Network Dynamics course
Project Report

HOMEWORKS REPORT

July 15, 2023

Alessandro Rizzardi
Universita di Trento
alessandro.rizzardi@stundenti.unitn.it

Contents

1 Homework 1

1.1 Pressure Sensors

Considering the the graph in Figure 1 representing a network of pressure sensors we firstly want to compute some centralities.

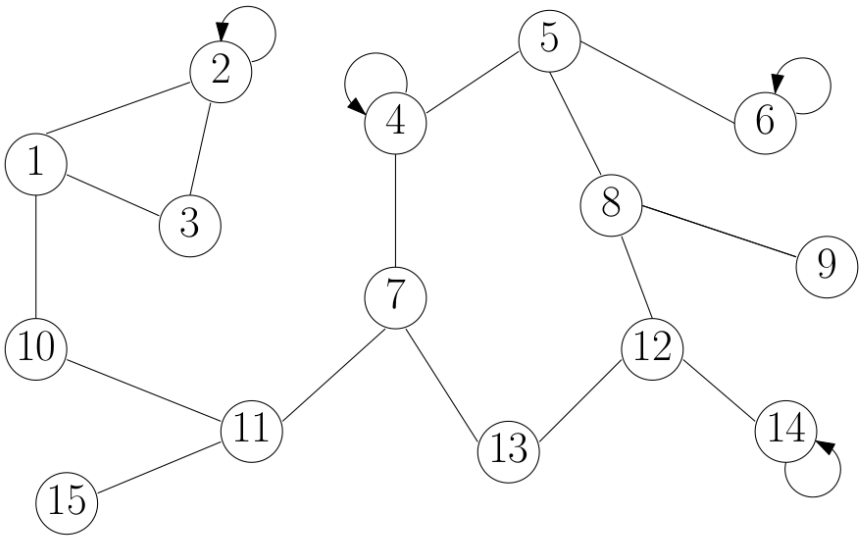

Figure 1: Network of pressure sensors

The invariant centrality π is easily computed considering the problem:

$$\pi = P^T \pi \tag{1}$$

where the matrix P is the normalized weight matrix of the matrix W representing the network.

The results are shown in Table 1.

node	π
1	0.083
2	0.083
3	0.056
4	0.083
5	0.083
6	0.056
7	0.083
8	0.083
9	0.083
10	0.028
11	0.056
12	0.083
13	0.056
14	0.056
15	0.028

Table 1: Invariant centrality for each node

The invariant centrality of a node clearly depends of the number of link of each node, so the nodes with the same number of links have the same invariant centrality value.

The closeness centrality is the inverse average distance from i to all other nodes j and for a single node:

$$closeness(i) = \frac{n}{\sum_j dist(i,j)} \tag{2}$$

The betweenness centrality is the fraction g_{ij}^k of all the minimum-distance paths from i to j that pass through the node k:

$$betweenness(k) = \frac{1}{n^2} \sum_{i,j} g_{ij}^k \tag{3}$$

For the closeness centrality the higher the value the higher is the centrality of the node's position, while for betweenness centrality higher values are associated to the more central node crossed by more paths and some nodes have betweenness centrality equal to zero because they are very far away from the center and there are no paths crossing them.

In Table 2 the results obtained for this network.

node	closeness	betweenness
1	0.278	0.264
2	0.227	0
3	0.227	0
4	0.375	0.242
5	0.349	0.253
6	0.268	0
7	0.429	0.56
8	0.312	0.209
9	0.246	0
10	0.333	0.363
11	0.395	0.538
12	0.349	0.253
13	0.375	0.242
14	0.268	0
15	0.294	0

Table 2: Closeness and betweenness centrality for each node

The PageRank centrality, also called Bonacich centrality, gives the relative importance of a node. The centrality is given by:

$$\pi_B = (1-\beta)\, P^T \pi_B + \beta\, \mu \tag{4}$$

with $0 < \beta \leq 1$

The solution can be computed analytically or numerically as:

$$\pi_B(k+1) = (1-\beta)\, P^T \pi_B(k) + \beta\, \mu \tag{5}$$

In Table 3 the results obtained.

1.1.1 DeGroot dynamics

We now consider each node to have an initial state given by $x_i(0) = \theta + \xi_i$, where $\theta = 101,325$ Pa and the Gaussian noise has zero mean and $\sigma = 0.5Pa$ as standard deviation.

We can used the DeGroot dynamics to compute a consensus value that provides us a weighted average of the initial nodes' state. So the described dynamics, for discrete-time case, has form:

$$x(k+1) = P\, x(k) \tag{6}$$

and an analytic results can be computed as:

$$\alpha = \pi^T x(0) \tag{7}$$

so, all the nodes converge to the consensus value alpha.

node	$\beta = 0.15$ $\mu_i = 1$	$\beta = 0.6$ $\mu_i = 0 \forall i = 7$ $\mu_7 = 1$	$\beta = 0.6$ $\mu_i = 0 \forall i = 11$ $\mu_1 1 = 1$
1	1.19	0.003	0.029
2	1.164	0.001	0.008
3	0.817	0	0.006
4	1.161	0.101	0.022
5	1.204	0.015	0.004
6	0.854	0.002	0.001
7	1.203	0.644	0.107
8	1.278	0.005	0.002
9	0.512	0.001	0
10	0.853	0.013	0.104
11	1.292	0.093	0.593
12	1.241	0.019	0.005
13	0.842	0.088	0.019
14	0.872	0.003	0.001
15	0.516	0.012	0.099

Table 3: PageRank centralities for each conditions

node	$x_i(0)$
1	101324.804
2	101325.742
3	101324.607
4	101324.712
5	101325.638
6	101324.236
7	101324.486
8	101324.79
9	101324.717
10	101324.216
11	101323.847
12	101324.662
13	101325.001
14	101324.549
15	101324.518

Table 4: Initial status of nodes in DeGroot dynamics

The initial states are computed randomly every time, an example of initial state is shown in Table 4.

The consensus value found analytically is 101324.736 $^\circ C$, the same of the one obtained numerically.

If we add a self loop for each nodes, setting the diagonal of the weight matrix to 1 we obtained a slightly different result equal to 101324.719 $^\circ C$

To compute the exact average we run to consensus algorithm in parallel considering the following two initial states:

$$x_i'(0) = \frac{x_i(0)}{w_i} \quad x_i''(0) = \frac{1}{w_i}$$

Once we run the consensus algorithm for both we can combining the two results to get the exact average given by the ratio:

$$\lim_{t\to\infty} \frac{x_i'(t)}{x_i''(t)}$$

If we consider to set some stubborn nodes to a fixed input value we need to change a bit the dynamics equation where:

$$P = \begin{pmatrix} Q & B \\ 0 & I \end{pmatrix}$$

resulting in:

$$x(k+1) = Q\,x(k) + B\,u \tag{8}$$

Considering the nodes $3, 12$ as stubborn with input value $u_3 = 12$ and $u_{12} = 6$, the equilibrium vector is $x_{eq} = (11.2, 11.6, 7.2, 6.8, 6.8, 7.6, 6.4, 6.4, 10.0, 8.8, 6.8, 6.0, 8.8)$.

In Figure 2 an example of evolution of the value of each node during a simulation.

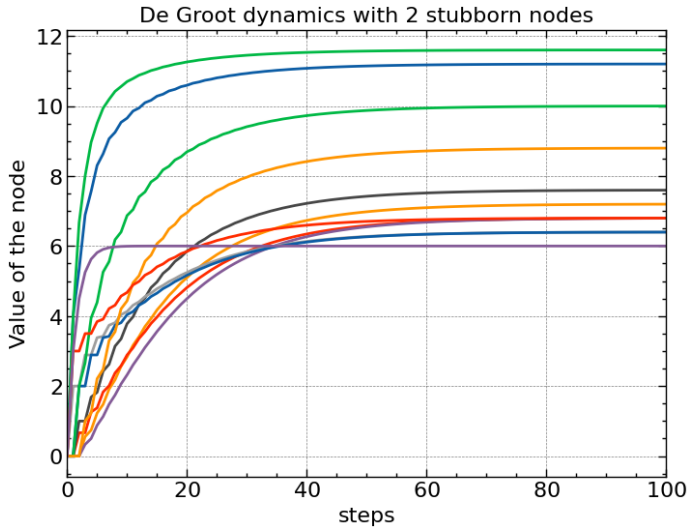

Figure 2: DeGroot dynamics evolution with node 3 and 12 as stubborn

If now we consider only one stubborn as the nodes 11 with input value $u_{11} = 15$, we can compute the results analytically as:

$$x_{eq} = (I - Q)^{-1} B\, u \tag{9}$$

or numerically as shown above, in each case the equilibrium value for all the nodes is 15, the same of the only stubborn nodes.

1.2 Robotic coordination

We consider a network of 16 robots with the following dynamics:

$$\dot{p}_i = u_i \quad i = 1, \ldots, n \tag{10}$$

where $p_i \in \mathbb{R}$ is the position of the i-th robot.

Given a random initial position to each robotwe want to check the behaviour of the control law given by:

$$u_1 = \frac{1}{2}(p_1 + p_2) - p_1 u_i = \frac{1}{3}(p_{i-1} + p_{i+1}) - p_i u_n = \frac{1}{2}(p_{n-1} + p_n) - p_n \tag{11}$$

The dynamics equation is:

$$\dot{p} = -Lp \tag{12}$$

This can be numerically solved using Euler method:

$$p_{k+1} = p_k - Lp\, dt \tag{13}$$

In those equation the matrix L has been computed from the control law and it looks like:

$$L = \begin{pmatrix}
\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & \frac{1}{2}
\end{pmatrix}$$

From the control law we can also derive the degree matrix D considering that every robots interact with the previous and the next, except the particular case of the first and last one, and the adjacency matrix W cam be computed just as $D - L$, so they result in:

$$D = \begin{pmatrix} 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 \end{pmatrix}$$

$$W = \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{4}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & \frac{4}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & \frac{4}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{4}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{4}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{4}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{4}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{4}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{4}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{4}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{4}{3} & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{4}{3} & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{4}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{4}{3} & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

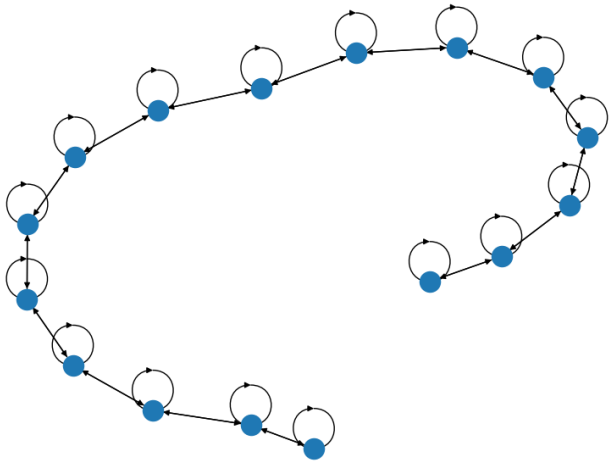

Figure 3: Graph of the robotic network

The adjacency matrix found correspond to a network with a topology shown in Figure 3.

Given the dynamics at (13) we simulate the system considering random initial position and the Figure 4 shown the results. As we can see the consensus is reached with all the robots having final position around a common point.

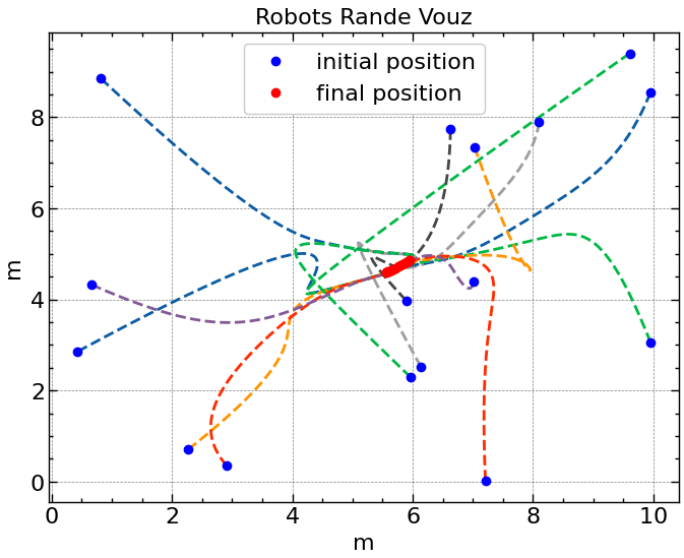

Figure 4: Paths of the robot for reaching rande vouz

The problem could have been solved also analytically computing the consensus value as:

$$\bar{\alpha} = x^T(0)\,\bar{\pi} \tag{14}$$

Where $\bar{\pi}$ is the normalized invariant distribution centrality equal to:

$$\bar{\pi} = \frac{D^{-1}\,\pi}{\mathbb{1}^T\,D^{-1}\,\pi} \tag{15}$$

The result obtained is $\alpha_x = 5.327$, $\alpha_y = 5.329$, close to the one obtained numerically.

If we fixed two robots in the opposite vertices of the squared box containing the robots, imposing the control law:

$$u_i = \frac{1}{3}(p_{i-1} + p_i + p_{i+1}) - p_i \quad i = 1, \ldots, n \tag{16}$$

the result we obtained running the simulation numerically brigs to the one shown in Figure 5.

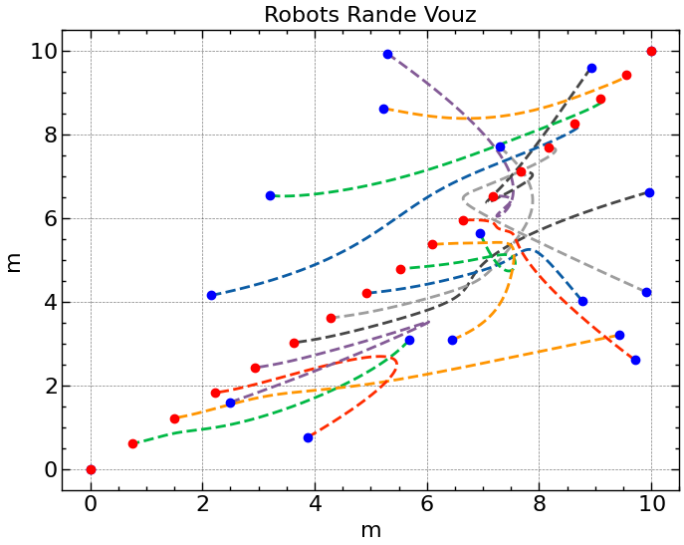

Figure 5: Simulation considering two fixed robots

As shown in the figure with two fixed robots is impossible to reach the the randevouz with the robots posing themself on a line between the fixed two.

1.3 Epidemics

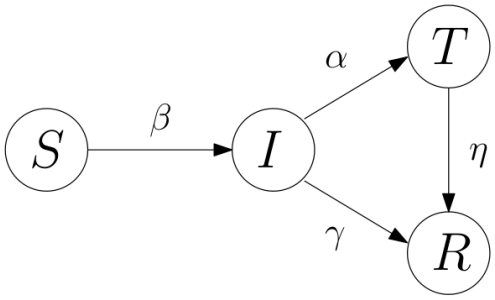

Figure 6: Compartimental network model of SITR

We consider a SITR(Susceptible-Infectious-Treated-Recover) compartmental model (Figure 6) , which has the following equation.

$$\dot{S} = -\beta\, S\,(I + \delta\, T) \tag{17}$$
$$\dot{I} = \beta\, S\,(I + \delta\, T) - (\alpha\, + \gamma)\, I \tag{18}$$
$$\dot{R} = \gamma\, I + \eta\, T \tag{19}$$
$$\dot{T} = -\eta\, T + \alpha\, I \tag{20}$$
$$\tag{21}$$

where $\beta, \gamma, \alpha, \delta, \eta$ are the contagion rate, the recovery rate, percentage of treated, the percentage of infectivity reduction and the percentage of treated recovered.

To analyze the behaviour of the system we run some simulation with different set up. The initial state of each compartment is the same for all the simulation with an initial number of susceptible and infectious respectively equal to the 70 % and the 30 % of the total.

In the first simulation we consider the case of $R_0 < 1$, where R_0 is defined as:

$$R_0 = \frac{\beta}{\gamma} \tag{22}$$

Defining $R_0 < 1$, we expected a quick drop of the infectious and a steady state condition stabilize at an equilibrium in which $R_{eq} = \bar{R_1}$, $S_{eq} = \bar{S_1}$, while $I_{eq} = T_{eq} = 0$.

Results of the simulation are shown in Figure 7.

The second scenario we consider we set $R_0 > 1$. In this scenario we expect the infectious to raise at the beginning, to reach a peak, and the slowly stabilize at an equilibrium that is the same of the first scenario, $R_{eq} = \bar{R_2}$, $S_{eq} = \bar{S_2}$, while $I_{eq} = T_{eq} = 0$.

Results of the simulation are shown in Figure 8.

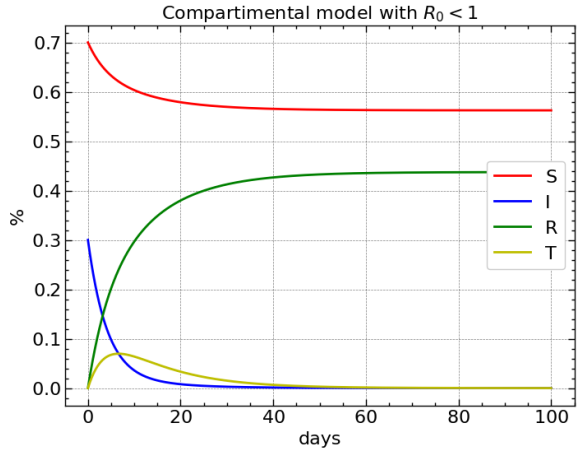

Figure 7: Compartimental model considering $R_0 < 1$

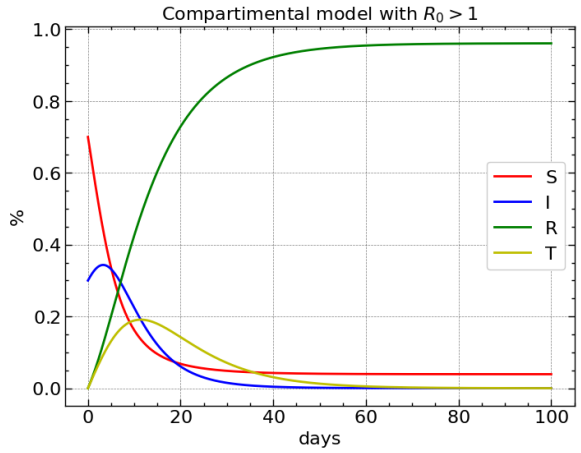

Figure 8: Compartimental model considering $R_0 > 1$

In these simulation we consider the treated to be 50% less contagious but we can study what happen to the simulation we we push the δ value close to its limits as shown in Figure 9.

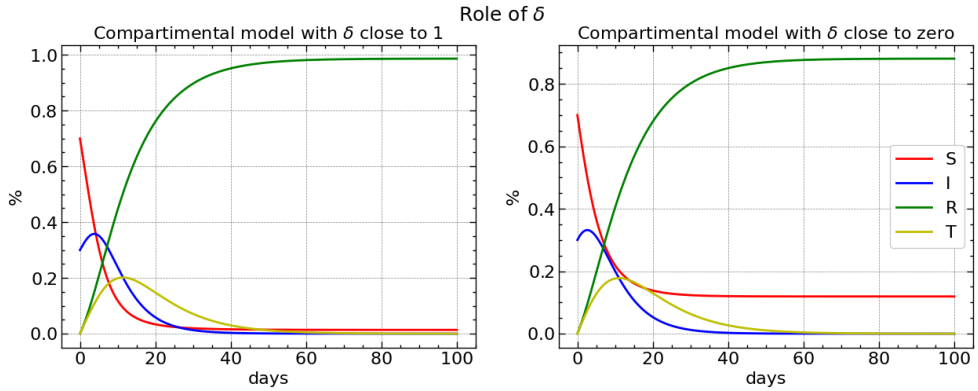

Figure 9: Compartimental model considering δ with limit values

An alternative approach could be consider a network of people, randomly connected, created with the same initial percentage of the compartmental model($S = 0.7$, $I = 0.3$). The networked based simulation for every days check the status of every individual and the status is changed based on the parameters $\beta, \gamma, \delta, \alpha, \eta$ and the topology of the graph.

We simulate both scenario ($R_0 < 1$ and $R_1 > 1$) and the results of simulations can be seen is Figure

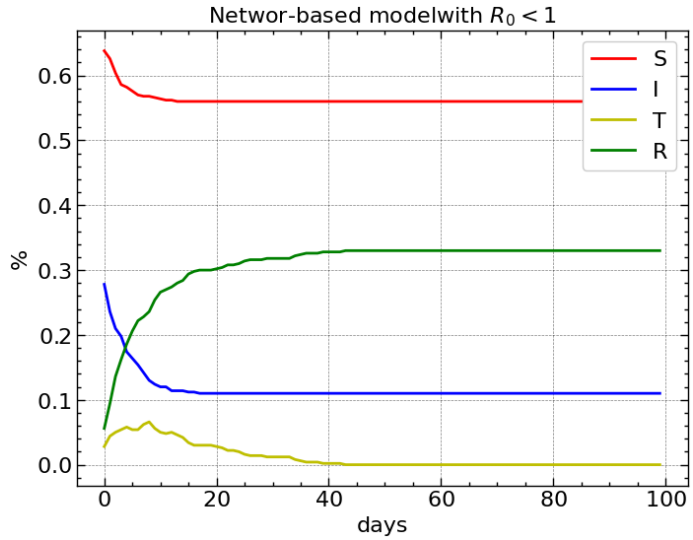

Figure 10: Networked based model with $R_0 < 1$

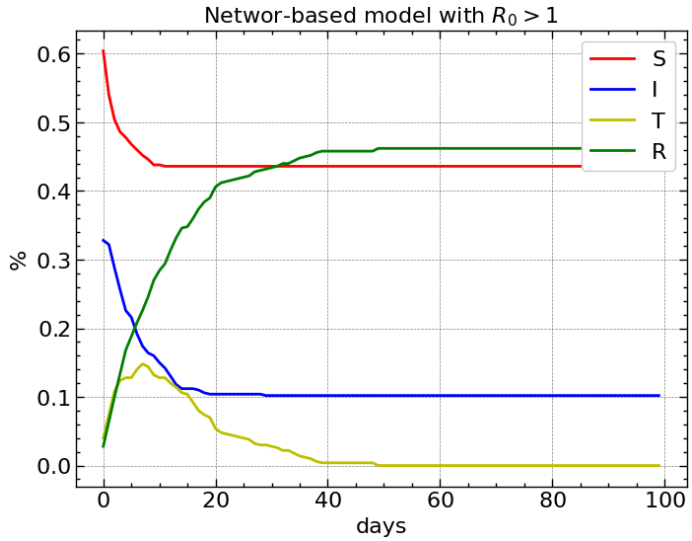

Figure 11: Networked based model with $R_0 > 1$

In Figure 12 and Figure 13 we compare the two appraoch and we can observe that the results for the scenario $R_0 > 1$ are quite similar with no strange difference, while for the case $R_0 < 1$ we observed a different behaviour of the susceptible class.

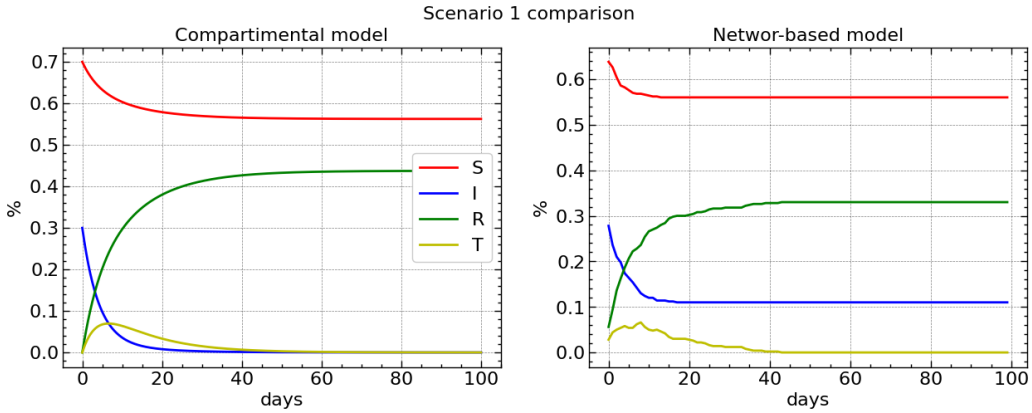

Figure 12: Comparison of the two models with scenario $R_0 > 1$

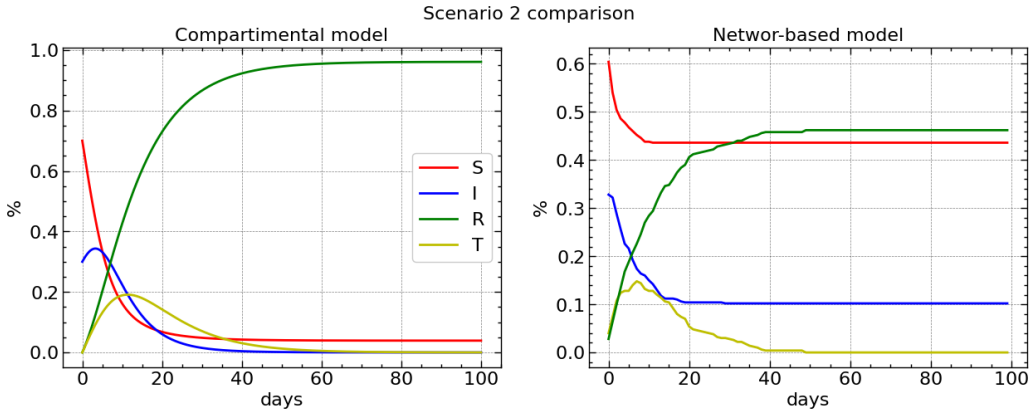

Figure 13: Comparison of the two models with scenario $R_0 < 1$

2 Homework 2

2.1 Chemical reaction

Considering a Chemical Reaction Networked assumed having a mass action kinetics, given its stochiometric matrix S we must check for the weakly reversibility of the network and the deficiency of the reaction.

The deficiency of a CNR is equal to :

$$\Delta = C - L - R \tag{23}$$

where C,L,R are respectively the number of complexes, the number of linkage classes and the reaction rank.

The reaction rank value can be easily recover by the rank of the matrix S.

To find the number of complexes we must consider the columns of the matrix S, and for each columns we associate the negative value to a reaction complex and the positive one to a product complex. After removing the repeated complexes counted we finally have the total number of complexes.

We can look at the columns of the stoichiometric matrix as a reaction and so we can pair reaction complexes to product complexes in order to have pairs representing a reaction. Then we can used these pairs to build a matrix representing the networked model of the CNR. Once find this matrix, the number of connected components is equal to the number of linkage classes.

The adjacency matrix of the networked can also be used to check if the CNR is weakly reversible computing the number of strictly connected components and if that is equal to the number of connected components then the CNR is weakly reversible.

Once we found R, L, C the deficiency is easily computed.

2.2 Traffic in a NY's district

Let consider a network representing a district of a city in which there are moving vehicles. The network shown in Figure 14 identifies nodes as point of interest and links as roads. The district has 16 roads each of them with a maximum flow capacity C_e and moreover each link has a certain length. Considering a speed of $14m/s$ we can easily recover the minimum travelling time t_e for each link.

The node-link incidence matrix B considers the relation between each node and link. So this matrix as dimension n, m where n is the number of nodes and m is the number of links and the value to the B_{ij} element is assigned the value of $+1$ if the $j-th$ link start from node i, -1 if the link is directed towards i or 0 if the node and the link are not related.

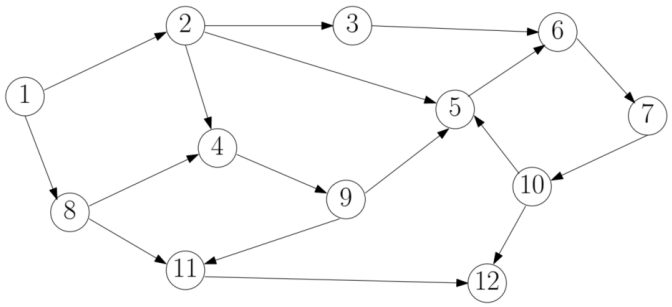

Figure 14: Network of the district

$$B = \begin{pmatrix}
1.0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1.0 & 0 & 1.0 & 1.0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1.0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1.0 & 0 & 0 & 1.0 & 0 & 0 & 0 & -1.0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1.0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & -1.0 & 0 & -1.0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1.0 & 0 & -1.0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 1.0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1.0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 1.0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.0 & 0 & 0 & 0 & 0 & 1.0 & 1.0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.0 & 0 & -1.0 & 0 & 0 & 1.0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.0 & -1.0
\end{pmatrix}$$

The shortest path, in term of time, to go from node 1 to node 12 considering an empty network pass trough nodes $2, 4, 9, 11$ with travel time of $504, 1$ seconds.

Considering the capacities associated to each link the maximum flow between the nodes 1 and 12 is $2024, 8$.

To each link is assigned a flow value and so we can recover the net exogenous flow ν thank to the relation $\nu = B\,f$, where f is the flow vector containing the flow for each link. Knowing ν the external inflow and outflow are easily computed as:

$$\lambda_i = max(\nu_i, 0) \tag{24}$$

$$\mu_i = max(-\nu_i, 0) \tag{25}$$

The length of the segments, the flow vector and the maximum capacities have been computed randomply every time we run the simulation, an example of results for inflows and outflows are shown in Table 5.

node	λ	μ	ν
1	1105.16	0	1105.16
2	1232.84	0	1232.84
3	2.87	0	2.87
4	0	587.67	- 587.67
5	0	1205.64	-1205.64
6	0	568.11	- 568.11
7	39.04	0	39.04
8	527.27	0	527.27
9	652.56	0	652.56
10	477.38	0	477.38
11	0	618.54	- 618.54
12	0	1057.14	- 1057.14

Table 5: External outflow and inflow and net flow

To compute the social optimum traffic flow value f^* we must solve the following optimization problem:

$$\begin{aligned} \underset{f}{\text{minimize}} \quad & \sum_{e \in E} c_e(f_e) \\ \text{subject to} \quad & B\,f = \nu = \lambda - \mu, \\ & 0 \leq f \leq C \end{aligned} \tag{26}$$

where C is the vector of the capacities and $c_e(f_e)$ is the cost function corresponding to:

$$c_e(f_e) = f_e \cdot d_e(fe) \tag{27}$$

with $d_e(f_e)$ the delay function.

For this example it is assumed that all the net inflows are zero except for one at node 1, where the computed value above is kept.

The delay function consider is the following:

$$d_e(f_e) = \frac{t_e}{1 - f_e/C_e} \tag{28}$$

An example of social cost computed in a simulation is equal to 795018.6, while the social optimum flow f^* computed is :

node	f^*
1	311.61
2	793.55
3	81.75
4	0
5	229.86
6	81.75
7	402.02
8	229.86
9	311.61
10	311.61
11	402.02
12	391.53
13	0
14	402.02
15	0
16	311.61
17	793.55

Table 6: Social optimum flow

To compute the Wardrop value we must change the cost function to:

$$c_e(f_e) = \int_0^{f_e} d_e(x)\, dx = -t_e\, c_e \ln(\frac{C_e - f_e}{C_e}) \tag{29}$$

The resulting Wardrop flow values f^W for the simulation are:

The Price of Anarchy is the ratio between the cost at the Wardrop equilibrium and the cost at social optimum. The computed Price of Anarchy is 1.113 and as expected is grater than zero because the cost of the Wardrop equilibrium should be higher.

In order to make the Price of Anarchy equal to 1 we can introduce tolls. The new cost becomes:

$$c_e(f_e) = \int_0^{f_e} d_e(x) + \omega_e\, dx = \tag{30}$$

where toll on link e is computed as:

$$\omega_e = \frac{d}{d\, f_e} c_e(f_e^*) - d_e(f_e^*) = f_e^*\, \frac{d}{d\, f_e} d_e(f_e^*) \tag{31}$$

node	f^W
1	173.21
2	931.95
3	0
4	0
5	173.21
6	0
7	464.75
8	173.21
9	173.21
10	173.21
11	464.75
12	467.20
13	0
14	464.75
15	0
16	173.21
17	931.95

Table 7: Wardrop optimum flow

With these tolls we can observe as the cost computed at Wardrop is now equal to the social optimum with Price of Anarchy equal to 1.

For the optimization problem we can use another different cost, where instead of considering the total delay we use the total additional delay with respect to free flow, so the cost now is:

$$c_e(f_e) = f_e(d_e(f_e) - t_e) = \frac{f_e\, t_e}{1 - f_e/C_e} - f_e\, t_e \tag{32}$$

If we want the Wardrop equilibrium coincides with the new f^* social optimum we must use tolls like:

$$\omega_e = \frac{d}{d\, f_e} c_e(f_e^*) - (d_e(f_e^*) - t_e) = f_e^*\, \frac{d}{d\, f_e} d_e(f_e^*) \tag{33}$$

Using the new tolls in a simulation we see as the Price of Anarchy result equal to 1.

2.3 Water Distribution

Let consider the model of a water system distribution shown in Figure 15, where each node is a subsystem with internal dynamics. Different subsystem are connected by pipes whose flow can be controlled. For some nodes the network has a constant demand d.

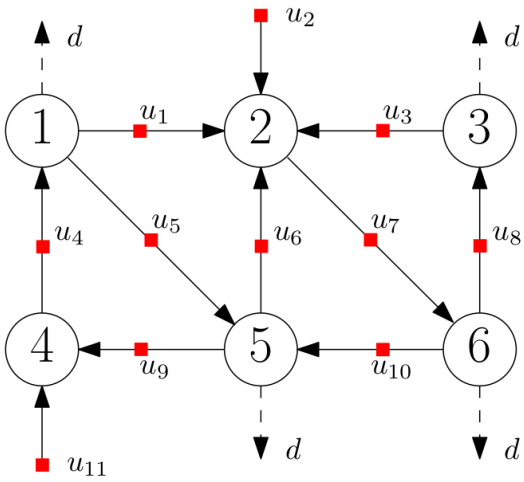

Figure 15: Water system network topology

Each node has a double integrator dynamics given by:

$$A_i = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad B_{ij} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad i \in \{1, \dots, 6\}, j \in \{1, \dots, 11\} \tag{34}$$

The complete system has form:

$$\dot{x}(t) = A\,x(t) + B\,u(t) + Ed(t) \tag{35}$$

where $A = blockdiag\{A_i\}$, $d = 1$.

The matrix B and E depend on the network topology. In particular to build the matrix B we must look at each link of the network: based on the direction of the link we assign $\pm B_{ij}$ while if the node and the link are not related we put a zero-block matrix. In this scenario the matrix B assumes form:

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

where $B_n = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

In similar way we can recover the E matrix considering in which nodes there is the external fixed demand. In the scenario we are considering these nodes are $1, 3, 5, 6$ so the matrix results in:

$$E = - \begin{pmatrix} B_n \\ 0 \\ B_n \\ 0 \\ B_n \\ B_n \end{pmatrix} \tag{36}$$

According to [1] given a marginal stable system like this one is possible to find a matrix S, positive define, with the same block-diagonal structure as A, which satisfy the linear matrix inequalitiy (LMI) condition:

$$S\,A^T + AS - 2\,\gamma\,B\,B^T < 0 \tag{37}$$

then there exist a stabilizing decentralized control law.

The control law decentralized and stabilizing has for $u = -K\,x$, where $K = \gamma\,B^T\,P$ and $P = S^{-1}$.

We then simulate the system using the controlled law computed numerically. Results are shown in Figure 16.

The initial conditions of the states are all zeros and we can see as there are two group of states: the odd states are asymptotically constant around values -20 and -40, while the even states recover the initial value of zero asymptotically.

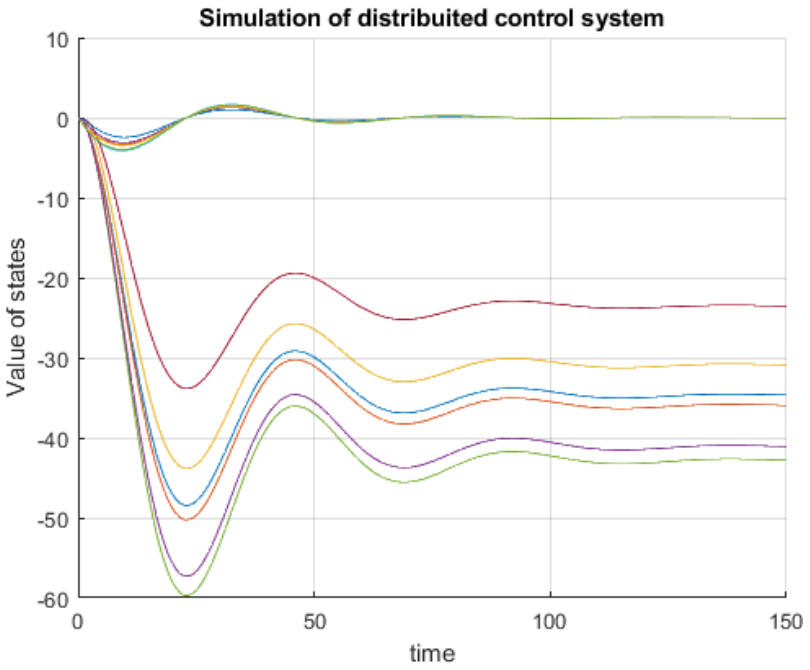

Figure 16: Simulation of the water distribution system

References

[1] F. Blanchini, E. Franco and G. Giordano, "Structured-LMI conditions for stabilizing network-decentralized control," 52nd IEEE Conference on Decision and Control, Firenze, Italy, 2013, pp. 6880-6885, doi: 10.1109/CDC.2013.6760979.