Homework 2

Giuseppe Galilei s295620

12 December 2021

0 Preface

The following homework is submitted alongside three Jupyter notebooks, one for each exercise.

1 Problem 1

We are given the Graph G in figure 1 along with the transition rate matrix Λ, we are asked to simulate the behaviour of a single particle moving around the network as a random walk in continuous time. Questions A and C are solved using the same function $hitting\_time\_compute$, which computes hitting time in a network starting from a certain node A and ending in a node B. The same code can be reused, with minor adjustments and a flag $returnTime\_compute$, to compute return times (i.e. hitting times where node A and B coincide). Particles move following a random walk with global Poisson clock, where

$$t_{next} = -\frac{\log(u)}{w^*}$$

with u a uniformly distributed random variable in $[0, 1)$ and $\omega^* = \max_i \omega_i, \omega = \Lambda \mathbb{1}$. At each tick of the clock the particle chooses the node to move to, based on the transition probability matrix, until the chosen node is the destination one, then the elapsed time is returned.

1.1 A

To answer this question the function $hitting\_time\_compute$ is called with $pos\_start = 'a'$, $pos\_dest\_array = ['a']$ and the flag $returnTime\_compute = True$. The return time is computed and averaged over 100.000 iterations, obtaining as result an average return time of 6.74.

1.2 B

To compute theoretical return time, the following formula is used:

$$\mathbb{E}_i\left[\bar{T}_i^+\right] = \frac{1}{\omega_i \bar{\pi}_i}, \quad i \in \mathcal{X}$$

where $\bar{T}_i^+ = \inf\{t \geq 0 : X(t) = i \text{ and } X(s) \neq i \text{ for some } s \in (0,t)\}$

The graph G is strongly connected, thus the usage of such formula is justified by $Theorem 7.2$ from the lecture notes.

The theoretical return time computed is 6.75, almost equal to the iterative one.

1.3 C

To answer this question the function $hitting\_time\_compute$ is called with $pos\_start = 'o'$, $pos\_dest\_array = ['d']$ and the flag $returnTime\_compute = False$. The hitting time is computed and averaged over 100.000 iterations, obtaining as result an average hitting time of 8.76.

1.4 D

If $\mathcal{S} \subseteq \mathcal{X}$ is a subset of states that is globally reachable in G_Λ, then the expected hitting times

$$\bar{\tau}_i^S = \mathbb{E}_i[T_S], \quad i \in \mathcal{X}$$

are the unique solutions of

$$\bar{\tau}_s^S = 0, \quad s \in \mathcal{S}, \quad \bar{\tau}_i^S = \frac{1}{\omega_i} + \sum_{j \in \mathcal{X}} P_{ij} \bar{\tau}_j^S, \quad i \in \mathcal{X} \backslash \mathcal{S}$$

With P being the normalized transition matrix and Q the portion of P related to nodes not in the "hitting set", in our case composed only of the node d. Thus the latter equation can be seen as

$$\bar{\tau}^S = \frac{1}{\omega} + Q\bar{\tau}^S$$

Then

$$\bar{\tau}^S = (I - Q)^{-1} \frac{1}{\omega}$$

It is obtained that $\bar{\tau}_o^S$, the theoretical hitting time from node o to d, is 8.79

1.5 E

Considering the matrix Λ as the weight matrix for G (figure 1),we are asked to simulate the French-DeGroot dynamics on an arbitrary initial condition.

After 100 iterations of the dynamics, a consensus is obtained. Such result is justified by $Corollary 5.1$ from the lecture notes, which states that:

Let G be a graph such that $s_g = 1$ and the connected component of $\mathcal{G}$ corresponding to the unique sink of its condensation graph is aperiodic. Let P be the normalized weight matrix of G and let $\pi = P'\pi$ be its unique invariant probability. Then, the averaging dynamics satisfies

$$\lim_{t \to +\infty} x(t) = \alpha \mathbb{1}, \quad \alpha = \pi' x(0)$$

The graph G satisfies the requirements of the corollary, being strongly connected and aperiodic, as a further proof it is shown in the notebook that the consensus value determined using the corollary coincides with simulation results.

1.6 F

As random initial condition $x(0)$, five values are sampled from a normal distribution, thus with a variance $\sigma^2 = 1$ The variance of the consensus state can be computed as

$$\sigma_{\bar{x}}^2 = \sigma^2 \sum_i \pi_i^2$$

Using this formula, a variance of 0.213 is obtained. Simulating the evolution of the system, we obtain a variance of 0.209, almost equal.

1.7 G

Now, the graph G' is considered (figure 2). In this case, node D constitutes the only sink component for G', this means that the system will reach a consensus state which coincides to the initial one for node D. (In order to be able to simulate the system, a self loop has been added on node D). Such behaviour is also expected by looking at the invariant distribution $\pi = [0, 0, 0, 0, 1]$ for nodes o, a, b, c, d.

From simulation the predictions are confirmed.

In this case the computed variance for the consensus state coincides with the initial variance of node states, this result can be explained theoretically by looking at the distribution π and at the formula used to compute variances introduced in the previous section. Indeed, in this case we end up with

$$\sigma_{\bar{x}}^2 = \sigma^2$$

1.8 H

Now, the graph G'' is considered (figure 3).

This time nodes C and D constitute the unique sink component, such component is connected but not aperiodic. The invariant distribution $\pi = [0, 0, 0, 0.5, 0.5]$ for nodes o, a, b, c, d. This means that the evolution of opinions in the system will depend only on opinions of nodes C and D, but an equilibrium will not be reached.

The expected behaviour is confirmed in simulation.

2 Problem 2

The considered network is graph G (figure 1), with transition rate matrix Λ.

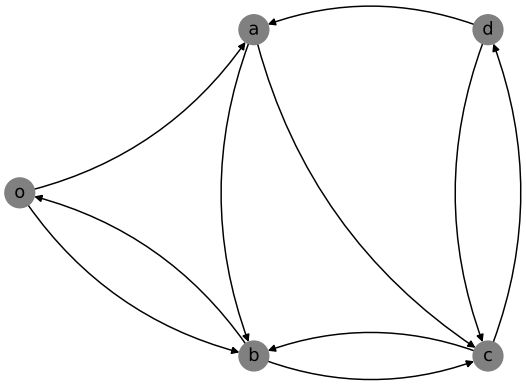

Figure 1: Graph G for Problem 1

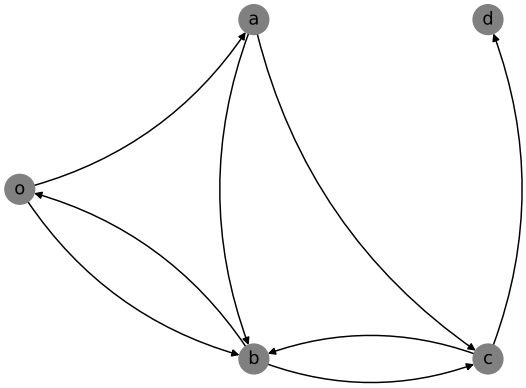

Figure 2: Graph G' for Problem 1, question g

2.1 Particle perspective

A particle at a time (chosen uniformly among particles) moves at the tick of a global Poisson clock of rate equal to the number of particles, 100. The node to which the particle will move to, is chosen accordingly to the transition probability matrix. The function used to compute average return times over all particles is $returnTime\_NparticlePersp$, it resembles the function adopted in problem 1. Over 1000 iterations it is obtained an average return time of 6.78, which is comparable to the theoretical average, 6.75, computed for the single particle in problem 1, this can be explained by the fact that particles do not interact with each other.

2.2 Node perspective

2.2.1 Question 1

Here the simulation happens from the node perspective, we are interested in the number of particles on each node. A node at a time (chosen proportionally to the number of particles on it) passes a particle at the tick of a global Poisson clock of rate equal to the number of particles, 100. The destination node of the particle is chosen accordingly to the transition probability matrix. The simulation starts with 100 particles on node o and has a duration of 60 time units, the function used is $particlePlacement\_nodePersp$. Over 100 simulations it is obtained an average particle placement of $[18.3, 15.4, 22.3, 21.1, 22.8]$ for nodes $[o, a, b, c, d]$.

2.2.2 Question 2

A plot of the evolution of particle placements in a single simulation is shown in figure 4, in this case the final placement is $[16, 17, 25, 17, 25]$.

2.2.3 Question 3

The average particle placement, obtained over 100 simulations in question 1, looks similar to the stationary probability vector $\bar{\pi} = [0.19, 0.15, 0.22, 0.22, 0.22]$. The difference in absolute value, computed with all decimal digits at disposal is $[0.18, 0.62, 0.11, 1.11, 0.56]$. Such resemblance is explained by the equivalent of the *ergodic theorem* (*Theorem* 7.2 point *iii*, from the lecture notes) for continuous-time Markov chains, which states that: being the graph G strongly connected, it holds that for every given observable function $f : \mathcal{X} \to \mathbb{R}$, one has that

$$\lim_{t \to +\infty} \frac{1}{t} \int_0^t f(X(s))\mathrm{d}s = \sum_{i \in \mathcal{X}} \bar{\pi}_i f(i)$$

with probability 1, where $\bar{\pi}_i(t) = \mathbb{P}(X(t) = i)$ is the stationary probability vector. This means that the empirical frequency of visits of a Markov chain X(t) in any node i converges to the stationary probability $\bar{\pi}_i$ of that node.

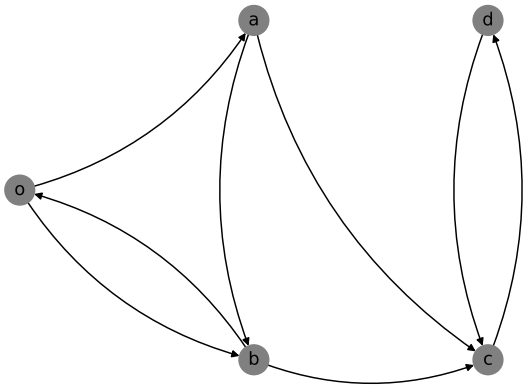

Figure 3: Graph G'' for Problem 1, question h

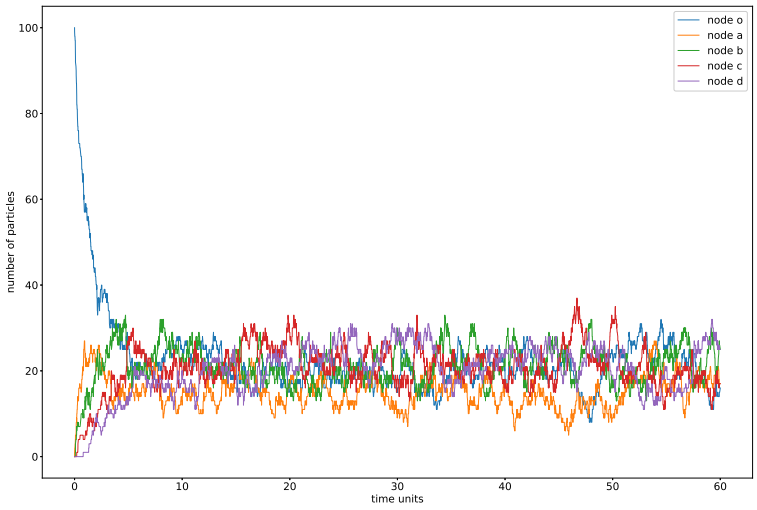

Figure 4: Evolution of particle placements in a single node perspective simulation

3 Problem 3

The graph G''' is considered (figure 5), along with the transition matrix Λ_{open}. It is asked to simulate the behaviour of a system having a fixed rate 1 to regulate the entrance of particles from node o, and a fixed/proportional rate to regulate the movement of particles in the network. Particles which move from node d will exit the network. To model such situation it has been chosen to use two different clocks:

- A clock for the entrance of particles
- A global clock for moving particles already in the network

The ticks from the two clocks are compared:

- if entrance clock ticks first, a particle enters the system and is placed on node o
- if moving clock ticks first, a node is chosen with a probability which is uniform in the "fixed rate" case or proportional to the number of particles on it, in the "proportional rate" case. Then a particle is moved to another node (or exits the system in case we are on node d) which is chosen accordingly to the transition probability matrix.

3.1 Proportional rate

The function $particlePlacement\_nodePersp\_prop$ is used.

- The entrance ticks are computed as $t_{next} = -\frac{\log(u)}{rate_{entrance}}$
- The moving ticks are computed as $t_{next} = -\frac{\log(u)}{n_{particles}}$

with u being a uniformly distributed random variable in [0,1), $rate_{entrance}$ a parameter which assumes value 1 for the first question and then will be changed to evaluate blow-up conditions, $n_{particles}$ the number of particles in the network at the moment each t_{next} is computed.

3.1.1 Question 1

The evolution of the system, for $rate_{entrance} = 1$ and 60 time units available is displayed in figure 6.

3.1.2 Question 2

To evaluate "blow-up" conditions, we can look at figure 7. From the simulations appears that "blow-up" doesn't happen, even for high input rates, this can be explained by the "proportional" nature of the moving rate i.e. the more the particles in the network, the more the moving rate increases and allows the network to manage them.

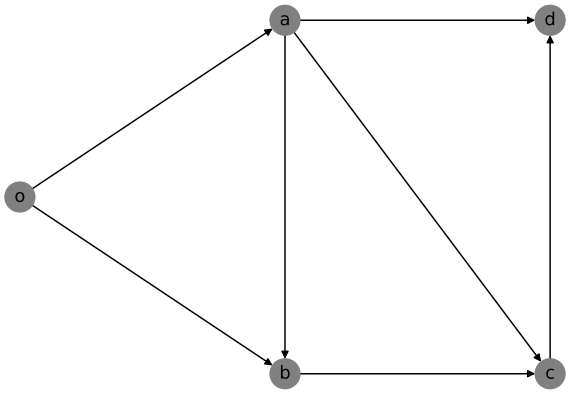

Figure 5: Graph G"'

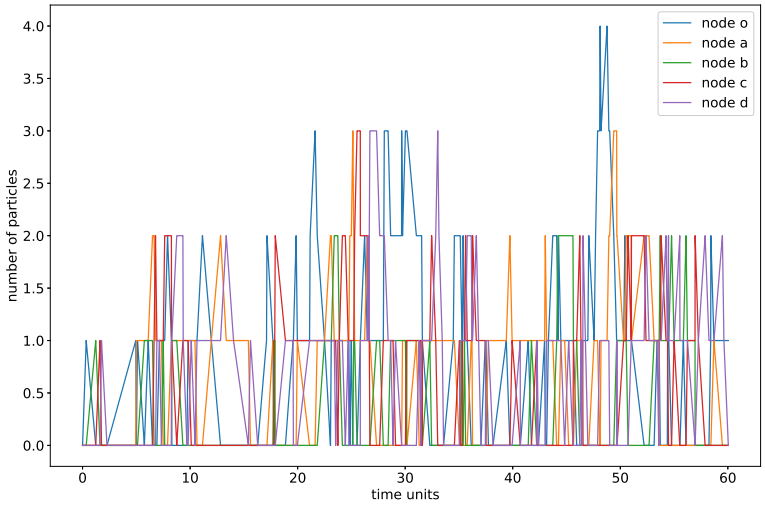

Figure 6: proportional rate evolution with entrance rate 1

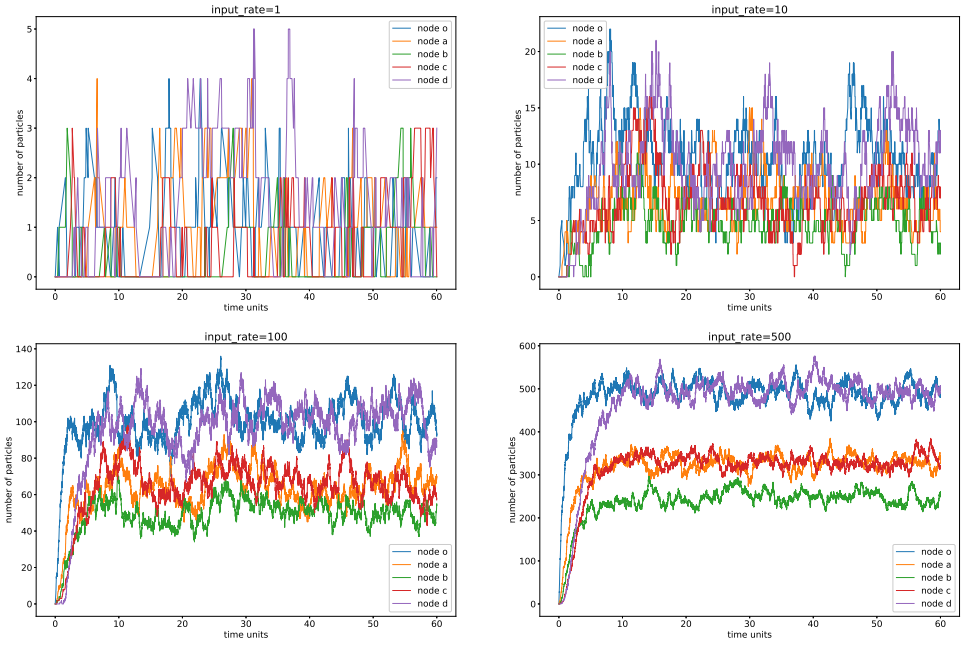

Figure 7: Blow-up evaluation for proportional rate

3.2 Fixed rate

The function $particlePlacement\_nodePersp\_fixed$ is used.

- The entrance ticks are computed as $t_{next} = -\frac{\log(u)}{rate_{entrance}}$
- The moving ticks are computed as $t_{next} = -\frac{\log(u)}{n_{nodes}}$

with u being a uniformly distributed random variable in [0,1), $rate_{entrance}$ a parameter which assumes value 1 for the first question and then will be changed to evaluate blow-up conditions.

3.2.1 Question 1

The evolution of the system, for $rate_{entrance} = 1$ and 60 time units available is displayed in figure 8.

3.2.2 Question 2

To evaluate "blow-up" conditions, we can look at figure 9. It is evident that the "blow-up" happens with very low values of entrance rate, indeed it can be seen that, with a entrance rate of 1, the network is not able anymore to handle and reduce the number of particles on node o, which will keep increasing. This happens because of the fixed rate in particle moving, which doesn't allow the system to effectively respond to increases in entrance rate.

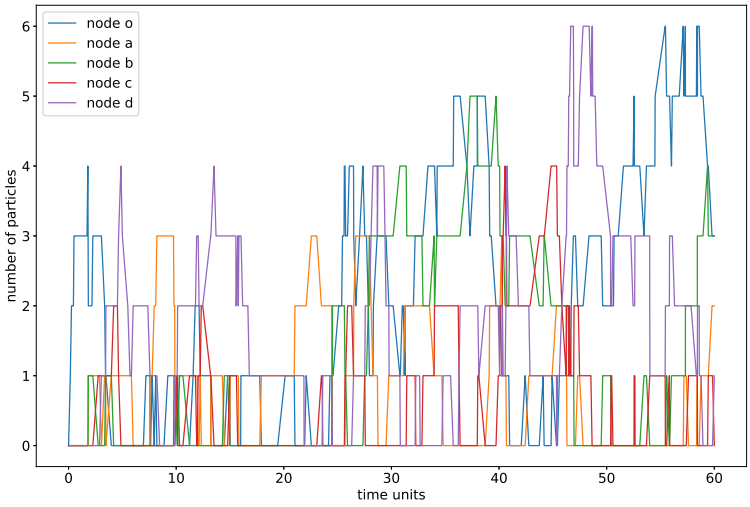

Figure 8: fixed rate evolution with entrance rate 1

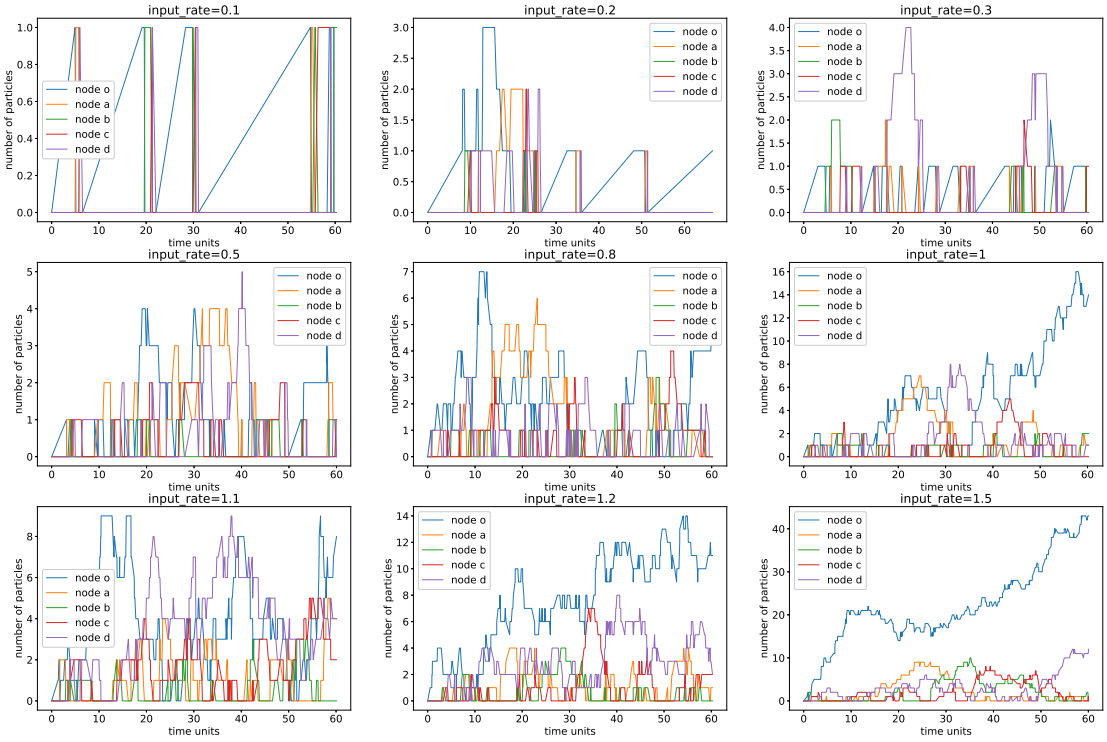

Figure 9: Blow-up evaluation for fixed rate