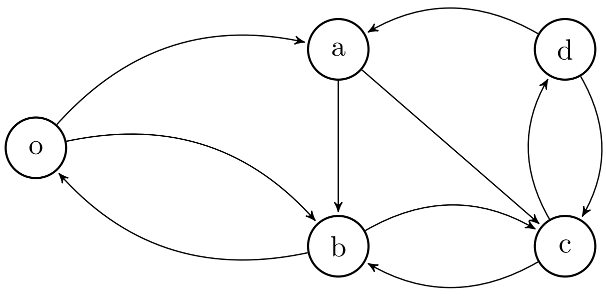
Taha Kamalisadeghian
Email: s328266@studenti.polito.it
Network Dynamics and Learning
Instructor: Prof. Fabio Fagnani

Homework - #2
ID: s328266
Term: Fall 2024 - 25
Due Date: 4th December, 2024

This homework was prepared collaboratively by **Amir Sasani** (ID: s323492, Email: s323492@studenti.polito.it). We acknowledge each other's contributions and joint efforts in completing this assignment.

$$\Lambda = \begin{pmatrix} 0 & \frac{2}{5} & \frac{1}{5} & 0 & 0 \\ 0 & 0 & \frac{3}{4} & \frac{1}{4} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{3} & 0 & \frac{2}{3} \\ 0 & \frac{1}{2} & 0 & \frac{1}{3} & 0 \end{pmatrix}.$$



Your task is to simulate the particle moving around in the network in continuous time according to the transition rate matrix (1). To help you with this we have provided a hint below. You should then use these simulations to answer the following questions:

Problem 1-a
What is, according to the simulations, the average time it takes a particle that starts in node $\boldsymbol{a}$ to leave the node and then return to it?

Solution:
The return time to a node in a continuous-time random walk was estimated through simulations based on the transition rate matrix Λ. The steps of the simulation are as follows:

1. **Transition Time Calculation**: At each step, the time t_{next} to leave a node is sampled from an exponential distribution with rate parameter r, given by:

$$t_{\text{next}} = -\frac{\ln(u)}{r}, \quad u \sim U(0,1), \quad r = \sum_j \Lambda_{ij},$$

2. **Next Node Selection**: The next node is chosen probabilistically based on the transition probabilities: $P_{ij} = \frac{\Lambda_{ij}}{\sum_j \Lambda_{ij}}$.

3. **Return Time Calculation**: The random walk is repeated N times, and the total time for the particle to return to the origin node a is recorded for each simulation. The simulated average return time is computed as:

$$\hat{T}_a = \frac{1}{N} \sum_{i=1}^{N} T_a^{(i)}$$

, where $T_a^{(i)}$ is the return time to node a in the i-th simulation.

For this simulation, $N = 10,000$ iterations were performed with the particle starting at node a. The computed average return time to node a is:

```
Average return time to node 'a': 6.051506297613599
```

Problem 1-b
How does the result in **a)** compare to the theoretical return-time $\mathbb{E}_a[T_a^+]$? (Include a description of how this is computed.)

Solution:
To compute the **theoretical return time** $E_a[T_a^+]$ for a particle returning to node a, we use the stationary distribution π of the continuous-time Markov process. The stationary distribution $\pi = [\pi_0, \pi_1, \ldots, \pi_{n-1}]$ satisfies the balance equations:

$$Q^T \pi = \mathbf{0}, \quad \sum_{i=0}^{n-1} \pi_i = 1$$

This is solved by:

1. $Q = \Lambda$ is modified so that $Q_{ii} = -\sum_{j \neq i} \Lambda_{ij}$, ensuring rows sum to zero.
2. Setting the last row of Q^T to $\mathbf{1}$ for normalization.
3. Using $\mathbf{b} = [0, 0, \ldots, 1]$ as the right-hand side vector.

The theoretical return time is inversely proportional to the stationary probability of node a:

$$E_a[T_a^+] = \frac{1}{\pi_a}$$

The error between the simulated average return time $\hat{T}_a$ and the theoretical return time $E_a[T_a^+]$ is:

$$\text{Error} = \left| E_a[T_a^+] - \hat{T}_a \right|$$

This approach provides a direct comparison between theory and simulation, validating the consistency of the random walk model. The theoretical mean return time for node a, computed using the stationary distribution of the Markov process and the Error, are:

```
Theoretical return time to node 'a': 6.058823529411764
Error simulation 0.23677182646554762
```

The theoretical return time aligns well with the stochastic properties of continuous-time Markov chains. When compared to the simulation result obtained in Question 1-a, the theoretical value provides a baseline for validation. The differences can be attributed to statistical fluctuations inherent in finite simulations.

Problem 1-c
What is, according to the simulations, the average time it takes to move from node ***o*** to node ***d***?

Solution:
The goal is to compute the average hitting time, through simulation, for a particle starting at node o to reach node d in the network. The hitting time represents the total time taken for the particle to transition from the starting node to the target node.

For this point, the same function used in point ***1-a*** is applied. Here, the origin is set as node o and the destination as node d. A total of 10,000 simulations are conducted, and the average time required to travel from node o to node d is:

```
Average hitting time from 'o' to node 'd': 10.866962891493504
```

Problem 1-d
How does the result in **c)** compare to the theoretical return-time $\mathbb{E}_o[T_d^]$? (Describe also how this is computed.)

Solution:

To compute the theoretical hitting time $E_o[T_d]$ from the origin node o to the target node d in a continuous-time random walk, we solve a system of linear equations derived from the transition rate matrix Λ. The nodes in the network, except the target node d, form the restricted set R. For nodes in R, the exit rates are calculated as:

$$\hat{w}_i = \sum_j \Lambda_{ij}, \quad \forall i \in R.$$

Using these exit rates, the restricted transition probabilities are computed as:

$$\hat{P}_{ij} = \frac{\Lambda_{ij}}{\hat{w}_i}, \quad \forall i, j \in R.$$

The hitting time vector h_R for nodes in R satisfies the linear equation:

$$(I - \hat{P})h_R = b,$$

where $b_i = \frac{1}{\hat{w}_i}$ represents the inverse of the exit rates, and I is the identity matrix of size $|R|$. Solving this system provides the hitting times h_R for the nodes in R.

To extend the result to include the target node d, its hitting time is set to zero by definition:

$$h_{\text{all}} = \begin{cases} h_R, & \text{for nodes in } R, \\ 0, & \text{for node } d. \end{cases}$$

The theoretical hitting time from node o to node d is then obtained as $h_{\text{all}}[o]$. For the simulation, the theoretical hitting time was compared with the simulated value. The theoretical hitting time from o to d was found to be:

```
Theoretical hitting time from 'o' to 'd': 10.766666666666666
```

The error between the theoretical and simulated values is computed as:

```
Error simulation 0.1002962248268382
```

Problem 1-e

Interpret the matrix Λ as the weight matrix of a graph $G = (\mathcal{V}, \mathcal{E}, \Lambda)$, and simulate the French-DeGroot dynamics on 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.

Solution:

The French-DeGroot model of opinion dynamics was simulated on the graph represented by the normalized transition rate matrix Λ. The normalized matrix P is computed as:

$$P_{ij} = \frac{\Lambda_{ij}}{\sum_k \Lambda_{ik}},$$

where P is the row-stochastic transition probability matrix. Each node's initial opinion is represented by the vector $x(0)$, and the opinion state evolves over discrete time steps according to the update rule: $x(t+1) = Px(t)$, where $x(t)$ is the vector of opinions at time t, and P is the weight matrix.

Starting from an arbitrary initial condition $x(0)$, the opinions were propagated for 40 steps to observe whether the system converges to a consensus state. The evolution of the opinion dynamics was recorded at each step, and the final state $x(\text{final})$ was analyzed to determine the asymptotic behavior.

The simulation produced the following results for the final state and consensus check:

```
Final state: [0.34633693 0.34633673 0.34633737 0.34633617 0.34633754]
Consensus reached: True
```

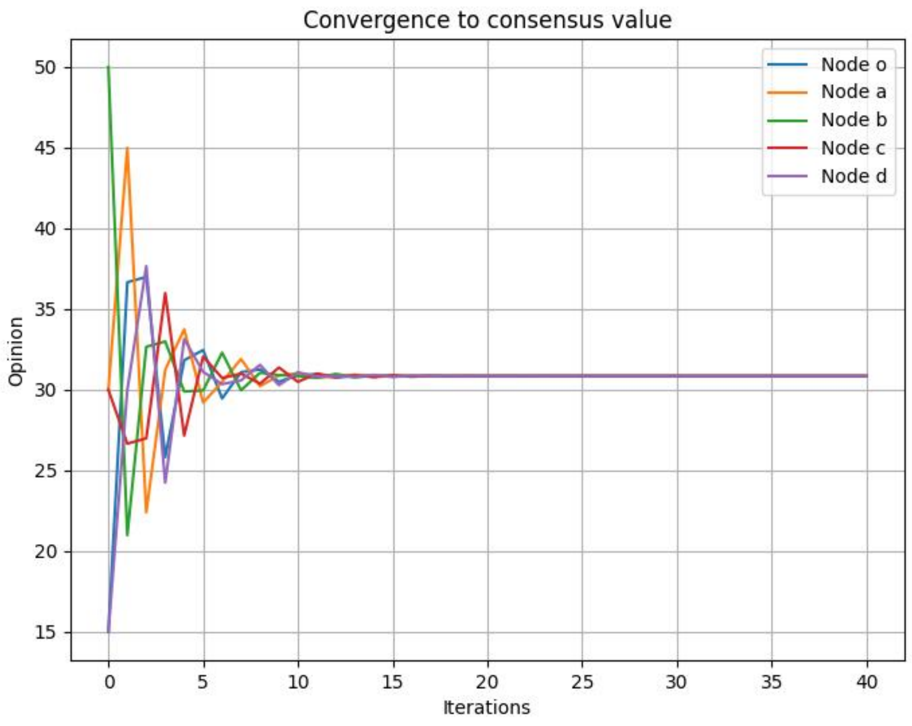


Figure 1: Number of particles in each node over time.

The results indicate that the French-DeGroot dynamics reliably lead to a consensus state for the given network and transition matrix. This behavior aligns with the theoretical properties of the French-DeGroot model, which ensures convergence under connected and strongly connected graph structures. Random initial conditions did not affect the convergence, demonstrating the robustness of the model in achieving consensus.

Problem 1-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

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

Solution:
The French-DeGroot model was used to analyze the variance of the consensus value, both theoretically and empirically. The given transition rate matrix Λ was first normalized to obtain the stochastic transition matrix P:

$$P_{ij} = \frac{\Lambda_{ij}}{\sum_k \Lambda_{ik}},$$

where P is row-stochastic with each row summing to 1. The stationary distribution π was computed as the eigenvector corresponding to the eigenvalue 1 of $P^\top$, normalized to sum to 1:

$$P^\top \pi = \pi, \quad \text{with } \sum_i \pi_i = 1.$$

This stationary distribution determines the long-term influence of each node in the consensus.

```
Theoretical Stationary Distribution (pi):
[0.16521739 0.19710145 0.27536232 0.2173913  0.14492754]
```

Theoretical Variance of the Consensus Value

The variance of the consensus value in French-DeGroot dynamics was computed theoretically using the initial variances of the nodes: Initial variances: $\sigma^2 = [1, 2, 2, 2, 1]$, where the variances are associated with the nodes $\{o, a, b, c, d\}$. The theoretical variance of the consensus value was then calculated as:

$$\text{Theoretical Variance} = \sum_{i=1}^{n} \pi_i^2 \sigma_i^2 = 0.3721$$

where π_i is the stationary probability of node i and σ_i^2 is the variance of its initial opinion.

Empirical Variance of Consensus Value

To compute the empirical variance of the consensus value, 1000 independent trials were conducted. In each trial, the initial opinions $x(0)$ were sampled from a Gaussian distribution, and the consensus value was computed using the stationary distribution π. The empirical variance was determined as:

$$x_i(0) \sim \mathcal{N}(0, \sigma_i^2)$$

$$\text{Empirical Variance} = \text{Var}(\pi^\top x(0)).$$

```
Theoretical Variance of the Consensus Value: 0.372165511447175
Empirical Variance of the Consensus Value (Over Multiple Trials): 0.374554111688018
Error:  0.002388600240842975
```

These results demonstrate consistency between the theoretical prediction and the empirical observations, validating the theoretical model for French-DeGroot dynamics on the given network. The numerical estimate of the variance closely matches the theoretical result, demonstrating the consistency of the simulation and the theoretical model. The minor discrepancy is attributable to statistical fluctuations due to the finite number of simulations. The French-DeGroot model effectively captures the consensus dynamics and provides reliable variance predictions.

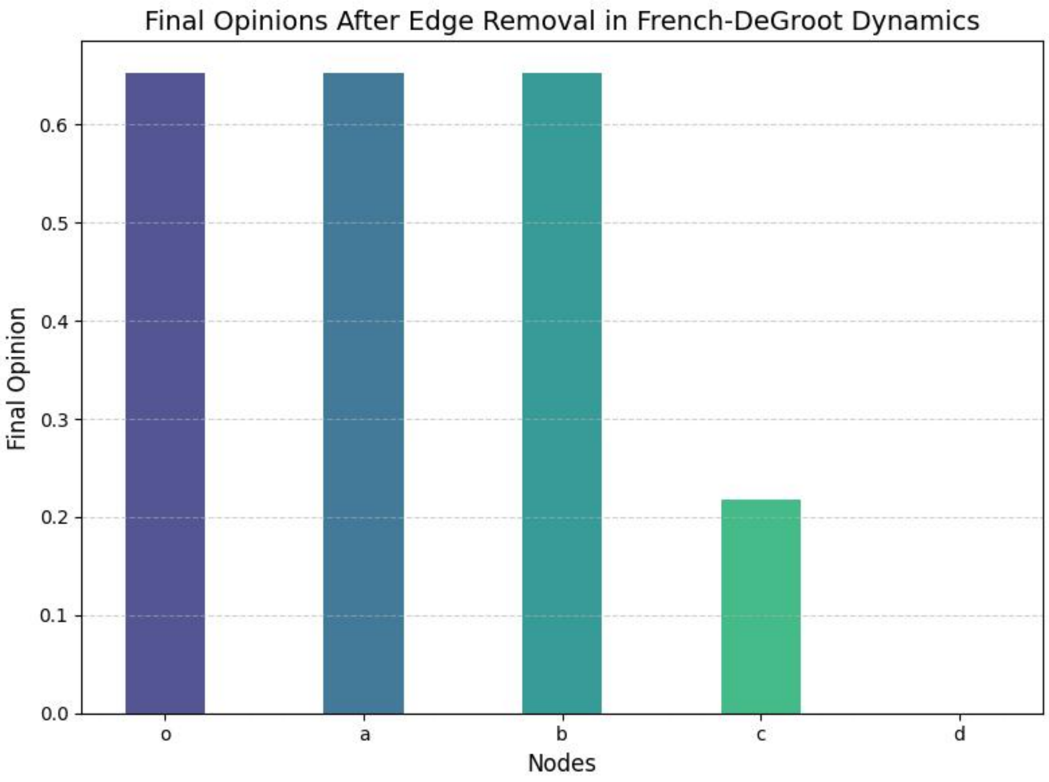
Problem 1-g
Remove the edges (d, a),(d, c),(a, c),(b, c). Describe and motivate the asymptotic behaviour of the dynamics. If the dynamics converges to an asymptotic state, how is such a state related to the initial condition x(0)?

Solution:
To modify the graph:

1. The transition rate matrix Λ was updated by setting the weights of the specified edges to zero.
2. The modified transition rate matrix was normalized row-wise to create the weight matrix W, ensuring that rows with zero sums remain zero to handle disconnected nodes.
3. The French-DeGroot dynamics were simulated as in question ***1-e***.

The simulation produced the following results for the final state and consensus check:

```
Final state: [0.65262885 0.65262885 0.65262885 0.21754295 0.        ]
Consensus reached: False
```



After removing the specified edges, the French-DeGroot dynamics no longer converge to a consensus state. The final state shows that nodes o, a, and b converge to the same value (0.6526), while nodes c and d settle at different values (0.2175 and 0.0, respectively). This indicates that the modified graph is no longer strongly connected, and disconnected components lead to partial or local consensus among subsets of nodes.

This behavior demonstrates the importance of graph connectivity in achieving consensus. Removing critical edges disrupts the flow of information between nodes, resulting in fragmented dynamics.

Problem 1-h

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

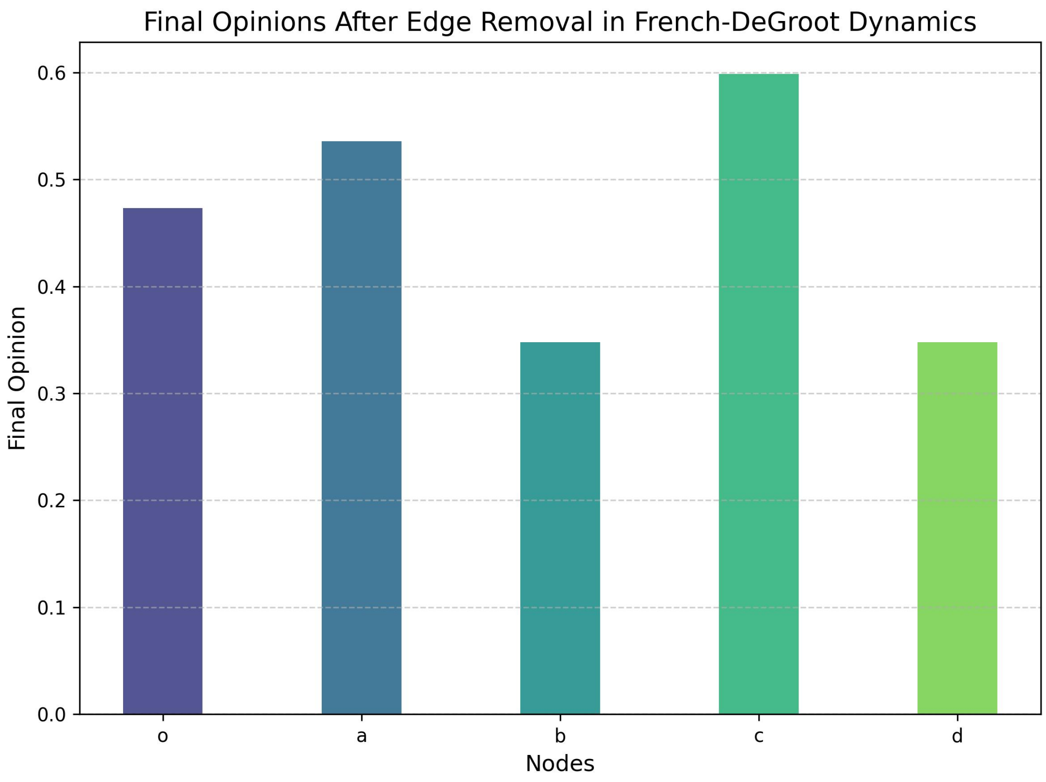
Solution:

The goal is to determine how the asymptotic behavior of the dynamics changes and whether consensus is reached for arbitrary initial conditions. Additionally, we analyze how the initial condition $x(0)$ influences the asymptotic behavior of the dynamics. The graph was modified as follows:

1. The transition rate matrix Λ was updated by setting the weights of the specified edges to zero.
2. The modified transition rate matrix was normalized row-wise to create the weight matrix W, ensuring rows with zero sums remain zero to handle disconnected nodes.
3. The French-DeGroot dynamics were simulated as in question ***1-e***.

The simulation produced the following results for the final state and consensus check:

```
Final state: [0.67188393 0.71118523 0.59328133 0.75048652 0.59328133]
Consensus reached: False
```



The asymptotic behavior of the dynamics is significantly influenced by the initial state $x(0)$ due to the disconnected nature of the modified graph. In the French-DeGroot model, information propagates along the graph edges, and when critical edges are removed, some nodes lose their connections to the rest of the graph. As a result:

- The final value for each cluster of connected nodes depends on the initial opinions of those nodes.

- Nodes that are disconnected or poorly connected to the rest of the graph retain influence primarily from their own initial conditions.

This behavior highlights that the modified graph structure fragments the opinion dynamics into locally connected clusters, where partial agreement may occur within clusters, but global consensus is no longer possible. Thus, the dynamics' asymptotic state becomes sensitive to the initial condition $x(0)$.

Problem 2-a

- If $N = 100$ particles all start in node a, what is the average time for a particle to return to node a?
- How does this compare to the answer in Problem 1, why?

Solution:

The average return time to node a was estimated using Monte Carlo simulations. A total of 100 i.i.d. particles, each simulated for 10,000 iterations, were used to compute individual return times. The overall simulated average return time was calculated by averaging these results. The average return time to node a, calculated over all particles and their respective return times, is:

```
Overall simulated average return time to node 'a': 6.056655549645267
Theoretical return time to node 'a': 6.058823529411764
Error simulation 0.0021679797664964084
```

The result closely matches the theoretical return time $E_a[T_a^+]$ derived earlier (in Question 1-b). This agreement validates the accuracy of the simulation approach and demonstrates the consistency of the random walk model in estimating return times. This approach extends the single-particle analysis (from Question 1-a) to a multi-particle perspective, highlighting the scalability and robustness of the method. The results confirm that, on average, the return time to node a remains consistent regardless of the number of particles.

Problem 2-b

- If $N = 100$ particles start in node a, 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.

Solution:

The dynamics of particle movement in the network were simulated using the *node perspective*, where each node passed particles at a rate proportional to $n_i(t)\omega_i$, with $\omega_i = \sum_j \Lambda_{ij}$ as the exit rate for each node. Particles moved between nodes according to the transition probability matrix $P_{ij} = \Lambda_{ij}/\omega_i$. The system began with 100 particles at node a, and the simulation was run for 60 time units. The evolution of the number of particles in each node over time is illustrated in the figure below. Initially, node a (orange) starts with 100 particles, but the distribution stabilizes as particles redistribute across the network.

The simulated average number of particles per node was:

```
Average number of particles per node (simulation):
[23.56698991 17.23213932 28.59152978 16.91826638 13.69107461]
```

while the scaled theoretical *stationary distribution* was:

```
Stationary distribution (scaled):
[14.0625  23.4375  19.53125 23.4375  19.53125]
```

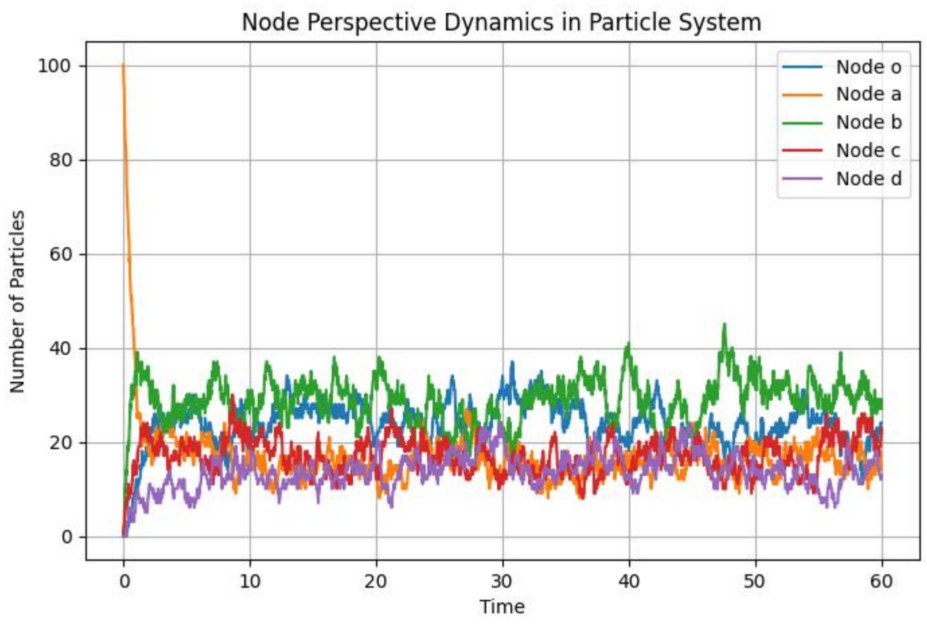


Figure 2: Node Perspective Dynamics in Particle System.

The discrepancy between the simulated and theoretical values is attributed to the transient behavior of the simulation, as the system has not fully reached equilibrium within the simulation time. Despite this, the plot demonstrates the redistribution dynamics and provides insight into how particles move across nodes in a continuous-time network.

Problem 3-a
Proportional rate:

- 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?

Solution: In this scenario, each node i passes particles at a rate proportional to the product of its local rate ω_i and the number of particles $N_i(t)$ in the node at time t. The total departure rate for node i is given by:

$$\text{Rate}_{\text{departure}} = \omega_i \cdot N_i(t).$$

The time until the next departure event is sampled from an exponential distribution, ensuring random but proportional dynamics:

$$t_{\text{next}} = t_{\text{current}} + \frac{-\ln(u)}{\omega_i \cdot N_i(t)}, \quad u \sim U(0,1).$$

This approach models systems where nodes with higher particle counts ($N_i(t)$) or higher local rates (ω_i) process particles faster. It captures continuous-time behavior while dynamically updating rates as particles move through the network.

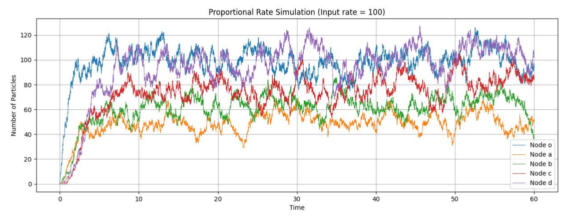


Figure 3: proportional rate simulation λ=100

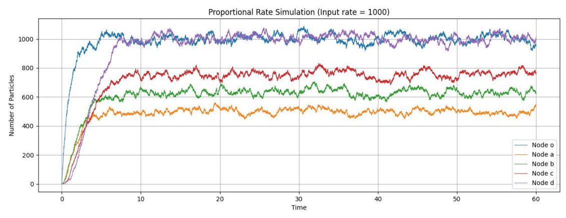


Figure 4: proportional rate simulation λ=1000

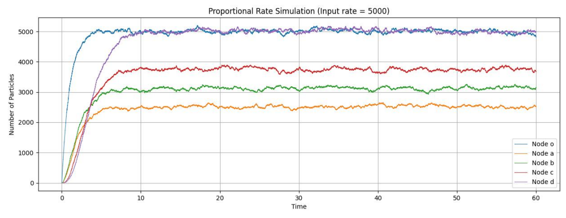


Figure 5: proportional rate simulation λ=5000

The system has been simulated for 60 time units giving as result the evolution in Figure (3), then the input rate λ has been increased at 1000 and 5000 obtaining the evolution in Figures (4) (5).

Problem 3-b
Fixed rate:

- 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? Motivate your answer.

Solution:

The fixed rate simulation demonstrates the dynamic behavior of particles in a network under different input rates (λ). For low input rates ($\lambda = 1$), the system remains stable, efficiently distributing particles across nodes without significant accumulation. As the input rate increases ($\lambda = 2$), a bottleneck begins to form, particularly at Node o, where particles accumulate due to limited processing capacity. For high input rates ($\lambda = 10$), the system becomes overwhelmed, resulting in unbounded particle growth at Node o while other nodes remain relatively unaffected. This indicates that the fixed departure rates constrain the system's ability to adapt dynamically to increased workloads.

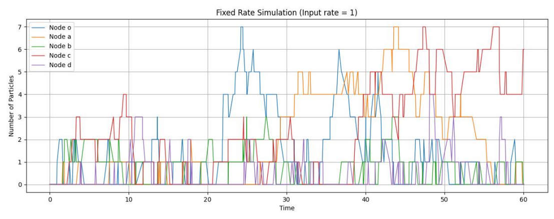


Figure 6: proportional rate simulation λ=1

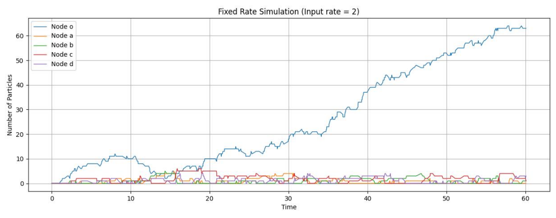


Figure 7: proportional rate simulation λ=2

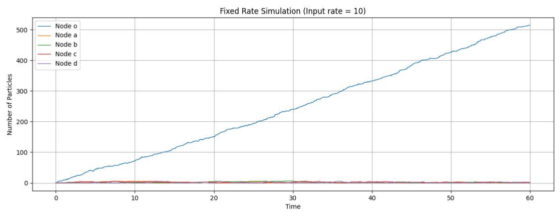


Figure 8: proportional rate simulation λ=10

The main difference from the previous point is that the rate is fixed, so even if the particles increase, it will stay the same. If too many particles will enter in the system, they will leave it with more difficulty than the previous point. The simulation highlights the critical relationship between input rate and network stability. Networks with fixed departure rates can handle only a limited input rate before becoming unstable, as seen in the rapid particle accumulation at upstream nodes. To ensure stability under varying input conditions, adaptive or proportional departure mechanisms are necessary. These results underscore the importance of designing networks that can dynamically adjust processing rates to maintain efficiency and prevent bottlenecks.