

# IE 370, Spring 2024 Homework 8

Chris Lee

April 24, 2024

```
[ ]: import numpy as np
```

## 1 Problem 1. (25 points)

The answers in this problem have to be actual numbers. And you have to show clearly how you derive/prove your answers. Buses A and B arrive at a bus stop as mutually independent Poisson processes with rates 2 buses/hour and 3 buses/hour, respectively. Alice and Bob arrive at the bus stop simultaneously at 4pm; Alice needs to take bus A, and Bob needs to take bus B.

- 1.1. (10 points) What is the probability that both Alice and Bob will take a bus before 4:20pm?
- 1.2. (15 points) What is the probability that Alice and Bob will take their buses within 20 min of each other?

*Solution:*

```
[ ]: lambda_A = 2
lambda_B = 3
t = 1 / 3

P_A_before_420 = 1 - np.exp(-lambda_A * t)
P_B_before_420 = 1 - np.exp(-lambda_B * t)
P_both_before_420 = P_A_before_420 * P_B_before_420
f"1.1: Probability both Alice and Bob will take a bus before 4:20pm:␣
↪{P_both_before_420:.4f}"
```

```
[ ]: '1.1: Probability both Alice and Bob will take a bus before 4:20pm: 0.3076'
```

```
[ ]: # Computes the integral for one term of the probability that |X-Y| <= t where X~
↪exp(lambda_x) and Y ~ exp(lambda_y)
def integration_term(lambda_x, lambda_y, t):
    return lambda_x / (lambda_x + lambda_y) * (1 - np.exp(-(lambda_x +
↪lambda_y) * t))

P_within_20min = integration_term(lambda_A, lambda_B, t) + integration_term(
    lambda_B, lambda_A, t
)
```

```
f"1.2: Probability that Alice and Bob will take their buses within 20 min of_
↵each other: {P_within_20min:.4f}"
```

```
[ ]: '1.2: Probability that Alice and Bob will take their buses within 20 min of each
other: 0.8111'
```

## 2 Problem 2. (25 points)

In this problem you are not given numerical values of the parameters. The answers have to be formulas, and you have to show clearly how you derive/prove your answers. Mary and John are employed by a small insurance agency, to handle customer calls. Customer calls are of three types: 1, 2 and 3. Type 1 calls arrive as Poisson process of rate  $\lambda_1$  (calls/hour); they are processes by John, with a processing time having exponential distribution with mean  $1/1$  (hours). Type 3 calls arrive as Poisson process of rate  $\lambda_3$  (calls/hour); they are processes by Mary, with a processing time having exponential distribution with mean  $1/3$  (hours). Type 2 calls arrive as Poisson process of rate  $\lambda_2$  (calls/hour); each type 2 call needs to be processed by John and Mary jointly (so they are both busy processing it), with a processing time having exponential distribution with mean  $1/2$  (hours). A type 1 call is accepted for processing (by John) only if John is free – otherwise this call is lost; a type 3 call is accepted for processing (by Mary) only if Mary is free – otherwise this call is lost; a type 2 call is accepted for processing (by John and Mary jointly) only if both John and Mary are free – otherwise this call is lost. All arrival processes and all processing times are independent

2.1. (10 points) Model this process as a continuous-time Markov chain. What are the states and transition rates? Draw the transition rate diagram.

2.2. (5 points) Write the system of linear equations, from which the stationary distribution  $\{\pi_i\}$  can be found for this Markov chain. ( $\pi_i$  is the stationary probability of state  $i$ , whatever your definition of states is.) You do not have to solve these equations, just make sure the equations are correct and the set of equations is sufficient to find the stationary distribution.

2.3. (10 points) Suppose Mary and John are paid at the rates of  $c_M$  and  $c_J$  dollars/hour, respectively. And they are only paid for time they actually busy processing calls (they are not paid for their idle time). In terms of the stationary distribution  $\{\pi_i\}$  and parameters  $c_M$  and  $c_J$ , write expressions for the long-run average rates at which Mary and John earn money (one expression for Mary and one for John).

### ***Solution:***

2.1:

Use a pair of variables  $(x, y)$  to represent the state of the availability of John and Mary to handle calls. - Say  $x = 0$  if John is free,  $x = 1$  if John is busy with a type 1 call,  $x = 2$  if John is busy with a type 2 call - Say  $y = 0$  if Mary is free,  $y = 1$  if Mary is busy with a type 1 call,  $y = 2$  if Mary is busy with a type 2 call

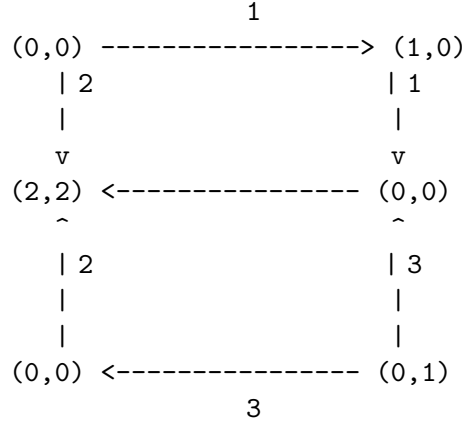
All possible states: -  $(0, 0)$  -  $(1, 0)$  -  $(0, 1)$  -  $(2, 2)$

Transition Rates:

- From  $(0, 0)$  (Both free)

- To  $(1,0)$  at rate  $\lambda_1$  (arrival of type 1 call)
- To  $(0,1)$  at rate  $\lambda_3$  (arrival of type 3 call)
- To  $(2,2)$  at rate  $\lambda_2$  (arrival of type 2 call)
- From  $(1,0)$  (John busy with type 1, Mary free)
  - To  $(0,0)$  at rate  $\mu_1$  (completion of type 1 call)
- From  $(0,1)$  (John free, Mary busy with type 3)
  - To  $(0,0)$  at rate  $\mu_3$  (completion of type 3 call)
- From  $(2,2)$  (Both busy with type 2)
  - To  $(0,0)$  at rate  $\mu_2$  (completion of type 2 call)

Transition rate diagram:



2.2:

The system of linear equations, from which the stationary distribution  $\{\pi_i\}$  can be found for this Markov chain: -  $\pi_{00}$  as the probability that both John and Mary are free. -  $\pi_{10}$  as the probability that John is busy with a type 1 call and Mary is free. -  $\pi_{01}$  as the probability that John is free and Mary is busy with a type 3 call. -  $\pi_{22}$  as the probability that both are busy with a type 2 call.

1. State  $(0,0)$  (both free):

$$\lambda_1\pi_{00} + \lambda_3\pi_{00} + \lambda_2\pi_{00} = \mu_1\pi_{10} + \mu_3\pi_{01} + \mu_2\pi_{22}$$

$$(\lambda_1 + \lambda_3 + \lambda_2)\pi_{00} = \mu_1\pi_{10} + \mu_3\pi_{01} + \mu_2\pi_{22}$$

2. State  $(1,0)$  (John busy with type 1, Mary free):

$$\lambda_1\pi_{00} = \mu_1\pi_{10}$$

3. State  $(0,1)$  (John free, Mary busy with type 3):

$$\lambda_3\pi_{00} = \mu_3\pi_{01}$$

4. State  $((2,2))$  (both busy with type 2):

$$\lambda_2\pi_{00} = \mu_2\pi_{22}$$

5. Normalization condition:

$$\pi_{00} + \pi_{10} + \pi_{01} + \pi_{22} = 1$$

2.3:

Mary earns money when she is busy processing calls. This occurs in the states where: - Mary is busy with a type 3 call  $\pi_{01}$  - Both are busy with a type 2 call  $\pi_{22}$

Mary's average earnings rate  $R_M$  is given by the sum of the probabilities of her being busy, each weighted by her hourly rate  $c_M$ . Her earnings rate can be expressed as:

$$R_M = c_M \cdot (\pi_{01} + \pi_{22})$$

John earns money when he is busy processing calls. This occurs in the states where: - John is busy with a type 1 call  $\pi_{10}$  - Both are busy with a type 2 call  $\pi_{22}$

John's average earnings rate  $R_J$  is given by the sum of the probabilities of him being busy, each weighted by his hourly rate  $c_J$ . His earnings rate can be written as:

$$R_J = c_J \cdot (\pi_{10} + \pi_{22})$$

### 3 Problem 3. (25 points)

The answers in this problem have to be actual numbers. And you have to show clearly how you derive/prove your answers. Customers arrive at a bank as a Poisson process of rate 10 customers per hour. The customers are served by one teller, Chris, with exponentially distributed service time with mean equal to 1/12 hours. There is infinite waiting room, so customers are never turned away – they wait in line, if necessary

3.1. (10 points) Suppose that at the closing time the “system” is in stationary regime. (In other words, assume that at the closing time the distribution of the number of customers is the stationary distribution.) What is the probability that at the closing time there are at least 8 customers in total at the bank?

3.2. (15 points) Suppose that at the closing time there happen to be exactly 8 customers in total still at the bank. After the closing time no new customers are admitted, but those 8 remaining customers still need to be served. So, at the closing time a manager – Julia – starts working as a second teller; her customer service time is exponential with mean equal to 1/10 hours. Chris and Julia do not “pass” customers to each other and do not help each other with any given customer – when they take a customer for service they process this customer alone until service completion. What is the average time it will take to clear all 8 remaining customers (until they all are served)?

***Solution:***

This queuing system can be characterized as an  $M/M/1$  queue where the arrival rate ( $\lambda$ ) of customers is 10 per hour, and the service rate ( $\mu$ ) of customers is 12 per hour (since the mean service time is 1/12 hour per customer).

```
[ ]: # less than one so system is stable
rho = 10 / 12

# stable M/M/1 queue has a geometric distribution with probability mass
  ↪function:
#  $P(n=k) = (1-\rho)\rho^k$ 
# then use the formula for the sum of a geometric series to find
# the probability that there are at least 8 customers in the system
sum_first_7_terms = (1 - rho**8) / (1 - rho)
P_n_ge_8 = 1 - (1 - rho) * sum_first_7_terms
P_n_ge_8
```

```
[ ]: 0.23256803936137804
```

3.1: The probability that at the closing time there are at least 8 customers in total at the bank is 23.26%

```
[ ]: mu_C = 12 # Chris's service rate
mu_J = 10 # Julia's service rate

# harmonic mean of the service rates when both are active
harmonic_mean_rate = 2 * mu_C * mu_J / (mu_C + mu_J)

# Expected time to serve one customer when both tellers are active (in hours)
expected_time_one_customer = 1 / harmonic_mean_rate

# Approximate the expected time to clear all 8 customers
# Assuming both tellers can fully serve 6 customers (3 each simultaneously) at
  ↪the faster combined rate,
# then deal with the remaining 2 customers individually at an average of their
  ↪individual rates.
time_for_6_customers = 6 / (mu_C + mu_J)
time_for_last_2_customers = (
    1 / mu_C + 1 / mu_J
)

total_expected_time = time_for_6_customers + time_for_last_2_customers
print(total_expected_time, "hours")
print(total_expected_time * 60, "minutes")
```

```
0.45606060606060606 hours
27.363636363636363 minutes
```

3.2: The average time it will take to clear all 8 remaining customers is 27.36 minutes

## 4 Problem 4. (25 points)

The answers in this problem have to be actual numbers. And you have to show clearly how you derive/prove your answers. You have a call center, with 4 agents, labeled 1,2,3,4, which can answer different kind of questions. Each agent receives exogenous calls as a Poisson process of rate 2; these Poisson processes are independent. Each agent has an infinite buffer for calls waiting to be processed. The service times of the agents are exponentially distributed with parameters 3, 5, 5, and 6 for agents 1,2,3,4, respectively; all these service times are mutually independent. (So, the mean service times are 1/3, 1/5, 1/5, 1/6, for agents 1,2,3,4, respectively.) Each call processed by agent 1 is routed to agent 2 or 3 with equal probabilities 1/2. Each call processed by agent 2, either leaves the system or routed to agent 3 or routed to agent 4 with equal probabilities 1/3. Each call processed by agent 3, either leaves the system or routed to agent 2 or routed to agent 4 with equal probabilities 1/3. Each call processed by agent 4 leaves the system

- 4.1. (5 points) Model this system as continuous time Markov chain, and determine whether or not it has a stationary distribution. If it does have a stationary distribution, what is it?
- 4.2. (5 points) What is the long-run fraction of time when agents 1 and 4 are idle simultaneously?
- 4.3. (5 points) What is the long-run fraction of time when there are at least 2 calls in total at agents 2 and 3? (A call is 'at agent i' when it is either being served by agent i or is in the queue waiting to be served by agent i.)
- 4.4. (10 points) Among those calls which will receive any service by agents 2 or 3, what is the expected total time they spend at agents 2 and 3?

### **Solution:**

CMTC Model: Let  $X(t) = (X_1(t), X_2(t), X_3(t), X_4(t))$  where  $X_i(t)$  denotes the number of calls at agent i at time t.

Arrival process: Each agent receives calls as an independent Poisson process with rate 2. Define with  $\lambda_{1,2,3,4} = 2$  calls per time unit

Rates:  $\mu_1 = 3, \mu_2 = 5, \mu_3 = 5, \mu_4 = 6$

Transition probabilities: - From Agent 1:  $P_{12} = 1/2, P_{13} = 1/2$  - From Agent 2:  $P_{23} = 1/3, P_{24} = 1/3, P_{20} = 1/3$  - From Agent 3:  $P_{32} = 1/3, P_{34} = 1/3, P_{30} = 1/3$  - From Agent 4: All calls leave the system.

Transition Matrix: Define the rate matrix  $Q$  for the CTMC where each element  $q_{ij}$  represents the transition rate from state  $i$  to state  $j$ . State  $i$  would represent a vector indicating the number of calls at each agent. We use transition probabilities to derive transition rates for calls transferring between agents or leaving the system.

Stationary Distribution: System's stationary distribution  $\pi$  satisfies  $\pi Q = 0$  and  $\sum \pi_i = 1$ . For each agent i, equilibrium equation is:

$$\lambda_i \pi(\text{prev state}) + \mu_{j \rightarrow i} \pi(\text{from } j) = \mu_i \pi(\text{curr state}) + \lambda_i \pi(\text{next state})$$

Stability: Need to check if system is stable and therefore a systems stationary distribution. the traffic intensity  $\rho_i$  at each agent must be less than 1, where  $\rho_i = \frac{\lambda_i^{\text{eff}}}{\mu_i} \cdot \lambda_i^{\text{eff}}$  is the effective arrival rate considering incoming routed calls. Then we Compute each  $\rho_i$  and check if it's less than 1.

$$\lambda_1^{\text{eff}} = \lambda_1 = 2$$

$$\lambda_2^{\text{eff}} = \lambda_2 + \frac{1}{2}\mu_1 = 2 + \frac{1}{2} \times 3 = 3.5$$

$$\lambda_3^{\text{eff}} = \lambda_3 + \frac{1}{2}\mu_1 + \frac{1}{3}\mu_2 = 2 + \frac{1}{2} \times 3 + \frac{1}{3} \times 5 = 5.166$$

$$\lambda_4^{\text{eff}} = \frac{1}{3}\mu_2 + \frac{1}{3}\mu_3 = \frac{1}{3} \times 5 + \frac{1}{3} \times 5 = 3.332$$

$$\rho_1 = \lambda_1^{\text{eff}} / \mu_1$$

$$\rho_2 = \lambda_2^{\text{eff}} / \mu_2$$

$$\rho_3 = \lambda_3^{\text{eff}} / \mu_3$$

$$\rho_4 = \lambda_4^{\text{eff}} / \mu_4$$

```
[ ]: mu = [3, 5, 5, 6]
lambda_external = [2, 2, 2, 2]

# Effective arrival rate calculations
lambda_eff = {
    "Agent 1": lambda_external[0], # Agent 1 only receives external calls
    "Agent 2": lambda_external[1] + 0.5 * mu[0], # Agent 2 receives calls from
    ↪ external and half from Agent 1
    "Agent 3": lambda_external[2] + 0.5 * mu[0] + 1/3 * mu[1], # Agent 3
    ↪ receives calls from external, half from Agent 1, and a third from Agent 2
    "Agent 4": 1/3 * mu[1] + 1/3 * mu[2] # Agent 4 receives a third of calls
    ↪ processed from Agents 2 and 3
}

rho = {f"Agent {i+1}": lambda_eff[f"Agent {i+1}"] / mu[i] for i in range(4)}
rho
```

```
[ ]: {'Agent 1': 0.6666666666666666,
      'Agent 2': 0.7,
      'Agent 3': 1.0333333333333332,
      'Agent 4': 0.5555555555555555}
```

Agents 1, 2, and 4 are stable, as their traffic intensities are less than 1, indicating that under typical conditions, queues are unlikely to build indefinitely. Agent 3 has a traffic intensity greater than 1, indicating instability. This suggests that queues could grow indefinitely at Agent 3, which could potentially affect the overall stability of the system, despite the stability of the other agents.

```
[ ]: mu_1 = 3
lambda_1 = 2
mu_4 = 6
lambda_4_eff = lambda_eff["Agent 4"]

P_idle_1 = 1 - (lambda_1 / mu_1)
P_idle_4 = 1 - (lambda_4_eff / mu_4)

# agents operate independently, the joint probability of both being idle
P_idle_1_and_4 = P_idle_1 * P_idle_4
```

```
P_idle_1, P_idle_4, P_idle_1_and_4
```

```
[ ]: (0.3333333333333337, 0.44444444444444453, 0.1481481481481482)
```

4.2:

Probability that Agent 1 is idle:  $P(\text{Idle } 1) \approx 0.333$

Probability that Agent 4 is idle:  $P(\text{Idle } 4) \approx 0.445$

Probability that Agents 1 and 4 are idle simultaneously:  $P(\text{Idle } 1 \text{ and } 4) \approx 0.15$

The long-run fraction of time when agents 1 and 4 are idle simultaneously is around 15.0%.

```
[ ]: def simulate_queue(total_time):
    lambda_ = 2
    # service rates for agents 2 and 3
    mu = [5, 5]
    # keep track of queue lengths for agents 2 and 3
    queue = [0, 0]
    # Event queue
    events = []
    # time counter for at least two calls in queue
    time_at_least_two = 0

    for i in range(2):
        next_arrival = np.random.exponential(1 / lambda_)
        events.append((next_arrival, 'arrival', i)) # indexing from 0 for
        ↪ agents 2 and 3

    current_time = 0
    while current_time < total_time:
        events.sort()
        time, event_type, agent_idx = events.pop(0)
        time_elapsed = time - current_time
        current_time = time

        if sum(queue) >= 2:
            time_at_least_two += time_elapsed
        if event_type == 'arrival':
            queue[agent_idx] += 1
            events.append((current_time + np.random.exponential(1 / lambda_),
            ↪ 'arrival', agent_idx))
            events.append((current_time + np.random.exponential(1 /
            ↪ mu[agent_idx]), 'departure', agent_idx))
        elif event_type == 'departure':
            queue[agent_idx] -= 1

    return time_at_least_two / total_time
```



```
[ ]: display(f"simulation with a small number: {simulate_queue(1000)}")
display(f"simulation with a medium number: {simulate_queue(10000)}")
display(f"simulation with a big number: {simulate_queue(1000000)}")
```

'simulation with a small number: 0.18705072459654054'

'simulation with a medium number: 0.1898609199761089'

'simulation with a big number: 0.1912111765108328'

4.3: The long-run fraction of time when there are at least 2 calls in total at agents 2 and 3 is  $\approx 0.19$

```
[ ]: mu_2, mu_3 = mu[1], mu[2]
lambda_2_eff, lambda_3_eff = lambda_eff["Agent 2"], lambda_eff["Agent 3"]

if mu_2 > lambda_2_eff:
    T_2 = 1 / (mu_2 - lambda_2_eff)
else:
    # Indicates instability
    T_2 = float('inf')

if mu_3 > lambda_3_eff:
    T_3 = 1 / (mu_3 - lambda_3_eff)
else:
    # Indicates instability
    T_3 = float('inf')

total_expected_time = T_2 + T_3
total_expected_time
```

```
[ ]: inf
```

4.4: Since Agent 3 is unstable and therefore the system is unable, the expected total time they spend at agents 2 and 3 is infinite