

IE 370, Spring 2024 Homework 7

Chris Lee

April 9, 2024

0.1 Problem 1. (20 points)

Consider a call center with three identical agents (servers) and four phone lines. Call arrivals follow a Poisson process with rate 2 per minute. An arriving call that finds all lines busy is lost (blocked). Call processing times are i.i.d., exponentially distributed with mean 1 minute. An admitted call that finds all agents busy will wait until getting service, occupying a phone line. Model this system as a CTMC. Find the long-term throughput of the system (the rate of served customers leaving the system). Find the average waiting time (in queue) among those who are served. Find the average number of calls in the entire system.

Solution:

- 3 identical agents (servers)
- 4 phone lines
- Call arrival rate: $\lambda = 2$ calls per minute
- Service rate per server: $\mu = 1$ call per minute

$X(t)$ is a CTMC with state space $\{0, 1, 2, 3, 4\}$ that models the number of calls in the system at time t . The call center can be modeled as a $M/M/3/4$ system. We first solve the system of equations for the stationary distribution values $\{\pi_i\}$. Then, we calculate the long-term throughput γ which is the dot product of the stationary distribution values and the corresponding service rates. This tells us the effective average arrival rate λ_{ef} . Then, using Little's Law, we can find the average waiting time W_q . The average number of calls in the entire system L is just the summation of state * probability.

```
[ ]: from sympy import symbols, Eq, solve

pi0, pi1, pi2, pi3, pi4, lam, mu = symbols('pi0 pi1 pi2 pi3 pi4 lam mu')

lam_value = 2
mu_value = 1

eq1 = Eq(lam * pi0, mu * pi1)
eq2 = Eq(lam * pi1, 2 * mu * pi2)
eq3 = Eq(lam * pi2, 3 * mu * pi3)
eq4 = Eq(lam * pi3, 3 * mu * pi4)
eq5 = Eq(pi0 + pi1 + pi2 + pi3 + pi4, 1)

solution = solve((eq1, eq2, eq3, eq4, eq5), (pi0, pi1, pi2, pi3, pi4))
```

```
solution = {k: v.subs({lam: lam_value, mu: mu_value}) for k, v in solution.
            ↪items()}
solution
```

```
[ ]: {pi0: 9/65, pi1: 18/65, pi2: 18/65, pi3: 12/65, pi4: 8/65}
```

```
[ ]: pi_values = list(solution.values())
gamma = sum(pi_values[i] * min(i, 3) * mu_value for i in range(1, 5))
lambda_eff = gamma
Lq = pi_values[4] # Only calls in state 4 are waiting
Wq = Lq / lambda_eff
L = sum(i * pi_values[i] for i in range(5))

gamma, Wq, L
```

```
[ ]: (114/65, 4/57, 122/65)
```

- Long-term throughput of the system (the rate of served customers leaving the system): $114/65$
- Average waiting time (in queue) among those who are served: $4/57$
- Average number of calls in the entire system: $122/65$

0.2 Problem 2. (30 points)

Consider a service system in which customers arrive as a Poisson process of rate 3 (per unit time). There are two servers in the system, labeled 1 and 2. For both, the service time has the same exponential distribution with mean $1/2$ (time units). All service times are i.i.d. When both servers are busy, an arriving customer waits in a queue, whose capacity is assumed to be infinite. The queue is served in the first-in-first-out order. The service policy is such that server 1 is the "main" one, and server 2 is a "helper." This means that: (a) if a customer arrives in the empty system, it goes to server 1, not 2; (b) if a customer arrives in the system when server 1 is working and server 2 is idle, this customer goes to server 2; (c) when any server completes service, it will take for service the next customer waiting in queue if any; (d) if server 1 completes a service when there is nobody waiting in the queue, but server 2 is working on a customer, then server 1 "takes" this customer from server 2, and serves it to completion as if it were a new customer. In particular, these rules imply that if there is exactly one customer in the system, it is served by server 1, not 2; if there are two or more customers in the system, both servers are working.

2.1 (10 points) Model this system as a CTMC. Find its stationary distribution.

Solution:

$X(t)$ is a CTMC with state space $\{0, 1, 2, 3, \dots\}$ that models the total number of customers in the system at time t . Has - Arrival rate: $\lambda = 3$ - Service rate: $\mu = 2$

We know that stationary distribution of this process exist b/c $\rho = \lambda/2\mu = 3/4 < 1$

```
[ ]: lambda_arrival = 3
mu_service = 2
k_servers = 2
```

```
rho = lambda_arrival / (2 * mu_service)

pi_0 = (1 - rho) / (1 + rho)
pi_1 = 2 * rho * pi_0
pi_2 = 2 * (rho) ** 2 * pi_0

rho, pi_0, pi_1, pi_2
```

```
[ ]: (0.75, 0.14285714285714285, 0.21428571428571427, 0.1607142857142857)
```

- $\rho = 0.75$
- $\pi_0 = 0.14285714285714285$, (probability of the system being empty)
- $\pi_1 = 0.21428571428571427$, (probability of there being exactly one customer in the system, served by server 1)
- $\pi_2 = 0.1607142857142857$, (probability of there being exactly two customers in the system, one with each server)

2.2 (20 points) In the long run: What is the fraction of all customers that will leave the system from server 1? What is the fraction of all customers that will leave the system from server 2? What is the fraction of all customers that will start service at server 2, but will leave the system from server 1?

Solution:

```
[ ]: # frac of time the server 1 is working: 1 - pi_0
fraction_leaving_server_1 = mu_service * (1 - pi_0) / lambda_arrival

# frac of time the server 1 is working: 1 - pi_0 - pi_1, can also easily see
↳ that this is 1 - fraction_leaving_server_1
fraction_leaving_server_2 = mu_service * (1 - pi_0 - pi_1) / lambda_arrival

# this happens only when a customer goes from 2 to 1
fraction_moved_server_1 = mu_service * pi_2 / lambda_arrival

fraction_leaving_server_1, fraction_leaving_server_2, fraction_moved_server_1
```

```
[ ]: (0.5714285714285715, 0.4285714285714286, 0.10714285714285714)
```

- Fraction of all customers that will leave the system from server 1: 0.5714285714285715
- Fraction of all customers that will leave the system from server 2: 0.4285714285714286
- Fraction of all customers that will start service at server 2, but will leave the system from server 1: 0.10714285714285714

0.3 Problem 3. (20 points)

You are running a small insurance agency. Customer calls arrive as a Poisson process of rate 2 calls/hour. (Time unit is 1 hour.) You need to decide whether to employ one or two agents

answering the calls. A call processing time by an agent has exponential distribution with parameter 2 (so that the mean call processing time is $1/2$ hours); call processing times are independent. Each employed agent costs \$25/hour. If a customer call arrives when there is an agent to take it, the call is processed, and the agency receives \$40 commission (revenue). If a call arrives when there is no available agent to take it (all agents are busy), the call is lost. What is more profitable for the agency (in the sense of long-term average profit), to employ one or two agents answering the calls?

Solution:

We will compare the long term average profit from employing one agent, which can be modeled as a M/M/1/1 system and the long term average profit from employing two agents which can be modeled as a M/M/2/2 system.

```
[ ]: lambda_rate = 2
mu_rate = 2
agent_cost_per_hour = 25
commission_per_call = 40

pi1_mm11 = lambda_rate / mu_rate / (1 + lambda_rate / mu_rate)
profit_mm11 = lambda_rate * (1 - pi1_mm11) * commission_per_call - 1 *
    ↪ agent_cost_per_hour

pi2_mm22 = (lambda_rate / (2 * mu_rate)) / (1 + lambda_rate / mu_rate +
    ↪ (lambda_rate / (2 * mu_rate)))
profit_mm22 = lambda_rate * (1 - pi2_mm22) * commission_per_call - 2 *
    ↪ agent_cost_per_hour

profit_mm11, profit_mm22
```

```
[ ]: (15.0, 14.0)
```

- **Employ one agent, you make bigger profit**

1 Problem 4. (30 points)

A DMV office has three people serving customers. Alice (A) is at the front desk, Bob (B) is making photos, Carol (C) takes payments and issues driver licenses. Customers arrive a Poisson process with rate 2 per minute. Each customer first goes to A. After being served by A, each customer (independently of each other, and of anything else) goes to B with probability 0.4 or to C with probability 0.6. (The latter means that a new photo is not needed.) After being served by B, a customer goes to C. After being served by C, a customer leaves the DMV office. Service times by A are i.i.d. exponential with mean $1/4$ min. Service times by B are i.i.d. exponential with mean 1 min. Service times by C are i.i.d. exponential with mean $1/3$ min.

4.1 (6 points) Model this system as a CTMC. 4.2 (6 points) Does it have a stationary distribution? If so, what is it? 4.3 (6 points) What is the long-run fraction of time when A is working while both B and C are idle? 4.4 (6 points) What is the long-run average time a customer spends at DMV? 4.5 (6 points) What is the long-run average time a customer spends at B (waiting or being processed), among those customer that do require service by B?

Solution:

```
[ ]: def calculate_stationary_distribution(A, B, C):
    = lambda i, j, m: (1 - A) * A**i * (1 - B) * B**j * (1 - C) * C**m
    return

def long_run_fraction_A_working_B_C_idle(A, B, C):
    return A * (1 - B) * (1 - C)

def long_run_average_time_DMV(A, B, C, ):
    W = 1 / (A * (1 - A) + B * (1 - B) + C * (1 - C))
    return W

def long_run_average_time_B(B, B):
    WB = 1 / (B * B * (1 - B))
    return WB
```

```
[ ]: = 2
μA, μB, μC = 4, 1, 3
A =
B = 0.4 * A
C = 0.6 * A + B
A, B, C = A / μA, B / μB, C / μC

= calculate_stationary_distribution(A, B, C)

# Long-run fraction of time A working, B and C idle
fraction_A_working = long_run_fraction_A_working_B_C_idle(A, B, C)

# Long-run average time a customer spends at DMV
average_time_DMV = long_run_average_time_DMV(A, B, C, )

# Long-run average time a customer spends at B among those requiring B's service
average_time_B = long_run_average_time_B(B, B)

fraction_A_working, average_time_DMV, average_time_B
```

```
[ ]: (0.03333333333333333, 3.5000000000000004, 5.000000000000001)
```

4.1 (CTMC Model): - The process is $(X(t) = (X_A(t), X_B(t), X_C(t)), t \geq 0)$, where $(X_k(t))$ is the total number of customers at server $(k = A, B, C)$. This is a CTMC since the arrival process is Poisson, and the service times are exponential (from problem statement)

4.2 (Stationary Distribution): - The system can be modeled as a Jackson network, which has a stationary distribution if the load $(\rho_k = \frac{\lambda_k}{\mu_k} < 1)$ for each server $(k = A, B, C)$. Given $(\lambda_A = 2, \lambda_B = 0.8, \lambda_C = 2)$, and $(\mu_A = 4, \mu_B = 1, \mu_C = 3)$, we find $(\rho_A = \frac{1}{2}, \rho_B = \frac{4}{5}, \rho_C = \frac{2}{3})$. The

stationary distribution exists and is unique: $(\pi_{i,j,m} = (1 - \rho_A)\rho_A^i(1 - \rho_B)\rho_B^j(1 - \rho_C)\rho_C^m)$.

4.3 (Long-run Fraction of Time): - The long-run fraction of time when A is working while both B and C are idle is $(\rho_A(1 - \rho_B)(1 - \rho_C) = \frac{1}{30})$.

4.4 (Long-run Average Time at DMV): - We can use Little's Law, so the long-run average time a customer spends at DMV is $W = \frac{7}{2} = 3.5$ minutes.

4.5 (Long-run Average Time at B): - The long-run average time spent at B, including waiting and processing, is $W_B = 5$ minutes.