Network Dynamics and Learning Homework 2

Giovanni Sciortino, Giuseppe Suriano

s302959@studenti.polito.it, s296605@studenti.polito.it

March 10, 2023

The code of the homework can be found in the following link here

1 Exercise 1

In this exercise, we have to study a single particle performing a continuous-time random walk in the network described by the graph in Figure 1 and by the transition rate matrix Λ given in the assignment.

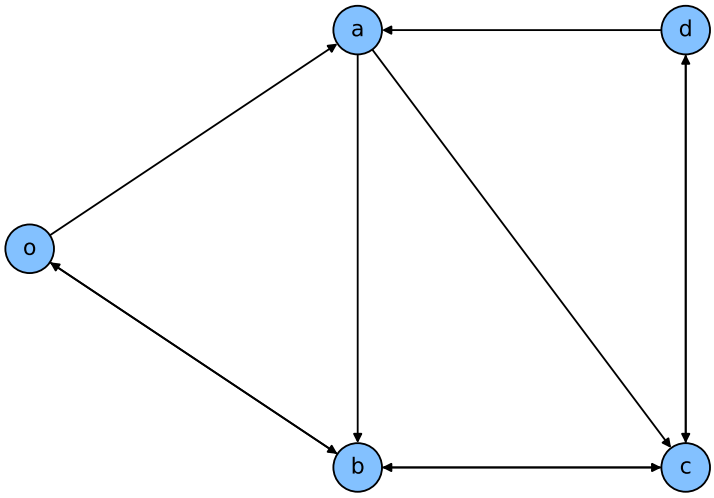

Figure 1: Graph Exercise 1.

1.1 Point a

According to the simulations, the average time it takes a particle that starts in node a to leave the node and return to it with time $E_a\left[T_a^+\right]$ is equal to 6.73. This is done by simulating a random walk, for each node v waiting $X \sim \exp(\text{outdeg}(v))$ time and then moving to an edge u with probability $P_{v,u}$, ending when $u = a$. This estimation was performed on a sample of $10,000$ simulations.

1.2 Point b

The second question is about comparing the previous result with the theoretical return time. From the theory we know that the theoretical return time is equal to:

$$E_a\left[T_a^+\right] = \frac{1}{w_a \bar{\pi}_a}$$

In our case the invariant distribution computed is equal to:

$$[0.18518519 \quad 0.14814815 \quad 0.22222222 \quad 0.22222222 \quad 0.22222222] \ .$$

So considering the formula, $\bar{\pi}_a$ node a component of the invariant distribution and w_a is the degree of node a.

1.3 Point c

From point a we defined a function that records for each simulation(the number of simulation is the same of point a) the hitting times specifying the origin and the destination nodes. So we decide to exploit the same code but specifying different source and destination node(previously both set to a, and now from o to d. The estimation is $E_o\left[T_d\right] = 8.77$.

1.4 Point d

The expected hitting times $\{E_i\left[T_{\mathcal{S}}\right]\}_{i\in\mathcal{X}}$ are the unique solution of the following linear system:

$$\begin{array}{ll} E_i\left[T_{\mathcal{S}}\right] = 0 \text{ if} & i \in \mathcal{S} \\ E_i\left[T_{\mathcal{S}}\right] = \frac{1}{\omega_i} + \sum_j P_{ij} E_j\left[T_{\mathcal{S}}\right] \text{ if} & i \notin \mathcal{S} \end{array}$$

The solution of this problem is: $(I - P)\, E[T_d] = 1/w_i$; where P_{ij} is set to 0 when i is equal to the node d because the hitting times does not depend on any other node reached d, and $1/w_i$ is set to 0 when i is equal to d because the hitting time starting from node d is 0. The result is $E_o[T_d] = 8.79$.

1.5 Point e

Since the graph is aperiodic and strongly connected, the dynamics converge to a consensus state for every initial condition $x(0)$.

So simulating the French-DeGroot dynamics on G with a random initial condition we can observe that the states of the node reach the consensus according to the theory.

1.6 Point f

Assuming $x_i(0) = \xi_i, \{\xi_i\}_{i\in\mathcal{V}}\, iid, V[\xi_i] = \sigma^2$

$$V[\alpha] = V\left[\sum_{i\in\mathcal{V}} \pi_i x_i(0)\right] \overset{\text{ind.}}{=} \sum_{i\in\mathcal{V}} V[\pi_i x_i(0)] = \sum_{i\in\mathcal{V}} \pi_i^2 V[x_i(0)] = \sum_{i\in\mathcal{V}} \pi_i^2 \sigma^2$$

For example, assuming $\sigma^2 = 0.25, V[\alpha] = 0.053$ Simulating the dynamics, we obtain 0.054 as consensus variance.

1.7 Point g

Removing the edges (d,a) and (d,c) from the graph in Fig. 1, we obtain the graph in Fig. 2. This graph is not strongly connected anymore.

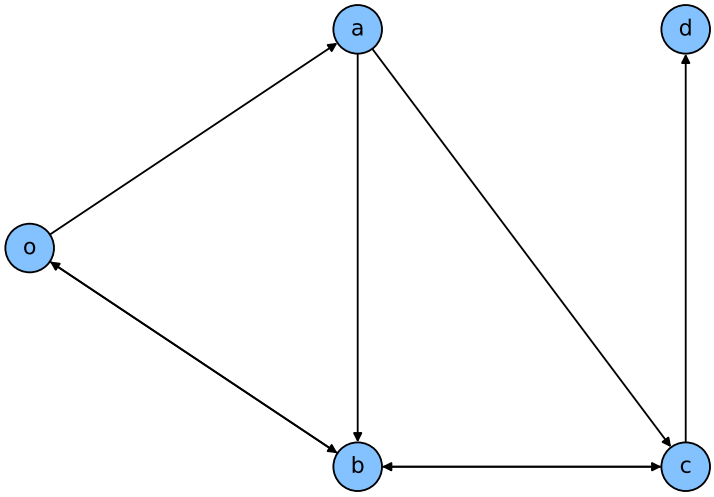

Figure 2: Graph Exercise 1 point g.

Now the asymptotic behaviour of the dynamics is determined only by the initial state of the nodes in the sink of the condensation graph, because they are not influenced by the rest of the graph. Moreover, the sink contain the support of the invariant distribution, which determines the consensus value.

As in this case there is only one attracting component and it is aperiodic, a consensus will be reached, and its value will be the initial opinion of d.

In this case the invariant distribution is: $[0 \quad 0 \quad 0 \quad 0 \quad 1]$. So as explained in the Point f in this case the variance of the consensus will be:

$$V[\alpha] = V\left[\sum_{i\in\mathcal{V}} \pi_i x_i(0)\right] \overset{\text{ind.}}{=} \sum_{i\in\mathcal{V}} V\left[\pi_i x_i(0)\right] = \sum_{i\in\mathcal{V}} \pi_i^2 V\left[x_i(0)\right] = \sigma^2$$

1.8 Point h

Removing the edges (d,a) and (c,b) from the graph in Fig. 1, we obtain the graph in Fig. 3. This graph is not strongly connected. Moreover the sink component is not aperiodic and is composed by node c and d. So a consensus is reached only if the initial opinion of the node c and d are equal. Otherwise, a consensus is not reached.

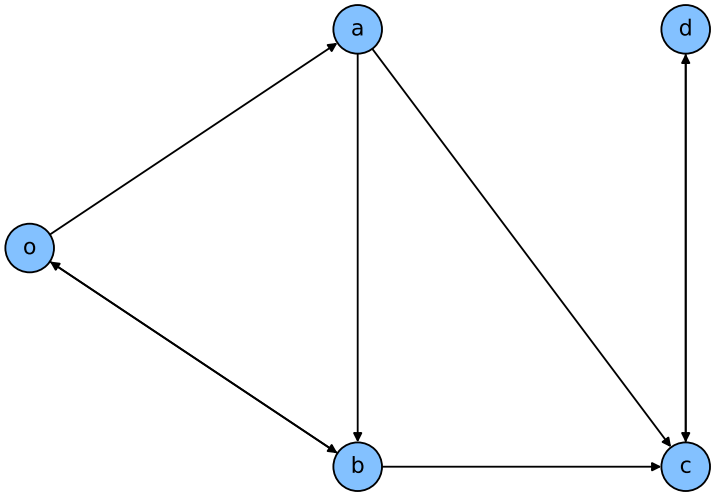

Figure 3: Graph Exercise 1 point h.

2 Exercise 2

In this exercise we continue to consider the graph in Fig.1 and the same associated transition rate matrix Λ. This time we are going to consider multiple particles moving around the network in continuous time, where each particle will move as in the exercise 1 point a[1.1]. We will analyze the system from two different prospective: particle and node.
To compute these simulation we used:

- w^* that is the maximum out-degree among the nodes of the graph, in our case is equal to 1.
- $\bar{P}$ that is a transition probability matrix definded in the following way:

$$\bar{P}_{ij} = \frac{\Lambda_{ij}}{\omega_*}, \quad i \neq j, \quad \bar{P}_{ii} = 1 - \sum_{j \neq i} \bar{P}_{ij}$$

2.1 Point a: Particle perspective

To simulate the average return-time of 100 particles starting in node a we used a system-wide Poisson clock with rate 100 times w^*. We simulated the system for 10.000 unit of times and at each tick we randomly select a

particle in the system which is moved according to the transition matrix $\bar{P}$ to another node. Once the particle return in node a we save the return time and increment a counter. Once the simulation ended, we can sum all the return times and divide it by the counter.
The average return time computed is 6.75.

We can notice that this result is very similar to the return time simulated in the exercise 1 point a. In fact, at each tick we take a random particle and moving it only according to $\bar{P}$, this fact basically makes each particle independent from the other ones.

2.2 Point b: Node perspective

To compute the simulation we followed the hint in the assignment. In fact, we have a system-wide Poisson clock with rate 100 times w^*. 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 $\bar{P}$.

The average number of particles in the different nodes at the end of the simulation is :

$$[18.89844015 \quad 15.73312987 \quad 21.5598508 \quad 22.2550017 \quad 21.55357748] \ .$$

The number of particles in each node during the simulation time is illustrated in Fig. 4.
Comparing the simulation result above with the stationary distribution $\bar{\pi}$:

$$[0.18518519 \quad 0.14814815 \quad 0.22222222 \quad 0.22222222 \quad 0.22222222] \ .$$

we can see as the average number of particles in the different nodes at the end of the simulation is very close to the $\bar{\pi}$ for each node.

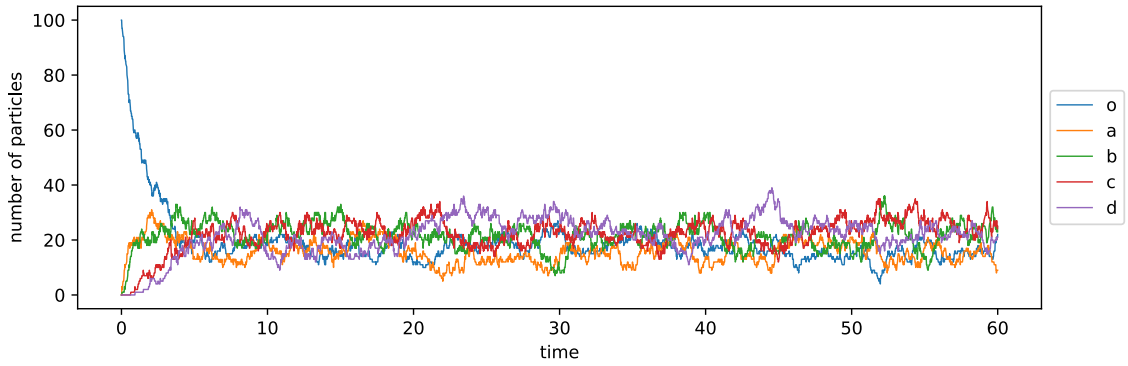

Figure 4: Plot of the simulation in Point b[2.2] showing the number of particles in each node during the simulation time.

3 Exercise 3

In this exercise we study how different particles affect each other when moving around in a network in continuous time. We consider the open network of Fig. 5, with transition rate matrix Λ_{open} according to the assignment.

The goal of the exercise is to simulate the system in two different scenario, both for 60 time unit.

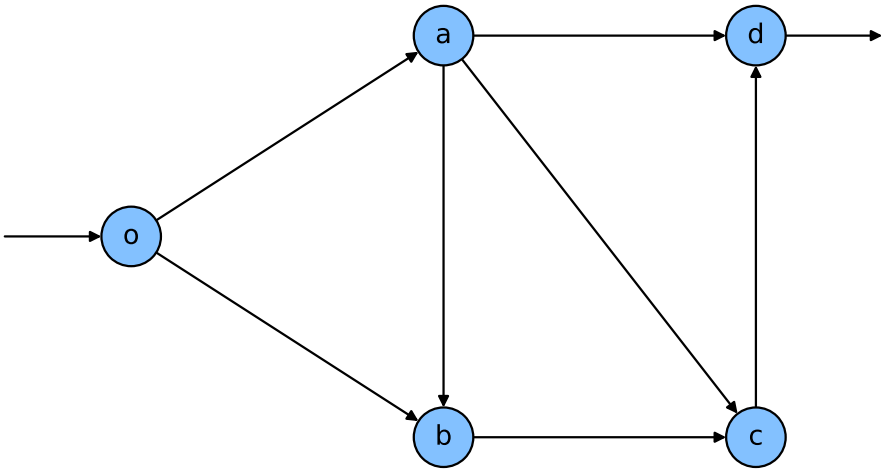

Figure 5: Graph Exercise 3

3.1 Point a: Proportional rate

For the simulation in this problem we consider two distinct Poisson process. The first is the one that regulates the when a particle enter in the system, more in specific in the node o, and it have rate $\lambda = 1$.
While the second one regulates how each node i will pass along particles according to a Poisson process with rate equal to the number of particles in the node times the rate of the local Poisson clock of node i(out-degree of the node i), i.e., the node i will pass along particles rate with rate $\omega_i N_i(t)$.

The algorithm proposed for the simulation take in consideration that each time a particle is moved from a node to another or a particle is added to node o according to the processes described before, the state of the system change and with that how each node i will pass along particles.

The evolution of the number of particles in each node over time is represented in Fig. 6.

Since each node i pass along particles with a rate proportional to number of particles in node, this system can handle any input rate without blowing up.

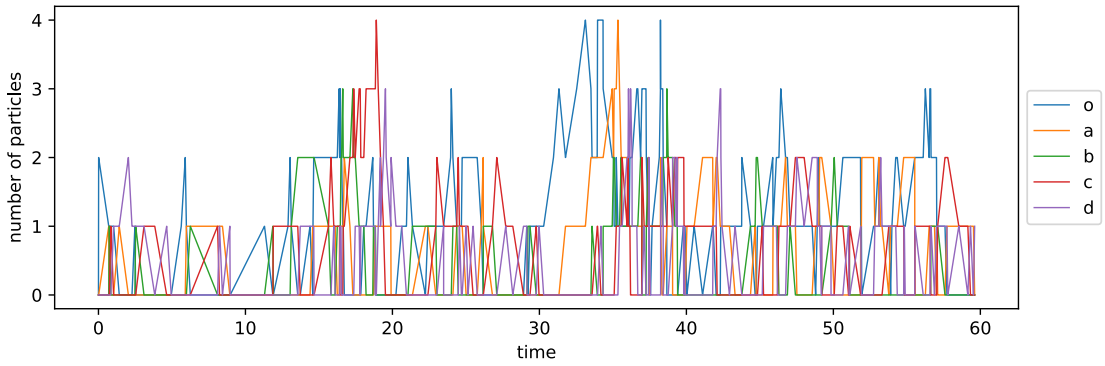

Figure 6: Plot of the simulation in the proportional rate showing the number of particles in each node during the simulation time.

3.2 Point b: Fixed Rate

As in the previous point, we consider two distinct Poisson process; the first one as before regulates the input of particles with rate $\lambda = 1$. The second one meanwhile differ from the point before, in fact in this case each node i will instead pass along particles with a fixed rate ω_i. Except for this fact, we basically deployed the same algorithm as in the proportional rate scenario by handling in the right manner the evolution of the simulation.

The evolution of the number of particles in each node over time is represented in Fig. 7.

Running the simulation with different input rates, we saw that in most of the case when the input rate λ is grater than $\omega_o = 1.125$ which is the rate of the local Poisson clock in node o. The number of particles in o is proportional to the input rate, but the output rate of o is not proportional to number of particles it contains, so node o is not able to handle the particles it receives. Node o can be able to handle the particles only if the input rate is equal to its output rate, but this time its output rate is fixed to 1, so, the only input rates for which the system will not blow up, are for input rates ≤ 1.125 . For all the other input rates, the system will blow up as faster as the rate increase as we can see in Fig.8 and in Fig. 9.

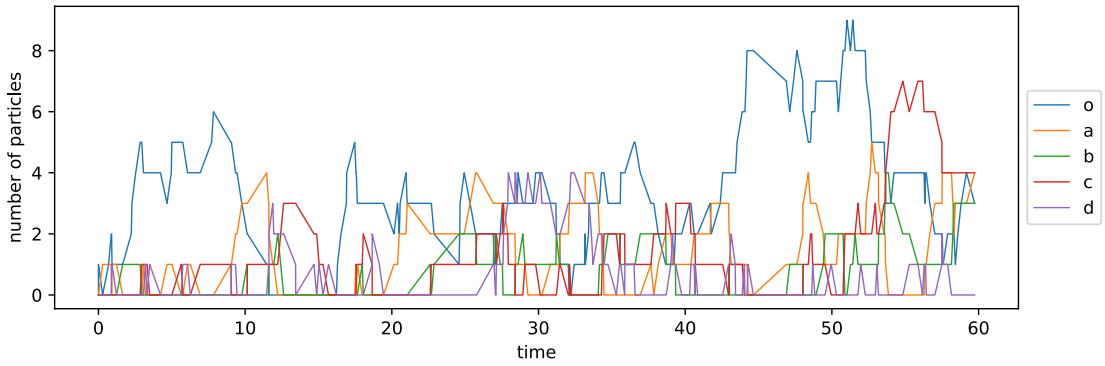

Figure 7: Plot of the simulation in the fixed rate scenario showing the number of particles in each node during the simulation time. With input rate $\lambda = 1$.

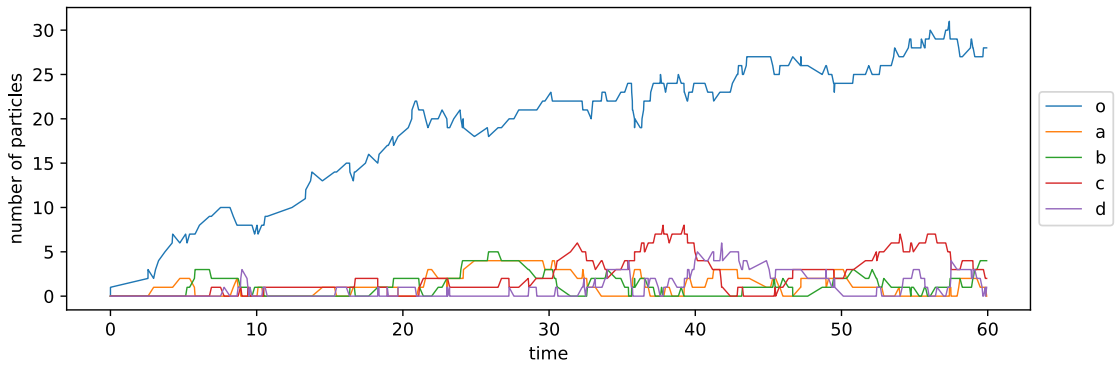

Figure 8: Plot of the simulation in the fixed rate scenario showing the number of particles in each node during the simulation time. With input rate $\lambda = 1.5$.

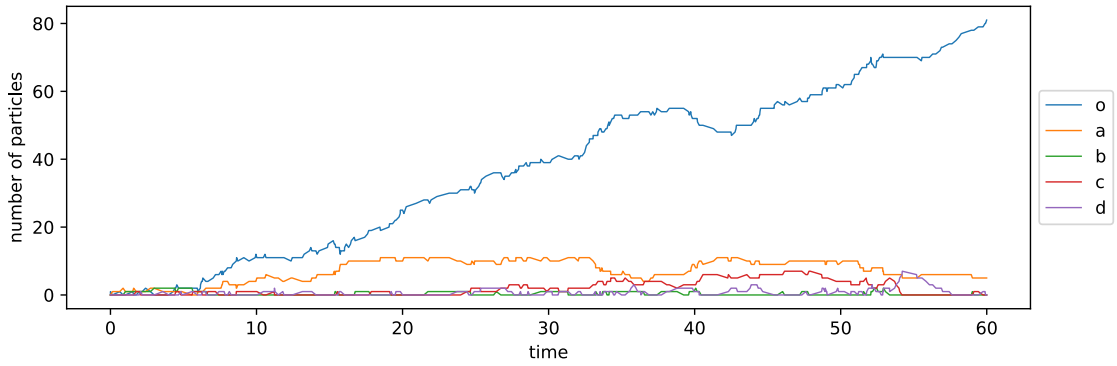

Figure 9: Plot of the simulation in the fixed rate scenario showing the number of particles in each node during the simulation time. With input rate $\lambda = 2$.