z_1 + z_2 + z_3 > 0$ . Con questo link i ritardi dei tre percorsi diventano

- $\Delta_1 = 3(z_1 + z_3) + 2z_1 + 2$
- $\Delta_2 = 2z_2 + 2 + 3(z_2 + z_3)$
- $\Delta_3 = 3(z_1 + z_3) + 2 + \alpha z_3 + 3(z_2 + z_3)$

Calcolo il nuovo equilibrio di Wardrop:

- $z_1 > 0$  se

–  $\Delta_1 \leq \Delta_2$  questo si verifica se

$$\begin{aligned} 3(z_1 + z_3) + 2z_1 + 2 &\leq 2z_2 + 2 + 3(z_2 + z_3) \\ 5z_1 + 3z_3 + 2 &\leq 5z_2 + 3z_3 + 2 \\ z_1 &\leq z_2 \end{aligned}$$

–  $\Delta_1 \leq \Delta_3$  questo si verifica se

$$\begin{aligned} 3(z_1 + z_3) + 2z_1 + 2 &\leq 3(z_1 + z_3) + 2 + \alpha z_3 + 3(z_2 + z_3) \\ 2z_1 &\leq \alpha z_3 + 3\chi - 3z_1 \\ 5z_1 &\leq \alpha z_3 + 3\chi \end{aligned}$$

- $z_2 > 0$  se

–  $\Delta_2 \leq \Delta_1$  questo si verifica se

$$\begin{aligned} 2z_2 + 2 + 3(z_2 + z_3) &\leq 3(z_1 + z_3) + 2z_1 + 2 \\ 5z_2 &\leq 5z_1 \\ z_1 &\geq z_2 \end{aligned}$$

–  $\Delta_2 \leq \Delta_3$  questo si verifica se

$$\begin{aligned} 2z_2 + 2 + 3(z_2 + z_3) &\leq 3(z_1 + z_3) + 2 + \alpha z_3 + 3(z_2 + z_3) \\ 2z_2 &\leq 3\chi - 3z_2 + \alpha z_3 \\ 5z_2 &\leq 3\chi + \alpha z_3 \end{aligned}$$

- $z_3 > 0$  se



–  $\Delta_3 \leq \Delta_1$  questo si verifica se

$$\begin{aligned} 3(z_1 + z_3) + 2 + \alpha z_3 + 3(z_2 + z_3) &\leq 3(z_1 + z_3) + 2z_1 + 2 \\ \alpha z_3 + 3\chi - 3z_1 &\leq 2z_1 \\ 5z_1 &\geq \alpha z_3 + 3\chi \end{aligned}$$

–  $\Delta_3 \leq \Delta_2$  questo si verifica se

$$\begin{aligned} 3(z_1 + z_3) + 2 + \alpha z_3 + 3(z_2 + z_3) &\leq 2z_2 + 2 + 3(z_2 + z_3) \\ 3(z_1 + z_3) + \alpha z_3 &\leq 2z_2 \\ 3\chi - 3z_2 + \alpha z_3 &\leq 2z_2 \\ 5z_2 &\geq 3\chi + \alpha z_3 \end{aligned}$$

Mettendo tutto insieme,  $z_1 > 0$ ,  $z_2 > 0$  e  $z_3 > 0$  se e solo se

$$\begin{cases} z_1 = z_2 \\ 5z_1 = \alpha z_3 + 3\chi \\ 5z_2 = \alpha z_3 + 3\chi \\ z_1 + z_2 + z_3 = \chi \end{cases}$$

da cui ottengo il vettore

$$z = \begin{pmatrix} \frac{(\alpha+3)\chi}{5+2\alpha} \\ \frac{(\alpha+3)\chi}{5+2\alpha} \\ \chi - \frac{2(\alpha+3)\chi}{5+2\alpha} \end{pmatrix}$$

Calcolo adesso l'ottimo sociale. Per farlo devo risolvere il seguente problema

$$\begin{aligned} \text{minimize } & z_1(5z_1 + 3z_3 + 2) + z_2(5z_2 + 3z_3 + 2) + z_3(3z_1 + 3z_2 + (6 + \alpha)z_3 + 2) \\ \text{s.t } & z_1 + z_2 + z_3 = \chi \end{aligned}$$

posso scrivere  $z_3$  in funzione di  $z_1$  e  $z_2$  come  $z_3 = \chi - z_1 - z_2$ . Così facendo ottengo il nuovo problema di ottimizzazione

$$\begin{aligned} \text{minimize } & (5 + \alpha)z_1^2 + (5 + \alpha)z_2^2 - (6\chi + 2\alpha\chi)z_1 - (6\chi + 2\alpha\chi)z_2 + 2\alpha z_1 z_2 \\ \text{s.t } & 0 \leq z_1 + z_2 \leq \chi \end{aligned}$$

Considero il gradiente della funzione obiettivo:

$$\nabla f(z_1, z_2) = \begin{pmatrix} \frac{\partial f}{\partial z_1} \\ \frac{\partial f}{\partial z_2} \end{pmatrix} = \begin{pmatrix} 2(5 + \alpha)z_1 - 6\chi - 2\alpha\chi + 2\alpha z_2 \\ 2(5 + \alpha)z_2 - 6\chi - 2\alpha\chi + 2\alpha z_1 \end{pmatrix}$$

e la sua matrice hessiana

$$H = \begin{pmatrix} \frac{\partial^2 f}{\partial z_1^2} & \frac{\partial^2 f}{\partial z_1 \partial z_2} \\ \frac{\partial^2 f}{\partial z_2 \partial z_1} & \frac{\partial^2 f}{\partial z_2^2} \end{pmatrix} = \begin{pmatrix} 2(5 + \alpha) & 2\alpha \\ 2\alpha & 2(5 + \alpha) \end{pmatrix}$$

analizzando la matrice hessiana posso dire che  $f(z_1, z_2)$  è convessa perché  $\alpha \geq 0$  per ipotesi e posso dunque trovare il suo minimo uguagliando a zero il gradiente ottenendo il sistema

$$\begin{cases} 2(5 + \alpha)z_1 - 6\chi - 2\alpha\chi + 2\alpha z_2 = 0 \\ 2(5 + \alpha)z_2 - 6\chi - 2\alpha\chi + 2\alpha z_1 = 0 \end{cases}$$

che ha soluzione

$$z = \begin{pmatrix} \frac{(\alpha+3)\chi}{5+2\alpha} \\ \frac{(\alpha+3)\chi}{5+2\alpha} \\ \chi - \frac{2(\alpha+3)\chi}{5+2\alpha} \end{pmatrix}$$

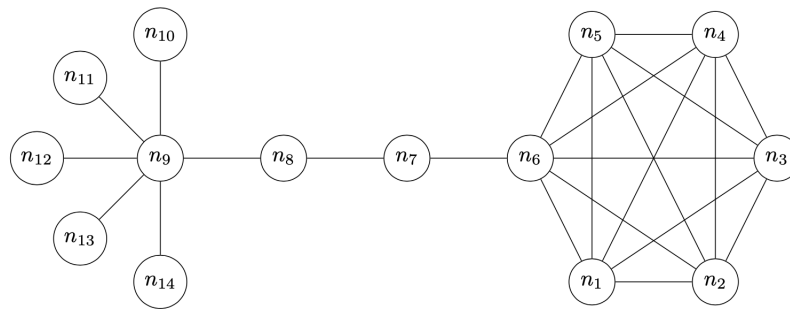
a cui è associato un flusso  $f = Az$  valido poiché  $Bf = BAz = \chi(\delta^{(o)} - \delta^{(d)})$ .

Poiché il vettore  $z$  coincide sia nell'equilibrio di Wardrop che all'ottimo sociale, significa che i due costi sono uguali e dunque, indipendentemente dalla scelta vettore di tool, ho che  $PoA^{(o)} = 1$ . Posso dunque definire in modo arbitrario un *optimal toll vector* indipendente sia da  $\chi$  che da  $\alpha$ .

# Capitolo 3

## Esercizio 3

Consider the simple graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$



- (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 an iterative algorithm in Matlab or Python for the computation of Katz centrality, with  $\beta = 0.15$  and uniform intrinsic centrality  $\mu$ .
- (c) Write a distributed algorithm in Matlab or Python for the computation of Page-rank centrality, with  $\beta = 0.15$  and uniform intrinsic centrality  $\mu$ .
- (d) Analyse the results of points (b) and (c), focusing in particular on the centrality of nodes  $n_6$  and  $n_9$ .

### 3.1 Punto A

Come prima operazione ho definito la matrice di adiacenza relativa al grafo in esame:

$$W = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Riporto ora di seguito il codice Matlab per il calcolo della *degree centrality*:

---

```
degrees = zeros(length(W), 1);
for i = 1:length(W)
    degrees(i) = sum(W(:, i));
end
degreeCentrality = (1 / sum(degrees)) * degrees; % normalized vector
```

---

Il codice riproduce esattamente la definizione di *degree centrality* secondo cui l'importanza di ciascun nodo è semplicemente il suo grado. Poiché il grafo è *undirected* è anche *balanced*, il che significa che *out-degree* e *in-degree* coincidono per ciascuno nodo:  $w_i = w_i^-$ . Il grado di ciascun nodo è stato quindi calcolato come la somma dei pesi (in questo caso unitari) dei links che coinvolgono quel nodo:

$$degrees = (5, 5, 5, 5, 5, 6, 2, 2, 6, 1, 1, 1, 1, 1)^T$$

come ultimo passo il vettore è stato normalizzato dividendo ciascun elemento per la somma di tutti gli elementi del vettore in modo che il vettore risultante sia a somma unitaria:

$$degreeCentrality = \begin{pmatrix} 0.1087 \\ 0.1087 \\ 0.1087 \\ 0.1087 \\ 0.1087 \\ 0.1304 \\ 0.0435 \\ 0.0435 \\ 0.1304 \\ 0.0217 \\ 0.0217 \\ 0.0217 \\ 0.0217 \\ 0.0217 \end{pmatrix}$$

Per il calcolo della *eigenvector centrality* ho implementato il seguente codice:

---

```
[~, eigenvalues] = eig(W);
lambdaW = max(eigenvalues, [], 'all');
[eigenvectors, eigenvalues] = eig((1 / lambdaW) * transpose(W));
eigenvalues = diag(eigenvalues);
eigenvector = eigenvectors(:, eigenvalues == max(eigenvalues, [], 'all')); %
    take the eigenvector associated with the eigenvalue 1
eigenvectorCentrality = (1 / sum(eigenvector)) * eigenvector; % normalized
```

---

Per calcolarla ho sfruttato la relazione

$$\lambda z = W'z$$

secondo cui  $z$  è un autovettore di  $W'$  di autovalore  $\lambda > 0$ . Scegliendo  $\lambda = \lambda_W$  autovalore dominante di  $W$ , la sua trasposta  $W'$  ammette un corrispettivo autovettore non negativo

$$z = \frac{1}{\lambda_W} W'z$$

per trovarlo ho calcolato autovalori e autovettori di  $\frac{1}{\lambda_W} W'$  e ho preso l'autovettore di autovalore  $\lambda = 1$ . Imponendo la normalizzazione  $z' \mathbf{1} = 1$  ho infine ottenuto la *eigenvector centrality*:

$$eigenvectorCentrality = \begin{pmatrix} 0.1583 \\ 0.1583 \\ 0.1583 \\ 0.1583 \\ 0.1583 \\ 0.1639 \\ 0.0340 \\ 0.0071 \\ 0.0018 \\ 0.0003 \\ 0.0003 \\ 0.0003 \\ 0.0003 \\ 0.0003 \\ 0.0003 \end{pmatrix}$$

Infine, ho implementato il calcolo della *invariant distribution centrality*:

---

```
D = diag(degrees);
P = D^(-1) * W;
[eigenvectorsP, eigenvaluesP] = eig(transpose(P));
eigenvaluesP = diag(eigenvaluesP);
eigenvectorP = eigenvectorsP(:, eigenvaluesP == max(eigenvaluesP, [],
    'all')); % take the eigenvector associated with the eigenvalue 1
invariantDistributionCentrality = (1 / sum(eigenvectorP)) * eigenvectorP;
```

---

Per calcolare questa centralità ho sostituito la matrice di adiacenza  $W$  con la sua versione normalizzata, ovvero la matrice stocastica

$$P = D^{-1}W$$

con  $D$  la matrice diagonale dei gradi. Poiché l'autovalore dominante di  $P$  è 1, arrivo alla formula

$$z = P'z$$

per cui  $z$  è una distribuzione invariante. Imponendo nuovamente la normalizzazione  $z' \mathbf{1} = 1$  ho ottenuto la *invariant distribution centrality*:

$$invariantDistributionCentrality = \begin{pmatrix} 0.1087 \\ 0.1087 \\ 0.1087 \\ 0.1087 \\ 0.1087 \\ 0.1304 \\ 0.0435 \\ 0.0435 \\ 0.1304 \\ 0.0217 \\ 0.0217 \\ 0.0217 \\ 0.0217 \\ 0.0217 \end{pmatrix}$$

Analizzando i risultati ottenuti noto che la *degreeCentrality* e la *invariant distribution centrality* sono uguali. Questo è dovuto al fatto che il grafo  $\mathcal{G}$  in esame è un grafo non orientato e dunque è anche un grafo bilanciato, il che implica che per ogni nodo  $i \in \mathcal{V}$  il grado entrante è uguale al grado uscente:  $w_i = w_i^-$ . Ma posso vedere la centralità del generico nodo  $i$  come

$$z_i \propto \sum_{j \in N_i^-} \frac{1}{w_j} z_j = \sum_j P_{ji} z_j$$

da cui  $\pi = w = w^-$  nel caso di grafo bilanciato. Confrontando invece la *eigenvector centrality* con le altre due centralità noto in primo luogo che si mantiene la caratteristica di avere la medesima centralità per gli insiemi di nodi  $\{1, 2, 3, 4, 5\}$  e  $\{10, 11, 12, 13, 14\}$  ma nella *eigenvector centrality* viene meno l'uguaglianza di centralità per i nodi 6 e 9 anche se si conserva la centralità massima in corrispondenza del nodo 6 che, insieme al nodo 9, è quello avente grado entrante (e quindi anche uscente) più grande, cioè sono i due nodi a cui sono associati il maggior numero di links.

## 3.2 Punto B

Riporto di seguito il codice Matlab di un algoritmo *iterativo* per il calcolo della *Katz centrality* con  $\beta = 0.15$  e *uniform intrinsic centrality*  $\mu$ :

---

```

beta = 0.15;
mu = ones(length(W), 1);
katzCentrality = zeros(length(W), 1);
while true
    oldCentrality = katzCentrality;
    katzCentrality = (((1 - beta) / lambdaW) * transpose(W)) *
        katzCentrality) + (beta * mu);
    if katzCentrality == oldCentrality
        break;
    end
end
katzCentrality = (1 / sum(katzCentrality)) * katzCentrality;

```

---

Per poter implementare questo algoritmo ho utilizzato la definizione di *Katz centrality* secondo cui

$$z^{(\beta)} = \left( \frac{1 - \beta}{\lambda_W} \right) W' z^{(\beta)} + \beta \mu$$

dove  $\lambda_W$  autovalore dominante di  $W'$ . Partendo dal vettore nullo  $z^{(\beta)} = \mathbf{0}$ , ho iterato la formula fintanto che il vettore non si è stabilizzato, ovvero l'algoritmo termina nel momento in cui le ultime due iterazioni hanno prodotto il medesimo vettore  $z$ .

La *Katz centrality* ottenuta è:

$$katzCentrality = \begin{pmatrix} 0.1233 \\ 0.1233 \\ 0.1233 \\ 0.1233 \\ 0.1233 \\ 0.1299 \\ 0.0457 \\ 0.0335 \\ 0.0456 \\ 0.0258 \\ 0.0258 \\ 0.0258 \\ 0.0258 \\ 0.0258 \end{pmatrix}$$

### 3.3 Punto C

Riporto di seguito il codice Matlab di un algoritmo *distribuito* per il calcolo della *Page-rank centrality* con  $\beta = 0.15$  e *uniform intrinsic centrality*  $\mu$ .

---

```

beta = 0.15;
mu = 1;
pageRank = zeros(length(W), 1);
degrees = zeros(length(W), 1);
for i = 1:length(W)
    degrees(i) = sum(W(:, i));
end
while true
    oldPageRank = pageRank;
    for j = 1:length(W)
        sigma = 0;
        for k = 1:length(W)
            sigma = sigma + (((1/degrees(k, 1)) * W(k, j)) * pageRank(k, 1));
        end
        pageRank(j, 1) = ((1 - beta) * sigma) + (beta * mu);
    end
    if pageRank == oldPageRank
        break;
    end
end
pageRank = (1 / sum(pageRank)) * pageRank;

```

---

ho utilizzato l'uguaglianza

$$z^{(\beta)} = (1 - \beta)P'z^{(\beta)} + \beta\mu$$

scomponendo il prodotto  $P'z^{(\beta)}$  in sommatoria così da poter calcolare ciascuna componente di  $z$  come

$$z_j^{(\beta)} = (1 - \beta) \sum_k \frac{W_{kj}}{w_k} z_k^{(\beta)} + \beta\mu_j$$

al fine di rendere l'algoritmo distribuito.

La *page-rank centrality* ottenuta è:

$$\text{pageRankCentrality} = \begin{pmatrix} 0.0766 \\ 0.0766 \\ 0.0766 \\ 0.0766 \\ 0.0766 \\ 0.0973 \\ 0.0507 \\ 0.0616 \\ 0.2072 \\ 0.0401 \\ 0.0401 \\ 0.0401 \\ 0.0401 \\ 0.0401 \\ 0.0401 \end{pmatrix}$$

### 3.4 Punto D

Anche per la *Katz centrality* e la *page-rank centrality* si mantiene la caratteristica di avere la medesima centralità per gli insiemi di nodi  $\{1, 2, 3, 4, 5\}$  e  $\{10, 11, 12, 13, 14\}$ . Inoltre, la *Katz centrality* è maggiore rispetto alla *page-rank centrality* nei primi sei nodi, ma minore nei rimanenti otto. Vi è però una considerevole differenza nelle centralità dei nodi 6 e 9: mentre nella *Katz centrality* il nodo avente centralità maggiore è il nodo 6, nella *page-rank centrality* è invece il nodo 9 quello con maggior centralità. La *Katz centrality* inserisce una centralità intrinseca ai nodi che può essere trasferita ai loro vicini, per cui si ha centralità elevata quando un nodo è collegato a molti nodi e/o questi hanno elevata centralità. È questo il motivo per cui nella *Katz centrality* il nodo avente centralità maggiore è il nodo 6. Al contrario, la *page-rank centrality* del nodo 9 è quella maggiore poiché i suoi vicini sono collegati ad esso soltanto, al contrario del nodo 6 che invece ha dei vicini collegati con altri nodi e tra loro stessi.

### 3.5 Note

#### **CODICE MATLAB COMPLETO:**

[github.com/athos-innocenti/Homeworks/blob/main/Dinamiche%20su%20Network/homework\\_1.m](https://github.com/athos-innocenti/Homeworks/blob/main/Dinamiche%20su%20Network/homework_1.m)