POLITECNICO DI TORINO

Master Degree course in Data Science and Engineering

Network Dynamics and Learning

Homework 2

Professors
Prof. Fabio Fagnani
Prof. Giacomo Como
PhD. Leonardo Cianfanelli

Student
Gustavo Nicoletti Rosa
Claudio Fantasia

Academic Year 2023-2024

Contents

Chapter 1

Exercise 1

1.1 Part (a)

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

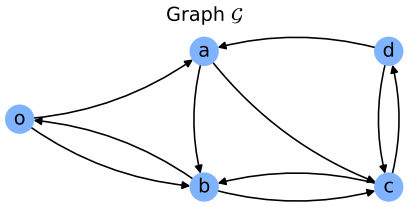

Figure 1.1: Graph $\mathcal{G}$

We considered a Markov Chain $X(t)$ with transition rate matrix Λ on a finite state space $\mathcal{X}$ described by the different nodes in our closed network 1.1.

Assuming that our single particle started in node b and performed a random walk, on average how many time units are required for the particle to return in b? We can rewrite this more rigorously

$$T_b^+ := \inf\{t \geq 1 : X(t) = b\}$$

$$E_b[T_b^+]$$

We defined for each node i a local *Poisson Clock* ω_i computed summing up the value along the rows in the transition rate matrix (*i.e* $\omega_i = \sum_j \Lambda_{ij}$). We defined the time units as $t_{next} = \frac{-ln(u)}{r}$ with $u \sim \mathbf{U}(0,1)$ and r equal to the local Poisson Clock of the node ω_i in which the particle is situated in that instant of time.
After the tick of the local Poisson clock, the particle is moved to an adjacent node using the transition probability matrix P.
We ran 100.000 simulations and we found out that on average the returning time for the particle starting in b is **4.56**.

1.2 Part (b)

How does the result in a) compare to the theoretical return-time $E_b[T_b^+]$? (Include a description of how this is computed.)

There are several ways in which we can compute the theoretical expected return time. We can do it recursively with the formula $E_i[T_b^+] = \frac{1}{\omega_i} + \sum_j P_{ij} E_j[T_j^+]$ if $i \neq b$ and $E_i[T_b^+] = 0$ if $i = b$. Otherwise we can use a more compact formula: $E_b[T_b^+] = \frac{1}{\omega_b \cdot \bar{\pi}_b}$. Firstly we define $\bar{P}$

$$\begin{aligned} \bar{P}_{ij} &= \frac{\Lambda_{ij}}{\omega_*}, i \neq j \\ \bar{P}_{ii} &= 1 - \sum_{j \neq i} \bar{P}_{ij} \end{aligned} \tag{1.1}$$

later we computed $\omega = \Lambda \mathbb{1}$, $\bar{\pi}$ as the solution of $\bar{P}^{'} \bar{\pi} = \bar{\pi}$ and then we found the theoretical return-time as **4.6**

1.3 Part (c)

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

For this task, we have used the same algorithm developed for 1.1.
We used the same local Poisson clock given by $\omega = \Lambda \mathbb{1}$ and the same time units $t_{next} = \frac{-ln(u)}{r}$ for taking into account the passage of time. The particle movement was determined by the transition probability matrix P. While in task 1.1 we started from b and ended in b after some time units, here the particle is starting in o and the algorithm ends when the particle reaches d.
The average time found after 100.000 simulations is **10.67**

1.4 Part (d)

How does the result in c) compare to the theoretical hitting-time $E_o[T_d]$? (Describe also how this is computed.)

As we said before we can solve this task using the recursive formulation of the problem characterized by its own linear system.

$$E_i[T_{\mathcal{S}}] = 0 \quad i \in \mathcal{S}$$

$$E_i[T_{\mathcal{S}}] = \frac{1}{\omega_i} + \sum_j P_{ij} E_j[T_{\mathcal{S}}] \quad i \notin \mathcal{S}$$

Assuming $\mathcal{S} = \{d\}$, $E_i[T_{\mathcal{S}}] = \tau_i$, $\mathcal{R} = \mathcal{V} \setminus \mathcal{S}$, $Q = P|_{R \times R}$, we can prove that:

$$\tau_i - \sum_j P_{ij} \tau_j = \frac{1}{\omega_i}$$

And we notice that $\tau_j = 0$ if $j \in \mathcal{S}$ so we can rewrite as:

$$\tau_i - \sum_j Q_{ij}\tau_j = \frac{1}{\omega_i}$$

And everything as:

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

And we found that the $E_o[T_d] = \tau_o = \mathbf{10.76}$

1.5 Part (e)

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

The linear dynamics of a learning model is given by $x(t+1) = Px(t)$ and $x(t) = P^t x(0)$. Also if the graph is str. connected and **aperiodic** the consensus is achieved by $\lim_{t\to+\infty} x(t) = \alpha\mathbb{1}$ with $\alpha = \pi^{'} x(0)$.
We know that the graph $\mathcal{G}$ in Fig. 1.1 is aperiodic, so for every random distribution of value that starts at $x(0)$, the dynamics will converge to the consensus vector α.
Usually the upper bound of the convergence rate for this Graph's Topology is $O(n^2)$, so in our case should be $O(5^2)$. Indeed, the root mean square error between the components of $x(25)$ and the consensus α was of

$$\text{RMSE}(x(25), \alpha) = \mathbf{3.94 \cdot 10^{-4}}$$

1.6 Part (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 independent random variables with variance

$$\sigma_a^2 = \sigma_b^2 = \sigma_c^2 = 1, \sigma_o^2 = \sigma_d^2 = 2$$

Compute the variance of the consensus value, and compare your results with the numerical simulations.

When $x_i(0) = \xi_i$ are random variables, $x(0) = \xi$ is a multivariate random variable, hence the consensus $\alpha = \pi^{'} x(0)$ is also a random variable. We can calculate the variance of α:

$$
\begin{aligned}
Var(\alpha) &= Var(\pi^{'} x(0)) = Var(\pi^{'} \xi) \\
&= Var(\sum_{i \in \mathcal{V}} \pi_i \xi_i) = \sum_{i \in \mathcal{V}} Var(\pi_i \xi_i) \\
&= \sum_{i \in \mathcal{V}} \pi_i^2 Var(\xi_i) \\
&= \sum_{i \in \mathcal{V}} \pi_i^2 \sigma_i^2 = \mathbf{0.2593}
\end{aligned}
\tag{1.2}
$$

To compute the numerical simulations, we performed with two families of distributions, the first is the Normal distribution $N(\mu, \sigma^2)$ and the second is the Uniform distribution $U(A)$, with A the amplitude.
Given a million random $x(0) = N(0, \sigma^2)$, the average μ_α and variance σ_α^2 of the consensus were

$$\mu_\alpha = 0.0000$$

$$\sigma_\alpha^2 = 0.2591$$

Instead for the uniform distribution simulations, we first derived the amplitude parameter to achieve distributions with the desired variances, if the variance $Var(U(A)) = A^2/12$, the amplitude A_i of each $\xi_i = U(A_i)$ is achieved by setting $A_i = \sqrt{\sigma^2 \cdot 12}$. After a million simulations, we obtained

$$\mu_\alpha = 1.9585$$

$$\sigma_\alpha^2 = 0.2590$$

We observe that both variance simulations approximate very well the theoretical variance of the consensus. We further notice the convergence $\mu_\alpha \to \pi^{'} E[\xi]$.

1.7 Part (g)

Remove the edges (d,a) and (d,e). Describe and motivate the asymptotic behavior dynamics. If the dynamics converge 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 analytically the variance of the consensus value.

The starting Graph $\mathcal{G}$ in Fig. 1.1 is str. connected and aperiodic, meaning that the consensus value α is determined by π which has support on all the nodes.
We define a second graph $\mathcal{H}$ by removing the edges (d, a) and (d, e) from $\mathcal{G}$, making $\mathcal{H}$ not str. connected with a unique sink in it. In this way the support of π and so the value of consensus is determined just by the value in the sink node. Indeed it can be shown that the value of π in our case is a vector with value [0,0,0,0,1], so the initial random vector is multiplied by 1 only in the position of the sink node, hence the consensus will be $\alpha = x_d(0)$.
Notice also that we inserted by hand a self-loop in d in order to make the support of π

feasible, as in Fig. 1.2. From these observations and Eq. 1.2, if $x_i(0) = \xi_i$ are random variables, with a sink component with a single node d, the variance of the consensus will be:

$$\sigma_\alpha^2 = Var(\xi_d)$$

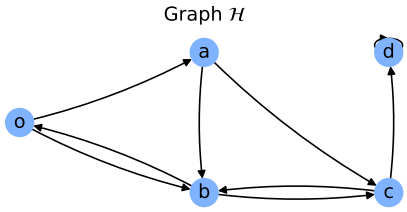

Figure 1.2: Graph $\mathcal{H}$

1.8 Part (h)

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

Removing the edges (c, b) and (d, a) from $\mathcal{G}$, we define the graph $\mathcal{J}$ which is not str. connected, with a globally reachable sink component composed by two nodes: d, e.

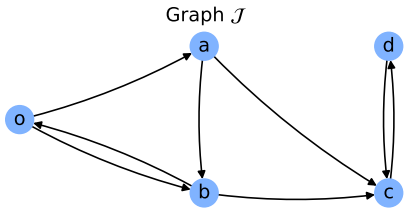

Figure 1.3: Graph $\mathcal{J}$

The opinions of nodes not belonging to the globally reachable component do not influence the final consensus value and the consensus will only depend on the values in the sink component. We observe that the P-invariant distribution is $\pi = [0, 0, 0, 0.5, 0.5]$, therefore one would expect the consensus to be $\alpha = 0.5 \cdot x_c(0) + 0.5 \cdot x_d(0)$, however, the sink component is periodic with period $T = 2$, therefore the system will not converge to a consensus unless $x_c(0) = x_d(0)$, rather it will oscillate around an equilibrium point with the same period. With $x(0) = [0.3745, 0.9507, 0.7320, 0.5987, 0.1560]$ we can observe the oscillations on Tab. 1.1.

t	$x_o(t)$	$x_a(t)$	$x_b(t)$	$x_c(t)$	$x_d(t)$
95	0.3572	0.3572	0.4779	0.156	0.5987
96	0.3975	0.3975	0.2767	0.5987	0.156
97	0.3572	0.3572	0.4779	0.156	0.5987
98	0.3975	0.3975	0.2767	0.5987	0.156
99	0.3572	0.3572	0.4779	0.156	0.5987
100	0.3975	0.3975	0.2767	0.5987	0.156

Table 1.1: Oscillations on time around equilibrium

Chapter 2

Exercise 2

Using the same network of Ex. 1, as in Fig. 1.1, we simulated the motion of a hundred particles and analyzed the results.

2.1 (a) Particle perspective

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

Since all particles' movements are independent of the other particles, we ran a hundred random walk simulations and stopped them when the particle completed the closed walk.

The average time of the closed walk from b to b, with a hundred simulations is 4.52. However, by the law of large numbers, we can better approximate the average to the expected value by computing more simulations. With a hundred thousand computations, the average return time is 4.6129. The result is comparable with the one obtained through simulations on Ex. 1.1 and the theoretical on Ex. 1.2.

2.2 (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.*

The algorithm is different in the node perspective for the simulation of the random walk of a hundred particles. We start by defining another transition probability matrix $\bar{P}$:

Where $\omega_* = \max \omega_i$, is the maximum node rate of the graph. The random walk is performed as follows: We simulate a global Poisson clock, with the rate $r = n\omega_*$, with $n = 100$ the number of particles in the system. When the clock hits, we randomly choose among all nodes with a probability distribution

$$\eta(i,t) = \frac{n_i(t)\omega_i}{\sum_k n_k(t)\omega_k} \tag{2.1}$$

To then randomly choose the next node j a single particle will move to, from node i, with the probability distribution

$$\bar{p}_i(j) = \bar{P}_{ij}$$

We start our simulation with a hundred particles in node o, and the results we obtain are in Fig. 2.1

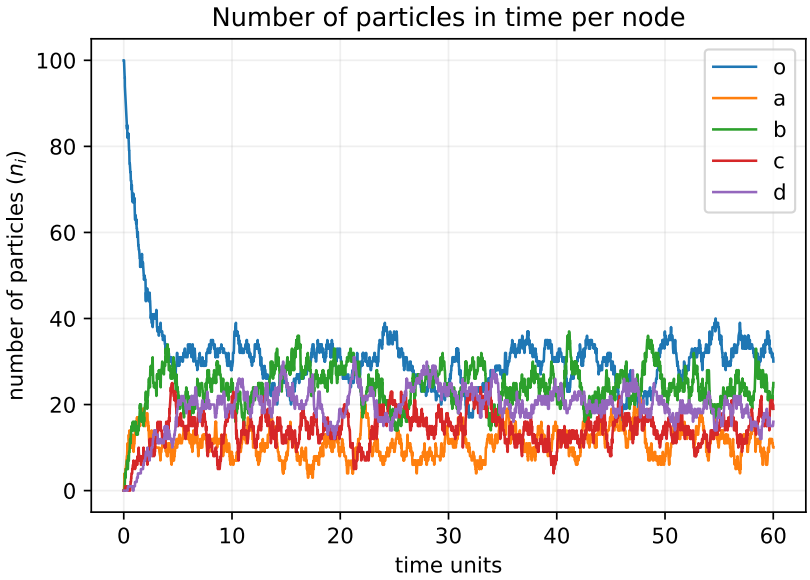

Figure 2.1: Node perspective simulation with a hundred particles $\mathcal{G}$

node i	avg	ω_i
o	29.2	3/5
a	11.0	1
b	24.6	5/6
c	14.6	1
d	20.6	2/3

Table 2.1: Average number of particles per node after time unit 10

We calculated the average in time of the number of particles in each node after the time unit 10 (Fig. 2.1), we observe and Tab. 2.1, the greater the ω_i, the faster a particle will move from the node.

Chapter 3

Exercise 3

In this part we consider the open network of 3.1, with transition rate matrix Λ_{open}

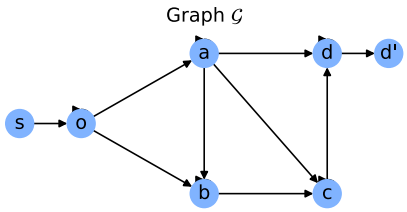

Figure 3.1: Graph $\mathcal{G}$

$$\Lambda_{open} = \begin{pmatrix} 0 & 3/4 & 3/4 & 0 & 0 \\ 0 & 0 & 1/4 & 1/4 & 2/4 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

As we did in Ex. 2 for the **Node perspective** we are overestimating the maximum velocity of the system, setting the global Poisson Clock as *maximum between num_particles in the system times* ω_{max} *and the input rate* λ. In our specific case the ω_{max} is given by the velocity of the node d (i.e. $w_d = 2$) where the particles are rejected from the system.

The overestimated system frequency depends on the input rate in the initial steps of the simulation and then it will depend on the other term when the number of particles grows too large. Notice that it is *almost* impossible that one node will reach the maximum frequency of the global Poisson Clock set, indeed it is a good upper bound.

In order to deal with this upper bounded velocity, every row in the transition matrix Λ_{open} is normalized using the $\bar{P}$ mentioned in 2. In this way, even the fastest node in the system will have a non-zero value in the position $\bar{P}_{ii}$.

We added a node called s to simulate the insertion of new particles in the system with $w_s = \lambda$ and n_particle of s equal to 1 to normalize the following random choice

among nodes. We also added the node d' to simulate the exit of the network obtaining this modified *transition matrix*:

$$\bar{\Lambda} = \begin{pmatrix} 0 & \lambda & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3/4 & 3/4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/4 & 1/4 & 2/4 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Every time the global Poisson Clock ticks we select one node in the system with a random choice assigning probabilities to be picked based on the number of particles and velocity of the node. The probability distribution is given by Eq. 2.1.

After selecting the node, we move our dynamics using the matrix $\bar{P}$ aforementioned, adding one particle in the following node and removing one particle in the selected node.

3.1 Part (a)

- *Simulate the system for 60 time units and plot the evolution of the number of particles in each node over time with input rate $\lambda = 100$.*
- *What is the largest input rate that the system can handle without blowing up?*

Given the assumption mentioned above that we did in the introduction, considering $global\_freq = max(\lambda, \omega_{max} \cdot n)$, we set λ as 100, max_{time} for the simulation as 60 and the passage of time using $t = -\frac{\ln(U(0,1))}{global\_freq}$.
The plots can be seen in Fig. 3.2 and Fig. 3.3

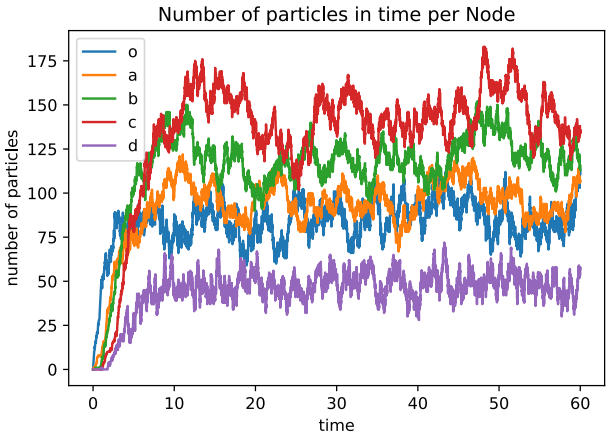

Figure 3.2: Particles per node for proportional rate

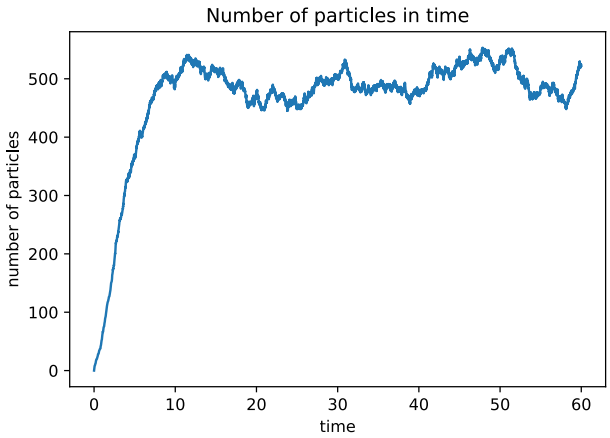

Figure 3.3: Total particles in the system for proportional rate

Note that $\omega_s = \lambda$, $\omega_o = 1.5$, $\omega_a = 1$, $\omega_b = 1$, $\omega_c = 1$ and $\omega_d = 2$. Indeed we can notice from Fig. 3.2 that the most congested nodes are c and b. This is mainly because the in-degree w^- of c and b is *2* and these nodes have small ω with respect to the others.
From Fig. 3.3 we notice that the dynamic regarding number of particles increases in the initial time units and then tend to converge. Indeed the dynamic is prevailed by the input rate λ in the first steps and then it will be prevailed by the frequency of the system, meaning that the number of particles that enter in the system is more or less equal to the number of particles that pass from o to d'.
As we discussed in the following Ex. 3.2 the system *converge* and does not blow up when the value $\frac{1}{\lambda}$ (i.e. the average time of a new particle's entrance) is greater than τ_o, so when it is greater than the hitting time for d' : $E_o[T_{d'}]$. So if the particles require more time to enter than the time to exit, the system will converge

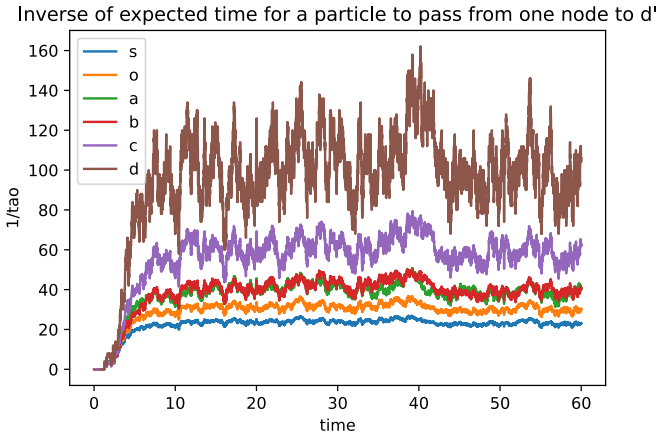

Figure 3.4: Inverse of Hitting time for proportional rate with $\lambda = 100$

We also plotted the graph for the Hitting time for the node $d^{'}$, so the expected time

for a particle to exit the system. We noticed that in the oscillating phase, where the dynamics oscillate around a stable number of particles in the system, we observed that τ_d is equal to the input rate λ

3.2 Part (b)

- *Simulate the system for 60 time units and plot the evolution of the number of particles in each node over time with input rate $\lambda = 1$.*
- *What is the largest input rate that the system can handle without blowing up? Motivate your answer.*

In this case, we have set $\lambda = 1$ and we changed the probability distribution of random assigning a node when the global Poisson Clock ticks to:

$$\eta(i,t) = \frac{\omega_i}{\sum_k \omega_k}$$

We can see this formula as a special case of 2.1 where the velocity of one node does not depend on the number of particles.

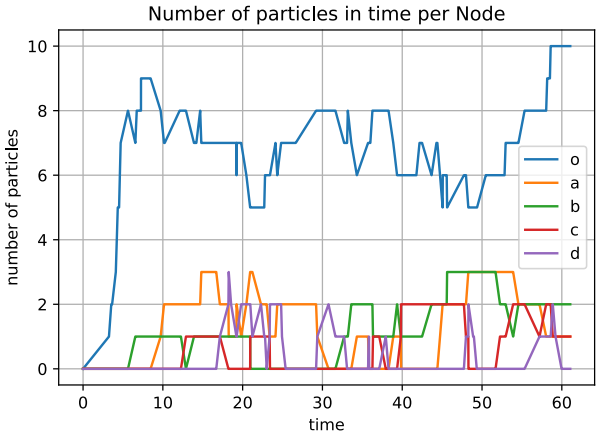

Figure 3.5: Total particles per node in time in the system for fixed rate $\lambda = 1$ and $t = 60$

We can notice from Fig 3.5 and 3.6 that the fixed rate system can not handle the input rate $\lambda = 1$. Note that we even increased the time horizont to be sure that the system diverge for $\lambda = 1$ as it can be seen in 3.7 and 3.8. So we ask ourselves what is the threshold for which the number of particles converge and does not diverge?
As said before in Ex. 3.1 we need that the average time for the particle to enter in the system (i.e passing from s to o) must be greater than the average for the particle to leave the system(i.e. passing from o to d'). In a more rigorous manner:

$$\frac{1}{\lambda} > \tau_o$$

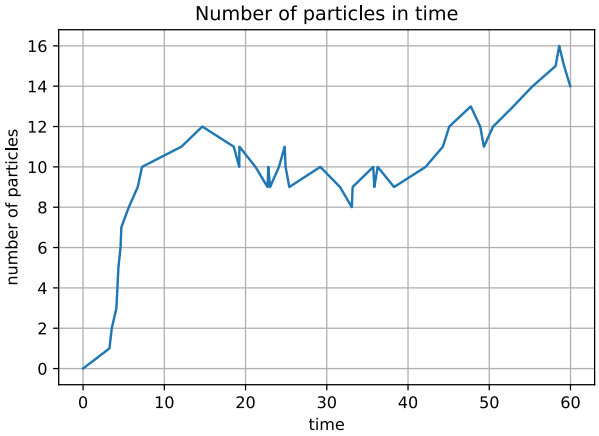

Figure 3.6: Total particles in time in the system for fixed rate $\lambda = 1$ and $t = 60$

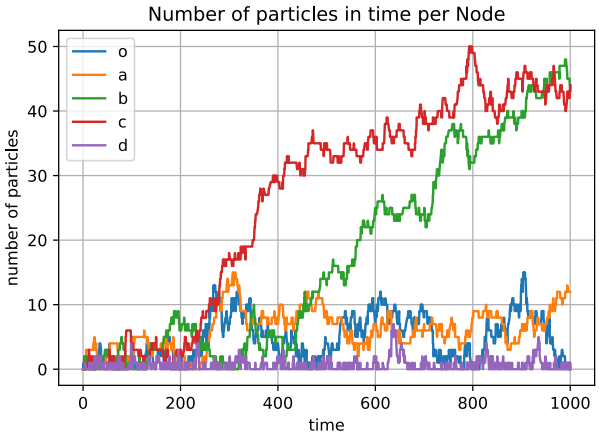

Figure 3.7: Total particles in time per node for fixed rate $\lambda = 1$ and $t = 1000$

With $\tau_o = E_o[T_{d'}]$ and $\frac{1}{\lambda} = E_s[T_o]$

We computed τ_o like we did in Ex. 1.4 using $P = diag(\omega)^{-1}\bar{\Lambda}$ and $\mathcal{S} = d'$. We found out that τ_o is equal to **3.041** and does not depend on λ as expected. So the input rate for which the system does not blow up is $\lambda < 0.327$ and we can notice in Fig. 3.9 and Fig. 3.10

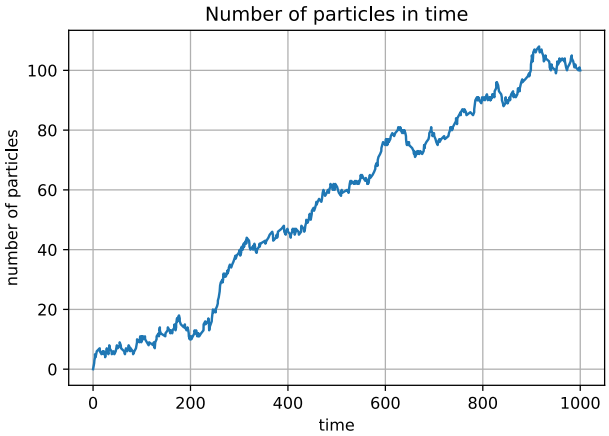

Figure 3.8: Total particles in time in the system for fixed rate $\lambda = 1$ and $t = 1000$

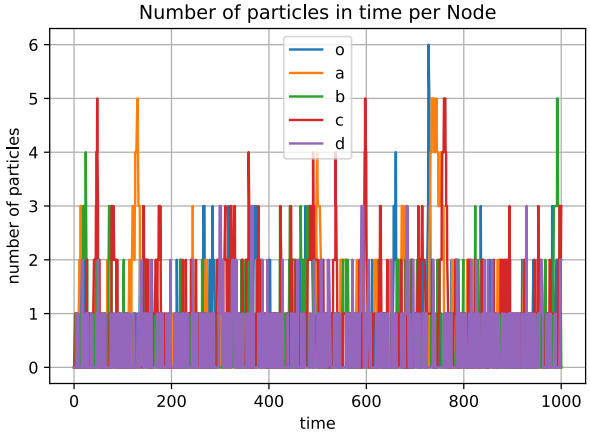

Figure 3.9: Total particles in time in the system for fixed rate $\lambda = 0.3$

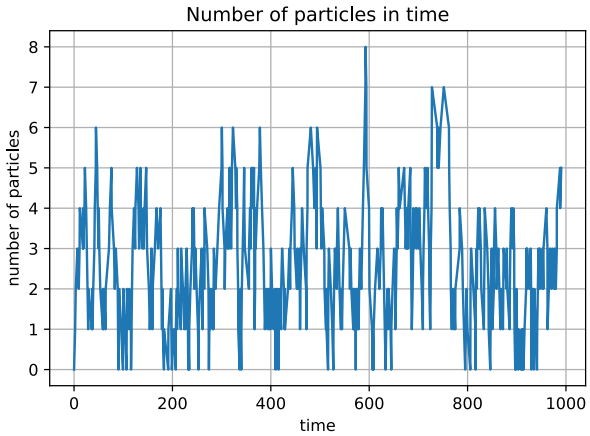

Figure 3.10: Total particles in time in the system for fixed rate $\lambda = 0.3$