Dublin City University
Ollscoil Chathair Bhaile Átha Cliath

DCU

Declaration

Name: Ryan McDyer	Student ID Number: 13431038
Programme: CASE 4	
Module Code: CA4011	
Assignment Title: Part A (Simulation)	
Submission Date: 19 March 2017	

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying is a grave and serious offence in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion, or copying. I have read and understood the Assignment Regulations set out in the module documentation. I have identified and included the source of all facts, ideas, opinions, viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study
I have read and understood the referencing guidelines found at
http://www.library.dcu.ie/citing&refguide08.pdf

and/or recommended in the assignment guidelines.

Name: Ryan McDyer
Date: 19 March 2017

If submitting on-line, please tick the following box to indicate that you have read and that you agree with the statements above ☑

My system

My system works as follows:

- Prompt the user to enter the input params from the standard input
- Populate the list of Clients with their arrival times and service times
    - The arrival time distribution will depend on whether the user chooses an Exponential Distribution or a Poisson Distribution.
- Populate the list of Servers (with their break times if applicable)
- Run x simulations of the model, incrementing time by 1 minute per iteration
    - On every iteration, check a few things
        - Is there a new customer due in the queue?
        - Is someone finished with their Client?
        - Is someone due to start or finish lunch?
        - Is a free Server who can take someone in the queue
- Then print out some data to a txt file.

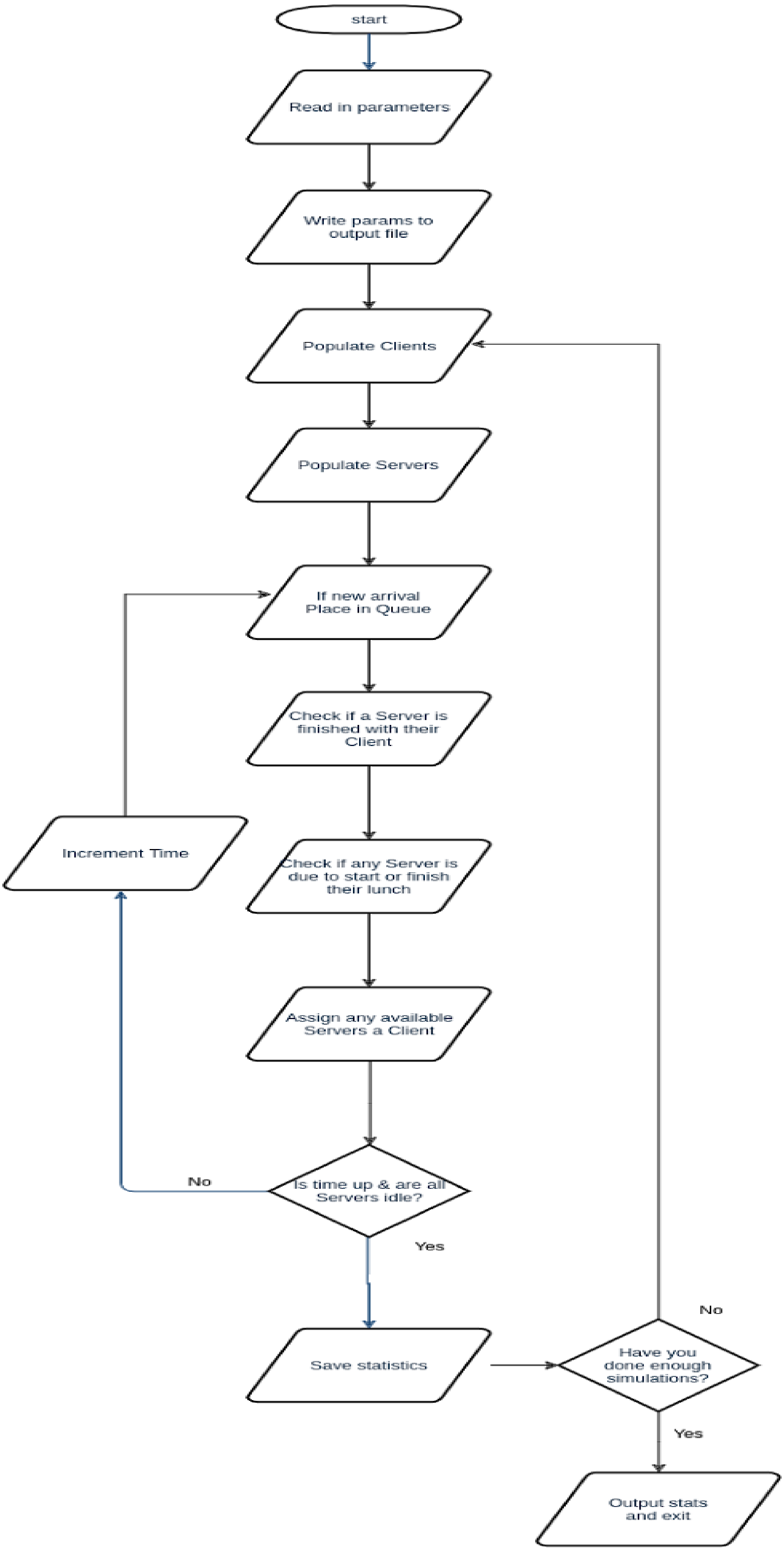
There is a diagram outlining my system on the next page.

The algorithm I have used to calculate the exponential distributions has been taken from an online source that is cited in the method body.

I have attached in a zip file the outputs of my simulations. The file naming format is:

taskNum-typeOfDistribution-lengthOfSim-lamdba-mu-numServers-anythingExtra.txt

My system is written in Java and requires Java 8 (or above) to run. The times in my system are zero-indexed. As a result, 9:00 AM is represented as 0 in my system, and 5:30 PM is represented as 510 (ie: 510 minutes after 9:00 AM).


start
Read in parameters
Write params to output file
Populate Clients
Populate Servers
If new arrival Place in Queue
Check if a Server is finished with their Client
Check if any Server is due to start or finish their lunch
Assign any available Servers a Client
Is time up & are all Servers idle?
No
Yes
Increment Time
Save statistics
Have you done enough simulations?
No
Yes
Output stats and exit

Results

Task 1: Compare random and scheduled patterns of customer arrival

I ran the simulation 250 times with params:

- 510 minutes
- Arrival rate: 6 per hour
- Service rate: 4 per hour, exponentially distributed
- 2 Servers

This yields a traffic intensity of ¾.

The first time I ran the simulation the arrival times were distributed exponentially. The second time, they were distributed with Poisson.

I found some interesting differences between the 2 distributions. The maximum average waiting time (ie: the average waiting time over 1 day) for Clients in the Exponentially distributed system was 0.06 minutes (mean 0.0059, median 0), but in the Poisson distributed system the maximum average waiting time was 0.5102 (mean 0.046, median 0.02).
All of these values are larger in the poisson distributed system. Indeed, the maximum and mean are nearly 8.5 times and 8 times larger respectively than the results in the exponentially distributed system.
It is my opinion that this occurred because in the poisson distributed system, it is more likely that a number of consecutive Clients will have their arrival times distributed with a very small gap between them. Indeed, the longest individual waiting time for a Client in the exponential system was 3 minutes, and in the Poisson system it was 9 minutes.

It is reasonable to assume that because the waiting times for Poisson are longer than those in Exponential, the Servers spend less time idle (that is, they will commonly see a new Client immediately after releasing their previous Client). However, this is not the case. The median time a Server spent idle in the Poisson simulation was 150 minutes per day, and it was 146 minutes per day in the Exponential simulation.

Both simulations had a median finishing time of 521 minutes (ie: the final Client left at 5:41 PM, 11 minutes after the ‘clinic’ was due to close).

I ran the simulations again with the following parameters:

- 510 Minutes
- Arrival rate: 6 per hour
- Service rate: 3 per hour, exponentially distributed
- 2 Servers

This yields a traffic intensity of 1.

It is reasonable to assume that because the traffic intensity is larger that the Servers will spend less time idle and that the time a Client spends Queueing will be longer.
Indeed, this is the case. The single longest waiting time from for a Client was 31 mins in the Poisson, and 14 in the Exponential system. This is a rise of 340% and 466% over the simulation with an intensity of ¾.

From my experiments, I haven't found a system that works considerably better than the other. Both distributions have very similar results in terms of the time-efficiency for Servers and the time Clients spend waiting to be served.

Task 2: Is one experienced server better than two novices?

In this section, I first ran the simulation with as an M/M/2 system with 2 inexperienced Servers as follows:

- 510 Minutes
- Arrival rate: 3 per hour, exponentially distributed
- Service rate: 2 per hour, exponentially distributed
- 2 Servers

This yields a traffic intensity of ¾.
I also ran the simulation as an M/M/1 system with 1 experienced Server with the following parameters:

- 510 Minutes
- Arrival rate: 3 per hour, exponentially distributed
- Service rate: 4 per hour, exponentially distributed
- 1 Server

This yields an identical traffic intensity of ¾.

While the M/M/2 system has a slight advantage regarding Queueing times (avg: 0.2221 mins v 0.8387 mins M/M/1), the Clients will spend a far longer time in the System overall (15.837 minutes in M/M/1 versus 30.2254 in M/M/2 because the service rate is distributed around a lambda of 30 minutes and not around 15 minutes. Also, because of how the arrivals are distributed evenly throughout the entire day, a Client will still arrive at around 5.15 PM, 15 minutes before the scheduled close. This results in the median closing time in the M/M/2 system being 6.01 PM (541 mins), while in the M/M/1 system the median closing time is the planned time of 5.30 PM. Indeed, the M/M/2 system never finished on time in my simulations (the earliest closing time was 5.32 PM), while the M/M/1 system finished on time 135 of the 250 times the simulation was ran (54% of the time).

M/M/1 is better too regarding the time-efficiency of the Server(s). In M/M/1, the Server was idle for a median of 150 minutes in the day, while each Server in M/M/2 was idle for 169 minutes in the day (a total of 338 median idle minutes between them).

In my opinion, from a financial viewpoint, the M/M/1 system would be the best for a medical clinic. Assuming the wage paid in the M/M/1 system is the same as is paid in the M/M/2 system, not only will they need to pay less overtime to the experienced Doctor, they will also be getting more bang for their buck as they will be paying 1 Doctor who will be idle for 150 minutes a day, while they would be paying 2 Doctors who would be idle for an 169 minutes each. Going by the median figures, between overtime and the increased idleness, they would be paying for an extra 50 minutes of wages per day. They would also need to rent 1 less room for Patients to be served in.

I ran the simulation again with the following parameters:

- 510 Minutes
- Arrival rate: 6 per hour, exponentially distributed
- Service rate: 3 per hour, exponentially distributed
- 2 Servers

This yields a traffic intensity of 1.
I also ran it with these parameters:

- 510 Minutes
- Arrival rate: 6 per hour, exponentially distributed
- Service rate: 6 per hour, exponentially distributed
- 1 Server

This yields an identical traffic intensity of 1.

These tests re-affirmed the results of the previous simulation. Our M/M/2 Doctors spent a median 29 minutes each idle while our M/M/1 Doctor only had 20.5 median idle minutes. Our M/M/1 Doctor spent a median 500 minutes serving patients, and finished work at 5:40 PM (520 minutes) while our M/M/2 clinic closed at 5:49 PM (529 mins).

I also ran the simulation with the original parameters (traffic intensity = ¾) except with the arrivals distributed with Poisson. This change made quite a difference to the output. The M/M/1 system lost its advantage of having less Server idle time (median 155.5 mins idle v 154 mins in M/M/2). Furthermore, the median finishing times for the simulations became closer (5:52 PM in M/M/2 versus 5:51 PM in M/M/1).

Task 3: More realistic pattern of service (with scheduled arrivals)

In this section, I started by running the simulation with the parameters given in the assignment brief (ie: Breaks last 30 minutes, Server 1 breaks at 1030 and 1430, Server 2 breaks at 1130 and 1530, or whenever they finish with a Client). In my system, these times are 90, 330, 150 and 390 respectively.
In my simulation, I do not count time spent on lunch as time idle.

I ran the simulation 250 times with params:

- 510 minutes
- Arrival rate: 6 per hour, exponentially distributed
- Service rate: 4 per hour, exponentially distributed
- 2 Servers
- The break times given above with break length of 30 minutes.

This yields a traffic intensity of ¾.

Adding breaks of this size into this particular system results in some interesting differences. As one would expect, the time Clients spend queueing increases (Median with breaks: 1.5 mins in queue, without breaks: 0 mins). The maximum queueing time increased from 3 to 15, and the maximum time a Client spent in the system increased from 21 to 31.

The time each Server spent idle reduced from a median of 146 to 98, but surprisingly so did the time spent serving Clients, which reduced from a median of 375 to 361.

However, it is surprising to note that the median finishing time remains at 5:41 PM even with breaks added. Perhaps if the breaks were moved to later in the day, the Servers would not be able to pick up the slack generated during their breaks 5:30 PM.

I ran the simulation again with the same parameters but I changed the lunch times to:

- Server 1: 1130 and 1530
- Server 2: 1230 and 1630
- The break times given above with break length of 30 minutes.

In my system, these times are 150, 390, 210 and 450 respectively.

The only difference I expected to see here was in the finishing times. However, this wasn’t the case. The median finishing time remained at 5:41 PM.

Task 4: Customers not all the same – how best to manage?

I ran the simulation 250 times with params:

- 510 minutes
- Arrival rate: 6 per hour, exponentially distributed
- Service rate: 4 per hour, exponentially distributed
- Special service rate: 2 per hour, exponentially distributed, 1 in 6 chance
- Special Clients randomly distributed.
- 2 Servers

As expected, because some of the Clients now take twice as long to be served, the system runs slower. The median average waiting time is now 1.69 minutes (0 without Special Clients). The median closing time is now 5:45 PM (5:42 PM without Special Clients). The time spent in the system rises too on account of the longer queues and longer service times for some Clients. Perhaps the biggest change occurs when a number of Special Clients arrive early into the system, delaying the remaining Clients. The largest queueing time is now 34 minutes (without Special Clients it was 3 minutes), and one Special Client took 63 minutes in the system between queueing and being served.

I now tested the system with the following parameters:

- 510 minutes
- Arrival rate: 6 per hour, exponentially distributed
- Service rate: 4 per hour, exponentially distributed
- Special service rate: 2 per hour, exponentially distributed, 1 in 6 chance
- Special Clients arrive last
- 2 Servers

As one could expect, this heavily impacted on the finishing times. The median closing time is now 6:26 PM (566), as opposed to 5:45 PM with randomly distributed Special Clients and 5:42 PM without Special Clients. The average queueing time increased from 1.69 mins with randomly distributed Special Clients to 2.42 mins. The rates of Server business did not change.

It is best in my opinion to handle the Special Clients throughout the day. Efficient Servers can build work through the normal Clients and when they encounter a random Special Client, they can deal with them and then get back on pace quite quickly if a normal Client follows.

Conclusions

In conclusion

- Random v Scheduled arrivals
    - Poisson distribution results in longer queueing times
    - Surprisingly, the Servers in Poisson spend slightly more time idle
- One experienced Server v two novice Servers (M/M/1 v M/M/2)
    - M/M/2 has slightly smaller queueing times
    - M/M/1 has earlier finishing times, and the Server is more efficient with his time
    - M/M/1 would be more financially viable in my opinion
- Lunch breaks
    - Adding in lunch breaks as prescribed doesn’t have much of an impact on the finishing time
    - It does however increase the time-efficiency of the Servers
    - Moving the lunch breaks back one hour has no effect on closing times
- Special Clients
    - Adding Special Clients throughout the simulation increases the average queueing times.
    - However it doesn’t have much of an impact on the finishing times.
    - If the Special Clients enter the simulation near the end, the finishing times get pushed back significantly.

Appendix

My code

Unfortunately Google Docs didn't co-operate when I pasted my code in here, so it's contained in the zip file attached. The zip file also contains the output .txt files I generated, from which I took the statistics I use above.