SYSEN 5200: Systems Analysis Behavior and Optimization

Queueing Theory

Nick Kunz [NetID: nhk37] nhk37@cornell.edu

March 16, 2023

1. Consider a two-server queueing system, where the customers arriving into the system join a single queue, wait for their turn, receive service from either one of the two servers and leave the system. The first server is faster than the second one. If an arriving customer finds both servers available, then it uses the first server. Otherwise, it simply uses whichever server is available. We are interested in simulating the behavior of this system with the intention of estimating:

   - the proportion of time that the first server is busy
   - the proportion of time both servers are busy
   - the proportion of time one or more customers in the queue over the first 100 minutes.

   The interarrival times for the customers are exponentially distributed with mean 0.5 minutes. The service times at the first server are exponentially distributed with mean 0.8 minutes and the service times at the second server are exponentially distributed with mean 0.9 minutes. We assume that the queue can accommodate infinite number of customers and the system starts empty.

   (a) The state of the system can be described as:

   $$\begin{aligned} n &= \text{Number of customers queueing for service.} \\ s_1 &= \text{Number of customers being served by the first server.} \\ s_2 &= \text{Number of customers being served by the second server.} \end{aligned} \tag{1}$$

   The possible events that change the state of the system can be described as:

   $$\begin{aligned} A &= \text{Arrival of a new customer.} \\ D_1 &= \text{Departure of customer upon first server } s_1 \text{ completion.} \\ D_2 &= \text{Departure of customer upon second server } s_2 \text{ completion.} \end{aligned} \tag{2}$$

   (b) The pseudo-code that describes how different events change the state of the system and what other events are scheduled in response can be exhibited as:

Algorithm 1 Event A

$t = 0$
while $t < T$ **do**
  generate inter-arrival time $t + t_A$ $Expo(0.5)$ and schedule event A
  if $s_1 = 0$ and $s_2 = 0$ **then**
    $s_1 = s_1 + 1$
    generate service time $t + t_{D1}$ $Expo(0.8)$ and schedule event D_1
  end if
  if $s_1 > 0$ and $s_2 = 0$ **then**
    $s_2 = s_2 + 1$
    generate service time $t + t_{D2}$ $Expo(0.9)$ and schedule event D_2
  end if
  if $s_1 = 0$ and $s_2 > 0$ **then**
    $s_1 = s_1 + 1$
    generate service time $t + t_{D1}$ $Expo(0.8)$ and schedule event D_1
  end if
  if $s_1 > 0$ and $s_2 > 0$ **then**
    $n = n + 1$
  end if
  $t = t + 1$
end while

Algorithm 2 Event D_1

$s_1 = s_1 - 1$
if $n > 0$ **then**
  $n = n - 1, s_1 = s_1 + 1$
  generate service time t_{D1} $Expo(0.8)$ and schedule event D_1
end if

Algorithm 3 Event D_2

$s_2 = s_2 - 1$
if $n > 0$ and $s_1 > 0$ **then**
  $n = n - 1, s_2 = s_2 + 1$
  generate service time t_{D2} $Expo(0.9)$ and schedule event D_2
end if

(c) Using a worksheet to manually simulate the system over 10 minutes, we have the interarrival times: 1.3, 1.1, 0.6, 1.5, 2.2, 1.1, 0.5, 0.7, 0.1, 1.4, 1.2, 1.4, 1.4, 1.2, 2.1. For the service times at the first server, we have: 2.1, 1.3, 0.9, 1.4, 1.2, 2.1, 2.2. For the service times at the second server, we have: 2.2, 1.6, 1.1, 1.7, 2.1, 1.3, 0.9.

Time	Event	State	Future Event List	$t : s_1$	$t : s_2$	$t : s$
0	Init	$n = 0, s_1 = 0, s_2 = 0$	$(A, 1.3)$	0	0	0
1.3	A	$n = 1, s_1 = 1, s_2 = 0$	$(A, 2.4), (D_1, 3.4)$	0	0	0
2.4	A	$n = 2, s_1 = 1, s_2 = 1$	$(A, 3.0), (D_1, 3.4), (D_2, 4.6)$	1.1	0	0
3.0	A	$n = 3, s_1 = 1, s_2 = 1$	$(A, 4.5), (D_1, 3.4), (D_2, 4.6)$	1.7	0.6	0
3.4	D_1	$n = 2, s_1 = 1, s_2 = 1$	$(A, 4.5), (D_1, 4.7), (D_2, 4.6)$	2.1	1.0	0.4
4.5	A	$n = 3, s_1 = 1, s_2 = 1$	$(A, 6.7), (D_1, 4.7), (D_2, 4.6)$	3.2	2.1	0.4
4.6	D_2	$n = 2, s_1 = 1, s_2 = 1$	$(A, 6.7), (D_1, 4.7), (D_2, 6.2)$	3.3	2.2	0.5
4.7	D_1	$n = 1, s_1 = 0, s_2 = 1$	$(A, 6.7), (D_2, 6.2)$	3.4	2.3	0.5
6.2	D_2	$n = 0, s_1 = 0, s_2 = 0$	$(A, 6.7)$	3.4	2.3	0.5
6.7	A	$n = 1, s_1 = 1, s_2 = 0$	$(A, 7.8), (D_1, 7.6)$	3.4	2.3	0.5
7.6	D_1	$n = 0, s_1 = 0, s_2 = 0$	$(A, 7.8)$	4.3	2.3	0.5
7.8	A	$n = 1, s_1 = 1, s_2 = 0$	$(A, 8.3), (D_1, 9.2)$	4.3	2.3	0.5
8.3	A	$n = 2, s_1 = 1, s_2 = 1$	$(A, 9.0), (D_1, 9.2), (D_2, 9.4)$	4.8	2.3	0.5
9.0	A	$n = 3, s_1 = 1, s_2 = 1$	$(A, 9.1), (D_1, 9.2), (D_2, 9.4)$	5.5	3.0	0.5
9.1	A	$n = 4, s_1 = 1, s_2 = 1$	$(A, 10.5), (D_1, 9.2), (D_2, 9.4)$	5.6	3.1	0.6
9.2	D_1	$n = 3, s_1 = 1, s_2 = 1$	$(A, 10.5), (D_1, 10.4), (D_2, 9.4)$	5.7	3.2	0.7
9.4	D_2	$n = 2, s_1 = 1, s_2 = 1$	$(A, 10.5), (D_1, 10.4), (D_2, 11.1)$	5.9	3.4	0.9
10.0	End	-	-	6.5	4.0	0.9

The proportion of time that the first server is busy is $6.5/10 = 0.65$. The proportion of time both servers are busy is $4.0/10 = 0.40$. The proportion of time that there are one or more customers in the queue over the first 100 minutes is $9.0/10 = 0.90$.

(d) Exhibited here is a program that simulates the two-server queueing system with the given probability distributions.

```python
## library
import numpy as np

## discrete event simulation
def des_sim(a = 0.5, d_1 = 0.8, d_2 = 0.9, end = 100):

    """
    Desc:
        Priority Queueing Theory.

    Args:
        - a: Mean of the interarrival time.
        - d_1: Mean of the service times at first server.
        - d_2: Mean of the service times at second server.
        - end: Max simulation time (mins).

    Return:
        - p_first_busy: Proportion of time that the first server is busy.
        - p_both_busy: Proportion of time both servers are busy.
        - p_queue: Proportion of time one or more customers in the queue.
    """

    ## time
    T = 0
    t = 0  # sim time (mins)
```

```python

    ## state vars
    fel_time = [t + np.random.exponential(a), end]
    fel_type = ['A','E']
    s_1_busy = False
    s_2_busy = False

    ## metrics
    num_arr = 0
    tot_time_s = 0
    tot_time_q = 0
    tot_time_b = 0

    eve_type = ''
    while(eve_type != 'E'):
        eve_next = np.argmin(fel_time)
        eve_time = fel_time.pop(eve_next)
        eve_type = fel_type.pop(eve_next)
        tot_time_s += T*(eve_time-t)

        if T > 0:
            tot_time_q += (T-1)*(eve_time-t)

        if s_1_busy and s_2_busy:
            tot_time_b += (eve_time-t)

        t = eve_time

        ## arrivals
        if eve_type == 'A':
            T += 1
            num_arr += 1
            t_A = np.random.exponential(a)
            fel_time.append(t+t_A)
            fel_type.append('A')

            if not s_1_busy:
                t_S = np.random.exponential(d_1)
                fel_time.append(t+t_S)
                fel_type.append('d_1')
                s_1_busy = True

            elif not s_2_busy:
                t_S = np.random.exponential(d_2)
                fel_time.append(t+t_S)
                fel_type.append('d_2')
                s_2_busy = True

        ## departs from first server
        elif eve_type == 'd_1':
            s_1_busy = False

            if T >= 1:
                t_S = np.random.exponential(d_1)
                fel_time.append(t+t_S)
                fel_type.append('d_1')
                s_1_busy = True

            else:
                s_2_busy = False

        ## departs from second server
        elif eve_type == 'd_2':
            s_2_busy = False
```

```python

            if T >= 1:
                t_S = np.random.exponential(d_2)
                fel_time.append(t+t_S)
                fel_type.append('d_2')
                s_2_busy = True

            else:
                s_1_busy = False

    ## compute results
    p_first_busy = (d_1*tot_time_s)/t
    p_both_busy = (100*tot_time_b)/t
    p_queue = tot_time_q/t

    return p_first_busy, p_both_busy, p_queue
```

Fig. Python 1(d)

Proportion of time that the first server is busy: 72%
Proportion of time that both servers are busy: 99%
Proportion of time one or more customers: 92%

2. Six dump trucks are used to haul coal from the entrance of a small mine to the railroad. Each truck is loaded by one of the two loaders. After loading, a truck immediately moves to the scale to be weighed. Both the loaders and the scale have a first-come-first-serve queue for the trucks. The trucks are weighed one by one. Travel time from the loader to the scale is negligible. After being weighed, a truck begins a travel time (during which it unloads) and then afterwards returns to the loader queue. The purpose of the simulation is to compute:

   - the average number of loaders that are busy
   - the proportion of time that the scale is busy over the first 75 minutes.

   All queue capacities are larger than six. At the beginning of the simulation, five of the trucks are at the loaders and one of the trucks is at the scale.

   (a) The state of the system can be described as:

   $$\begin{aligned} l_x &= \text{Number of trucks being loaded by either loader.} \\ s &= \text{Number of trucks being weighed by the scale.} \end{aligned} \tag{3}$$

   The possible events that change the state of the system can be described as:

   $$\begin{aligned} A_l &= \text{Arrival of a truck at either loader } l_x. \\ D_l &= \text{Departure of a truck upon either loaders } l_x \text{ completion.} \\ D_s &= \text{Departure of a truck upon scale } s \text{ completion.} \end{aligned} \tag{4}$$

   (b) The pseudo-code that describes how different events change the state of the system and what other events are scheduled in response can be exhibited as:

Algorithm 4 Event A_l

$t = 0$
while $t < T$ **do**
  generate inter-arrival time $t + t_{Al}$ and schedule event A_l
  if $l_1 = 0$ and $l_2 = 0$ **then**
    $l_x = Rand(l_1, l_2)$
    $l_x = l_x + 1$
    generate service time $t + t_{Dl}$ and schedule event D_l
  end if
  if $l_1 > 0$ **then**
    $l_2 = l_2 + 1$
    generate service time $t + t_{Dl}$ and schedule event D_l
  end if
  if $l_2 > 0$ **then**
    $l_1 = l_1 + 1$
    generate service time $t + t_{D1}$ and schedule event D_l
  end if
  if $l_1 > 0$ and $l_2 > 0$ **then**
    $n_l = n_l + 1$
  end if
  $t = t + 1$
end while

Algorithm 5 Event D_l

$l_x = l_x - 1$
$s = s + 1$
if $n_l > 0$ **then**
  $n_l = n_l - 1, l_x = l_x + 1$
  generate service time t_{D1} and schedule event D_1
end if
if $s > 0$ **then**
  $n_l = n_l - 1, l_x = l_x + 1$
  generate service time t_{D1} and schedule event D_1
end if

Algorithm 6 Event A_s

$t = 0$
while $t < T$ **do**
  generate inter-arrival time $t + t_{As}$ and schedule event A_s
  if $s = 0$ **then**
    $s = s + 1$
    generate service time $t + t_{Ds}$ and schedule event D_s
  end if
  if $s > 0$ **then**
    $n_s = n_s + 1$
  end if
  $t = t + 1$
end while

Algorithm 7 Event D_s

$s = s - 1$
if $n_s > 0$ **then**
  $n_s = n_s - 1, s = s + 1$
  generate service time t_{Ds} and schedule event D_s
end if

(c) Using a worksheet to manually simulate the system over 75 minutes, we have the loading times: 10, 5, 6, 10, 15, 10, 10, 15, 10, 5, 5. For the weighing times, we have: 12, 12, 12, 16, 12, 16, 9, 11. For the travel times, we have: 21, 21, 23, 24, 40, 40.

Time	Event	State	Future Event List	$t : l_s$	$t : s$
0	Init	$l_x = 5, s = 1$	$(D_l, 10), (D_l, 5), (D_s, 12)$	0	0
5	D_l	$l_x = 4, s = 2$	$(D_l, 10), (D_l, 11), (D_s, 12)$	10	5
10	D_l	$l_x = 3, s = 3$	$(D_l, 20), (D_l, 11), (D_s, 12)$	20	10
11	D_l	$l_x = 2, s = 4$	$(D_l, 20), (D_l, 26), (D_s, 12)$	22	11
12	D_s	$l_x = 2, s = 3$	$(D_l, 20), (D_l, 26), (D_s, 24), (A_l, 33)$	24	12
20	D_l	$l_x = 1, s = 4$	$(D_l, 26), (D_s, 24), (A_l, 33)$	40	20
24	D_s	$l_x = 1, s = 3$	$(D_l, 26), (D_s, 36), (A_l, 33), (A_l, 45)$	44	24
26	D_l	$l_x = 0, s = 4$	$(D_s, 36), (A_l, 33), (A_l, 45)$	46	26
33	A_l	$l_x = 1, s = 4$	$(D_s, 36), (D_l, 43), (A_l, 45)$	46	33
36	D_s	$l_x = 1, s = 3$	$(D_s, 52), (D_l, 43), (A_l, 45), (A_l, 59)$	49	36
43	D_l	$l_x = 0, s = 4$	$(D_s, 52), (A_l, 45), (A_l, 59)$	56	43
45	A_l	$l_x = 1, s = 4$	$(D_s, 52), (D_l, 55), (A_l, 59)$	56	45
52	D_s	$l_x = 1, s = 3$	$(D_s, 64), (D_l, 55), (A_l, 59), (A_l, 76)$	63	52
55	D_l	$l_x = 0, s = 4$	$(D_s, 64), (A_l, 59), (A_l, 76)$	66	55
59	A_l	$l_x = 1, s = 4$	$(D_s, 64), (D_l, 74), (A_l, 76)$	66	59
64	D_s	$l_x = 1, s = 3$	$(D_s, 80), (D_l, 74), (A_l, 76)$	71	64
74	D_l	$l_x = 0, s = 4$	$(D_s, 80), (A_l, 76)$	81	74
75	End	-	-	81	75

The average number of loaders that are busy is $81/75 = 1.08$. The proportion of time that the scale is busy over the first 75 minutes is $75/75 = 1$.