Homework II

- Solutions have to be uploaded on the course website by h 23:59 of Sunday December 12, 2021 under the name Homework2.
- You may implement your solutions in any language you see fit, but the teacher assistant can only guarantee you support with MATLAB and Python. Your code should be written in a quite general manner, i.e., if a question is slightly modified, it should only require slight modifications in your code as well. Upload a PDF lab-report together with your code.
- The PDF should read like a standard lab-report, including a description of what you are doing, and proper presentation of results (including readable figures with axis labels, if any). Writing the report in Latex is strongly encouraged. Clarity of the presentation (especially if the report is hand-written) and ability to synthesize are part of the evaluation of the homework.
- Comment your code well. Clarity is more important than efficiency.
- Collaboration such as exchange of ideas among students is encouraged. However, every student has to submit her/his own report (in PDF format) and code, and specify whom she/he has collaborated with and on what particular part of the work.

Problem 1. The first part of this assignment consists in studying a single particle performing a continuous-time random walk in the network described by the graph in Fig. 1 and with the following transition rate matrix.

$$\Lambda = \begin{array}{c} \\ \\ \\ \\ \\ \end{array}\begin{array}{cc} \begin{array}{ccccc} o & a & b & c & d \end{array} & \\ \begin{pmatrix} 0 & 2/5 & 1/5 & 0 & 0 \\ 0 & 0 & 3/4 & 1/4 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/3 & 0 & 2/3 \\ 0 & 1/3 & 0 & 1/3 & 0 \end{pmatrix} & \begin{array}{c} o \\ a \\ b \\ c \\ d \end{array} \end{array} . \qquad (1)$$

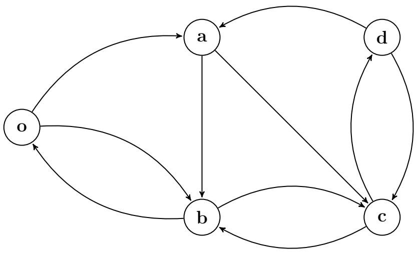

Figure 1: Closed network in which particles move according to the transition rate matrix (1).

Your task is to simulate the particle moving around in the network in continuous time according to the transition rate matrix (1). To help you with this we have provided some hints below. You should then use these simulations to answer the following questions:

a) *What is, according to the simulations, the average time it takes a particle that starts in node a to leave the node and then return to it?*

b) *How does the result in a) compare to the theoretical return-time $\mathbb{E}_a[T_a^+]$? (Include a description of how this is computed.)*

c) *What is, according to the simulations, the average time it takes to move from node o to node d?*

d) *How does the result in c) compare to the theoretical hitting-time $\mathbb{E}_o[T_d]$? (Describe also how this is computed.)*

Hint 1: To simulate a rate-r Poisson process, one can simulate the time t_{next} between two ticks of the Poisson clock, which is a random variable with rate-r exponential distribution. In fact, if we draw a uniformly distributed random variable $u \sim \mathcal{U}(0,1)$, we can compute t_{next} as

$$t_{\text{next}} = \frac{-\ln(u)}{r}.$$

It might also prove useful to recall that an exponential random variable is memoryless:

$$\mathbb{P}(X \geq t+s \mid X \geq t) = \frac{\mathbb{P}(X \geq t+s)}{\mathbb{P}(X \geq t)} = \frac{e^{-r(t+s)}}{e^{-rt}} = e^{-rs} = \mathbb{P}(X \geq s).$$

Hint 2: Let's say that you wish to simulate the outcome of 6-sided dice roll, but where the dice has only 3 numbers $i = 1, 2, 3$, with **probability distribution** $p = (\frac{1}{6}, \frac{3}{6}, \frac{2}{6})$. The outcome could be simulated by drawing a uniform random variable $u \sim \mathcal{U}(0,1)$. You can then use the **cumulative probability distribution** of u:

$$\mathbb{F} = \left(\sum_{i \leq 1} p_i, \ \sum_{i \leq 2} p_i, \ \sum_{i \leq 3} p_i \right) = \left(\frac{1}{6}, \frac{4}{6}, \frac{6}{6} \right)$$

and choose the number $i = 1, 2$ or 3 for which $\mathbb{F}_{i-1} < u \leq \mathbb{F}_i$. You can use a similar method but with $p = (P_{i1}, P_{i2}, \dots, P_{in})$, i.e., row i in the probability matrix P for the discrete-time Markov chain, to randomly select which node the particle should visit next, if the particle currently is in node i.

The next part of the exercise focuses on opinion dynamics.

e) *Interpret the matrix Λ as the weight matrix of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \Lambda)$, and simulate the French-DeGroot dynamics on $\mathcal{G}$ with an arbitrary initial condition $x(0)$. Does the dynamics converge to a consensus state? Motivate your answer.*

f) *Assume that the initial state of the dynamics for each node $i \in \mathcal{V}$ is given by $x_i(0) = \xi_i$, where $\{\xi_i\}_{i \in \mathcal{V}}$ are i.i.d random variables with variance σ^2. Compute the variance of the consensus value, and compare your results with numerical simulations.*

g) *Remove the edges (d, a) and (d, c). Describe and motivate the asymptotic behaviour of the dynamics. If the dynamics converges to a consensus state, how is the consensus value related to the initial condition $x(0)$? Assume that the initial state of the dynamics for each node $i \in \mathcal{V}$ is given by $x_i(0) = \xi_i$, where $\{\xi_i\}_{i \in \mathcal{V}}$ are i.i.d random variables with variance σ^2. Compute the variance of the consensus value. Motivate your answer.*

h) *Consider the graph $(\mathcal{V}, \mathcal{E}, \Lambda)$, and remove the edges (c, b) and (d, a). Analyse the French-DeGroot dynamics on the new graph. In particular, describe and motivate the asymptotic behaviour of the dynamics in terms of the initial condition $x(0)$.*

Problem 2. In this part we will again consider the network of Fig. 1, with weights according to (1). However, now we will simulate many particles moving around in the network in continuous time. Each of the particles in the network will move around just as the single particle moved around in Problem 1: the time it will stay in a node is exponentially distributed, and on average it will stay $1/w_i$ time-units in a node i before moving to one of its outneighbors. The next node it will visit is based on the probability matrix $P = diag(\omega)^{-1}\Lambda$, where $\omega = \Lambda\mathbb{1}$.

Your task is to simulate this system from two different perspectives: the *particle perspective*, i.e. "follow the particle", and the *node perspective*, i.e. "observe from the node".

Simulating the system from a particle perspective is exactly as in Problem 1, but here you have to follow many particles instead. To simulate the system from the node perspective you instead have to observe the particles from the node. When doing this you do not have to care about each single particle, but only about the number of particles in the node. Note that at node i, each particle in that node will stay there on average $1/\omega_i$ time units. Therefore, the node will pass along particles at a rate proportional to the number of particles in the node. In fact, if at time t the number of particles in node i is $n_i(t)$, it will pass along particles at a rate of $n_i(t)\omega_i$. The departure times of the node can thus be seen as a Poisson process with rate $n_i(t)\omega_i$. At each tick of the Poisson clock of the node, it will move a particle to a neighboring node. The node to which the particle will move is again based on the normalized transition rate matrix P.

Simulate the system from the two perspectives, and then answer the following questions (we have again provided a hint below):

a) Particle perspective:

- *If 100 particles all start in node a, what is the average time for a particle to return to node a?*
- *How does this compare to the answer in Problem 1, why?*

b) Node perspective:

- *If 100 particles start in node o, and the system is simulated for 60 time units, what is the average number of particles in the different nodes at the end of the simulation?*
- *Illustrate the simulation above with a plot showing the number of particles in each node during the simulation time.*
- *Compare the simulation result in the first point above with the stationary distribution of the continuous-time random walk followed by the single particles.*

Hint: to simulate the particle perspective we could either attach a Poisson clock to each of the particles and move them individually, just as in Problem 1. Alternatively, we could have a single, system-wide Poisson clock with rate 100. Then, at every tick of the system-wide clock we randomly select which particle to move. This particle is then moved to a neighbor node (which can be itself) based on the transition probability matrix $\overline{P}$ (according to the notation used in the lecture notes). This means, that if a particle in node o is selected, there is a probability of $1 - \omega_o = 1 - 3/5 = 2/5$ that it will stay in node o, and a probability of $\omega_o = 3/5$ that it will leave the node.

For the node perspective we could have one Poisson clock attached to each node. But since the rate of this is proportional to the number of particles in the node, it would have to change rate during the simulation. An alternative way is to again have a system-wide Poisson clock with rate 100. Then, at each tick we randomly, and proportionally to the number of particles in the different nodes, select a node from which we should move a particle. Then a particle from the selected node will move according to the transition probability matrix $\overline{P}$.

Problem 3. In this part we study how different particles affect each other when moving around in a network in continuous time. We consider the open network of Figure 2, with transition rate matrix Λ_{open} according to (2).

$$\Lambda_{\text{open}} = \begin{array}{c} \\ \\ \\ \\ \\ \end{array}\begin{array}{cc} \begin{array}{ccccc} o & a & b & c & d \end{array} & \\ \begin{pmatrix} 0 & 2/3 & 1/3 & 0 & 0 \\ 0 & 0 & 1/4 & 1/4 & 2/4 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} & \begin{array}{c} o \\ a \\ b \\ c \\ d \end{array} \end{array} \tag{2}$$

For this system, particles will enter the system at node o according to a Poisson process with rate $\lambda = 1$. Each node will then pass along a particle according to a given rate, similar to what you did in Problem 2 with the "node perspective".

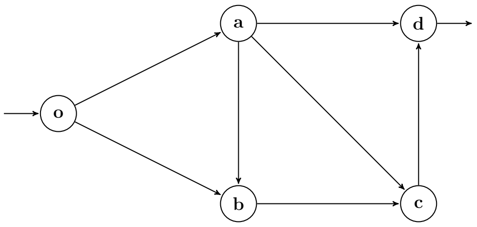

Figure 2: Open network.

You will simulate two different scenarios that differ by what rate the nodes will pass along particles: i) *proportional rate*, and ii) *fixed rate.* In scenario i), each node will pass along particles according to a Poisson process with rate equal to the number of particles in the node. In scenario ii), each node will instead pass along particles with a fixed rate of 1.

Note that since node d does not have a node to send its particles to, when the Poisson clock ticks for this node you could simply decrease the number of particles in the node by one (if there are any particles in the node). Equivalently think of another node d' connected to node d, such that at every tick of the Poisson clock of d, it sends a particle to node d'.

The goal is to simulate the two systems and answer the following questions:

a) Proportional rate:

The rate of the Poisson clock of each node is equal to the number of particles in it.

- *Simulate the system for 60 time units and plot the evolution of the number of particles in each node over time.*
- *What is the largest input rate that the system can handle without blowing up?*

b) Fixed rate:

The rate of the Poisson clock of each node is fixed, and equal to one.

- *Simulate the system for 60 time units and plot the evolution of number of particles in each node over time.*
- *What is the largest input rate that the system can handle without blowing up? Why is this different from the other case?*