Project Assignment 2

Dynamics in Systems and Networks: Network Dynamics
University of Trento
Spring 2023

Gianni Lunardi and **Giulia Giordano**

April 21, 2023

- You can choose to solve the problems with any software (Matlab, Mathematica, Python or others) but note that only Matlab is "supported".
- Please submit your Assignment as a **typeset PDF file**.
- When something has to be **computed analytically**, only use pen and paper. **All answers should be clearly motivated:** providing end results of calculations only is not sufficient.
- Please also send your **code** written to solve the Problems.
  Comment your code well. Clarity is more important than efficiency.
  Your code should be written in a general way: if a question is slightly modified, getting the new correct answer should only require slight modifications in your code as well.
  All answers should be clearly motivated: providing end results of calculations only is not sufficient.
- Submit your report (and the code) via e-mail to `gianni.lunardi@unitn.it` (and to `giulia.giordano@unitn.it` in cc).
- **Submission deadline: the day before you wish to take the exam!**.

1 Zero Deficiency, or Non-Zero Deficiency?

Write a Matlab code that, given a Chemical Reaction Network (CRN) assumed to have reaction kinetics of the mass action type, checks whether the CRN is weakly reversible and computes the number of complexes, the number of linkage classes, the reaction rank, and finally the deficiency of the CRN.

All the information about the CRN network is provided by supplying to the program the corresponding stochiometric matrix S.

2 Traffic in a NY's district

Consider a traffic problem, related to vehicles moving in a district of New York city. The network is represented in Figure 1: the points of interest (e.g. shops, companies ecc.) are identified by nodes, while the links are the roads in which the vehicle can travel.

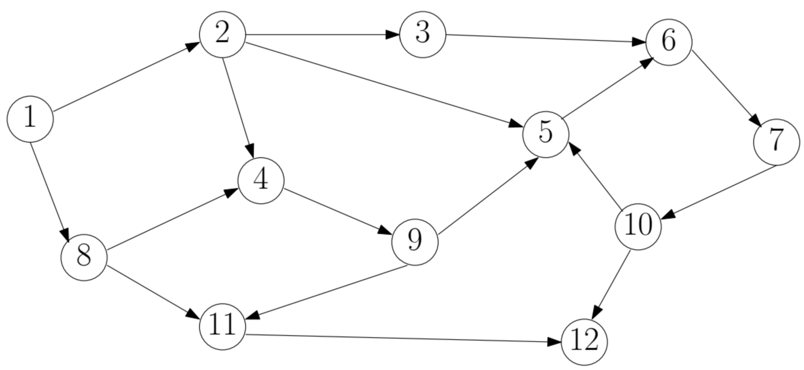

Figure 1: Road graph of a NY's district

Each link l_e, where $e = 1, \ldots, 16$, has a maximum flow capacity C_e. Moreover, each link has a minimum travelling time t_e, which the drivers experience when the road is empty. The quantities can be computed randomly in Matlab in the following way:

```
seg_length = 2000*rand(m,1);          % length of the road segment's in m
flw = 600 + 100*(2*rand(m,1) - 1);    % flow vector
C = 1000 + 100*(2*rand(m,1) - 1);     % max capacity
```

where $m = 16$. The minimum travel times can be retrieved by dividing the length of road's segment by the assumed speed of $14\,m/s$. To obtain the correct results, **be sure** to activate the seed for random numbers at the start of the Matlab file (for example `rng('deafult')`). For each link, we introduce the delay function:

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

a) Compute the node-link incidence matrix B.

b) Find the shortest path between node 1 and 12, with respect to travelling time in an empty network.

c) Find the maximum flow between node 1 and 12.

d) For the given flow vector, compute the external inflow/outflow at each node.

For the following subproblems you can use CVX, a Matlab-based convex optimization tool `cvxr.com/cvx/download`. Make sure you add the cvx-package to your path in MATLAB. The flow optimization problem:

$$\text{minimize} \sum_{e=1}^{M} f_e^2 \tag{2a}$$

$$\text{subject to } Bf = \lambda - \mu \tag{2b}$$

$$0 \le f \le C \tag{2c}$$

can be written for CVX in Matlab as:

```
cvx_begin
    variable f(M)
    minimize  sum(f.*f)
    subject to
            B*f == lambda - mu
            0 ≤ f ≤ c
cvx_end
```

Consult the CVX Users' Guide online for help if needed. For the following points, we assume that all net inflows are zero except for the one at node 1, where we keep the computed one from part **d)**. We also assume that all of the net inflow at node 1 leaves the network at node 12.

e) Using CVX, find the social optimum f^* with respect to the delays, i.e, minimize:

$$\sum_{e \in \mathcal{E}} f_e d_e(f_e) = \sum_{e \in \mathcal{E}} \frac{f_e t_e}{1 - f_e/C_e} = \sum_{e \in \mathcal{E}} \frac{t_e C_e}{1 - f_e/C_e} - t_e C_e \tag{3}$$

subject to the constraints on the flows. (Hint: Use the CVX-function `inv_pos`)

f) Using CVX, find the Wardrop equilibrium f^W.
(Hint: Use the cost function $\sum_{e \in \mathcal{E}} \int_0^{f_e} d_e(x)dx$.)

g) Compute the Price of Anarchy associated to the Wardrop equilibrium f^W.

h) Introduce tolls, such that the toll on link e is $\omega_e = f_e^* d_e'(f_e^*)$, where f_e^* is the flow at the system optimum. Now the delay on link e is given by $d_e(f_e) + \omega_e$. Use CVX to compute the new Wardrop equilibrium. What do you observe?

i) Instead of the total delay, let the cost be the total additional delay with respect to free flow, i.e.,

$$c_e(f_e) = f_e(d_e(f_e) - t_e).$$

Compute the system optimum f^* for the costs above and construct tolls such that the Wardrop equilibrium coincides with f^*. Verify your result with CVX.

3 Network-Decentralized Control Strategies

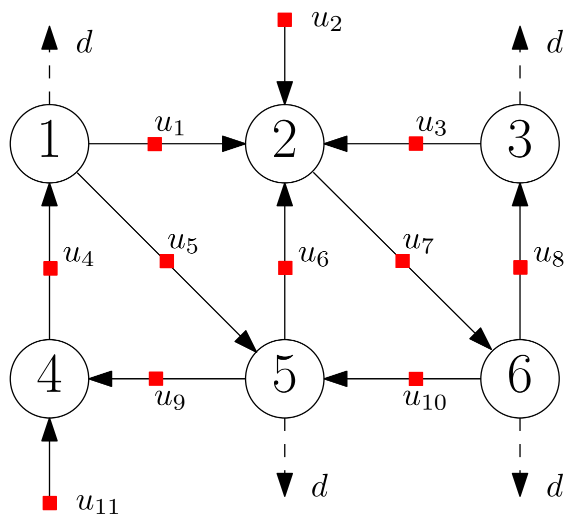

Figure 2: Network of a water distribution system

Consider the model of a water distribution system, shown in Fig. 2, where each node represents a subsystem with its internal dynamics. Different subsystems are connected by pipes whose flow u_j can be controlled. The network has a constant demand d (external signal that acts at the **same level** of the inputs) on nodes $1, 3, 5, 6$. Each node of the system has the following double-integrator dynamics:

$$A_i = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B_{ij} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \text{for } i \in \{1, \dots, 6\},\ j \in \{1, \dots, 11\} \tag{4}$$

Then, the overall system has the form:

$$\dot{x}(t) = Ax(t) + Bu(t) + Ed(t) \tag{5}$$

where $A = \text{blockdiag}\{A_i\}$, $d = 1$, while B and E are block-structured matrices that depend on the network's topology.

a) Compute the input matrix B and the external input matrix E;

b) Can we stabilise the system with a network-decentralised control[1] $u = -Kx$, where the feedback matrix K has the same structural zero blocks as $B^\top$? Why? Write the block structure of such a matrix K;

c) Compute a suitable network-decentralised feedback matrix K numerically, using the MATLAB LMI toolbox[2];

d) What it the difference between the evolution of the different states? Why?

[1] See the papers: F. Blanchini, E. Franco, G. Giordano, "Structured-LMI conditions for stabilizing network-decentralized control", Proceedings of the 52nd IEEE Conference on Decision and Control, pp. 6880-6885, Firenze, Italy, December 2013; F. Blanchini, E. Franco, G. Giordano, "Network-decentralized control strategies for stabilization", IEEE Transactions on Automatic Control, 60 (2), pp. 491-496, 2015.

[2] To enforce a certain speed of convergence, in the LMI A can be replaced with $A + \sigma I$, $\sigma > 0$, so that the closed loop eigenvalues have real part less than $-\sigma$. You can use $\sigma = 0.15$