Network Dynamics – Homework 2

14 December 2025

Abstract

In this report we present our solution to Homework 2. This report is the result of active collaboration between: Kevin Elezi (s316685), Marco Conte (s346753), Melika Ghasemi (s309495), Mohamed Hatem (s346906) and Xiangxi Li (s336719).

Exercise 1

This exercise analyzes the dynamics of particles and opinions on a network described by a specific directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \Lambda)$. The assignment is divided into two main sections: the first focuses on a single particle performing a Continuous-Time Random Walk (CTRW), and the second analyzes the French-DeGroot opinion dynamics model. The network (Figure 1) consists of 5 nodes $\mathcal{V} = \{o, a, b, c, d\}$ with transition rates defined by the matrix Λ. The goal is to compare numerical simulations (Monte Carlo methods) with theoretical predictions derived from Markov chain theory and algebraic graph theory.

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

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

Question b) How does the result in a) compare to the theoretical return-time $E_a[T_a^+]$?

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

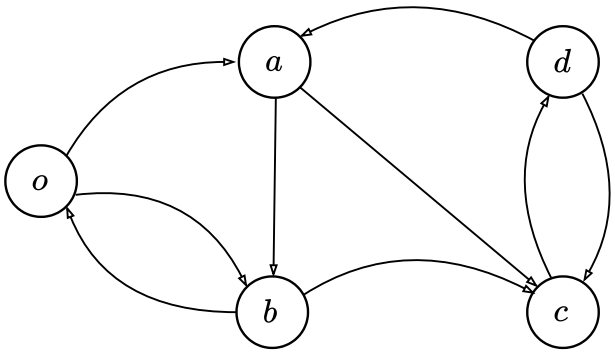

Figure 1: Closed network in which particles move according to the transition rate matrix

Question d) How does the result in c) compare to the theoretical hitting-time $E_o[T_d^+]$?

Question e) 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 $x(0)$. Does the dynamics converge to a consensus state for every initial condition $x(0)$?

Question 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 $\sigma_a^2 = \sigma_b^2 = \sigma_c^2 = 2, \sigma_o^2 = \sigma_d^2 = 1$. Compute the variance of the consensus value, and compare your results with the numerical simulations.

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

Question h) Consider the graph $(\mathcal{V}, \mathcal{E}, \Lambda)$, and remove the edges (b, o) 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.

Question a

The problem to be solved is the average time for a particle to start from node a and return to the same node. To address this, we will proceed in four steps:

1. The particle starts at node a.

2. It immediately leaves node a (departing on the first transition).

3. The time from the moment of departure until the first return to node a is recorded.

4. The average $\tilde{E}_a(T_a^+)$ is taken over $N_{sim} = 1000$ independent experiments.

In the continuous case, the time at which a jump is executed is determined by a Poisson process with rate r. We use a Poisson clock because the time between two ticks is an independent random variable with a Poisson distribution. The particle remains in node i for a random holding time τ which follows an exponential distribution with rate r that is the sum of the out-degrees of the node. To simulate this numerically, we use

$$t_{next} = -\frac{\ln u}{r}$$

where u is a random variable with uniform distribution $u \in \mathcal{U}(0, 1)$. After the holding time elapses, the particle moves to a neighbor j and the next node is selected based on the normalized transition probabilities.

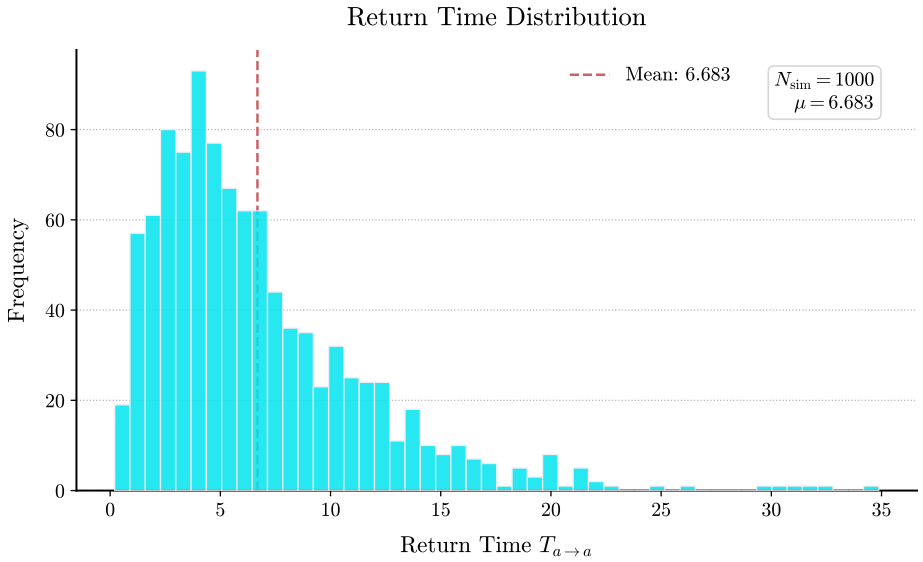

Figure 2: Return time distribution

As Figure 2 shows, the average simulated time $\tilde{E}_a(T_a^+) = 6.683$.

Question b

To compare the theoretical value with the simulated result, we first present the calculation process for the theoretical value. The theoretical return time for a recurrent state in a CTMC is given by

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

where ω_a is the total exit rate from a, and π is the stationary distribution of the embedded discrete-time Markov chain (DTMC). We compute the generator matrix $Q = \Lambda - diag(\Lambda \mathbf{1})$. Solving the linear system $\boldsymbol{\pi} Q = \mathbf{0}$ subject to the normalization constraint $\sum_i \pi_i = 1$ yields the steady-state distribution:

$$\boldsymbol{\pi} = \begin{bmatrix} 0.217 & 0.149 & 0.261 & 0.186 & 0.186 \end{bmatrix}.$$

Using the steady-state probability for node a (where $\pi_a \approx 0.149$) and the total departure rate λ_a^{out}, the theoretical expected return time is calculated as:

$$E_a[T_a^+] = \frac{1}{\pi_a \lambda_a^{\text{out}}} = 6.709\,\text{s}.$$

The calculated theoretical value shows little discrepancy from the simulated value $\tilde{E}_a(T_a^+)$, so they are quite closely aligned. Therefore, both the simulation and the theory are correct.

Question c

Now, we want to simulate the particle that is moving from node o to node d and so we want to compute the hitting time. We proceed with the following three steps:

1. Simulate the continuous-time random walk of a particle starting at node o.
2. Record the time of first arrival at node d.
3. Repeat the simulation N_{sim} times obtaining each time a measurement of the duration of the process to calculate the average hitting time $\tilde{E}_o(T_d)$.

As Figure 3 shows, the average hitting time $\tilde{E}_o(T_d)$ is around 10.833.

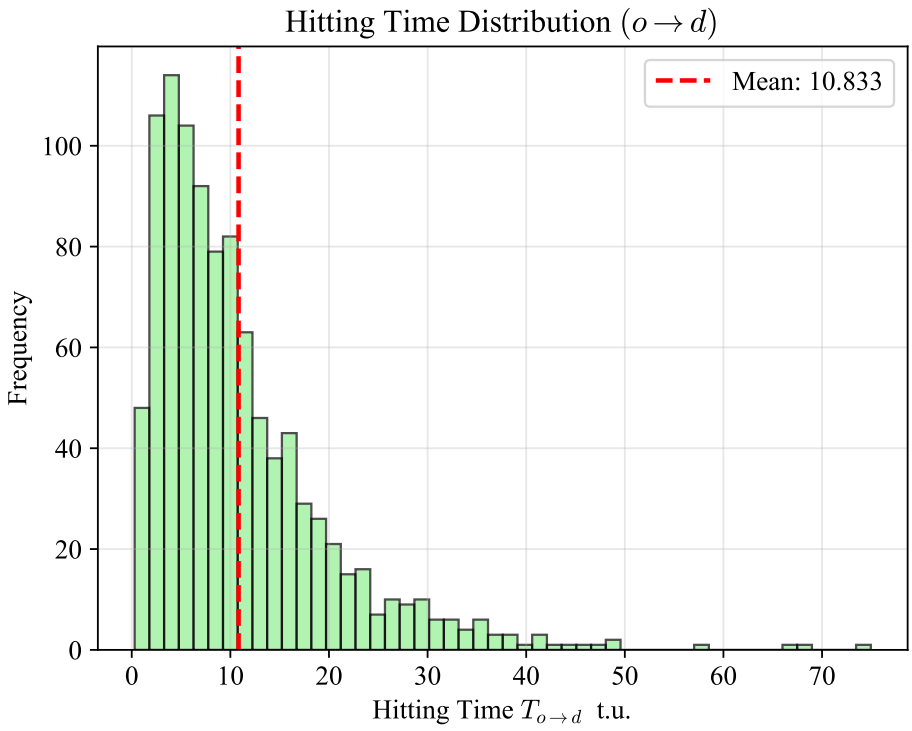

Figure 3: Hitting time distribution

Question d

The expected hitting time vector $E_o[T_d]$, is the solution to the linear system:

$$E_i[T_d] = \frac{1}{\omega_i} + \sum_{j \neq d} P_{ij} E_j[T_d] \quad \forall i \neq d$$

with the boundary condition Lambda for the rate of node i, ω_i. This can be written in matrix form as $(I - P_{-d})E_{-d}[T_d] = \mathbf{w}_{-d}$, because of the boundary condition $E_d = 0$. We compute the matrix P normalizing the matrix where $\mathbf{w}$ contains the mean holding times. Then solve the linear system to obtain the hitting times for all nodes.

Solving the linear system $(I - P_{-d})E_{-d}[T_d] = \mathbf{w}_{-d}$ yields the vector of expected hitting times. Specifically, for the origin node o, we obtain:

$$\text{d) Theoretical } E_o[T_d] = 10.7667\,\text{s}.$$

The theoretical solution yields a value of approximately 10.77, which closely matches the value obtained from our numerical simulation $\tilde{E}_o(T_d)$. This confirms the correctness of the simulation.

Question e

In this part, we interpret Λ as the weight matrix of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \Lambda)$. Opinion dynamics models describe processes where agents update their opinion over time with their neighbors' opinions, weighting the information with respect to the normalized weight matrix $P = D^{-1}\Lambda$. We simulate the French-DeGroot dynamics following the discrete-time update rule $x(t+1) = Px(t)$. Considering the graph in Figure 1, it is noticeable that $\mathcal{G}$ is strongly connected because there is a directed path between any pair of nodes. This ensures, from the Perron-Frobenius theorem, that the matrix P has a unique left eigenvector π associated with the dominant eigenvalue $\lambda = 1$. Moreover, the closed paths of nodes of $\mathcal{G}$ have different lengths, so the greatest common divisor is 1 that makes $\mathcal{G}$ aperiodic. For example $o \to a \to b \to o$ has length $= 3$ or $o \to b \to o$ has length $= 2$ so $per(o) = g.c.d\{2, 3\} = 1$. From a theorem, if $\mathcal{G}$ is connected and aperiodic, then

$$\lim_{x\to\infty} x(t) = \lim_{x\to\infty} P^t x(0) = \alpha \mathbf{1}$$

with

$$\alpha = \pi' x(0) = \sum_k \pi_k x_k(0) \quad \forall i$$

where $\pi = P'\pi$ is the unique invariant distribution of P and α is the consensus value. While the guarantee of convergence is intrinsic to the matrix P (for the properties we said before, the other eigenvalues of P are < 1), the specific consensus value is a convex combination of the initial opinions; the limit exists for any vector $x(0)$. Therefore, the initial opinion of every single agent influences the final consensus. Since the graph is strongly connected, every component of the stationary distribution vector π is strictly positive ($\pi_i > 0$). This means that every node influences the final outcome, that is a democratic weighted average of the initial conditions. According the previous equations, we compute the invariant distribution and the consensus value and in Figure 4 it can be visualize how the evolution of the simulated dynamics converge to the theoretical value of the consensus.

Question f

We analyze the variance of the consensus value by considering the initial state of the dynamics for each node $i \in \mathcal{V}$ as $x_i(0) = \xi_i$, where $\{\xi_i\}_{i\in\mathcal{V}}$ are independent random variables with variances defined as:

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

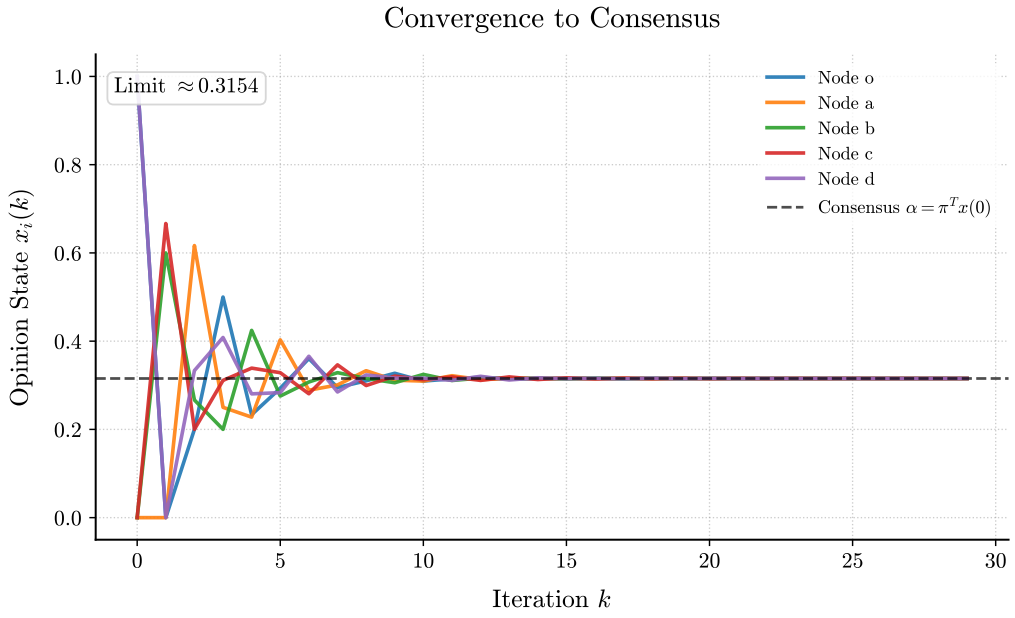

Figure 4: Convergence of the agents to the consensus value

The theoretical variance of the consensus value α depends on the invariant distribution $\boldsymbol{\pi}$ (computed in *Question e*) and is given by the weighted sum:

$$\mathrm{Var}_{\mathrm{th}}(\alpha) = \mathrm{Var}_{\mathrm{th}}\left(\sum_{i=1}^{n} \pi_i x_i(0)\right) = \sum_{i=1}^{n} \pi_i^2 \sigma_i^2. \tag{1}$$

To validate this, we performed $N_{sim} = 10,000$ Monte Carlo simulations. In each iteration, initial states $x(0)$ were sampled from normal distributions with the specified variances, and the resulting consensus variance was compared against the theoretical prediction. The results are summarized below:

$$\mathrm{Var}_{\mathrm{th}}(\alpha) = \sum_{i} \pi_i^2 \sigma_i^2 \approx 0.3694$$

$$\tilde{\mathrm{Var}}_{\mathrm{sim}}(\alpha) \approx 0.3817$$

$$\epsilon_{\mathrm{rel}} = \frac{|\tilde{\mathrm{Var}}_{\mathrm{sim}} - \mathrm{Var}_{\mathrm{th}}|}{\mathrm{Var}_{\mathrm{th}}} \times 100\% \approx 3.33\%$$

The relative error of approximately 3.33% confirms the theoretical derivation. We observed that the error fluctuates between simulation runs occasionally reaching higher percentages which is inherent to the stochastic nature of the Monte Carlo method. Reducing this discrepancy further would require increasing the sample size N_{sim} significantly.

Question g

Removing the edges $(d,a),(d,c),(a,c),(b,c)$ from $\mathcal{G}$ renders the graph not fully connected (Figure 5). In particular, node d becomes a sink node with no outgoing edges to other nodes. Consequently, its row in the weight matrix is all zero, though we assign a self-loop to d to satisfy the row-stochastic property:

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

We simulate the system dynamics $x(t+1) = Px(t)$ using the matrix P derived from this new Λ. Unlike the previous case, the graph now contains two distinct sink components: the isolated node $\{d\}$ and the strongly connected component $\{o,a,b\}$, which forms a cycle and has no outgoing edges to the rest of the graph.

Because the graph is reducible and has multiple sink components, the dynamics do not converge to a global consensus:

$$\lim_{t\to\infty} x_i(t) \neq \pi^\top x(0)$$

Figure 6 illustrates these distinct asymptotic behaviors. For a specific realization of random initial conditions, the system reached the asymptotic state:

$$\lim_{t\to\infty} x(t) \approx \begin{bmatrix} 0.622 & 0.621 & 0.624 & 0.809 & 0.903 \end{bmatrix}^\top.$$

Here, the component $\{o,a,b\}$ reaches a local consensus determined solely by the initial values of its members. Node d remains at its initial opinion $x_d(0)$ (or settles there effectively). Node c, which has outgoing edges to both b and d, is influenced by both groups and converges to a weighted average of their limits. Thus, nodes c and d exert zero influence on the consensus of the group $\{o,a,b\}$.

The removal of edges destroys the strong connectivity of the graph; the matrix P is now reducible and possesses the eigenvalue $\lambda = 1$ with algebraic multiplicity 2, corresponding to two independent stationary distributions.

Question h

Removing the edges (b,o) and (d,a) from $\mathcal{G}$ disrupts the strong connectivity of the graph. As shown in Figure 7, the nodes $\{b,c,d\}$ form a sink component.

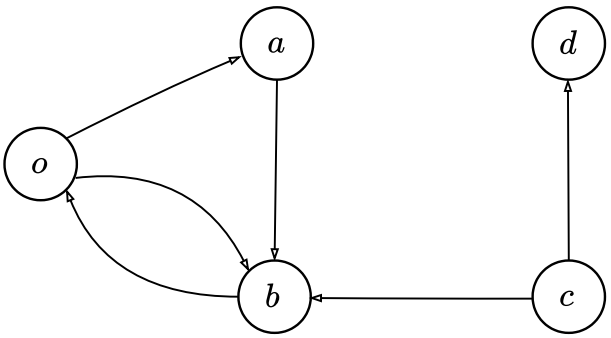

Figure 5: Network graph with the edges $(d, a), (d, c), (a, c), (b, c)$ removed

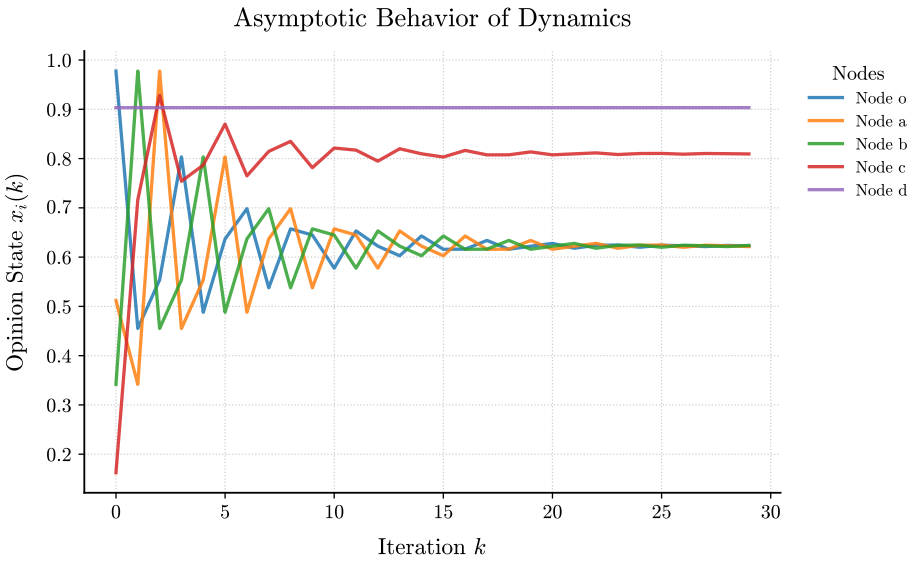

Figure 6: Asymptotic behavior

This component dictates the system's long-term behavior and is governed by the following transition matrix (derived from Λ):

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

Although the graph contains a unique closed component reachable from all nodes (implying a unique invariant distribution), this sink component $\{b, c, d\}$ is periodic with period 2. Specifically, nodes $\{b, d\}$ exchange mass with node c at every step. Consequently, the condition $\lim_{t\to\infty} P^t = \mathbf{1}\pi^\top$ does not hold, and the system exhibits asymptotic oscillations rather than convergence to a static state.

Despite this lack of convergence, a unique invariant distribution $\boldsymbol{\pi}$ exists, supported strictly on the sink component. Our numerical analysis yields:

$$\boldsymbol{\pi} \approx \begin{bmatrix} 0 & 0 & 0.167 & 0.5 & 0.333 \end{bmatrix}^\top$$

$$\text{Consensus Value (Mean)} = \boldsymbol{\pi}^\top x(0) \approx 0.5542$$

Figure 8 illustrates the asymptotic behavior. A snapshot of the system state at the final simulation step shows the distinct values for the oscillating nodes:

$$x(t_{\text{final}}) \approx \begin{bmatrix} 0.554 & 0.598 & 0.466 & 0.642 & 0.466 \end{bmatrix}^\top$$

It is important to note the behavior of the transient nodes $\{o, a\}$. While both technically oscillate as they track the sink component, Figure 8 shows that node a exhibits visible oscillations, whereas node o appears smooth. This is due to a topological filtering effect: node a is directly connected to the oscillating source (b and c) and immediately reflects their switching. In contrast, node o is "upstream" from a; it receives a signal that has already been averaged by a, effectively acting as a low-pass filter that dampens the oscillation amplitude to a negligible level.

In conclusion, the asymptotic dynamics are determined exclusively by the initial conditions of the leader nodes $x_b(0), x_c(0), x_d(0)$.

Exercise 2

We extend the analysis of the network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \Lambda)$ to a system of $N = 100$ particles moving simultaneously in continuous time. Each particle follows

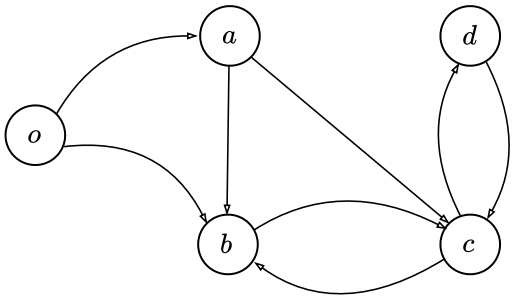

Figure 7: Network graph with the edges $(d, a), (b, o)$ removed

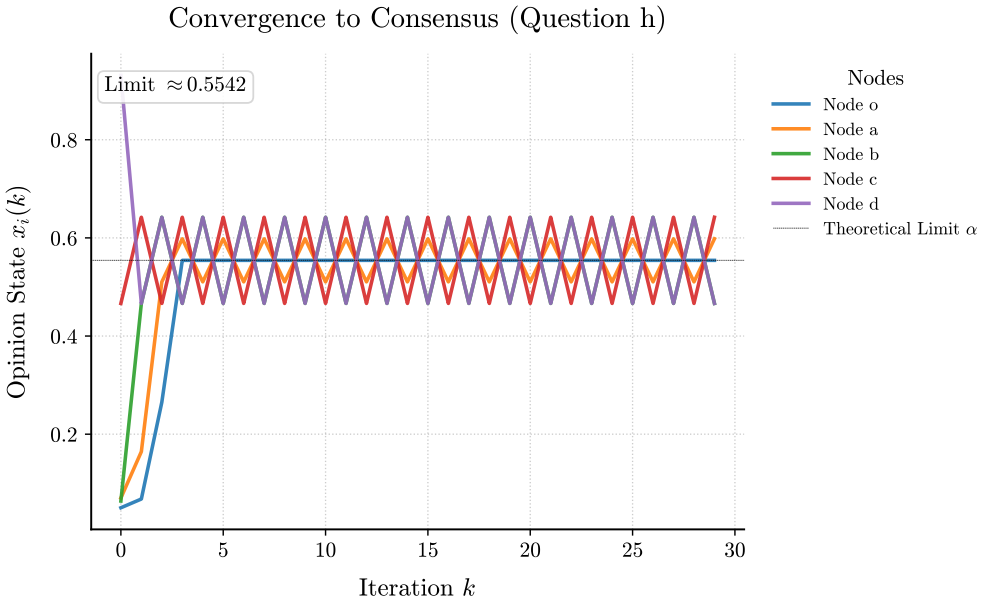

Figure 8: Asymptotic behavior showing the periodic oscillations of the sink component and the filtered response of the transient nodes.

the CTRW dynamics described in Exercise 1. All particles are initialized at node a.

Perform the simulations using the two perspectives described below and address the corresponding questions:

Question a) **Particle Perspective:** We track the N particles as independent random walkers.

- Compute the simulated average time for a particle to leave and return to node a.
- Compare this result to the single-particle return time found in Exercise 1. Explain the relationship between the two values.

Question b) **Node Perspective:** We observe the aggregate state $n_i(t)$, representing the number of particles in node i at time t. The node passes particles to neighbors at a rate proportional to its current occupancy, modeled as a Poisson process with intensity $\lambda_i(t) = n_i(t)\omega_i$. Simulate the system for $T = 60$ time units.

- Determine the average number of particles residing in each node at the end of the simulation.
- Generate a plot illustrating the time evolution of the number of particles $n_i(t)$ in each node over the interval $[0, T]$.
- Compare the particle distribution at $T = 60$ with the theoretical stationary distribution π derived in Exercise 1.

Question a: Particle Perspective

In the first part of the simulation, we adopted the *Lagrangian* perspective. To ensure robust statistical convergence, we did not merely simulate a single batch of particles; instead, we performed N_{sim} independent iterations, tracking $N = 100$ particles in each run.

The final estimate for the mean return time to node a was computed by averaging over the total pool of samples $M = N \times N_{\text{sim}}$:

$$\widehat{E}_a[T_a^+] = \frac{1}{M} \sum_{k=1}^{M} T_{a,k}^+ \approx 6.701 \,\text{t.u.}$$

Comparison with Single-Particle Dynamics: This value virtually coincides with the theoretical expectation for a single random walker derived in Exercise 1 ($E_a[T_a^+] = 6.701\,\text{t.u.}$). The high precision of this result stems

from the independence of the experiments. Since the particles do not interact, running N particles simultaneously is statistically equivalent to running N independent single-particle simulations. By repeating this process N_{sim} times, we generated a total of $M = N \times N_{\text{sim}}$ independent and identically distributed (i.i.d.) realizations of the return time. According to the *Law of Large Numbers*, the variance of our estimator decreases as $\frac{1}{M}$. Thus, this method leverages massive parallelization to suppress stochastic noise, providing a far more accurate approximation of the theoretical mean than a single-particle simulation could achieve in the same timeframe.

The error for the simulation with M particles is significantly lower than the error associated with a single particle (or a smaller batch of N_{sim} simulations). This reduction is a direct consequence of the *Law of Large Numbers*: since the variance of the estimator scales as $1/M$, increasing the total number of independent samples from N_{sim} (or 1) to $M = N \times N_{\text{sim}}$ drastically suppresses the stochastic noise.

$$\text{Error}_{1 \text{ Particle}} \approx 0.1323 \quad \Longrightarrow \quad \text{Error}_{100 \text{ Particles}} \approx 0.0067$$

Question b: Node Perspective

To analyze the system from the node perspective, we simulated the aggregate evolution of particle counts $\mathbf{n}(t)$. Unlike the particle perspective, which tracks individual trajectories, this approach treats the system as a continuous-time stochastic process driven by the transition rates of the nodes.

Methodology: Local Clocks and Event Selection

The state of the system is defined by the vector of particle counts $\mathbf{n}(t) = [n_o, n_a, n_b, n_c, n_d]$. The evolution is governed by the following principles:

1. **Local Clocks Technique:** Conceptually, we assign a local Poisson clock to each node i with a rate $\lambda_i(t) = \omega_i n_i(t)$. This rate represents the frequency at which node i emits a particle. The time until the *next event anywhere in the network* is determined by the minimum of these local clock times. According to the theory of competing exponential processes, the minimum of independent exponential variables is itself exponentially distributed with a rate equal to the sum of the individual rates:

$$R_{\text{total}}(t) = \sum_{j \in \mathcal{V}} \lambda_j(t) \quad \Longrightarrow \quad t_{\text{next}} \sim \text{Exp}(R_{\text{total}}(t)) \tag{2}$$

2. **Event Selection:** When the global clock ticks (at $t + t_{\text{next}}$), one node must fire. The probability that the event corresponds to a departure from node i is proportional to its rate:

$$P(\text{Node } i \text{ departs}) = \frac{\lambda_i(t)}{R_{\text{total}}(t)} \tag{3}$$

It is important to note that this vector always sums to 1 ($\sum_i P_i = 1$). For example, in an initial configuration where all particles are located in node o (i.e., $\mathbf{n} = [N, 0, 0, 0, 0]$), the rate vector is non-zero only for the first element. Consequently, the probability vector becomes $P_{\text{dep}} = [1, 0, 0, 0, 0]$, meaning the next event is deterministically a departure from node o with probability 1.

3. **State Update:** Once a departure node i is selected, the destination node j is chosen according to the random walk transition probabilities P_{ij}. The state is then updated ($n_i \to n_i - 1$, $n_j \to n_j + 1$), and the rates are recomputed for the next step.

Simulation Results

We kept $N = 100$, $T = 60$ time units and $N_{\text{sim}} = 1000$.

Figure 9 displays the particle evolution for a single representative run to illustrate the stochastic stepping inherent to the Gillespie algorithm. While the plot visualizes one trajectory to avoid interpolation artifacts, the numerical results (Table 1) are averaged over 1000 independent simulations to ensure statistical accuracy. The average number of particles in each node at $T = 60$ was found to be:

$$\mathbf{n}(60)_{\text{sim}} \approx [21.77, 15.08, 26.08, 18.74, 18.33]$$

The simulated results align perfectly with the theoretical predictions. This convergence is guaranteed by the **Ergodic Theorem** for continuous-time Markov chains. Since the network is strongly connected (irreducible), the system is ergodic, and the long-term time average of the particle distribution converges to the stationary distribution $\bar{\pi}$.

Exercise 3

In this final section, we analyze the network as an **Open System**. Unlike the previous exercises where the number of particles was conserved, we now introduce an external inflow. Particles arrive at node o according to a Poisson

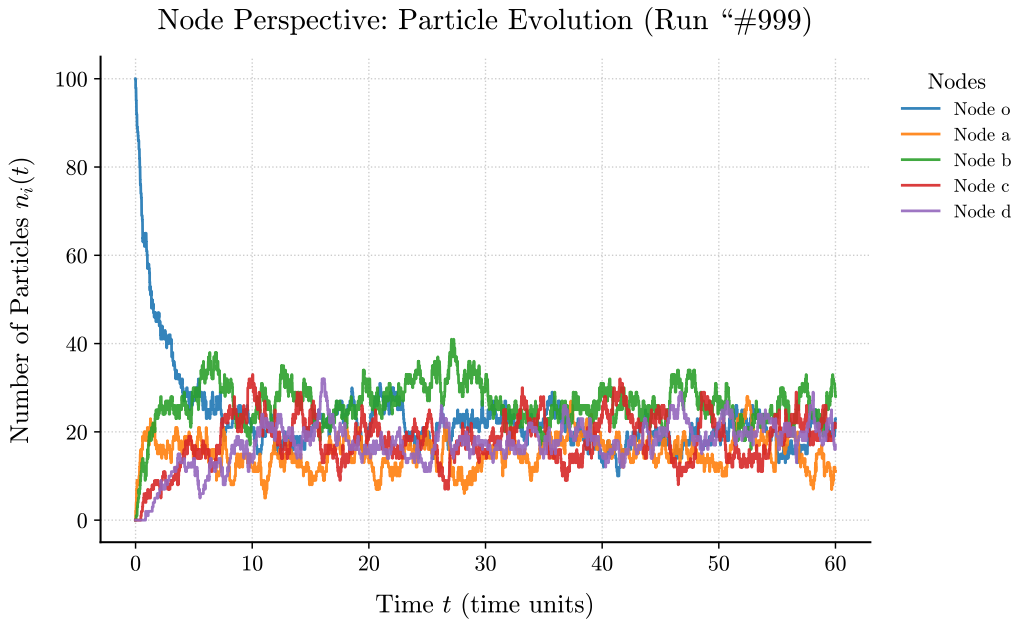

Figure 9: **Node Perspective:** Time evolution of the number of particles $n_i(t)$ in each node ($N = 100$). The system starts with all particles in node a and relaxes toward the stationary distribution $\bar{\pi}$.

Node	Stationary Dist. ($\bar{\pi}_i$)	Theoretical Mean ($N\bar{\pi}_i$)	Simulated Mean ($\bar{n}_i$)
o	0.217	21.70	21.77
a	0.149	14.90	15.08
b	0.261	26.10	26.08
c	0.186	18.60	18.74
d	0.186	18.60	18.33
Total	**1.000**	**100**	**100**

Table 1: Comparison of theoretical vs. simulated particle distribution at $T = 60$.

process with rate λ_{in}. Once inside the network, particles move according to the transition probabilities defined by the internal rate matrix Λ:

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

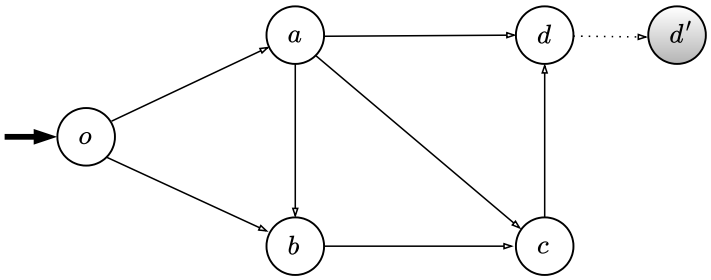

Figure 10: Open network topology with external inflow λ_{in} entering node o.

Perform the simulations and answer the following questions:

Question a) **Proportional rate:** In this model , the service rate of node i scales linearly with the number of particles present: $\mu_i(t) = n_i(t)\omega_i$.

- 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?

Question b) **Fixed rate:** In this model (Single Server), the nodes have limited capacity. The service rate is constant regardless of the queue size: $\mu_i(t) = \omega_i$ (provided $n_i(t) > 0$).

- Simulate the system for 6000 time units and plot the evolution of number of particles in each node over time with input rate $\lambda = 2$.
- What is the largest input rate that the system can handle without blowing up? Motivate your answer.

Question a: Proportional Rate (Infinite Server)

In this scenario, we modeled the network nodes as Infinite Servers. This implies that the processing capacity of a node scales linearly with the number of particles present. Specifically, if node i contains $n_i(t)$ particles, its total service rate is $\mu_i(t) = \omega_i n_i(t)$.

Since the transition rates depend on the state of the system $n_i(t)$, the inter-arrival times are not stationary. To simulate this accurately, we employed the Gillespie First Reaction Method (FRM).

The algorithm proceeds as follows:

1. **Rate Definition:** We define a rate vector $\mathbf{R}$ containing the propensities of all possible events. The first element corresponds to the external arrival, followed by the departure rates for each node:

   $$\mathbf{R} = [\lambda_{\text{in}}, R_o, R_a, R_b, R_c, R_d]$$

   where $R_i = \omega_i n_i(t)$ is the rate at which node i processes particles. *Note:* For this open network configuration, the service rate of the sink node d was set to $\omega_d = 7/4$ to facilitate system outflow, as per the assignment specifications.

2. **Tentative Time Generation:** For each event k, we generate a tentative time to occurrence T_k assuming it is the next event. These are drawn from exponential distributions with rates R_k:

   $$T_k = \frac{-\ln(u_k)}{R_k}, \quad u_k \sim \mathcal{U}(0, 1).$$

3. **Event Selection:** The actual time to the next event in the system is determined by the minimum of these tentative times:

   $$\Delta t = \min_k(T_k).$$

   The event index $k_{\min} = \arg\min_k(T_k)$ determines which reaction executes.

4. **State Update:**

   - If $k_{\min} = 0$ (Arrival): Increment n_o.
   - If $k_{\min} > 0$ (Departure): Decrement the corresponding node count. If the departure is from node d, the particle leaves the system; otherwise, it is routed to a neighbor j based on the probability matrix P.

Figure 11 shows the system evolution with a high input rate $\lambda_{\text{in}} = 100$. Starting from an empty state, the number of particles in each node grows rapidly but eventually stabilizes around an equilibrium level. The system handles the high load effectively, with node occupancies fluctuating around their respective means.

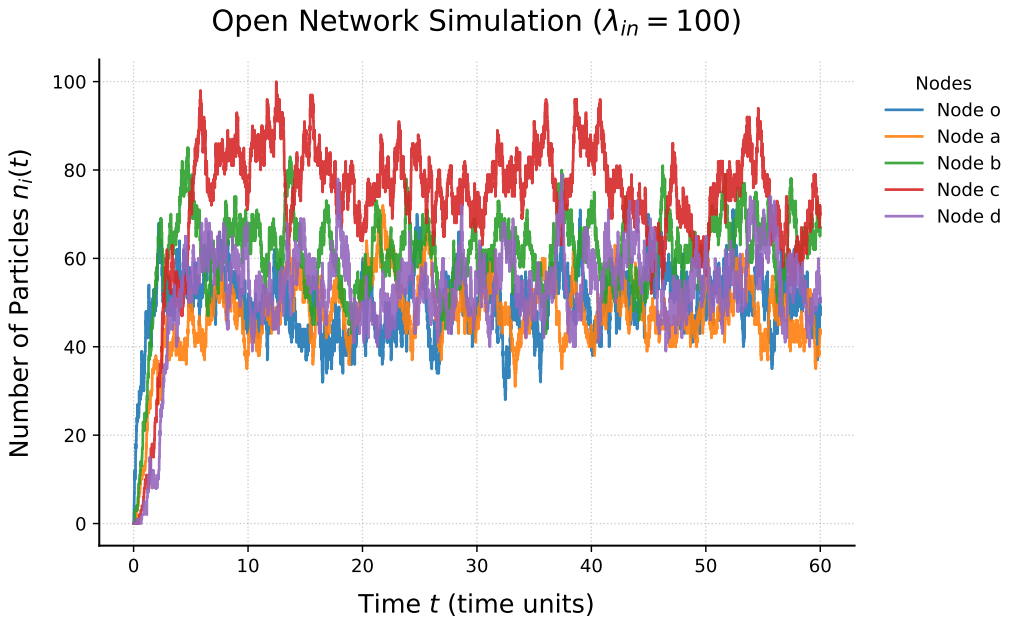

Figure 11: Evolution of particle counts in the open network with proportional rates ($\lambda_{\text{in}} = 100$). The system reaches a dynamic equilibrium without divergence.

What is the largest input rate λ_{max} that the system can handle without blowing up?

We investigated whether a "breaking point" exists for this system, i.e., a maximum input rate $\lambda_{\max}$ above which the queue lengths diverge to infinity.

For a proportional rate (Infinite Server) network, no finite breaking point exists. The system is stable for any finite arrival rate λ_{in}.

Consider the source node o. Particles arrive at a constant rate λ_{in}. The service rate at node o, however, is state-dependent: $R_{\text{out}}(t) = \omega_o n_o(t)$. Analyzing the asymptotic behavior:

$$\text{If } n_o(t) \to \infty \implies R_{\text{out}}(t) = \omega_o n_o(t) \to \infty.$$

Since the service rate grows linearly with the queue size while the arrival rate λ_{in} remains constant, there will always exist a threshold N^* such that for all $n_o(t) > N^*$, the service rate exceeds the arrival rate ($n_o(t)\omega_o > \lambda_{\text{in}}$). This negative feedback loop prevents the queue from growing indefinitely.

Since the source node o is stable, and it feeds downstream nodes which also operate as infinite servers, the stability propagates through the entire network. Thus, increasing λ_{in} only increases the equilibrium number of particles, but the system never "blows up."

Question b: Fixed Rate (Single Server)

In this second scenario, we modeled the nodes as Single Servers with fixed capacity. Unlike the previous case, the processing speed does not increase with the queue length. A node processes particles at a constant rate ω_i, but only if it is non-empty. This corresponds to an $M/M/1$ queueing discipline.

Methodology: Modified Gillespie Algorithm

The simulation strategy is similar to the First Reaction Method used in part (a), with a crucial modification to the rate definition.

1. **Rate Definition:** The rate vector $\mathbf{R}$ is now state-dependent in a binary fashion. The departure rate for node i is:

$$R_i(t) = \begin{cases} \omega_i & \text{if } n_i(t) > 0 \\ 0 & \text{if } n_i(t) = 0 \end{cases}$$

   This reflects the physical reality that an empty server cannot process jobs. The input rate remains constant: $R_0 = \lambda_{\text{in}}$.

2. **Event Selection:** We compute tentative times $T_k \sim \text{Exp}(R_k)$. If a node is empty ($R_k = 0$), the corresponding time to event is effectively infinite ($T_k = \infty$), ensuring that no departure event is scheduled for that node. The next event is selected via $t_{\text{next}} = \min(T_k)$.

Figure 12 displays the system evolution for $T = 6000$ time units with an input rate $\lambda_{\text{in}} = 2$.

Contrary to the Proportional Rate case, the Fixed Rate system is **unstable** at this input level. The simulation reveals a clear trend of unbounded growth, particularly for nodes b and c. The queues for these nodes accumulate particles indefinitely, indicating that the inflow of particles exceeds the constant service capacity of the servers.

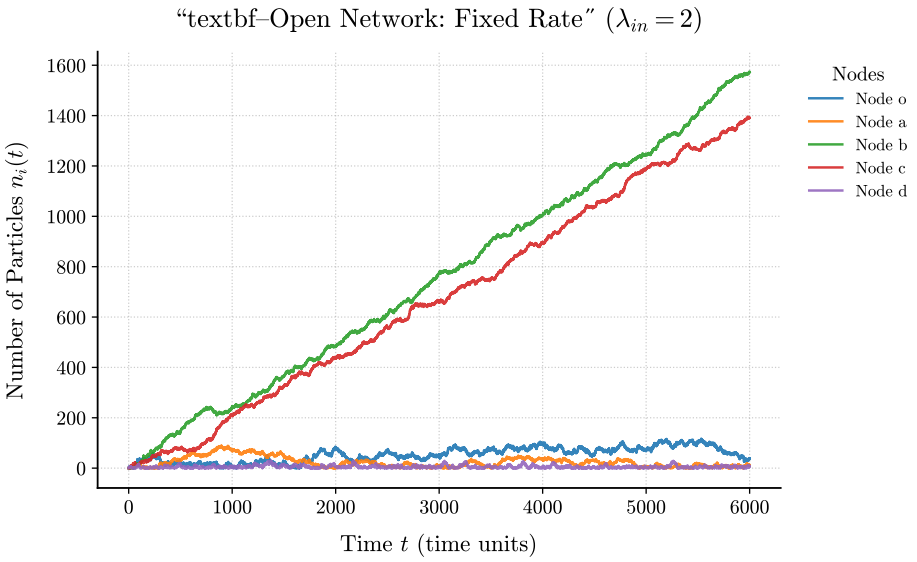

Figure 12: Evolution of particle counts in the open network with fixed rates ($\lambda_{\text{in}} = 2$). The system is **unstable**, as evidenced by the linear growth of particles in nodes o, a, and c.

What is the largest input rate $\lambda_{\max}$ that the system can handle without blowing up?

Unlike the proportional rate case, the fixed rate system possesses a distinct breaking point. The system becomes unstable if the utilization factor ρ_i of any single node exceeds 1. That is, if the mean arrival rate exceeds the fixed service rate ($\lambda_i > \omega_i$), the queue at that node will grow indefinitely.

To find the critical threshold $\lambda_{\max}$, we utilize the linearity of the Traffic Equations.

Solving the linear system $\boldsymbol{\lambda} = \mathbf{r}(I - P)^{-1}$ yields the steady-state arrival rates. We then calculate the utilization $\rho_i = \lambda_i/\omega_i$ for each node to assess the load relative to capacity:

$$\boldsymbol{\lambda} \approx [2.00, 1.00, 1.25, 1.50, 2.00]$$

$$\boldsymbol{\omega} = [2.0, 1.0, 1.0, 1.0, 1.75]$$

Computing the element-wise utilization ratio $\boldsymbol{\rho} = \boldsymbol{\lambda}/\boldsymbol{\omega}$:

$$\boldsymbol{\rho} \approx [1.00, 1.00, 1.25, \mathbf{1.50}, 1.14]$$

The stability of the entire system is dictated by the bottleneck node—the node with the highest utilization. From the vector $\boldsymbol{\rho}$, we identify node c (index 3) as the bottleneck:

$$\rho_{\max} = \rho_c = 1.50$$

Since $\rho_c > 1$, the system is currently overloaded. Specifically, Node c is receiving 150% of the traffic it can handle when the system input is $\lambda_{\text{in}} = 2$.

Because the Traffic Equations are linear, the internal flows scale proportionally with the external input. To find the maximum stable input $\lambda_{\max}$, we must scale our test $\lambda_{\text{test}} = \lambda_{\text{in}}$ input such that the bottleneck utilization reduces to exactly 1 (100% capacity).

The scaling relationship is defined as:

$$\lambda_{\max} = \frac{\lambda_{\text{test}}}{\rho_{\max}}$$

Substituting our values:

$$\lambda_{\max} = \frac{2.0}{1.50} \approx 1.333$$

The system's topology imposes a hard limit based on the capacity of Node c. Any input rate $\lambda_{\text{in}} > 1.333$ will cause the bottleneck to saturate ($\rho_c > 1$) and the system to blow up.