Network Dynamics and Learning
Homework II

Davide Gariglio
s292964@studenti.polito.it

1 Introduction

In this second *Homework* we are required to apply the theoretical knowledge acquired during the lectures and laboratory activities, such as: *Markov Chains, Random Walks, Distributed Averaging and Linear Flow Dynamics*. All the exercise have been solved in *python* language using the *NetworkX library*; the solutions of the exercises are provided with commented python notebook and explained in the following report.

2 Problem 1

In this first section, it is required to study a single particle performing continous-time random walk in the network described by the transition rate matrix Λ (Eq. 1) and reported in Fig. 1.

In order to perform the next simulation, we instantiate a rate-r Poisson process which will help in simulated a clock; the time between two "ticks" will be an independent random variable with Poisson distribution. In order to simulate the "ticks", we compute $t_{next} = -\frac{\ln(u)}{r}$, where $u \in \mathcal{U}\{0,1\}$.

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

2.1 Exercise 1.a

It is required to compute the average time for a particle to leave the node a and return to it. We performed the simulation of the Random Walk, keeping

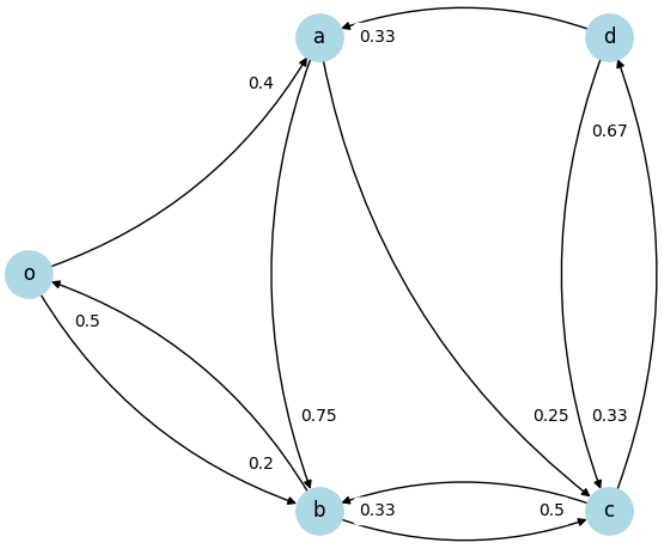

Figure 1: Graph representing the network with the corresponding transition rates set as edge weight.

track of the different time intervals elapsed between each move of the particle across the network; for each "jump", we verify if the particle arrived at the destination node, which is the one where started the simulation. In order to obtain a significant result, we performed 10000 simulation, then computed the average time giving as result:

$$\bar{\tau}_{a,a} \approx 6.75 \text{ time units}$$

2.2 Exercise 1.b

Now we compared the result found in Section 2.1 with the *theoretical return time* $\mathbb{E}_a[T_a^+]$. We can exploit the fact that the graph is strongly connected and apply **Theorem 7.2** *(iv)* from the lecture notes; in particular for every $i \in \chi$, where χ is the node set of the graph, it holds:

$$\mathbb{E}_i[T_i^+] = \frac{1}{\omega_i \pi_i} \tag{2}$$

By substituting the numbers in the formula of the Eq. 2 (and considering node $a = 1$ in the indexing) we obtain:

$$\mathbb{E}_1[T_1^+] = \frac{1}{\omega_1 \pi_1} = \frac{1}{1 \cdot 0.148} \approx 6.76 \text{ time units}$$

Another possible approach (the one done in the attached notebook) is to exploit again **Theorem 7.2**, but the sub-point *(v)*. In particular we can recursively apply the formula:

$$E_a\left[T_a^+\right] = \frac{1}{\omega_a} + \sum_j P_{a,j} E_j\left[T_a\right]$$

A more compact way, suitable for the coding part (introduced in laboratory activity 6), consists in defining the set of nodes $R = \chi/\mathcal{S}$ (where $\mathcal{S}$ is the set containing the node to hit, in this case a) and taking the reduced vector $\hat{\omega} = \omega_R$ and matrix $\hat{P} = P_{RxR}$; with these it is necessary to solve the linear system:

$$\hat{x} = (\mathbb{1} - \hat{P})^{-1} z \tag{3}$$

$$\text{with } z = \frac{1}{\omega_j} \text{ , } j \in R$$

For the two proposed approaches, the result is equivalent and the *theoretical return time* obtained is comparable with the *average return time* computed in Section 2.1.

2.3 Exercise 1.c

The next question requires to compute the Random Walk simulation in order to calculate the average time it takes the particle to move from node o to node d. This is analogous to what we have done in Section 2.1, but we have to set different origin and destination nodes.

As before, 10000 simulations are performed and the *average time* is:

$$\bar{\tau}_{o,d} \approx 8.69 \text{ time units}$$

2.4 Exercise 1.d

Now we compare the result obtained in Section 2.3 with the *theoretical hitting-time* $\mathbb{E}_o[T_d]$. This time origin and destination nodes are different (origin: o, destination: d). As we did in Section 2.2, we solve the Eq. 3 changing $\mathcal{S}$ and $\mathcal{R}$ node sets; the result of the *theoretical hitting-time* is:

$$\mathbb{E}_o[T_d] \approx 8.79 \text{ time units}$$

which is very similar to the value obtained empirically in Section 2.3.

2.5 Exercise 1.e

From now on, consider Λ as the *Weight matrix*; the resulting graph will be $\mathcal{G} = (V, \mathcal{E}, \Lambda)$. The goal is to perform Opinion Dynamics operations on it, in particular we are required to simulate the *French-DeGroot dynamics* with an arbitrary initial condition $x(0)$ and discuss the convergence with respect to $x(0)$.

The graph in Fig. 1 is strongly connected; moreover is aperiodic has only one *sink* in the associated condensation graph. These conditions allow to use **Corollary 5.1** in the lecture notes, in particular:

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

This Corollary states that the consensus is the asymptotic outcome of the model under certain connectivity assumptions; moreover, if these assumptions are satisfied, the consensus converges no matter the the initial condition $x(0)$ is.

In the attached notebook we instantiated randomly the initial states and let the simulation run for $t = 30$. In Fig. 2 it is graphically reported the convergence of the consensus.

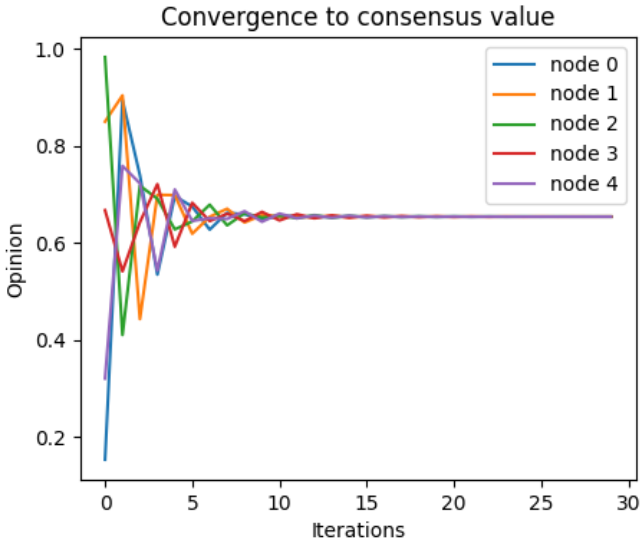

Figure 2: Visualization of the convergence of the consensus.

2.6 Exercise 1.f

Similarly to what we have done in Section 2.5, we randomly instantiate the initial condition $x(0)$, in particular: $x_i(0) = \xi_i$ where $\{\xi_i\}_{i\in\mathcal{V}}$ are *i.i.d. random variables* with variance σ^2; for this exercise we choose $\xi_i \in [0, 1)$ following a uniform distribution, so

$$\sigma^2 = \frac{(b-a)^2}{12} = \frac{(1-0)^2}{12} = \frac{1}{12} \approx 0.083$$

We have executed 100 *French-DeGroot* simulations and the calculated variance of the consensus was $\sigma_x^2 \approx 0.017$. The theoretical variance is given by Eq.4 and its result is $\sigma_{\bar{x}}^2 = 0.0178$, which is comparable with the value computed with the *French-DeGroot* model simulations.

$$\sigma_{\bar{x}}^2 = \sigma^2 \sum_i \pi_i^2 \tag{4}$$

Since $\pi_i < 1$ for every $i \in \mathcal{V}$, $\sum_i \pi_i^2 < \sum_i \pi_i = 1$; this implies that the variance of the consensus is always smaller than the variance of original opinion. This important result is called *Wisdom of the Crowd*. In particular this means that, under the assumption that the individuals have all the same measurements capabilities (this is the case because their opinions are i.i.d. random variables), it is possible to better estimate the state of the network θ with respect to the original estimate.

2.7 Exercise 1.g

In this task we remove the edges (d, a) and (d, c) and we discuss the asymptotic behaviour of the dynamics with the new graph (Fig. 3). We can see that the node d becomes a sink node of the network, therefore is not influenced by any other node and becomes a *stubborn node*. This implies that, no matter what is the initial opinion $x(0)$, the consensus will always converge to the initial opinion associated to node d.

As we did in Section 2.6, $x(0)$ has been initialized randomly from a uniform distribution defined in the interval $[0, 1)$ ($\sigma^2 = \frac{1}{12} \approx 0.083$).

After 100 simulations, the variance of the consensus is

$$\sigma_{\bar{x}}^2 \approx 0.079$$

and is much more closer to σ^2. This is because all the opinions of the nodes that are not stubborn will eventually follow the initial opinion of node d and the associated distribution.

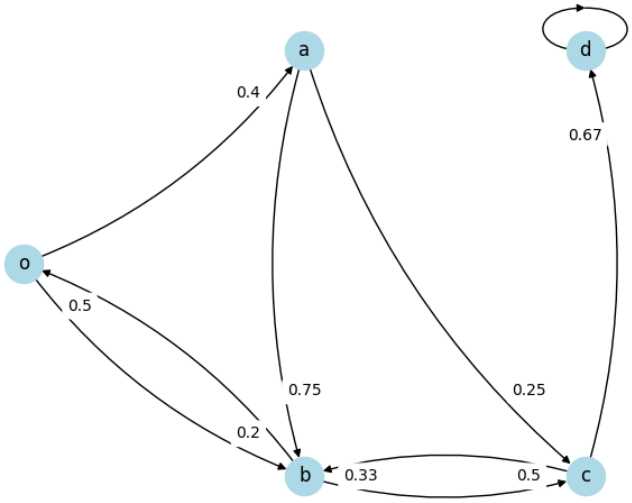

Figure 3: New graph without (d, a) and (d, c) edges. In the visualization the node d presents a self loop in order to simulate the particles that stay in there.

2.8 Exercise 1.h

In this exercise we start from the graph in Fig. 1 and we remove the edges (c, b) and (d, a), obtaining as result the graph shown in Fig. 4. We can see that now the graph is no longer connected; the condensed graph is composed by two connected component, therefore the consensus does not converge. This consideration has also been tested with the *French-DeGroot* simulation with $x(0)$ initialized randomly and the opinions of the nodes are reported in Fig.5

3 Problem 2

In this second problem we will work again with the graph in Fig. 1 and with the weights expressed by Λ (1). The goal is to simulate many particles (100) moving around the network in continuous time; each particle moves as studied in Section 2.

We have to simulate the system from two different perspectives: the particle perspective and the node perspective.

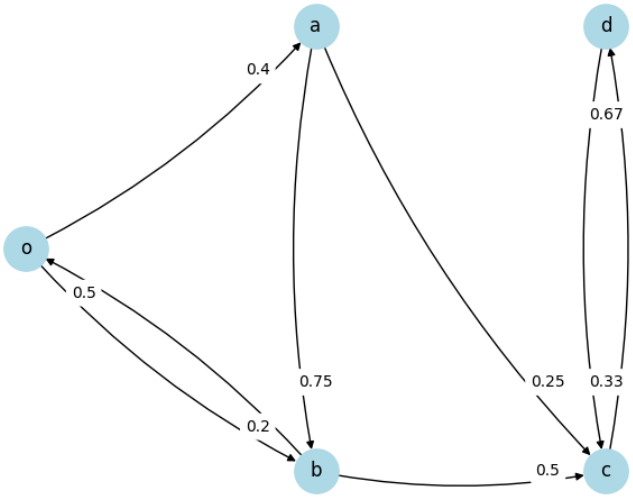

Figure 4: New graph without (c, b) and (d, a) edges. It is visible a *trapping set* composed by node c and d.

3.1 Exercise 2.a - Particle perspective

All the particles are i.i.d., the simulation of a random walk with 100 particles is equal to simulating a random walk with one particle 100 times.

As we did in the previous exercises, we have to perform multiple simulations in order to have a statistically significant result; therefore we executed 1000 simulations for 100 different particles. The average return time on the node a over 100 particles is $T_a^+ \approx 6,76$ which is very close to the theoretical return time found in Section 2.2 ($T_a^+ \approx 6,75$). This is reasonable since we just performed more simulations that were independent.

3.2 Exercise 2.b - Node perspective

In this second part we study the system from the node perspective, in particular we have to discuss the distribution of the particles across the nodes after 60 time units. We introduce a system-wide Poisson clock with rate equal to the number of particles in the network; when it ticks we randomly select the particle to move in the next node. After the 60 time units, we obtained the following average number of particles per node:

$$o = 20.86,\ a = 15.38,\ b = 22.05,\ c = 21.04,\ d = 20.67$$

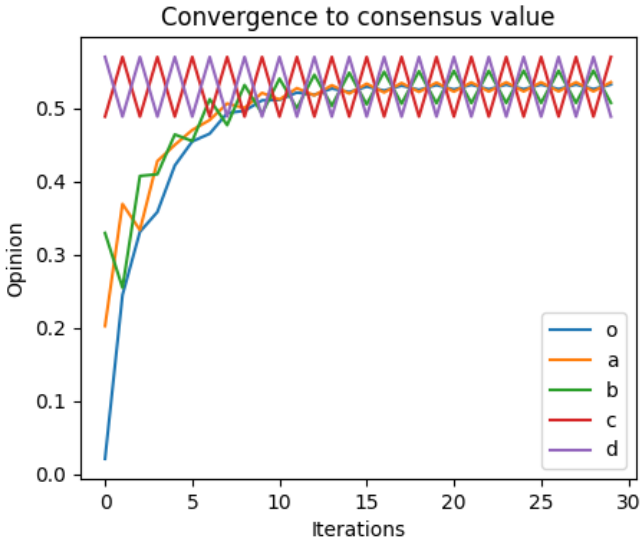

Figure 5: The first connected component composed by o, b and a influenced by the second one; however, since the latter one is a trapping set (composed by nodes c and d), the consensus is never reached. In particular it is bounded between the initial opinion of nodes c and d.

In Fig. 6 is reported the motion of the different particles across the network during the simulation.

4 Problem 3

In the last part of this assignment we study how different particles affect each other when moving around in a network in continuous time. We are given an open network shown in Fig. 7, with the associated transition rate matrix in 5. The particles are entering from node o according to a Poisson process with input rate $\lambda = 1$. We will treat the particle movement similarly to Section 3.2.

We will simulate two different scenarios: the first one in which the nodes pass along particles with *proportional rate*, while the second one with *fixed rate*. For both cases it is required to show the evolution of the particles across the graph and discuss if the network can handle different rates.

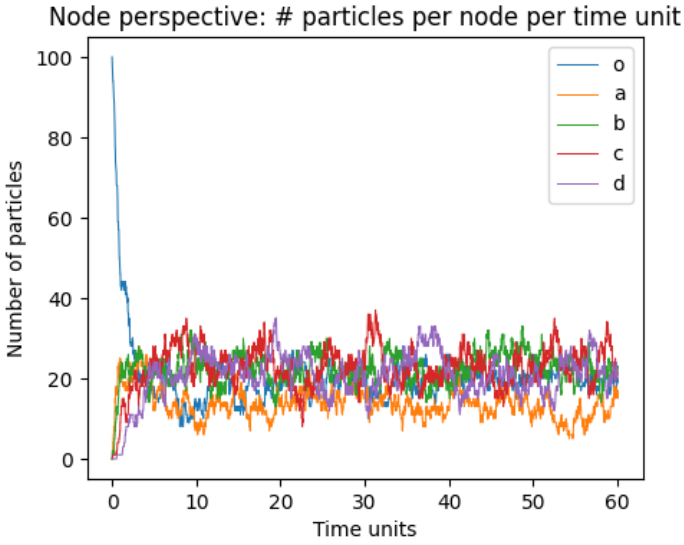

Figure 6: Graph showing the amount of particles inside each node of the network. It is visible that after few iterations the particles are able to leave node o and statistically they are almost evenly partitioned, without having a particular node in which are concentrated the majority of them.

$$\Lambda_{open} = \begin{pmatrix} o & a & b & c & d \\ 0 & 3/4 & 3/8 & 0 & 0 \\ 0 & 0 & 1/4 & 1/4 & 2/4 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{matrix} o \\ a \\ b \\ c \\ d \end{matrix} \tag{5}$$

4.1 Exercise 3.a - Proportional rate

In the first point of this exercise we set the input rate $\lambda = 1$ and run the simulation for 60 time units; the results are shown in Fig. 8. For this input rate the system does not "blow up"; in fact the number of particles remains bounded and there is not a "divergent behaviour".

Afterwards we tried to increase the value of the input rate λ. In particular the simulation has been executed with $\lambda = (5, 10, 50, 100)$ and the respective particles distribution across the time units are show in Fig. 9, 10, 11, 12. From these results we can see that the system does not "blow

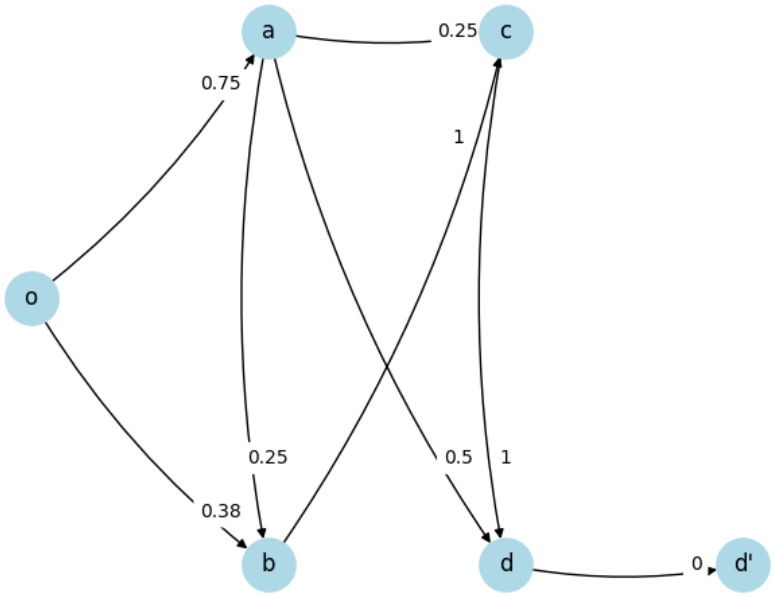

Figure 7: Graph associated with the exercises of Section 4. It has been added the node d' in order to let the particle leave the network through node d. For every edge (i, j) it is represented as weight the value retrieved from the transition rate matrix $\Lambda_{open_{i,j}}$ (5).

up" also with large values of λ; this is due to the Poisson clock of the single nodes which has been instantiated with *proportional rate*. A higher number of particles in a node implies an higher rate, so each node is able to pass along its particles to other nodes.

4.2 Exercise 3.b - Fixed rate

In this last part we test the second scenario, the one in which the rates of the clocks are fixed (equal to ω_i). We run the simulation for 60 time units with variable input rates in order to understand if the system "blows up". It has been observed that, in order to address the assignment, reaching high values of λ was not necessary. In particular with: $\lambda = (0.5, 0.7, 1, 1.25, 1.5)$ we obtained the respectively particle distributions across time shown in Fig. 13, 14, 15, 16, 17.

It is visible that the system can handle the input rate up to $\lambda = 1$; after that value, the node o is not able to send the particles quickly enough. This

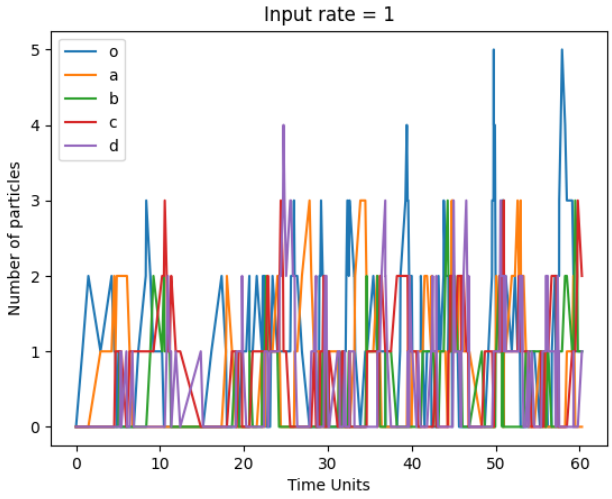

Figure 8: Particles distribution during the simulation for input rate $\lambda = 1$ and the nodes that pass along particles at a proportional rate $r = N_i(t)\omega_i$.

is caused by the fixed rate clock; in fact the node o has $\omega_o = 1.125$. In practice we will have more incoming particles in the network with respect to the amount that can be moved per time unit by the entry node o and the system will "blow up".

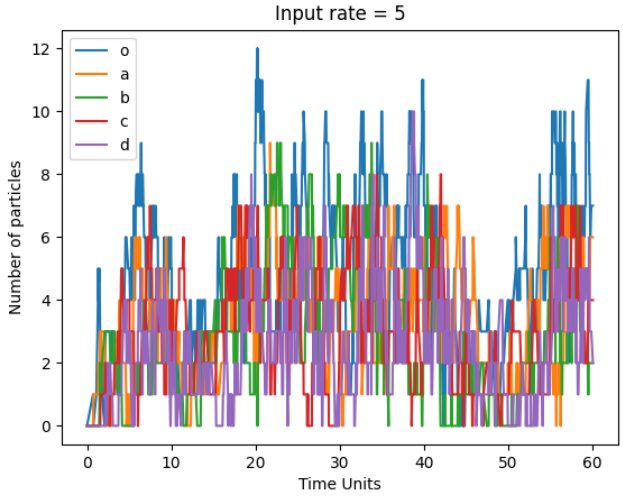

Figure 9: Particles distribution during the simulation for input rate $\lambda = 5$ and the nodes that pass along particles at a proportional rate $r = N_i(t)\omega_i$.

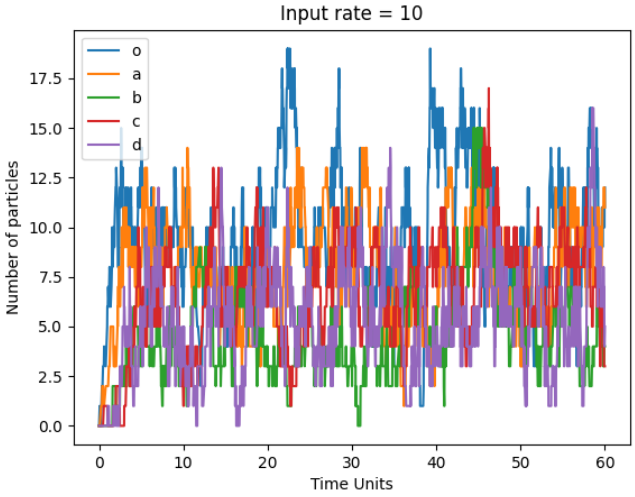

Figure 10: Particles distribution during the simulation for input rate $\lambda = 10$ and the nodes that pass along particles at a proportional rate $r = N_i(t)\omega_i$.

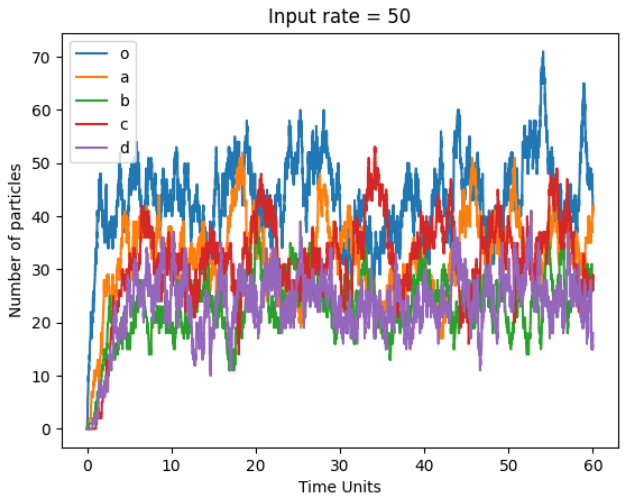

Figure 11: Particles distribution during the simulation for input rate $\lambda = 50$ and the nodes that pass along particles at a proportional rate $r = N_i(t)\omega_i$.

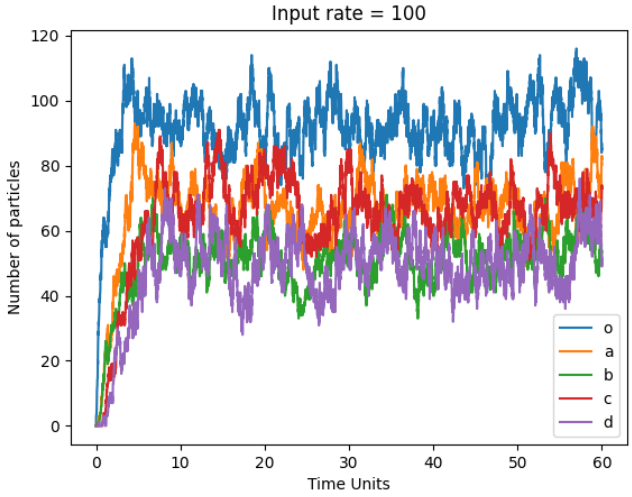

Figure 12: Particles distribution during the simulation for input rate $\lambda = 100$ and the nodes that pass along particles at a proportional rate $r = N_i(t)\omega_i$.

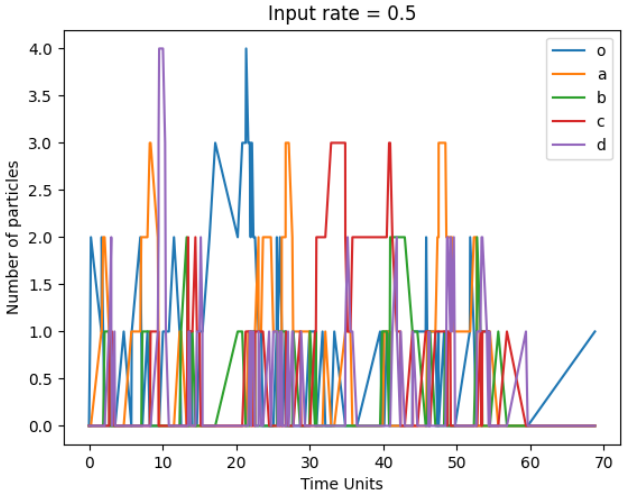

Figure 13: Particles distribution during the simulation for input rate $\lambda = 0.5$ and the nodes that pass along particles at a fixed rate $r = \omega_i$.

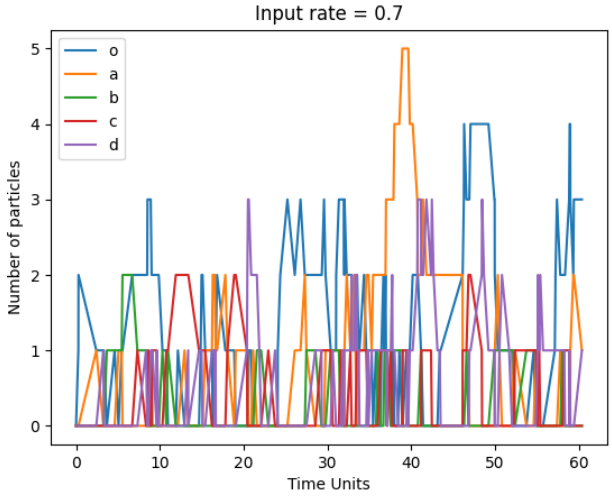

Figure 14: Particles distribution during the simulation for input rate $\lambda = 0.7$ and the nodes that pass along particles at a fixed rate $r = \omega_i$.

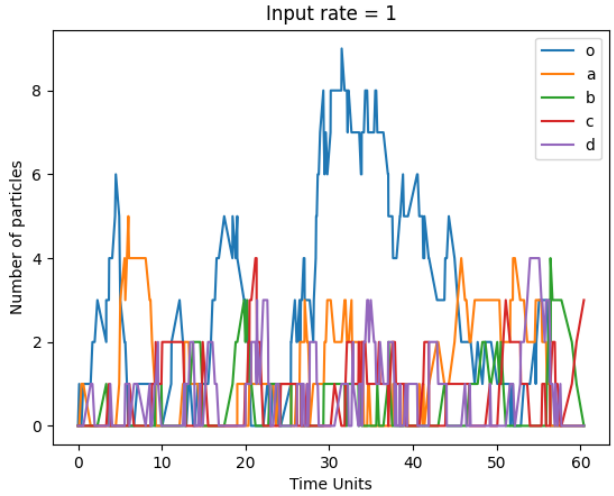

Figure 15: Particles distribution during the simulation for input rate $\lambda = 1$ and the nodes that pass along particles at a fixed rate $r = \omega_i$.

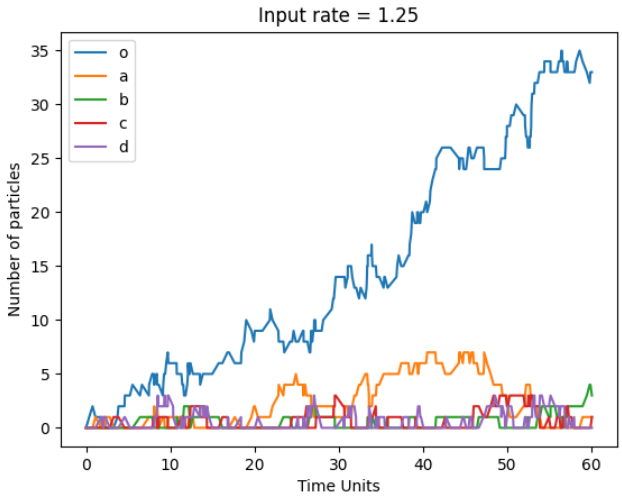

Figure 16: Particles distribution during the simulation for input rate $\lambda = 1.25$ and the nodes that pass along particles at a fixed rate $r = \omega_i$.

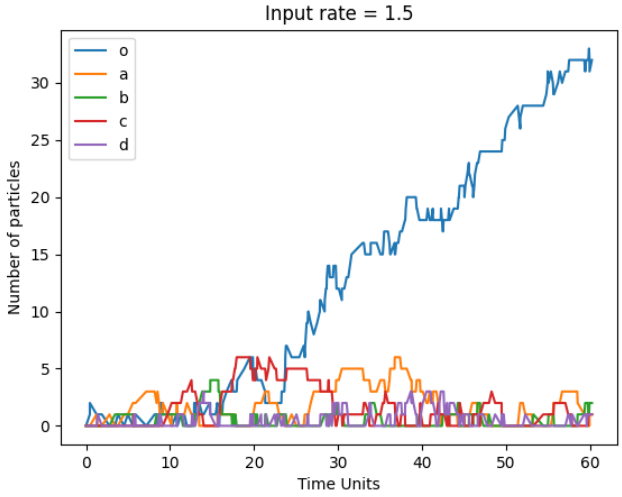

Figure 17: Particles distribution during the simulation for input rate $\lambda = 1.5$ and the nodes that pass along particles at a fixed rate $r = \omega_i$.