BS213 _simulation
Midterm Projects

Pick one question. If you would like to change any of the parameters, please explain how and why you are changing them. Also please explain any parameters that you are simulating. You will be required to show your code and your results in class.

1) Assume that banks open at 8am and close at 5pm. Customers arrive at the bank at a rate of 4/hr. Between the hours of noon and 1 that rate increases to 6/hr. Suppose we have 3 tellers each with their own rate of service. Teller 1 has rate G1 which is exponential with rate 6, Teller 2 has rate G2 ~ exponential with rate 5, Teller 3 has rate G3 ~ exponential with rate 4. We have studied 3 types of queueing systems 1) 1 line with parallel servers. 2) tandem service (DMV example) and 3) where each teller has their own line (grocery store). Given three tellers which queueing system has the shortest expected waiting time for the customers?

Report:

The first run was done with the initial skeleton before making any adjustment. That means, the initial run had 1000 repetition.

Average expected waiting time for the customers in the system (AWT) for each of the three queuing systems:

Sl.	Queuing Systems	Pass1: AWT (h)	Pass2: AWT (h)
1.	Single line with multiple servers	0.19	0.18
2.	Tandem service	2.26	2.25
3.	Teller with own line	0.2	0.19

Then we run the process after each adjustment (3 main adjustments) for 10,000 repetitions and two passes for each adjustment to make sure there is no unusual variation in the results.

1. **Intensity function**

I assumed rush hours when customers have tendency to visit the bank. First hour after the bank opens (8 am- 9 am) and the last before it closes (4 pm- 5 pm).

Sl.	Queuing Systems	Pass1: AWT /SD (h)	Pass2: AWT/SD (h)
1.	Single line with multiple servers	0.193/0.0372	0.194/0.0381
2.	Tandem service	2.54/0.9823	2.531/0.9705
3.	Teller with own line	0.197/0.04	0.198/0.0406

2. Variation in the teller's efficiency

Then, I assumed that each teller has different efficiency than what they have usually. It could be for many possible reasons like a teller has some issues with his/her health, mental issue, family issue, etc. In this case, the assumptions made are as follows:

Teller1 is 100% of his/her efficiency, while Teller2 is 95% (r2 = 95% of r2) of his/her usual efficiency, and Teller3 has 90% (r3 = 90% of r3_of his/her actual efficiency.

Sl.	Queuing Systems	Pass1: AWT /SD	Pass2: AWT/SD
1.	Single line with multiple servers	0.9845/0.2929	0.9804/0.2842
2.	Tandem service	124.884/25.5686	125.1671/25.7939
3.	Teller with own line	2.2404/1.1565	2.2761/1.1851

3. Abnormal system behavior

Abnormal system behavior might occur suddenly and possibly pause the system for a while. This may occur to any server of any teller. In this case, the assumption made like this:

There is 10% chance that any server might encounter an unusual behavior/failure for 0 to 30 minutes. So, every time a teller gets a customer, the probability of failure is determined and if the probability is greater than or equal to 0.9, only then unusual behavior will be taken into consideration. As a result, a delay (0-30 minutes) is added to the teller while serving a customer.

Sl.	Queuing Systems	Pass1: AWT /SD	Pass2: AWT/SD
1.	Single line with multiple servers	1.1406/0.3794	1.1527/0.3816
2.	Tandem service	124.6773/26.6289	129.1839/13.3416
3.	Teller with own line	22.7169/1.3952	2.7162/1.4546

As we can see for the initial set up and after each of the adjustments to the system, the single line with parallel server queuing system provides the shortest expected waiting time. For every adjustment made to the system, the single line parallel queuing system always has the shortest expected waiting time for the customers in the system.

There are so many ways that the system can be improved to be more realistic. The possible extensions could be adding priority for the senior citizen customers arriving in the system, assigning priority based on the service customers need, etc. This was really an interesting problem to work on and a good way to make this efficient would be the hybridization of multiple queues.