Network Dynamics and Learning - Homework #2

Hajali Bayramov - s288874@studenti.polito.it

December 12, 2021

Problem 1.

The first part of this assignment consists of 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:

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

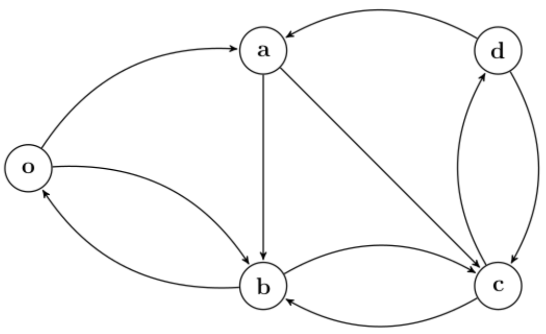

Figure 1: Closed network in which particles move according to the transition rate matrix (1)

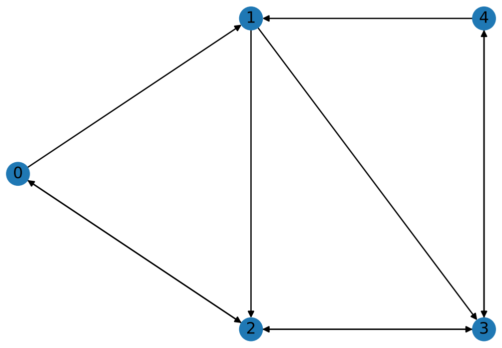

Figure 2: The same network drawn using netwrokX

We will simulate a particle moving around in the network in continuous time according to the transition rate matrix (1) with Poisson clocks whose rates are proportional to the out degrees of each node.

a) Average return time to node a

After simulating the random walk according to (1), starting from node o, for a large enough number of times so that the result would be statistically reliable, we calculate the mean over the return times to o and report the experimental result as below:

$$E_a^{exp}[T_+^a] = 6.724 \tag{2}$$

b) Comparison of the experimental and theoretical return times

We can compare the experimental result obtained in the previous section, with the theoretical formula for return time which is give in (3).

$$\mathbb{E}_i[\bar{T}_i^+] = \frac{1}{\omega_i \bar{\pi}_i} \tag{3}$$

where π_i is the i^{th} element of the unique invariant distribution of the graph. From theory, we know that this distribution is the left dominant eigenvector of the graph's transition probability matrix P. We can calculate this matrix by normalizing Lambda (1) and adding to it a diagonal part. The transition probability matrix is given below:

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

Therefore, we can calculate the invariant distribution as below:

$$\bar{\pi} = \begin{bmatrix} 0.185 \\ 0.148 \\ 0.222 \\ 0.222 \\ 0.222 \end{bmatrix} \tag{5}$$

Now, we can calculate the theoretical return time for node a by using equation (3):

$$\mathbb{E}_a[T_a^+] = \frac{1}{0.148} = 6.75 \tag{6}$$

It can be observed that the experimental and theoretical results are similar.

c) Average hitting time

In this part we calculate the average hitting time from node o to node d. To do so, we will again simulate the random walk several times, and calculate the mean over the simulations. The result is given below:

$$\mathbb{E}_o^{exp}[T_d^+] = 8.759 \tag{7}$$

d) Comparison of the experimental and theoretical hitting times

Now, we can validate the experimental result obtained above with the theoretical result. From theory, we know that the expected hitting time starting from a node in $\mathcal{V}$ to a node in a subset $\mathcal{S}$ can be found by solving the following linear system of equations:

$$(I - Q)\tau = \mathbb{1} \tag{8}$$

where $R = \mathcal{V} \backslash S$ and $Q = P_{|R \times R}$ and $\tau \in \mathbb{R}^R$. Here, τ_i represents the hitting time vector from nodes in S to the i^{th} node. By considering node d as the subset S, and solving the linear system, we will have the vector below:

$$\tau_d = \begin{bmatrix} 8.786 \\ 7.143 \\ 7.071 \\ 3.357 \\ 0. \end{bmatrix} \tag{9}$$

Therefore, the theoretical hitting time from o to d is the first element of this vector (8.786) which is similar to the experimental time calculated in part c.

e) French DeGroot Dynamics and Consensus Value

We can 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 $x(0)$. Since the graph is currently strongly connected and aperiodic, from theory we expect that the opinions converge to a single value. Indeed, we observe that this happens. For example, for the arbitrary initial vector in (10), we will reach the opinion vector in (11) which shows a single consensus value. Fig. 3 shows the simulation process.

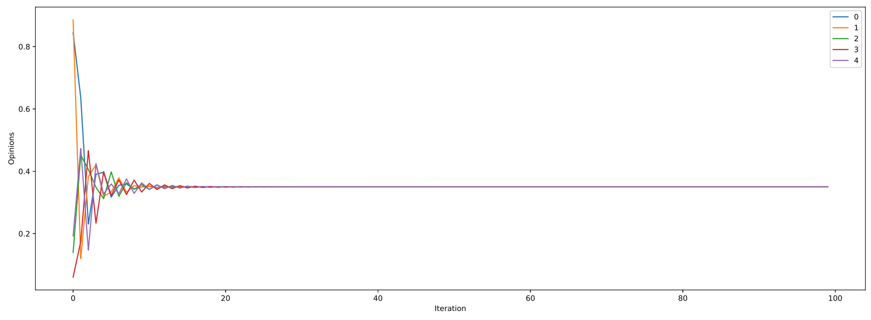

Figure 3: Opinions of nodes during the simulation

$$x_0 = \begin{bmatrix} 0.844 \\ 0.886 \\ 0.139 \\ 0.061 \\ 0.192 \end{bmatrix} \tag{10}$$

$$x_{conv} = \begin{bmatrix} 0.350 \\ 0.350 \\ 0.350 \\ 0.350 \\ 0.350 \end{bmatrix} \tag{11}$$

f) Wisdom of Crowds

Now, we 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. For example, $x_i(0) = \mu + N_i$ where $E[N_i] = 0$ and $Var(N_i) = \sigma^2$. We can calculate the variance of the consensus value by first calculating the graph's invariant distribution, and then we can form the theoretical consensus vector and compute its variance:

$$x_i(t) \to \bar{\pi} x(0) = \Sigma_k \pi_k (\mu + N_k) = \mu + \Sigma_k \pi_k N_k \tag{12}$$

and we have:

$$Var(\Sigma_k \pi_k N_k) = \sigma^2 \Sigma_k \pi_k N_k \tag{13}$$

This is the formula we will use for calculating the variance.

Furthermore, since $\mathcal{G}$ is strongly connected, for every i, $\pi_i > 0$, and we can say that $\Sigma \pi_k^2 < \Sigma \pi_k = 1$. Therefore, we will have:

$$Var(\Sigma_k \pi_k N_k) < \sigma^2 \tag{14}$$

which means that the crowd is always wiser than an individual (or we can say it is less noisy and more dogmatic!).

The interesting fact is that the based on (13), the decrease in variance is somehow proportional to the number of nodes in the graph. For example, when we chose the initial vector variance as 15, we reached a variance of 3.204 for the consensus variance calculated using (13) which is almost $1/5$ (one over the number of nodes) the original variance.

In the limit (when we have a very large population), we can say that we reach asymptotic wisdom because

$$\lim_{x \to \infty} max_k \pi_k = 0$$

and the variance in (13) will become zero.

We can also calculate the variance of the consensus value experimentally. We run the simulation for many times (in our case 10000 times) with different initial vectors (values sampled from a distribution with variance σ^2) and then compute the variance of the converged consensus values. Again, with initial variance of 15, we reach a consensus variance 3.205 that is similar to the theoretical results. Note that the probability distribution from which we sample the initial values does not matter (for example we tried both normal and uniform distributions), and it is only the variance that is important.

g) Consensus in Presence of a Sink with One Node

Now, we remove the edges (d, a) and (d, c) from the graph (Figure 5) and analyze the asymptotic behaviour of the dynamics. In this case, the graph is not strongly connected, however, because it possesses a globally reachable aperiodic sink component, from theory we know that

$$\lim_{t \to \infty} x_i(t) = \bar{\pi} x(0)$$

for every i. Since $\bar{\pi}$ has support only on the globally reachable component, the only non-zero components of this vector will be the ones belonging to the sink component. In other words, the consensus value only depends on the initial value of the sink nodes and other nodes have no influence. In this case, because the sink component has only one node, the consensus value will be the same as the initial value of the sink node. This statements can also be confirmed also by simulation. The following initial vector,

$$x(0) = \begin{bmatrix} 0.844 \\ 0.886 \\ 0.139 \\ 0.061 \\ 0.192 \end{bmatrix} \tag{15}$$

will result in the following (Note that for the new graph (Figure 5), $\bar{\pi} = [0, 0, 0, 0, 1]^T$):

$$\bar{\pi}x(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.192 \end{bmatrix} \tag{16}$$

and the results of the French DeGroot dynamics after 100 iterations will be:

$$x(t) = \begin{bmatrix} 0.192 \\ 0.192 \\ 0.192 \\ 0.192 \\ 0.192 \end{bmatrix} \tag{17}$$

Fig. 4 shows the simulation process.

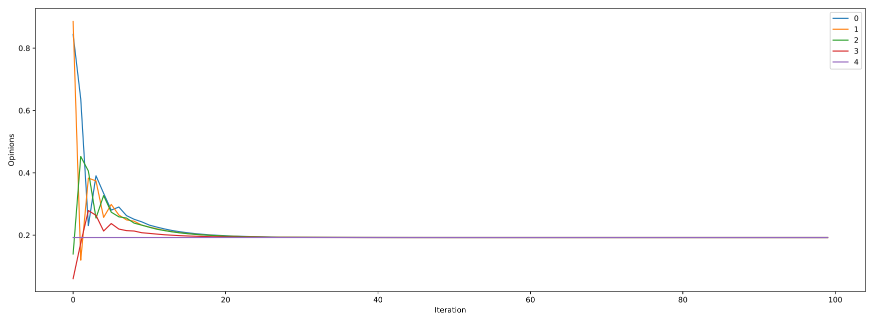

Figure 4: Opinions of nodes during the simulation

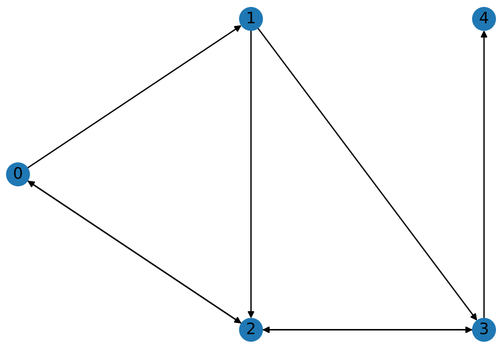

Figure 5: The graph with a single-node sink component drawn using networkX

Again, if we 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, we can rely on equation (13) to calculate the variance of the consensus value. In this case, since $\bar{\pi}$ has only support on a single node (it is a unit vector), using equation (13), we can say that $\Sigma\pi_k^2 = \Sigma\pi_k = 1$. Therefore, we will have:

$$Var(\Sigma_k \pi_k N_k) = \sigma^2 \tag{18}$$

In other words, when we have a single-node sink, wisdom of the crowd does not materialize and the initial variance does not increase because the final opinion of the crowd is dependent only on a single random variable that is the opinion of the sink node (the dictator), and its variance will also remain the same. This can be confirmed also by simulation. For example, when we chose the initial vector variance as 15, we reached a variance of 15.058 for the variance of the consensus values over 10000 simulations.

h) Consensus in Presence of a Sink with More than One Node

Now, we remove the edges (c, b) and (d, a) and again analyse the French-DeGroot dynamics on the new graph. Here, we again have a single globally reachable component in the graph (consisting of c and d), and the same arguments given in part g also apply here. The difference is that the sink component now has two nodes. Again, the value of other nodes does not influence the final consensus values, however this time we will not have a single consensus value. Instead, each sink node will converge to its own initial value and the rest of the nodes will fluctuate between the initial values of the sink nodes without convergence. An example simulation result is given below:

initial vector:

$$\bar{\pi}x(0) = \begin{bmatrix} 0.844 \\ 0.886 \\ 0.139 \\ 0.061 \\ 0.192 \end{bmatrix} \tag{19}$$

will result in the following (Note that for the new graph (Figure 6), $\bar{\pi} = [0, 0, 0, 0.5, 0.5]^T$):

$$x(t) = \begin{bmatrix} 0.121 \\ 0.116 \\ 0.162 \\ 0.061 \\ 0.192 \end{bmatrix} \tag{20}$$

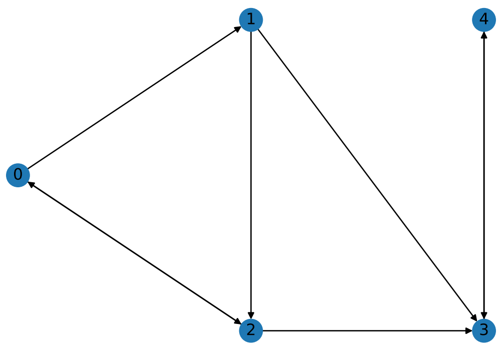

Figure 6: The graph with a two-node sink component drawn using networkX

Fig. 7 shows the simulation process.

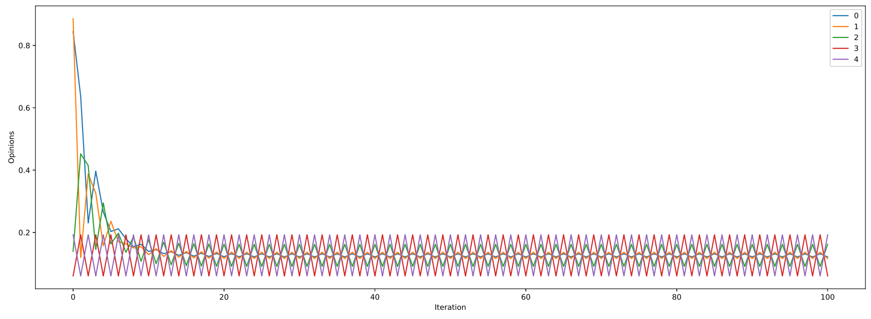

Figure 7: Opinions of nodes during the simulation

Problem 2.

In this part we will again consider the network of 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 $1/\omega_i$ time-units in node i before moving to one of its out neighbors. The next node it will visit is based on the probability matrix $P = diag(\omega)^1\Lambda$, where $\omega = \Lambda\mathbb{1}$.

a) Particle Perspective

We consider 100 particles starting from node a and moving around in the network. When doing so, we look at the network from the particle perspective. Therefore, we follow each of the particles, and calculate its return time to node a. The average over these 100 values will be 6.749. This number is the same as the experimental and theoretical results we obtained in problem 1. This is reasonable because particle movements are independent from each other, and moving different particles in the network and averaging over their return times is essentially the same as simulating a single particle random walk several times and calculating the average return time, or equivalently the theoretical return time calculated in problem 1.

b) Node Perspective

Now, again, we consider 100 particles starting in node o, and the system is simulated for 60 time units. In this simulation, we use dynamic Poisson clocks that determine when each state moves one particle to another state according to the transition matrix. The rate of these Poisson clocks is proportional to the number of particles currently in the node. Alternatively, we could have used a global clock as explained in the homework instructions. For analysis, we calculate the average number of nodes in each node over the last 1000 transitions of the simulation. Note that as will be illustrated in a plot later, the number of nodes in each node over the course of simulation stabilizes very soon (after around 5 time units or equivalently around 400 transitions), and the normalized values of these numbers fluctuate around the stationary distribution of the continuous random walk (unique invariant distribution of the graph or $\bar{\pi}$). Therefore, calculating the average number of particles over the final transitions will give us a vector almost proportional to $\bar{\pi}$ in (5):

$$states = \begin{bmatrix} 17.181 \\ 14.581 \\ 22.003 \\ 22.831 \\ 23.404 \end{bmatrix} \tag{21}$$

We can also normalize this vector, and get something close to $\bar{\pi}$:

$$states_{norm} = \begin{bmatrix} 0.171 \\ 0.146 \\ 0.220 \\ 0.228 \\ 0.234 \end{bmatrix} \tag{22}$$

Below, from figure 8 you can see a plot showing the number of particles in each node during the simulation time.

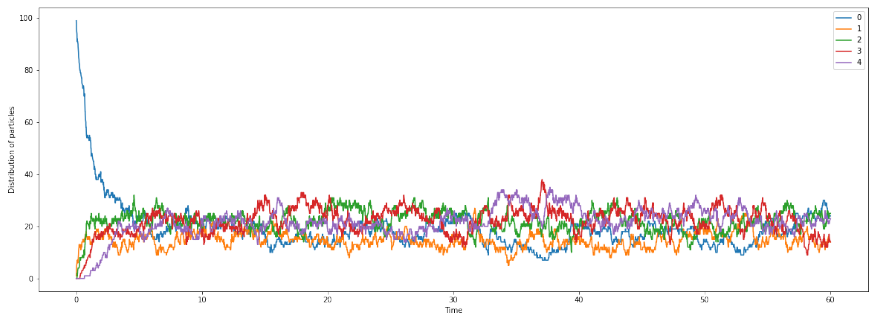

Figure 8: Number of particles in each node during the simulation

As discussed above, the number of particles stabilizes in a few hundred transitions, and the distribution of particles over the network fluctuates around the stationary distribution of the continuous-time random walk. This is reasonable because we can interpret the stationary distribution as a centrality measure or node importance, and also the probability distribution regarding the location of particles in the network in a random walk.

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 9, with transition rate matrix Λ_{open} according to (23).

$$\Lambda_{open} = \begin{array}{c} \begin{array}{ccccc} o & a & b & c & d \end{array} \\ \begin{pmatrix} 0 & 2/3 & 1/3 & 0 & 0 \\ 0 & 0 & 1/4 & 1/4 & 2/4 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \begin{array}{c} \\ o \\ a \\ b \\ c \\ d \end{array} . \tag{23}$$

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 we did in Problem 2 with the "node perspective". Input and output are simulated by following two main rules: if input clock ticks first new particle is added, if clock in node d ticks, number of particles is simply decreased by a unit.

a) Proportional Rate

In this part, the rate of the Poisson clock of each node is equal to the number of particles in it.

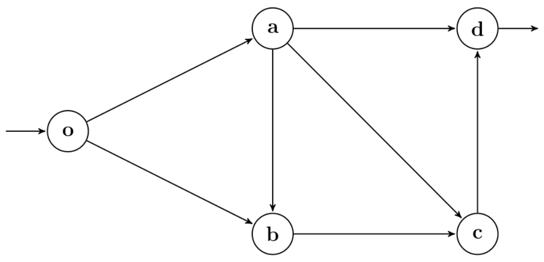

Figure 9: The open network

Now, we simulate the system for 60 time units and plot the evolution of the number of particles. In this simulation, each and every node has its own Poisson clock attached to itself and the rates are proportional to particles inside the node. Also, there is another Poisson clock with fixed rate which determines if a new particle shall enter the system or not. Decision is made depending on whose clock ticks first. So at the beginning, since there are no particles inside nodes, the input clock ticks first. When any node has at least the same number of particles as the input rate, its chance of moving its particles in the network will become equal to the input clock. In this manner, the system is never going to blow up because as soon as the first node has particles with the same number as the input rate, it starts to move its particles and eventually the particles will exit the network through node d. With a higher input rate, it would take longer to equalize the flow of the network's input and output, but sooner or later, it will happen and the nodes start to exchange particles and the total number of nodes in the network remains stable. This comparison can be shown more clearly in the following figures:

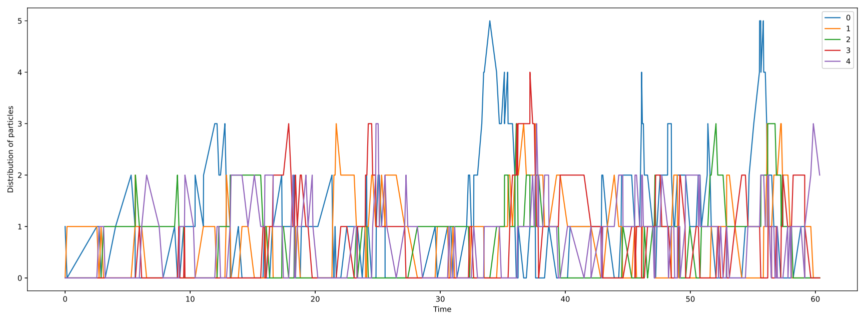

Figure 10: Distribution of particles among nodes with input rate 1

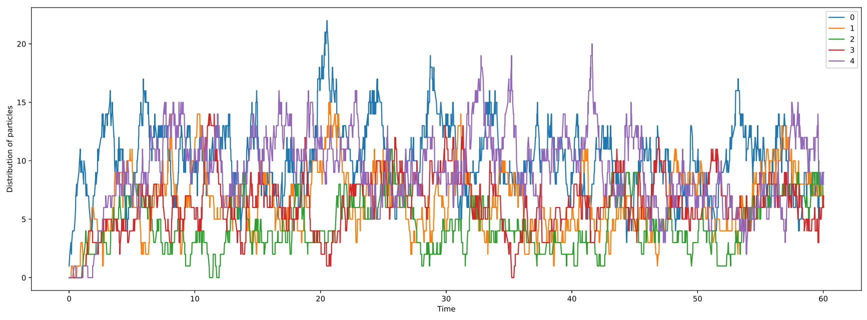

Figure 11: Distribution of particles among nodes with input rate 10

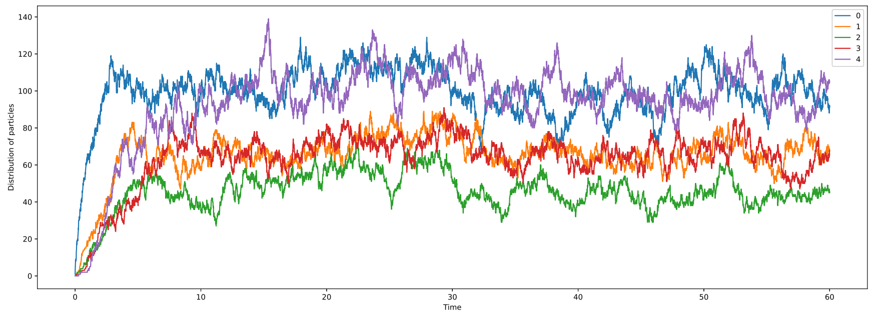

Figure 12: Distribution of particles among nodes with input rate 100

Difference can be felt from figure 12 where new particles are clearly added until node o (0) reaches 100 particles and the chance of moving particles inside the network starts to equalize the input rate.

b) Fixed Rate

In this part, the rate of the Poisson clock of each node will be fixed and equal to one.

The same simulation is done for the fixed rate case. However, in this particular case, clock rates of nodes is fixed at 1. For rates less than 1, the network may even become empty for some time points. For the input rate 1, there is not much difference from the first case, and the input-output flow will become equal. In this simulation, chances of ticking first for the clocks of inside nodes and input are equal. So in some sense movement of particles is randomly organized. This case is shown in 13.

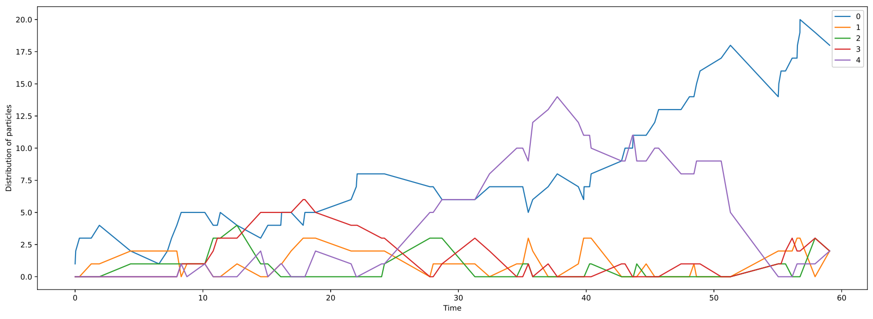

Figure 13: Distribution of particles among nodes with fixed rate 1

However, even a slight increase in the input rate to more than 1, results in the network blowing up. It can be explained by the fact that in this simulation, clock rates of nodes are not dynamic and when they have a value below the input rate they get almost no chance to tick first. Ergo, the system is always receiving new particles but not moving them to the exit node fast enough to stabilize the flow and the network will explode with particles. Fig. 14 visualizes the blow-up situation.

Notes

For drawing graphs in this assignment, the Python package NetworkX (Hagberg, Swart, & S Chult, 2008) was used.

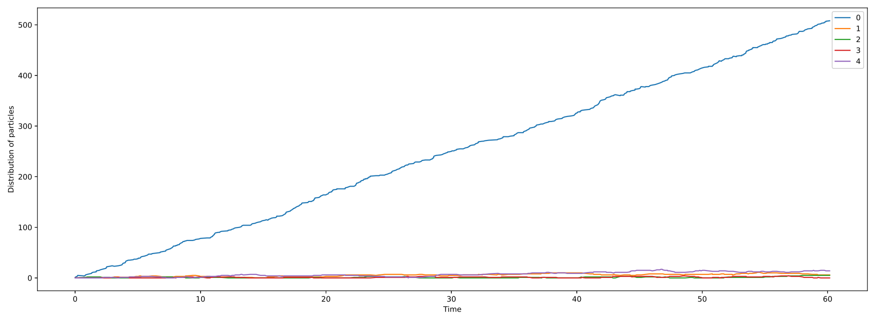

Figure 14: Distribution of particles among nodes with a fixed rate of 1 for nodes, and 10 for the input

I have collaborated with Hafez Ghaemi in solving the exercises and writing the report.

References

Hagberg, A., Swart, P., & S Chult, D. (2008). *Exploring network structure, dynamics, and function using networkx* (Tech. Rep.). Los Alamos National Lab.(LANL), Los Alamos, NM (United States).