

# HW2 - Network Dynamics

Luciano Scarpino - S346205

November 2025

## Collaborators (on all tasks):

S346378 Salvatore Nocita  
S329057 Shadi Mahboubpardahi  
S334015 Andrea Vasco Grieco

*We discussed the project and worked jointly on each exercise.*

## Exercise 1

### Graph and CTMC Structure

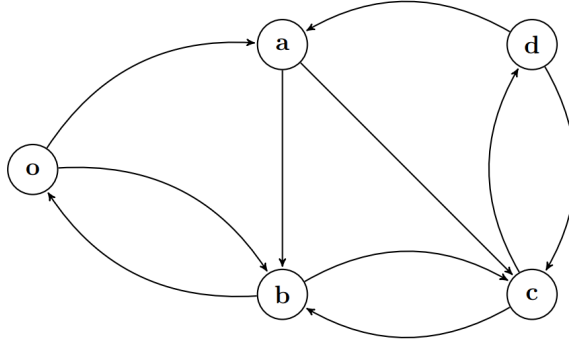


Figure 1: Closed network in which particles move according to the transition-rate matrix.

Figure 1 shows a directed network modeling a continuous time Markov chain (CTMC) defined on the graph  $G = (\mathcal{V}, \mathcal{E}, \Lambda)$ , with node set

$$V = \{o, a, b, c, d\}.$$

The dynamics are fully specified by the transition rate matrix  $\Lambda = (\lambda_{ij})_{i,j \in V}$ , whose nonzero entries identify the directed edges: a transition  $i \rightarrow j$  is allowed if and only if  $\lambda_{ij} > 0$ .

For each state  $i$ , the total outgoing rate is

$$\omega_i = \sum_{j \in V} \lambda_{ij}.$$

The process remains in state  $i$  for an exponentially distributed time with parameter  $q_i$ , and then jumps to a successor state  $j$  with probability

$$P_{ij} = \frac{\lambda_{ij}}{\omega_i}.$$

Since every node is reachable from any other through a directed path with positive rates, the chain is irreducible and admits a unique stationary distribution  $\pi$ , characterized by

$$\pi^\top \Lambda = 0, \quad \sum_{i \in V} \pi_i = 1.$$

In stationarity, the expected return time to a state  $i$  satisfies

$$\mathbb{E}_\pi[T_i^+] = \frac{1}{\pi_i}.$$

When the process is in state  $i$ , the next holding time is generated as

$$t_{\text{next}} = -\frac{\ln U}{\omega_i}, \quad U \sim \text{Uniform}(0, 1),$$

after which the next state is drawn according to  $P_{ij}$ . All simulations start from the initial node  $o$ . For implementation purposes, nodes are internally encoded as integers and mapped back to their symbolic labels for analysis.

**Estimation of the Invariant Distribution.** To approximate the invariant distribution of the CTMC, we rely on empirical sojourn times. Let

$$0 = T_0 < T_1 < T_2 < \dots < T_n$$

denote the cumulative transition times of a simulated trajectory, and

$$X_0, X_1, \dots, X_n$$

its corresponding sequence of visited states. The empirical stationary weight of node  $i$  is obtained by normalizing the total time spent in that state:

$$\hat{\pi}_i = \frac{1}{T_n} \sum_{k: X_k=i} (T_{k+1} - T_k).$$

Because the final visited state has no predefined departure time, one additional exponentially distributed holding time must be appended to ensure that its last sojourn is included. This correction is necessary for consistency of the estimator, since the invariant distribution of a CTMC is defined in terms of long-run proportions of *time*, not of jumps.

The resulting vector  $\hat{\pi}$  provides a Monte Carlo estimate of the true stationary distribution  $\pi$ , and converges to it as the simulation horizon increases.

**Reproducibility and Random Seed.** All simulations were performed using a fixed random seed to ensure reproducibility. Randomness affects both the CTMC evolution, through state transitions governed by  $P$  and exponential holding times with rates  $\omega_i$ , and the French-DeGroot dynamics, through the initialization of opinions. Without a fixed seed, repeated runs would generate different trajectories and numerical estimates. All reported results are therefore conditioned on the chosen seed, while alternative seeds would typically lead to small variations in the simulated values.

## Problem Specification

In this exercise we simulate the movement of a particle evolving on the directed network defined in Figure 1. The particle behaves as a CTMC: upon entering a state, it remains there for an exponentially distributed holding time whose rate depends on the total outgoing transition rate of that state. When the local Poisson clock ticks, the particle jumps to a new state according to the transition probabilities

of the embedded discrete chain. This mechanism generates a stochastic trajectory in which both the jump times and the visited states depend on the underlying rate matrix.

To visualize the early behaviour of the process, we record the sequence of visited states

$$X_0, X_1, \dots, X_{30},$$

together with the cumulative jump times

$$0 = T_0 < T_1 < \dots < T_{30},$$

where  $T_k$  denotes the time at which the  $k$ -th transition occurs. Plotting the pairs  $(T_k, X_k)$  provides an intuitive representation of how the particle moves across the network during the initial phase of the simulation.

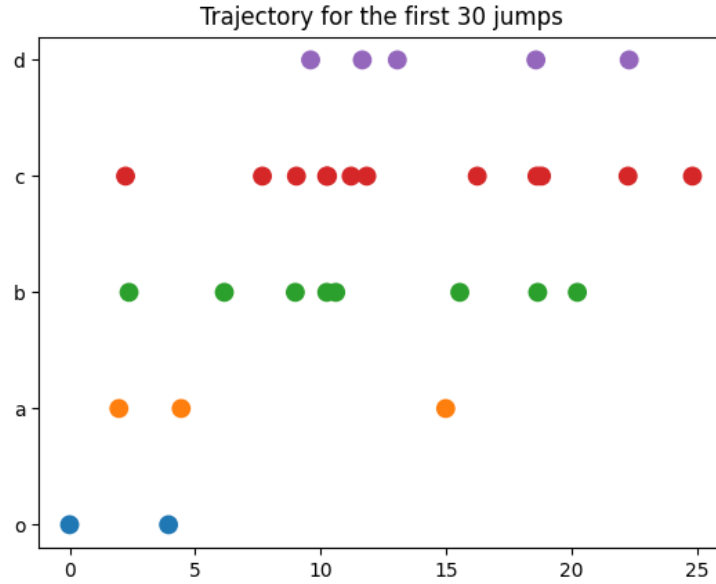


Figure 2: Trajectory of the first 30 jumps of the CTMC starting from node  $o$ . Each point corresponds to a transition event, located at the cumulative transition time on the horizontal axis and at the visited state on the vertical axis.

The trajectory in Figure 2 shows that the particle initially performs several fast returns to state  $o$  before progressively exploring the rest of the network. The irregular spacing of points along the time axis reflects the variability of the exponential holding times: states with larger outgoing rates produce shorter times, whereas states with smaller rates lead to longer intervals between jumps. Throughout the first 30 transitions, the particle visits all states of the graph, demonstrating the irreducible structure of the underlying CTMC.

### a) Simulated Return Time to Node $a$

To estimate the average return time to node  $a$ , we analyze a long simulated trajectory of the CTMC and record all arrival times to state  $a$ . Let  $\{T_k\}_{k \geq 0}$  denote the cumulative jump times and  $\{X_k\}_{k \geq 0}$  the associated sequence of visited states. Every index  $k$  such that  $X_k = a$  corresponds to a visit of the particle to node  $a$ . If these visits occur at times

$$T_{k_1}, T_{k_2}, \dots, T_{k_m},$$

then each difference  $T_{k_{i+1}} - T_{k_i}$  represents one realization of the return time to  $a$ , i.e. the time elapsed between leaving node  $a$  and reaching it again.

Since the last visit to  $a$  in the simulated path may not be followed by a subsequent jump back to  $a$ , we append an additional exponential holding time (with parameter  $\omega_a$ ) whenever the final state of the trajectory is  $a$ . This ensures that the final sojourn contributes a complete observation and prevents underestimation of the return-time statistics.

The empirical expected return time is therefore computed as

$$\widehat{\mathbb{E}}[T_a^+] = \frac{1}{m} \sum_{i=1}^m (T_{k_{i+1}} - T_{k_i}),$$

Using a simulation of 1000 jumps, we obtain the estimate

$$\widehat{\mathbb{E}}[T_a^+] \approx 7.22,$$

which represents the average time required for a particle starting in node  $a$  to leave the node and return to it according to the simulated CTMC dynamics.

## b) Comparison with the Theoretical Return Time

The theoretical return time to node  $a$  can be computed using the classical hitting-time formulation for discrete-time Markov chains derived from the embedded transition matrix  $P$ . Let  $S = \{a\}$  and let  $R = V \setminus S$  denote the set of states different from  $a$ . For every state  $i \in R$ , the expected time to hit  $a$  satisfies the linear system

$$x_i = \frac{1}{\omega_i} + \sum_{j \in R} P_{ij} x_j,$$

which compactly writes as

$$(I - P_{RR})x = \beta, \quad \beta_i = \frac{1}{\omega_i}.$$

Here  $P_{RR}$  is the submatrix of the embedded transition matrix restricted to the states in  $R$ , and  $x$  collects the expected hitting times from all states in  $R$  to  $a$ .

Solving this system yields the vector of expected hitting times  $x$ , from which the theoretical return time to  $a$  follows from the standard identity

$$\mathbb{E}_a[T_a^+] = \frac{1}{\omega_a} + \sum_{j \in R} P_{aj} x_j,$$

where the first term accounts for the expected holding time in  $a$ , and the second term corresponds to the expected time needed to return to  $a$  after jumping to a different state.

Applying this procedure to the chain under study, we obtain the theoretical return time

$$\mathbb{E}_a[T_a^+] \approx 6.71.$$

Comparing this value with the simulated estimate from point (a), we observe a discrepancy of approximately 0.50. This difference can be attributed to Monte Carlo variability: the simulation uses a trajectory of length 1000, and the occurrence of returns to  $a$  is relatively sparse. Increasing the length of the simulated path would reduce the variance of the estimator and make the empirical estimate converge toward the theoretical value.

### c) Simulated Hitting Time from $o$ to $d$

We now estimate the expected hitting time from node  $o$  to node  $d$ . In continuous time, the hitting time is defined as

$$T_d = \inf\{t \geq 0 : X_t = d \mid X_0 = o\},$$

where  $\{X_t\}_{t \geq 0}$  denotes the CTMC.

To extract empirical realizations of the hitting time from  $o$  to  $d$ , we process the path as follows. We scan the pairs  $(T_k, X_k)$  in chronological order and maintain an “open interval” whenever the chain is in state  $o$ :

- When the chain visits state  $o$  and no interval is currently open, we set

$$\text{start\_time} = T_k.$$

This marks the beginning of a potential hitting-time experiment starting from  $o$ .

- As soon as the chain subsequently visits state  $d$  while an interval is open, we close the interval and record

$$H_i = T_k - \text{start\_time},$$

which represents one realization of the time needed to reach  $d$  after the last entrance into  $o$ . The interval is then reset, waiting for the next visit to  $o$ .

If the simulation ends while an interval is still open (i.e., the last recorded state is  $o$  and  $d$  has not yet been reached), this partial observation is discarded.

Let  $H_1, \dots, H_m$  denote all collected intervals. The empirical estimate of the expected hitting time is then given by

$$\hat{\mathbb{E}}_o[T_d] = \frac{1}{m} \sum_{i=1}^m H_i.$$

Using a trajectory of 1000 jumps, this procedure yields

$$\hat{\mathbb{E}}_o[T_d] \approx 9.976,$$

which represents the average time, according to the simulations, for a particle starting in node  $o$  to reach node  $d$  for the first time.

### d) Comparison with the Theoretical Hitting Time from $o$ to $d$

The theoretical hitting time from node  $o$  to node  $d$  is computed using the same linear-system formulation adopted in point (b), applied now to the target set  $S = \{d\}$ . As before, we restrict the embedded transition matrix  $P$  to the complement  $R = V \setminus S$  and solve

$$(I - P_{RR}) h = \beta, \quad \beta_i = \frac{1}{\omega_i},$$

which yields the expected times to reach  $d$  starting from all states in  $R$ . The desired quantity is simply the component corresponding to state  $o$ :

$$\mathbb{E}_o[T_d] = h_o.$$

Applying this method to the given network provides the theoretical estimate

$$\mathbb{E}_o[T_d] \approx 10.77.$$

Comparing this with the simulated value from point (c), we observe a gap of approximately 0.79. This discrepancy is expected given the relatively small number of complete  $o \rightarrow d$  excursions available in a trajectory of length 1000. In contrast with the return-time case, where many visits to  $a$  were recorded, the hitting time to  $d$  is sampled far less frequently, which increases Monte Carlo variance. Increasing the simulation horizon would reduce this gap and yield a closer match to the theoretical value.

### e) French–DeGroot Dynamics on the Network

We investigate the evolution of opinions under the French–DeGroot (FDG) dynamics on the directed network defined by the stochastic matrix  $P$ . Given an arbitrary initial opinion vector  $x(0) \in \mathbb{R}^5$ , the FDG update is linear and deterministic:

$$x(t+1) = P^\top x(t).$$

Since the graph associated with  $P$  is strongly connected and aperiodic, the Perron–Frobenius theorem guarantees convergence toward a consensus value. In particular, there exists a unique invariant distribution  $\pi^\top$  of  $P$  such that

$$\lim_{t \rightarrow \infty} x(t) = \alpha \mathbf{1}, \quad \alpha = \sum_{i=1}^5 \pi_i x_i(0).$$

Thus, the limiting consensus depends on the initial condition only through the projection of  $x(0)$  onto the invariant distribution, meaning that nodes with higher invariant centrality exert stronger long-term influence on the group opinion.

To illustrate this behavior, we simulate 100 FDG iterations starting from a random initialization. As shown in Figure 3, the mean opinion  $\frac{1}{5} \sum_i x_i(t)$  rapidly stabilizes, while the opinion variance decreases monotonically, reflecting the contraction induced by repeated left-multiplication by a stochastic matrix. Numerically, convergence occurs within 28 iterations, fully consistent with the theoretical expectation of exponential decay toward consensus.

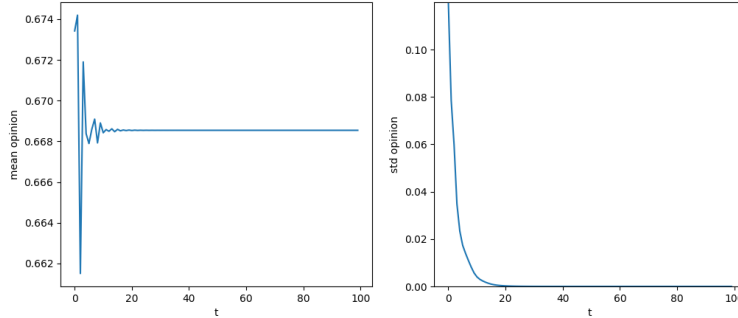


Figure 3: Evolution of the average opinion and opinion variance over the first 100 FDG iterations.

**Estimation Accuracy of the Consensus Value.** To quantify the robustness of the empirical estimation of the consensus  $\alpha$ , we repeat the simulation over  $M = 1000$  independent random initializations. For each run, we compute both the true value  $\alpha = \pi^\top x(0)$  and its empirical estimate given by the mean of  $x(t)$  at the final iteration. The average absolute estimation error over the  $M$  trials is found to be extremely small, confirming that the FDG dynamics reliably concentrates trajectories around the consensus value and that the sample-based estimate becomes highly accurate even with a modest number of iterations. This behavior follows from the linearity of the model and from the fact that the long-term dynamics is dominated by the single eigenvector associated with the Perron eigenvalue.

### f) Variance of the Consensus Value

In this part we assume that each node starts from an independent random initial opinion

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

with heterogeneous variances

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

Since the French-DeGroot dynamics converges to the consensus value, the random variable  $\alpha$  is a linear combination of independent zero-mean Gaussian variables. Therefore, the variance of the consensus value can be computed analytically as

$$\text{Var}(\alpha) = \sum_{i \in \{o, a, b, c, d\}} \pi_i^2 \sigma_i^2 = 2(\pi_a^2 + \pi_b^2 + \pi_c^2) + (\pi_o^2 + \pi_d^2),$$

which is strictly smaller than the variances assigned to the individual nodes. This reflects the smoothing effect of the weighted averaging performed by the invariant distribution.

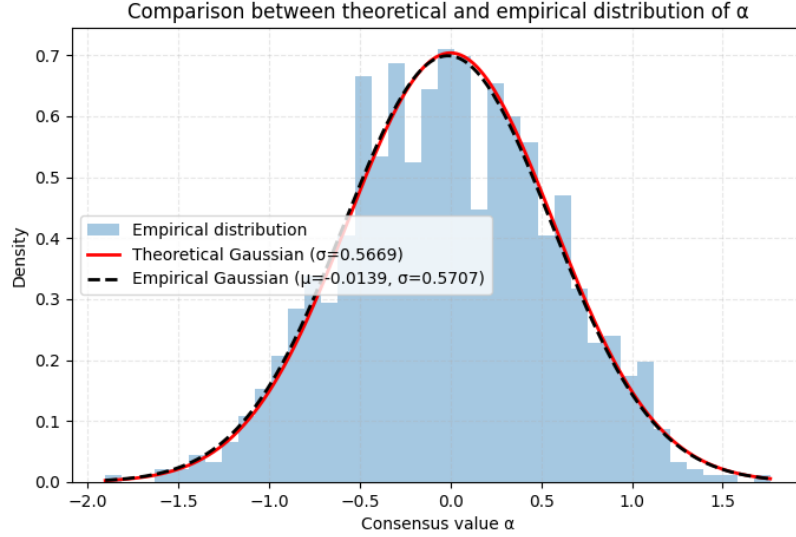


Figure 4: Comparison between the theoretical Gaussian distribution of  $\alpha$  and the empirical distribution obtained through Monte Carlo simulations.

**Comparison Between Theoretical and Empirical Distributions.** To validate the theoretical expression for  $\text{Var}(\alpha)$ , we generate a large number of independent realizations of the initial opinion vector  $x(0)$  and compute the corresponding consensus value  $\alpha = \pi^\top x(0)$  for each trial. The resulting empirical distribution is well approximated by a Gaussian density whose mean and variance are estimated directly from the Monte Carlo samples.

Figure 4 displays the histogram of the simulated  $\alpha$  values, together with the theoretical Gaussian curve and the empirical Gaussian fit. The two densities are nearly indistinguishable: the theoretical standard deviation matches the empirical standard deviation estimated from the samples. This confirms both the correctness of the analytical computation and the fact that the consensus value inherits a significantly reduced variability compared to the individual nodes. The close agreement between the curves demonstrates that the asymptotic averaging dynamics amplifies the influence of the invariant distribution while attenuating the heterogeneity of the initial opinions.

## g) Asymptotic behaviour after edge removal

**Graph modification.** We now consider the same set of nodes  $V = \{o, a, b, c, d\}$  and modify the topology of the network by removing the edges  $(d, a)$ ,  $(d, c)$ ,  $(a, c)$  and  $(b, c)$  from the original graph

$G$ , obtaining a new graph  $G_g = (V, \mathcal{E}_g, \Lambda_g)$ . The modified structure is shown in Fig. 5. The remaining non-zero entries of the weight matrix  $\Lambda_g$  are

$$\Lambda_g = \begin{bmatrix} 0 & 2/5 & 1/5 & 0 & 0 \\ 0 & 0 & 3/4 & 0 & 0 \\ 1/2 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 1/3 & 0 & 2/3 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

where we additionally introduce a self-loop on node  $d$  with weight 1 so that the last row of  $\Lambda_g$  corresponds to a stubborn agent. The associated row-stochastic matrix is  $P_g = D_g^{-1} \Lambda_g$ , where  $D_g = \text{diag}(\omega_g)$  and  $\omega_g$  collects the out-degrees of  $\Lambda_g$ .

Compared to the original network, the key structural change is that: (i) nodes  $o, a, b$  no longer have outgoing edges to  $c$  or  $d$ ; (ii) node  $d$  is stubborn; and (iii) node  $c$  still has outgoing edges (get influenced) to both  $b$  and  $d$ . These changes break the strong connectivity of the graph and create two sink components and one transient node.

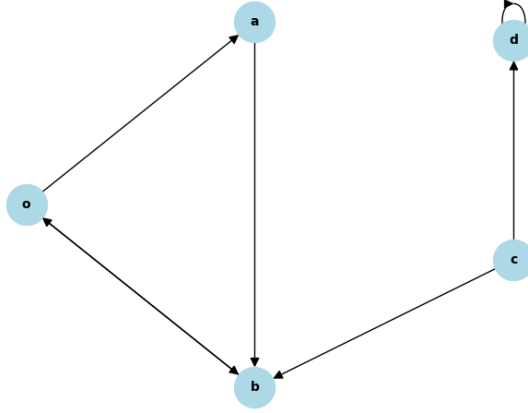


Figure 5: Modified graph  $G_g$  after the edge removal of point (g).

**Sink components and transient node.** In the modified graph  $G_g$  the strongly connected components are:

$$S_1 = \{o, a, b\}, \quad S_2 = \{d\}$$

The set  $S_1$  is a recurrent component: all nodes in  $S_1$  can reach each other and no edge leaves  $S_1$ . The set  $S_2$  is a singleton stubborn component, since node  $d$  only has a self-loop with probability 1. Hence,  $\{S_1, S_2\}$  are the sink components of  $G_g$ .

Node  $c$  is transient: from  $c$  the process can move to  $b$  or  $d$ , but once it enters either sink component it cannot come back to  $c$ . Both sink components are aperiodic (each contains a self-loop or cycles of coprime length), so the assumptions of the general convergence theorem for linear averaging dynamics are satisfied.

**Application of the convergence theorem.** Consider the French-DeGroot dynamics on the modified graph and let  $S = S_1 \cup S_2$  denote the union of all sink components, hence  $s = 2$  be their number. The theorem states that, if all sink components are aperiodic, then there exists a non-negative matrix  $H \in \mathbb{R}^{n \times s}$  with  $H\mathbf{1} = \mathbf{1}$  such that

$$\lim_{t \rightarrow \infty} x(t) = H \bar{x},$$



where

$$\bar{x}_k = \sum_{i \in S_k} \pi_i^{(k)} x_i(0), \quad k = 1, 2,$$

and  $\pi^{(k)}$  is the unique invariant distribution of the Markov chain restricted to sink component  $S_k$ . The matrix  $H$  is characterized by

$$H_{ik} = \begin{cases} 1, & i \in S_k, \\ 0, & i \in S \setminus S_k, \end{cases} \quad H_{ik} = \sum_{j \in V} P_{ij} H_{jk}, \quad i \notin S.$$

In our case,  $S_1 = \{o, a, b\}$  and  $S_2 = \{d\}$ . The invariant distribution on  $S_1$  is

$$\pi^{(1)} = \left( \frac{3}{8}, \frac{1}{4}, \frac{3}{8} \right)$$

(over the ordered nodes  $(o, a, b)$ ), while on  $S_2$  we simply have  $\pi^{(2)} = 1$ . Thus,

$$\bar{x}_1 = \frac{3}{8}x_o(0) + \frac{1}{4}x_a(0) + \frac{3}{8}x_b(0), \quad \bar{x}_2 = x_d(0).$$

The corresponding matrix  $H$  takes the form

$$H = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ \frac{1}{3} & \frac{2}{3} \\ 0 & 1 \end{bmatrix},$$

where rows are ordered as  $(o, a, b, c, d)$  and columns as  $(S_1, S_2)$ . For nodes belonging to sink components we have  $H_{ik} = 1$  while for the transient node  $c$ , the row  $(1/3, 2/3)$  is obtained by the recursive relation

$$H_{c\cdot} = P_{cb} H_{b\cdot} + P_{cd} H_{d\cdot} = \frac{1}{3}[1, 0] + \frac{2}{3}[0, 1] = \left[ \frac{1}{3}, \frac{2}{3} \right],$$

meaning that starting from  $c$  the process eventually ends in  $S_1$  with probability  $1/3$  and in  $S_2$  with probability  $2/3$ .

**Asymptotic behaviour of each component.** Collecting the previous expressions, the asymptotic opinion profile is given by

$$x(\infty) = H \bar{x} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_1 \\ \bar{x}_1 \\ \frac{1}{3}\bar{x}_1 + \frac{2}{3}\bar{x}_2 \\ \bar{x}_2 \end{bmatrix}.$$

Hence:

- The sink component  $S_1 = \{o, a, b\}$  reaches an internal consensus: all three nodes converge to the same limit value  $\bar{x}_1$ . This value depends only on the initial opinions of  $\{o, a, b\}$  through the invariant distribution  $\pi^{(1)}$ .
- The sink component  $S_2 = \{d\}$  behaves as a stubborn agent:  $x_d(t) = x_d(0)$  for all  $t$ , so  $\bar{x}_2 = x_d(0)$  and does not depend on the rest of the network.

- The transient node  $c$  does not reach a consensus with the rest of the nodes. Its asymptotic value is a convex combination of the two sink equilibria:

$$x_c(\infty) = \frac{1}{3}\bar{x}_1 + \frac{2}{3}\bar{x}_2,$$

which depends on the initial configuration only via  $\bar{x}_1$  and  $\bar{x}_2$ , and is completely independent of the initial opinion  $x_c(0)$ .

Therefore the dynamics no longer converges to a global consensus: instead, the network exhibits two distinct equilibrium levels associated with the two sink components, and the transient node converges to a weighted combination of these levels.

**Numerical evidence.** The theoretical predictions are confirmed by simulations of the French-DeGroot dynamics using the matrix  $P_g$ . For a representative random initialization  $x(0)$ , the theoretical asymptotic opinion vector computed from  $H\bar{x}$  is

$$x_{\text{theo}}(\infty) \approx [0.5334, 0.5334, 0.5334, 0.7582, 0.8706],$$

while the state of the system after  $10^3$  iterations is

$$x(10^3) \approx [0.5334, 0.5334, 0.5334, 0.7582, 0.8706],$$

with a discrepancy  $\|x(10^3) - x_{\text{theo}}(\infty)\|$  of order  $10^{-15}$ , which is numerically negligible.

In Figure 6 we report the time evolution of: (i) the average opinion of the sink component  $\{o, a, b\}$ ; (ii) the opinion of node  $c$ ; and (iii) the opinion of node  $d$ , for the same initialization  $x(0)$ . The component  $\{o, a, b\}$  rapidly collapses to its internal consensus  $\bar{x}_1$ , node  $d$  remains constant at  $x_d(0)$ , and node  $c$  converges to an intermediate value between these two levels, in agreement with the analytical expression for  $x_c(\infty)$ .

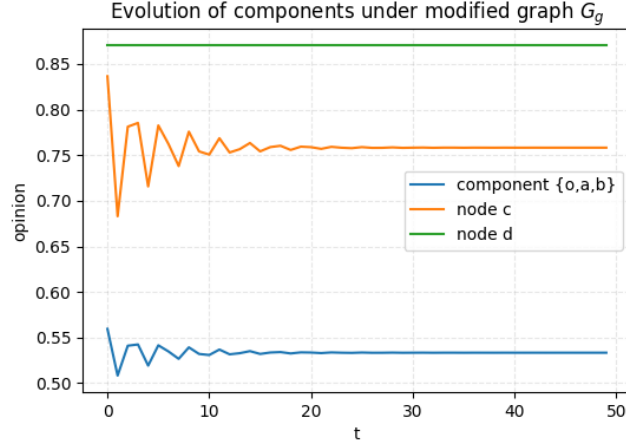


Figure 6: Time evolution of the average opinion on the sink component  $\{o, a, b\}$ , the opinion of the transient node  $c$ , and the opinion of the stubborn node  $d$  under the dynamics  $x(t+1) = P_g x(t)$ .

**Independence from the transient node initialization.** To explicitly illustrate the role of the transient node, we run additional simulations where the initial opinions of nodes  $\{o, a, b, d\}$  are kept fixed (same initialization as above), while  $x_c(0)$  is varied in the interval  $[0, 1]$ . For each initialization the dynamics is evolved until convergence and the final value  $x_c(\infty)$  is recorded.

According to the theoretical prediction, the asymptotic value of node  $c$  depends only on the equilibria of the two sink components and is given by

$$x_c(\infty) = \frac{1}{3}\bar{x}_1 + \frac{2}{3}\bar{x}_2 \approx 0.7582,$$

independently of the initial value  $x_c(0)$ . The numerical simulations confirm this behavior: for all tested values of  $x_c(0)$ , the node  $c$  converges to the same limit value, coinciding with the one predicted by the theory.

## h) Asymptotic behaviour after removing edges $(b, o)$ and $(d, a)$

**Graph modification and structure.** In this point we modify the original network by removing the edges  $(b, o)$  and  $(d, a)$  from the graph  $G$ , obtaining a new directed graph  $G_h = (V, \mathcal{E}_h, \Lambda_h)$  with the same node set  $V = \{o, a, b, c, d\}$ . The resulting topology is shown in Figure 7. Nodes  $o$  and  $a$  only send information towards the rest of the network, but they do not receive any edge back from  $b, c, d$ . Conversely, nodes  $b, c, d$  form a closed strongly connected component: from any of these three nodes it is possible to reach the other two, and once the dynamics enters this set it can never leave it. Therefore,  $\{b, c, d\}$  is the unique sink component of  $G_h$ , whereas  $\{o, a\}$  are transient nodes.

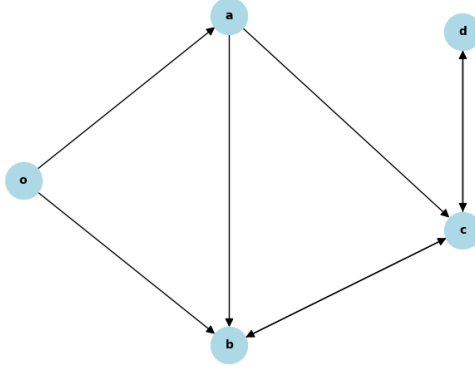


Figure 7: Modified graph  $G_h$  obtained by removing the edges  $(b, o)$  and  $(d, a)$ . The set  $\{b, c, d\}$  is the unique sink strongly connected component, while nodes  $\{o, a\}$  are transient.

**Periodicity of the sink component.** Let  $\Lambda_h$  denote the weight matrix of  $G_h$ . The corresponding row-stochastic matrix is

$$P_h = D_h^{-1} \Lambda_h, \quad D_h = \text{diag}(\omega_h), \quad (\omega_h)_i = \sum_j (\Lambda_h)_{ij}.$$

If we reorder the states so that the transient nodes come first, the matrix  $P_h$  admits the block decomposition

$$P_h = \begin{pmatrix} Q & R \\ 0 & P_{SS} \end{pmatrix},$$

where  $Q$  describes transitions among transient nodes,  $P_{SS}$  is the restriction of  $P_h$  to the sink strongly connected components, and  $R$  encodes the influence of sink components on transient nodes. The corresponding dynamics can be written as

$$x_T(t+1) = Q x_T(t) + R x_S(t), \quad x_S(t+1) = P_{SS} x_S(t).$$

where  $x_T = (x_o, x_a)^\top$  collects the opinions of transient nodes and  $x_S = (x_b, x_c, x_d)^\top$  those of the sink component  $\{b, c, d\}$ .

**Remark** The block decomposition of the transition matrix  $P_h$  does not rely on the presence of stubborn nodes, but follows from the reducible structure of the graph. Indeed, since the modified graph  $G_h$  contains a closed sink strongly connected component  $\{b, c, d\}$  and the remaining nodes  $\{o, a\}$  are transient with respect to it, the matrix  $P_h$  can be reordered in block triangular form. This decomposition is a purely structural property of any reducible row-stochastic matrix and holds independently of the specific dynamics on the sink component. In particular, models with stubborn agents correspond to the special case in which  $P_{SS} = I$ , whereas here  $P_{SS}$  is periodic.

Hence  $P_{SS}$  is the restriction of  $P_h$  to  $\{b, c, d\}$  and it is irreducible but periodic: all cycles inside  $\{b, c, d\}$  have even length and there are no self-loops. This involves on the fact that the assumptions of the general convergence theorem for French-DeGroot dynamics, which require all sink components to be aperiodic, are not satisfied. Consistently, in a representative run the last two opinion vectors before the final iteration,

$$x_h(n_{\text{it}} - 2) = [0.8128, 0.8247, 0.7892, 0.8365, 0.7892],$$

$$x_h(n_{\text{it}} - 1) = [0.8128, 0.8010, 0.8365, 0.7892, 0.8365],$$

differ by

$$x_h(n_{\text{it}} - 1) - x_h(n_{\text{it}} - 2) = [0, -0.0236, 0.0473, -0.0473, 0.0473],$$

showing that the opinions of the sink nodes keep oscillating instead of collapsing to a fixed point. As a consequence, the process  $x_S(t+1)$  does not converge to a fixed consensus value, but rather exhibits a two-cycle: the opinions of nodes  $b, c, d$  oscillate between two patterns instead of stabilising.

**Dynamics of transient nodes.** The transient subsystem  $x_T(t+1)$  is governed by the matrix  $Q$ , obtained by restricting  $P_h$  to transient nodes only. In our case  $Q$  is strictly sub-stochastic and has spectral radius  $\rho(Q) < 1$ , so the dynamics  $x_T(t+1) = Qx_T(t)$  is stable: if it were decoupled from the sink, it would drive the transient opinions to zero. The full solution can be written as

$$x_T(t) = Q^t x_T(0) + \sum_{k=0}^{t-1} Q^{t-1-k} R x_S(k).$$

As said, first term vanishes as  $t \rightarrow \infty$  while second term is a weighted sum of past values of the sink opinions  $x_S(k)$ . Since  $x_S(k)$  is bounded and periodic with period 2, the sequence of partial sums converges, and the transient opinions  $x_o(t)$  and  $x_a(t)$  approach constant limits. Intuitively, the transient block behaves like a low-pass filter: it does not inherit the oscillations of the sink component, but averages them out, producing stable asymptotic values for nodes  $o$  and  $a$ .

**Numerical behaviour and time evolution.** We now simulate the French-DeGroot dynamics on  $G_h$  using the matrix  $P_h$ , starting from a random initial condition  $x_0 \in [0, 1]^5$ . After  $n_{\text{it}} = 100$  iterations, the state of the system is

$$x_h(100) \approx [0.8128, 0.8010, 0.8365, 0.7892, 0.8365],$$

which is consistent with the oscillatory pattern discussed above. The time-averaged opinions of the individual nodes over the same run are

$$\bar{x}_o \approx 0.8104, \quad \bar{x}_a \approx 0.8123, \quad \bar{x}_b \approx 0.8128, \quad \bar{x}_c \approx 0.8128, \quad \bar{x}_d \approx 0.8128.$$

Although the instantaneous values of the nodes in the sink component do not converge, their long-run averages are almost identical and close to a common level. The same holds for node  $a$ , whose oscillations persist but remain centred around the same mean value.

The time evolution of the average opinion  $\frac{1}{5} \sum_i x_i(t)$  is reported in Figure 8. After an initial transient, the mean opinion settles around a constant level but keeps oscillating slightly with period 2, reflecting the persistent periodic behaviour of the sink component. Therefore, the network does not converge to a pointwise equilibrium: the global average opinion does not fully stabilise, but fluctuates around its limiting value.

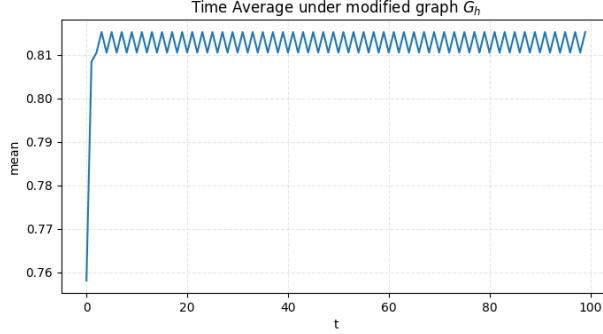


Figure 8: Mean opinion over time under the dynamics  $x(t+1) = P_h x(t)$ . The trajectory rapidly approaches a neighbourhood of its limiting level, but retains a small period-two oscillation induced by the periodic sink component.

A more detailed picture is given in Figure 9, where we plot the trajectory of each node during the first iterations. Nodes  $b, c, d$  exhibit a clear alternating pattern: their opinions oscillate with period 2, as predicted by the spectral properties of the sink submatrix  $P_{SS}$ . Node  $o$ , which belongs to the transient subsystem governed by the matrix  $Q$ , converges to a constant value, since repeated applications of  $Q$  progressively filter out any oscillatory input. In contrast, node  $a$  does not converge to a fixed limit: being directly connected to the periodic sink, it receives a fresh oscillatory signal at every iteration, before the action of  $Q$  can attenuate it. As a result,  $a$  inherits the period-two oscillation of the sink component, although with bounded amplitude.

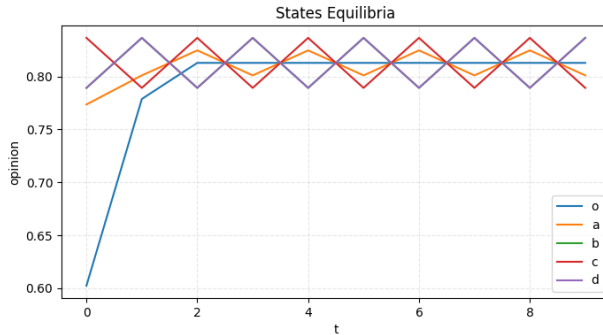


Figure 9: Evolution of the opinions of all nodes in the modified graph  $G_h$  during the first iterations. Nodes in the sink component  $\{b, c, d\}$  oscillate with period 2; node  $o$  converges to a constant value, while node  $a$  continues to oscillate due to its direct coupling with the periodic sink.

In summary, removing the edges  $(b, o)$  and  $(d, a)$  yields a single periodic sink component  $\{b, c, d\}$  and two transient nodes  $\{o, a\}$ , with qualitatively different behaviours. Because the sink component

is periodic, the standard convergence result for French–DeGroot dynamics does not apply and the process does not converge to a consensus vector. The sink nodes oscillate indefinitely on a two-cycle; node  $o$  converges to a fixed value due to the stability of the transient subsystem governed by  $Q$ ; and node  $a$ , lying at the interface between the sink and the transient block, continues to oscillate. The overall dynamics therefore consist of a mixture of periodic behaviour and asymptotic convergence, in full agreement with both the theoretical analysis and the numerical simulations.

## Exercise 2

### a) Particle perspective

We performed the simulation with a fixed Poisson clock of rate  $N\omega^*$ . At each iteration, we sampled the next time as

$$t_{next} = -\frac{\ln u}{N\omega^*}$$

where  $u \sim \mathcal{U}([0, 1])$  is sampled from the uniform distribution on the interval  $[0, 1]$ . Whenever a particle returned we removed it from the simulation, effectively decreasing  $N$ , and continued the process until no particle was left. Notice that we chose not to continue the simulation with all the particles for some fixed number of iterations or maximum time. This will be explored in the section comparing convergence times below. On average, a group of  $N = 100$  particles returned in  $a$  after 6.704 time units, with a standard deviation of 0.51. To estimate this range, we performed the experiment  $n_{exp} = 100$  times, and computed the empirical mean and standard deviation  $\hat{\mu}$  and  $\hat{\sigma}$  as follows:

$$\hat{\mu} \approx \frac{1}{n_{exp}} \sum_{i=1}^{n_{exp}} \bar{X}_i, \quad \hat{\sigma} \approx \sqrt{\frac{1}{n_{exp} - 1} \sum_{i=1}^{n_{exp}} (\bar{X}_i - \hat{\mu})^2}.$$

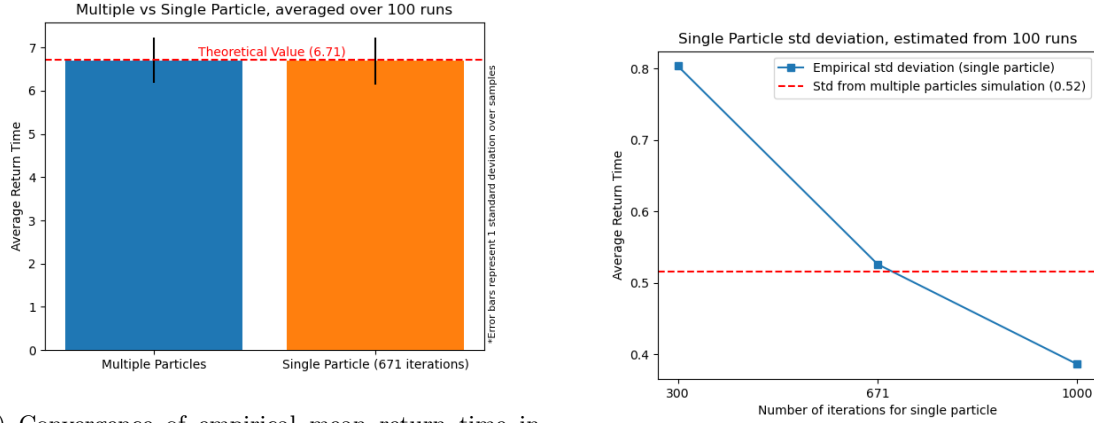
This is consistent with the expected convergence to the theoretical value of 6.708.

**Comparison between this simulation and the Exercise 1** The first observation to make in this context is that the method converges nicely to the theoretical value of 6.708. However, we can also compare this method with the one involving a single particle, as computed in Exercise 1a. Throughout this comparison we used a single global Poisson clock to simulate the case with a single particle.

We have verified experimentally that both methods converge to the real expected return value. However, we could ask ourselves the number of iterations needed for the single particle method, to obtain the same standard deviation as the one with  $N$  particles.

In theory, we would like to stop after exactly  $N$  returns, effectively performing an equivalent simulation. Given a theoretical return time  $\tau$  and a number  $N$  of particles, it is trivial to see that we would need to run the simulation about  $N\tau$  times to have an equivalent estimate. In our case, this corresponds to approximately  $100 \cdot 6.708 = 670.8$  units of time. Indeed, we can experimentally verify that the standard deviation decreases as we increase the number of iterations per experiment, and the two values cross around 671. These results are shown in Figure 10.

**Convergence comparison** We can also compare how convergence varies as we add more particles without removing any one of them, and fixing the number of iterations or the simulated time units. We tested this with 1, 10 and 100 particles and computed the mean value over 20 samples. The results are shown in Figures 11a and 11b. We can observe that the case with one particle is the faster to converge in terms of iterations, while, as we add more particles we tend to underestimate the real value. This is expected, as for a fixed number of iterations, we are effectively weighting more the particles that converge faster. This phenomenon is also true when we fix a maximum time, as it is shown in Figure 11b. However, we can see that as we add more particles, we also reduce the variance and the estimates become more stable.



(a) Convergence of empirical mean return time in both methods. The error bars indicate one standard deviation.

(b) Standard deviation of using a single particle as in Exercise 1.

Figure 10: Comparison of simulation outputs.

## b) Node perspective

Let  $n_t(i)$  be the number of particles contained in node  $i$  at time  $t$ , and  $N = \sum_i n_t(i)$ . Furthermore, let each node be endowed with a Poisson clock of rate  $\omega^* n_t(i)$ . At each tick of this clock, it lets one particle leave and move to a node  $j$ , which may be itself, with probability  $\bar{P}_{ij}$ . Due to the thinning property this system is equivalent to a system endowed with a single global Poisson clock of rate  $\omega^* N$  such that at each tick we sample a node  $i$  with probability  $n_t(i)/N$  and we move one particle from node  $i$  to node  $j$  with probability  $\bar{P}_{ij}$ .

Indeed to simplify the simulation, we chose to adopt a single global clock as defined above.

**Simulation with  $N = 100$  and 60 time units** We simulated the system with  $N = 100$  and 60 time unit, assuming they all start in node  $a$ . We expect the distribution of the particles between the nodes to converge to the invariant probability distribution  $\bar{\pi}$  s.t.  $\bar{P}^\top \bar{\pi} = \bar{\pi}$ , i.e.  $\bar{\pi}$  needs to be a left eigenvector.

We can estimate this eigenvector numerically, and obtain that

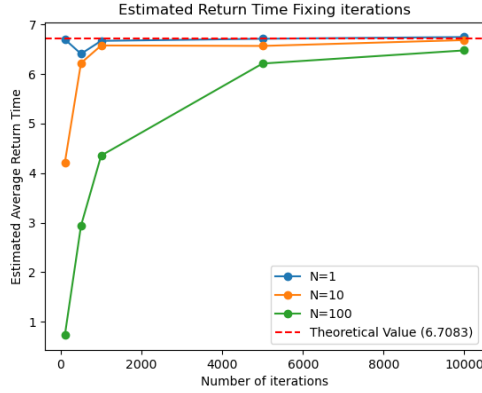
$$\bar{\pi} \approx (0.2173913, 0.14906832, 0.26086957, 0.1863354, 0.1863354)^\top.$$

After running our simulation once and we obtained this average particle position:

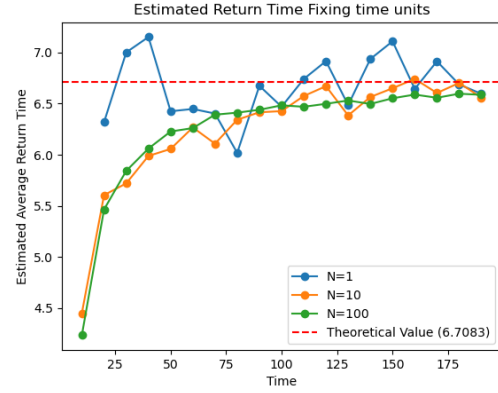
- Node o:  $20.6550 \pm 0.1280$  (theoretical:  $\sim 21.7$ )
- Node a:  $15.4637 \pm 0.2045$  (theoretical:  $\sim 14.9$ )
- Node b:  $26.3963 \pm 0.1135$  (theoretical:  $\sim 26.1$ )
- Node c:  $18.6839 \pm 0.1077$  (theoretical:  $\sim 18.6$ )
- Node d:  $18.8012 \pm 0.1235$  (theoretical:  $\sim 18.6$ )

where, given the empirical mean and standard deviation  $\hat{\mu}$  and  $\hat{\sigma}$ , the 95% confidence interval is obtained by

$$CI = \left[ -1.96 \frac{\hat{\sigma}}{\sqrt{n_{\text{samples}}}}, 1.96 \frac{\hat{\sigma}}{\sqrt{n_{\text{samples}}}} \right],$$



(a) Results fixing the number of iterations

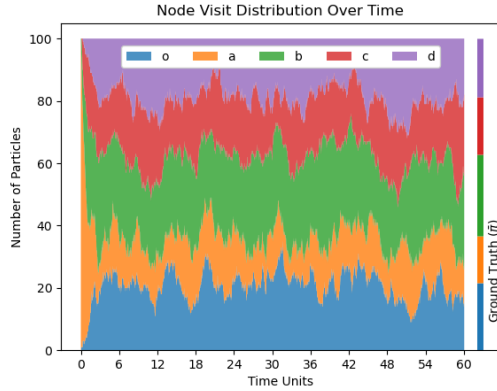


(b) Results fixing the number of iterations

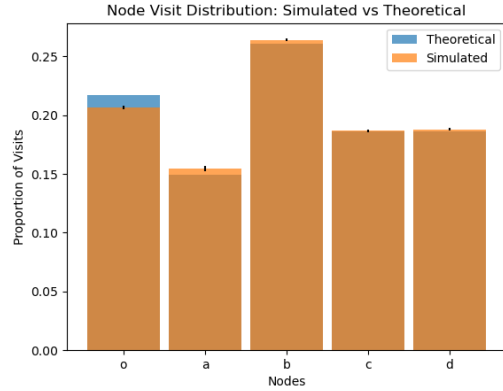
Figure 11: Convergence of the mean value over 20 samples to the expected values with different numbers of particles, fixing either the number of iterations or the number of time units.

under assumptions of normally distributed samples. These values are close to the theoretical  $\bar{\pi}$ , however only one confidence interval includes the theoretical value (obtained by multiplying  $\pi$  by  $N$ ).

In Figure 12a we show the distribution of the particles over time, while Figure 12b visually compares  $\bar{\pi}$  and the estimated distribution.



(a) Evolution of the particle distribution with  $N = 100$  over 60 time units



(b) Visual comparison of  $\bar{\pi}$  with the simulated estimate

Figure 12: Evolution of the particle distribution and comparison with  $\bar{\pi}$

## Convergence verification

The original question asked to compute the average of the samples over a single run, however this estimate is imprecise due to the initial values of the mean being distant from the converged value. Indeed, if we estimate the confidence intervals as above on the different samples we obtain this distribution:

the corresponding distribution is shown in Figure 13. Notice that in all the nodes but one (node  $c$ ) the real value is not included in the confidence interval.

Conversely, when running 1000 times the simulation and estimating the 95% CI with our method, we obtain these values:



- Node o:  $21.8540 \pm 0.2447$  (theoretical:  $\sim 21.7$ )
- Node a:  $14.8010 \pm 0.2175$  (theoretical:  $\sim 14.9$ )
- Node b:  $26.2530 \pm 0.2770$  (theoretical:  $\sim 26.1$ )
- Node c:  $18.8240 \pm 0.2414$  (theoretical:  $\sim 18.6$ )
- Node d:  $18.2680 \pm 0.2416$  (theoretical:  $\sim 18.6$ )

which are closer and correctly include the theoretical value in all the confidence intervals but one (node  $d$ ). The corresponding distribution is shown in Figure 13.

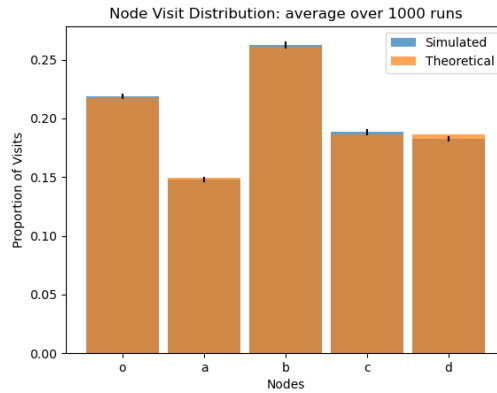


Figure 13: Average position of the final values over 1000 samples, compared with  $\bar{\pi}$ . Error bars indicate 95% CI.

## Exercise 3

### Graph specification

The system is modeled as an *open directed graph*  $G = (V, E)$  with node set

$$V = \{o, a, b, c, d\},$$

where particles enter the network externally at node  $o$  and may leave the system through node  $d$ . The graph is open since mass is not conserved: arrivals from the environment and departures from the system occur according to independent Poisson processes.

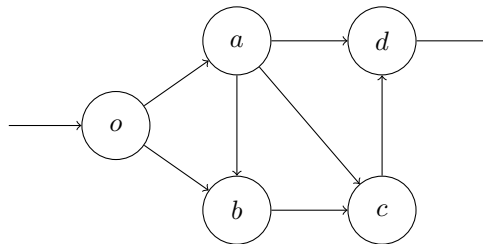


Figure 14: The open network of reference. External arrows indicate arrival at node  $o$  and departure from node  $d$ .

Node  $d$  is the unique *sink component* of the network and is *globally reachable* from all other nodes, ensuring that every particle eventually reaches the exit. Interaction with the external environment is modeled by assigning to node  $d$  a fixed service rate  $\omega_d = \frac{7}{4}$ : whenever its Poisson clock ticks, one particle (if present) leaves the system. Whereas, particles enter the network at node  $o$  according to a Poisson process with rate  $\lambda$ , which represents the external arrival intensity.

The routing structure of the internal network is encoded by the transition rate matrix

$$\Lambda_{\text{open}} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{2}{4} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

with rows and columns ordered as  $(o, a, b, c, d)$ . The associated internal Poisson rates are given by

$$\omega^{\text{int}} = \Lambda_{\text{open}} \mathbf{1},$$

where  $\mathbf{1}$  denotes the all-ones vector. Note that the last component of  $\omega^{\text{int}}$  is zero, since node  $d$  has no outgoing internal transitions.

To derive a corresponding transition probability matrix

$$P = D^{-1} \Lambda_{\text{open}},$$

with  $D = \text{diag}(\omega)$ , we temporarily introduce a self-loop at node  $d$  so that  $D$  becomes non-singular.

An equivalent and intuitive interpretation of the open network can be obtained by embedding it into a larger closed framework via the introduction of two auxiliary nodes. A fictitious source node  $o'$  is connected to node  $o$  with rate  $\lambda$ , modeling the external generation of particles, while a fictitious sink node  $d'$  is connected from node  $d$  with rate  $\omega_d$ , modeling departures from the system. Under this interpretation, entry and exit are represented explicitly as directed transitions.

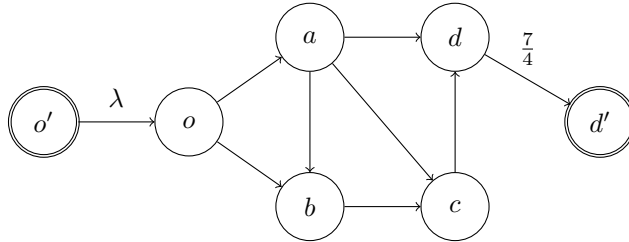


Figure 15: Open network with external source  $o'$  and sink  $d'$ .

### a) Proportional Rate Dynamics

**Modeling assumptions** In the proportional rate scenario, each node processes particles at a rate proportional to its instantaneous occupancy. Denoting by  $N_i(t)$  the number of particles at node  $i$  at time  $t$ , the total outgoing rate from node  $i$  is given by

$$\omega_i N_i(t),$$

where  $\omega_i$  is the local Poisson rate associated with node  $i$ .

This assumption admits a simple microscopic interpretation: each particle located at node  $i$  is associated with an independent exponential waiting time with parameter  $\omega_i$ . As a consequence, nodes with larger occupancies are activated more frequently, and the effective processing capacity of the network increases with congestion.

The proportional rate dynamics are simulated using a continuous-time Markov chain framework driven by a single global clock. At each event time, one among several competing Poisson processes is selected, representing either the arrival of a new particle from the environment or the activation of a node-level transition.

The system state is fully described by the vector

$$N(t) = (N_o(t), N_a(t), N_b(t), N_c(t), N_d(t)),$$

where the input event adds one particle to node  $o$ , while each internal event removes one particle from the chosen node and forwards it according to the routing probabilities encoded in the matrix  $P = D^{-1}\Lambda_{\text{open}}$ .

In the proportional rate case, the activation coefficient of node  $i$  coincides with its instantaneous particle count  $N_i(t)$ . Consequently, the vector of event rates takes the form

$$(\lambda, \omega_o N_o(t), \omega_a N_a(t), \omega_b N_b(t), \omega_c N_c(t), \omega_d N_d(t)),$$

where the first component corresponds to the external input process.

At each step of the simulation, the time to the next event is sampled from an exponential distribution with parameter equal to the sum of all rates. The specific event is then selected with probability proportional to its rate.

Whenever a node  $i \neq d$  is selected, one particle is removed from node  $i$  and routed to a successor node according to the transition probabilities specified by the internal routing matrix. If node  $d$  is selected, the corresponding particle exits the system.

**Numerical experiment** In accordance with the exercise instructions, the proportional rate system was simulated by setting the external input rate to  $\lambda = 100$  and running the dynamics for a time horizon of 60 time units. Figure 16 reports the evolution of the particle counts at each node over time.

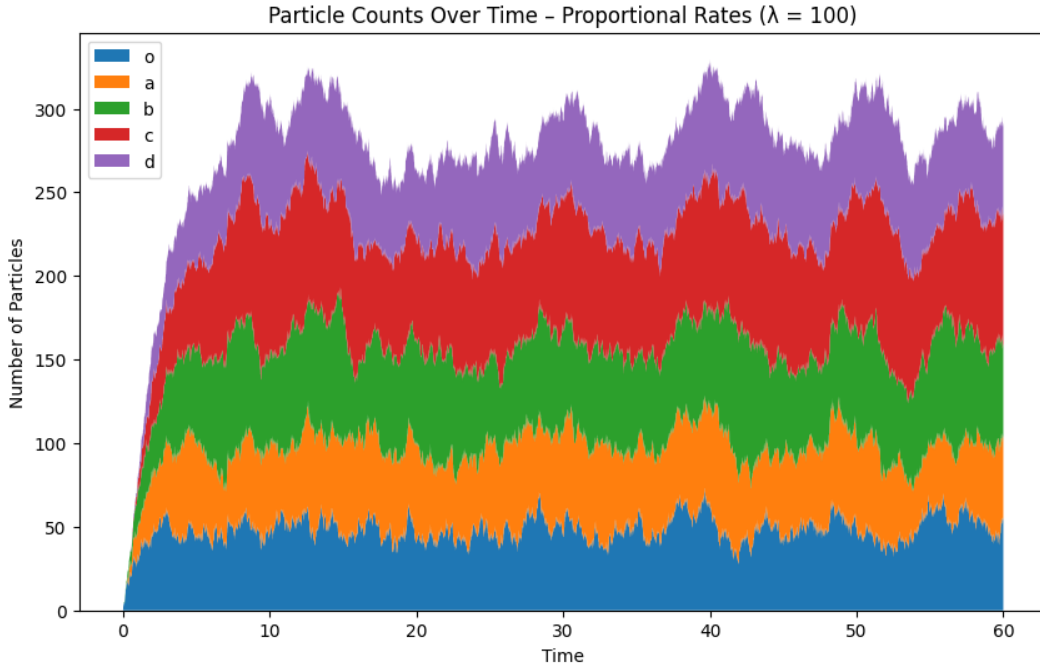


Figure 16: Particle counts at each node over time for the proportional rate case with  $\lambda = 100$ .

From the plot, it can be observed that after an initial transient phase the number of particles at each node fluctuates around a stable average level. As a consequence, the total number of particles in

the system also stabilizes, remaining close to a finite value of approximately 300. No long-term growth trend is observed, indicating that the system operates in a stable regime under proportional service.

The stacked representation further shows that particle populations are approximately evenly distributed across the nodes, with comparable average occupancies. The only notable exception is node  $c$ , which appears mildly more congested than the others. This behavior can be explained by analyzing the balance of average flows in the network.

Let  $x_i$  denote the average input rate through node  $i$  in stationarity. In the proportional rate case, the expected value of the total outgoing rate from node  $i$  is given by  $\omega_i \mathbb{E}[N_i]$ . Stability requires that, at equilibrium, the average incoming and outgoing rates at each node coincide, as each node represents an  $M/M/\infty$  queue. In fact, the presence of  $N_i$  in the expected outgoing rate, ensures that, if  $x_i < \omega_i N_i$ , then the number of particles going out would exceed the number of particles going, on average. This would therefore reduce the value of  $N_i$ . Similarly, if  $x_i > \omega_i N_i$ , then the number of particles going in would exceed the number of particles going out, increasing  $N_i$ . The only stable value is the one with equal number of particles entering and exiting, corresponding to  $x_i = \omega_i N_i$ . This leads to the system of balance equations

$$\omega_i \mathbb{E}[N_i] = x_i, \quad i \in V,$$

with the additional condition that the external input flow satisfies  $x_o = \lambda$ .

Solving these equations using the routing structure encoded in  $\Lambda_{\text{open}}$  yields the theoretical equilibrium occupancies

$$\mathbb{E}[N]_{\text{theory}} = (50, 50, 62.5, 75.0, 57.14).$$

These values reflect the progressive accumulation of particles along the direction of the flow and explain why node  $c$ , which collects traffic from multiple upstream paths, exhibits a higher average occupancy despite comparable local service rates.

The proportional dependence of the service rates on the occupancies ensures that the system does not blow up for any finite input rate  $\lambda$ . As particle counts increase, the aggregate forwarding and exit activity increases proportionally, allowing the network to continuously compensate the incoming flow without saturating any node.

The experimental averages computed from the simulation of length 60 are given by

$$\mathbb{E}[N]_{\text{exp}} = (48.7, 49.47, 59.48, 69.21, 54.18),$$

which are in close agreement with the theoretical prediction. The discrepancy between theoretical and experimental values, measured in Euclidean norm, is

$$\|\mathbb{E}[N]_{\text{theory}} - \mathbb{E}[N]_{\text{exp}}\|_2 \approx 7.31.$$

This small error (less than 6%) confirms that the simulation faithfully reproduces the theoretical equilibrium behavior derived from flow balance arguments, thereby validating both the modeling assumptions and the numerical implementation of the proportional rate dynamics.

## b) Fixed rate dynamics

**Modeling assumptions** In the fixed rate scenario, nodes forward particles according to Poisson clocks with rates that are independent of the number of particles currently present in the node. More precisely, each node  $i \in V$  is associated with a fixed activation rate  $\omega_i$ , regardless of its instantaneous occupancy.

From an operational perspective, the fixed rate assumption can be interpreted as introducing a finite service capacity along each outgoing edge of the network. While particles may accumulate at a node, the rate at which they can be forwarded remains constant, effectively limiting the maximum throughput that the node can sustain. In this sense, congestion is no longer self-regulating, and the network may fail to compensate increasing input flows.

Particles still enter the system at node  $o$  according to a Poisson process with rate  $\lambda$ , and internal transitions follow the same routing structure defined by the matrix  $P = D^{-1}\Lambda_{\text{open}}$ . The only modification concerns the definition of node activation rates. In the fixed rate regime, the activation coefficient associated with node  $i$  is given by a binary variable,

$$\mathbb{I}\{N_i(t) > 0\},$$

which equals 1 if at least one particle is present at node  $i$  and 0 otherwise. Consequently, the total activation rate of node  $i$  is

$$\omega_i \mathbb{I}\{N_i(t) > 0\},$$

so that no forwarding events can occur from an empty node, while a non-empty node activates at a constant rate independent of its occupancy.

As a result, the vector of instantaneous event rates takes the form

$$(\lambda, \omega_o \mathbb{I}\{N_o(t) > 0\}, \omega_a \mathbb{I}\{N_a(t) > 0\}, \omega_b \mathbb{I}\{N_b(t) > 0\}, \omega_c \mathbb{I}\{N_c(t) > 0\}, \omega_d \mathbb{I}\{N_d(t) > 0\}).$$

Aside from this change in rate activation, all other aspects of the simulation framework—including event selection, time sampling, and routing—remain identical to those described for the proportional rate dynamics.

**Numerical experiments** Following the specifications of the exercise, the fixed rate dynamics were first simulated by setting the external input rate to  $\lambda = 2$  and running the system for a time horizon of 6000 time units. Figure 17 reports the stacked evolution of the particle counts at each node over time.

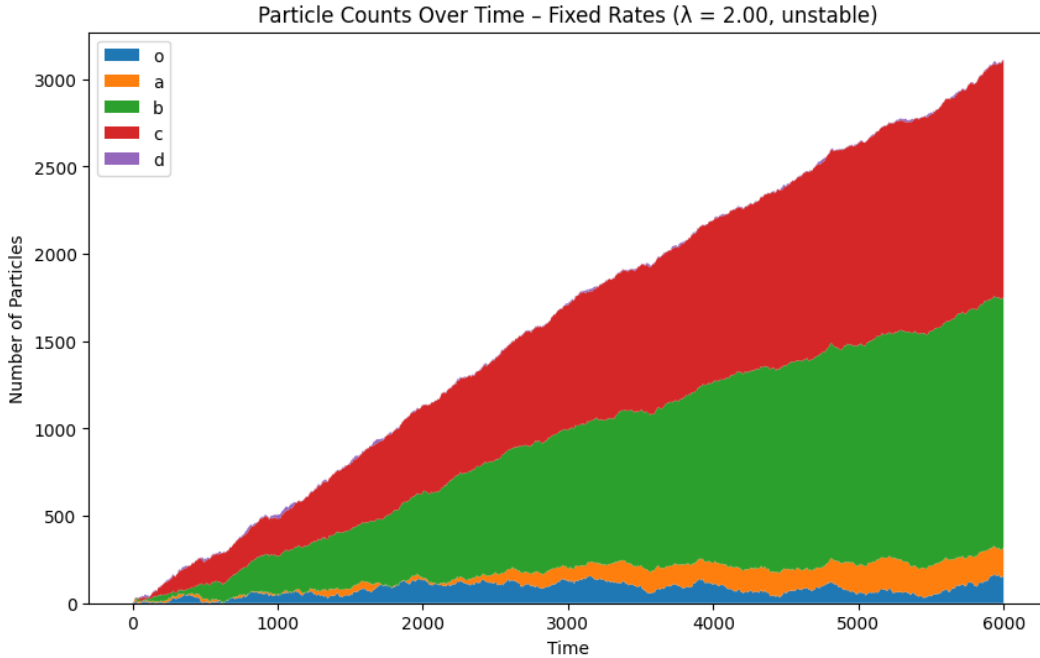


Figure 17: Stacked particle counts over time for the fixed rate case with  $\lambda = 2$ .

From the plot, it is immediately evident that, unlike in the proportional rate scenario, the total number of particles in the system does not stabilize. Instead, it grows steadily and without bound as time evolves, indicating that for this value of  $\lambda$  the network operates close to a blow-up regime.

A closer inspection of the stacked representation reveals a highly uneven distribution of congestion across the nodes. In particular, nodes  $b$  and  $c$  are by far the most congested ones: their particle counts increase monotonically and dominate the overall growth of the system. This behavior is a direct consequence of the network topology.

The observed instability can be traced back to the lack of adaptivity inherent in the fixed rate regime. While the incoming flow increases with  $\lambda$ , the service capacity of each node remains constrained by its fixed rate  $\omega_i$ . Once the incoming traffic exceeds the maximum sustainable throughput imposed by these rates, the system is no longer able to evacuate particles fast enough, and mass accumulates indefinitely.

This phenomenon can also be analyzed analytically using similar considerations to the case above. This time, we know that to avoid a blow-up the average input rate of each node  $x_i$  must not exceed the average out-going rate  $\omega_i$ . We can therefore solve the set of inequality constraints

$$x_i \leq \omega_i, \quad i \in V,$$

with equality corresponding to a critical, unstable equilibrium. Solving analytically the resulting system of balance equations induced by the routing structure of  $\Lambda_{\text{open}}$  yields a critical input rate

$$\lambda_c = \frac{4}{3},$$

beyond which the network cannot sustain the incoming traffic and necessarily blows up.

To validate this theoretical prediction, the experiment was repeated with an input rate slightly below the critical threshold, namely  $\lambda = 1.32$ . In this case, the system exhibits a markedly different behavior.

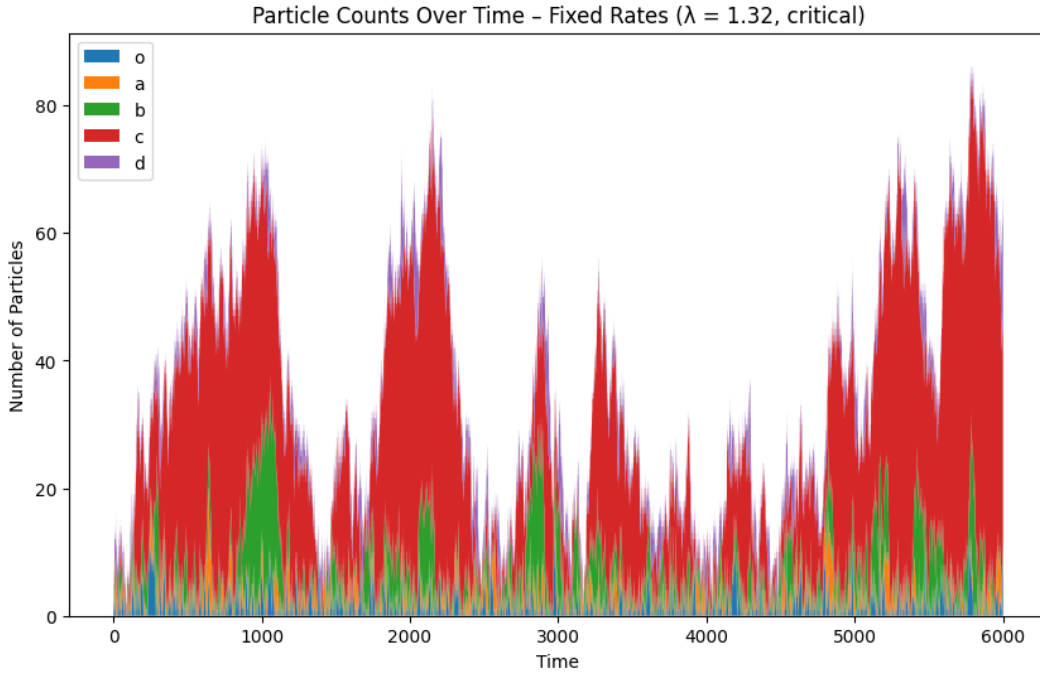


Figure 18: Stacked particle counts over time for the fixed rate case with  $\lambda = 1.32$ .

While the dynamics remain highly fluctuating and operate close to saturation, the total number of particles present in the system no longer diverges. Instead, large but finite oscillations are observed, with peak occupancies remaining below approximately 80 particles overall.

Even in this near-critical regime, node  $c$  consistently emerges as the most congested one, reflecting its central position in the routing structure and its limited forwarding capacity. The system thus reaches an operational point that is stable, yet extremely sensitive to perturbations in the input rate, in full agreement with the analytical characterization of the critical threshold.