Assignment 1
Discrete Event Simulator

FATMA ELGALLAL SAHITYA BOLLAVARAM

1. Introduction:

This report is an analysis of the discrete-event simulator with M/M/1 queue model, M/M/N queue model with Random selection, and supermarket models. Moreover, a comparison of the results of the simulator with theoretical results is made for two different scheduling policies for queues FIFO, SJF, and actual data token from a website.

2. Background

In the Markovian queuing model "M/M/c queue", the jobs arrive in the system one by one following the Poisson process with parameter λ and are served at service rate μ; all jobs are served in a memoryless fashion. To be in equilibrium; λ<μ is considered , otherwise, queue keeps growing.

In this model M/M/c, the number of servers at the queue node is a finite number represented with the letter c. The distribution of job arrivals in server queues can be done randomly or by checking the shortest queue like the supermarket model.

3. Implementation:

Single-server queuing model M/M/1 (FIFO):

Assuming one server (1 FIFO queue) with different arrival rate λ, services rate μ=1, and simulation time 1000000ms. The first thing that is analyzed is the mean time spent by jobs in the system, this is done by using Little's Law which is defined theoretically for mean spent time as follows.

$$\frac{1}{(1-\lambda)}$$

As shown in Table 1 and figure 1, the expected values from the simulation are close to theoretical values, and error increases with an increase in lambda because the waiting time of a job increases with lambda.

λ	Theoretical delay time	Simulated delay time	Relative Error
0.50	2.0	2.000387	0.000193
0.90	10.0	10.042570	0.004239
0.95	20.0	20.150665	0.007477
0.99	100.0	99.754669	0.002459

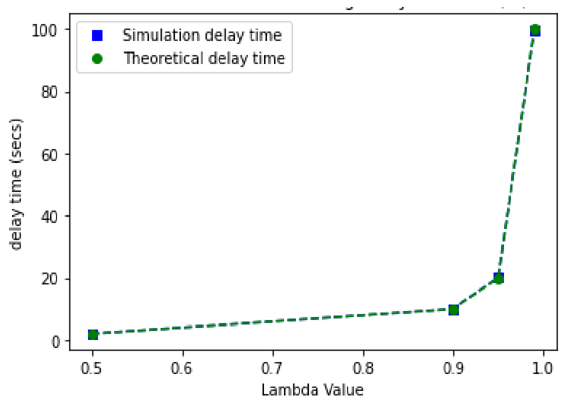


Figure 1Theoretical Vs Simulated delay time

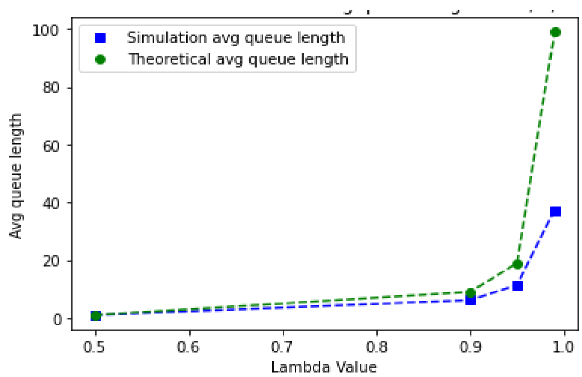
The second thing to consider is the mean queue length which is theoretically defined by the Little's Law as following.

$$\frac{\lambda}{(1-\lambda)}$$

As shown in Table 2 and figure 2, the obtained results are close to the theoretical results, and it can also be seen that with an increase in the value of λ, the meantime spent by the job in waiting in the queue also increases and this is also obvious in the pictures from the length of the queue.

λ	Theoretical queue length	Simulated queue length	Relative Error
0.50	1.0	1.000589	0.000588
0.90	9.0	6.167647	0.459227
0.95	19.0	11.412131	0.664895
0.99	99.0	37.304934	1.653804

Table 1 Theoretical Vs Simulated queue length M/M/1



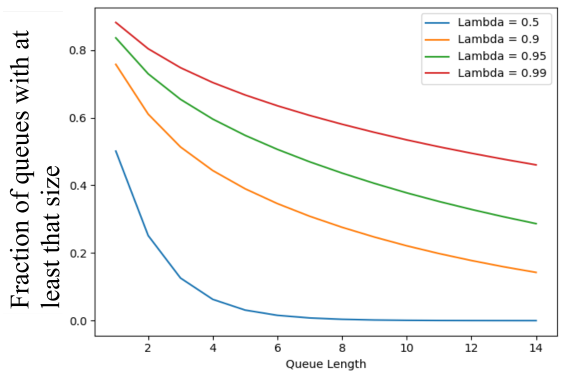


Figure 2 Theoretical Vs Simulated queue length

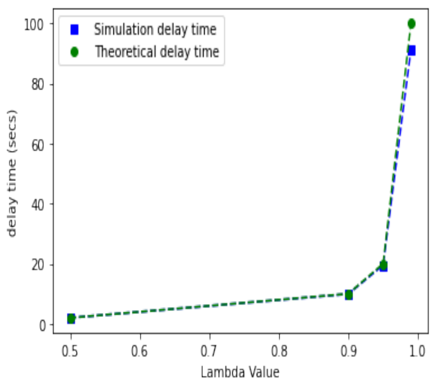
<u>Multi-server queuing model M/M/N</u>

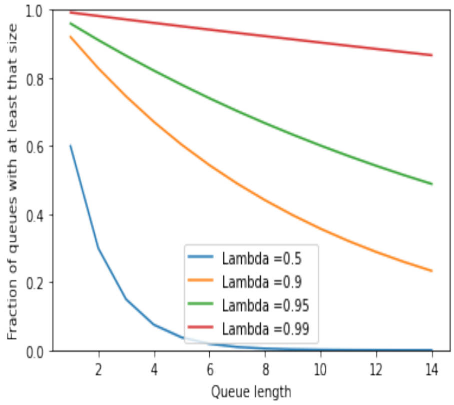
Random Selection – FIFO

Assuming that there are fifty servers (50 FIFO queue) and values for λ and μ, in this model, jobs need a way to select the right queue; the first study is done using Random selection strategy.

Similar to M/M/1, the mean time spent and the mean queue length for each server queue are analysed and compared with theoretical values.

The obtained delay time and queue length, increases with λ and is close to theoretical results much like the M/M/1 queuing model.





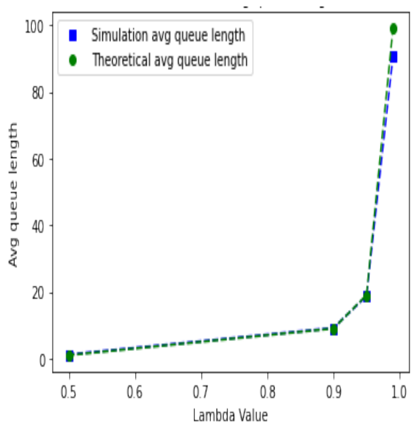


Table 2 Theoretical Vs Simulated delay time M/M/N-Random

l	Theoretical delay time	Simulated delay time	Relative Error
0.50	1.0	1.198675	0.165746
0.90	9.0	9.138334	0.015138
0.95	19.0	18.741692	0.013783
0.99	99.0	90.524731	0.093624

Table 4 Theoretical Vs Simulated queue length M/M/N-Random

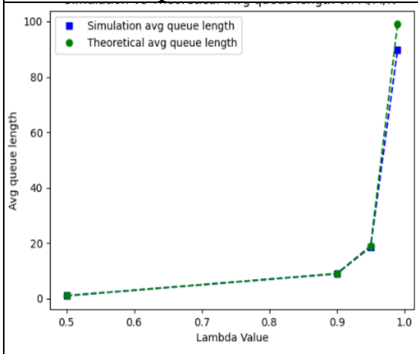
l	Theoretical queue length	Simulated queue length	Relative Error
0.50	2.0	1.999593	0.000203
0.90	10.0	9.932039	0.006843
0.95	20.0	19.524904	0.024333
0.99	100.0	91.254466	0.095837

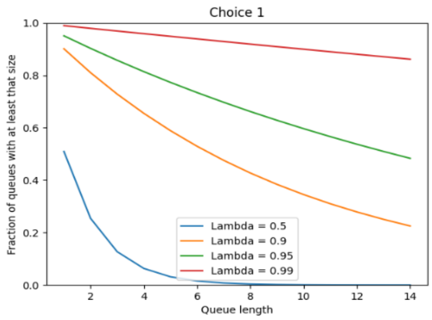
Super Market Strategy - FIFO

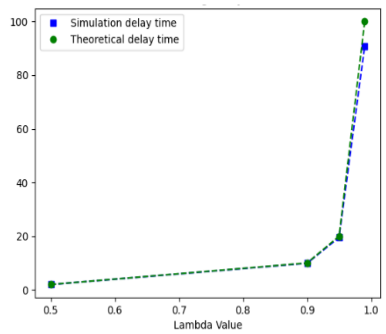
In this implementation, a sample of queues d is chosen from the total queues; the job is assigned to the shortest queue. Then, analysis is made for different values of "d" to find out if the number of choices impacts the queue length, assuming that this model is more efficient than random.

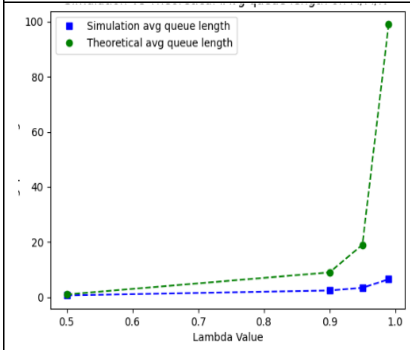
The figures below represent different d values 1,2,5,10 out of 50 servers.

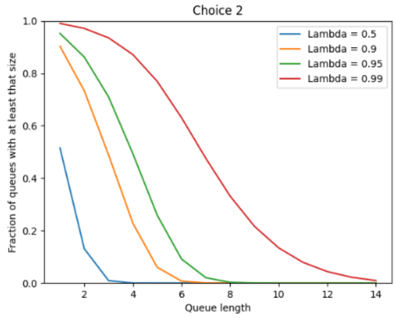
M/M/N Super Market with FIFO

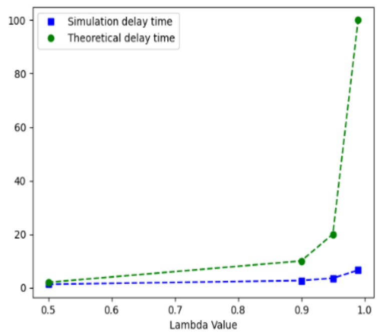


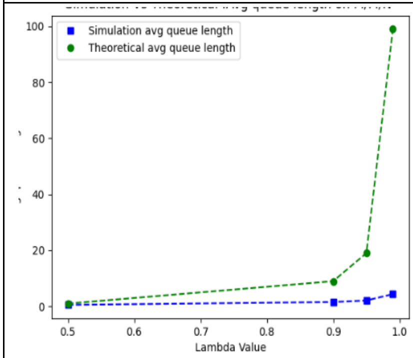


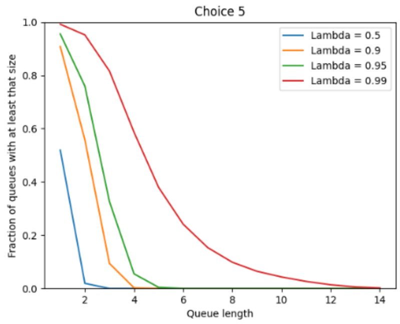


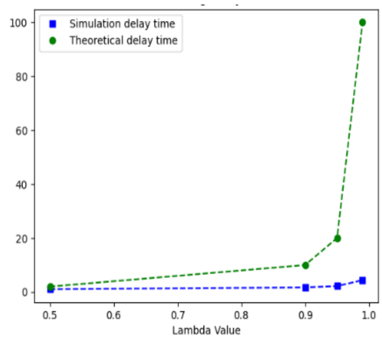


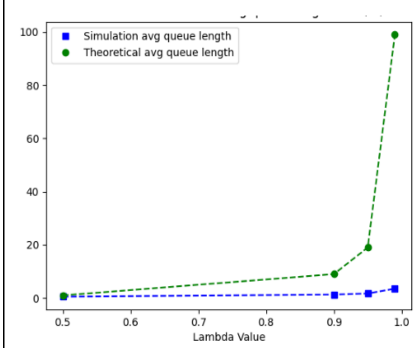


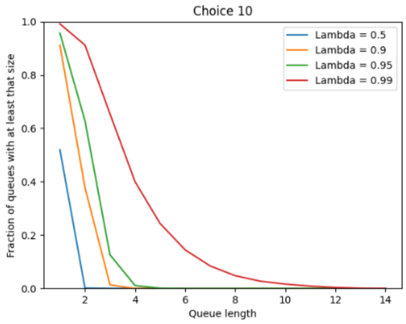


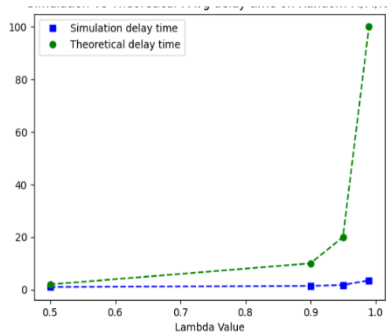












In general, this model is more efficient than the previous ones. This statement can be verified by considering the mean time spent by jobs in waiting and the queue's mean length. All these results are less than the random selection model and M/M/1. For 50 servers, the results show that with an increase in the number of choices, the queue length and the delay time are decrease. Charts1,2 show a comparison between Random and Super Market Model based on the mean queue length and the delay time for different choices. For choice1, it is quite clear from the charts that the queue length and the delay time are quite close which means that they behave in a similar fashion. However, for other choices, Random selection performs consistently bad when compared to supermarket model. Also, in chart 3 and 4, a comparison between Random and supermarket model is done based on different arrival rates. The queue length and delay time in this case, also seem to be better for supermarket model. All of these charts prove that Supermarket performs better than Random strategy.

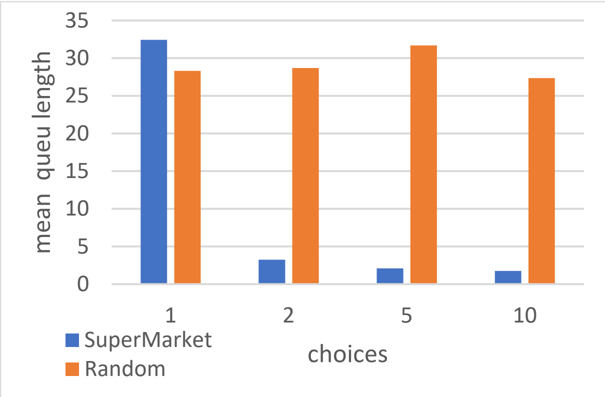


Chart 1 choices with mean queue length

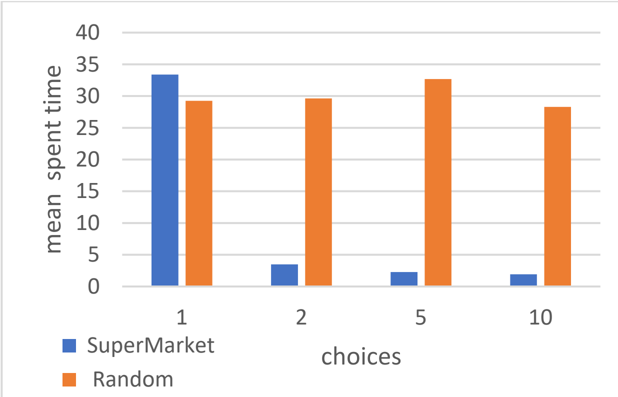


Chart 2 choices with mean spent time

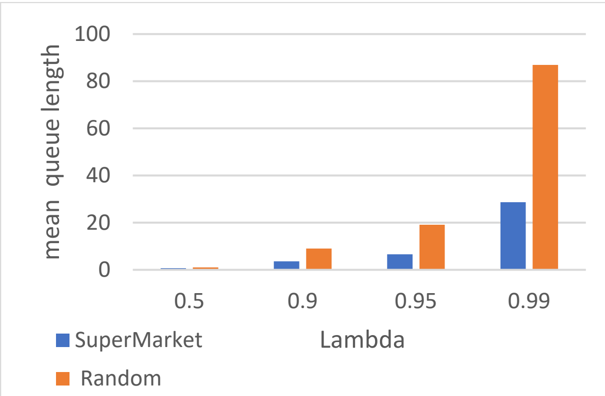


Chart 3 lambda with mean queue length

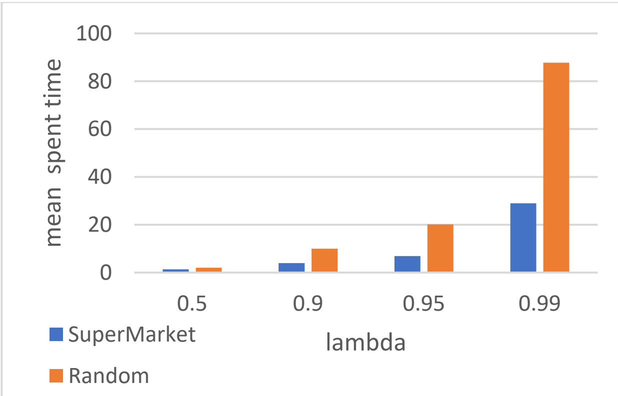
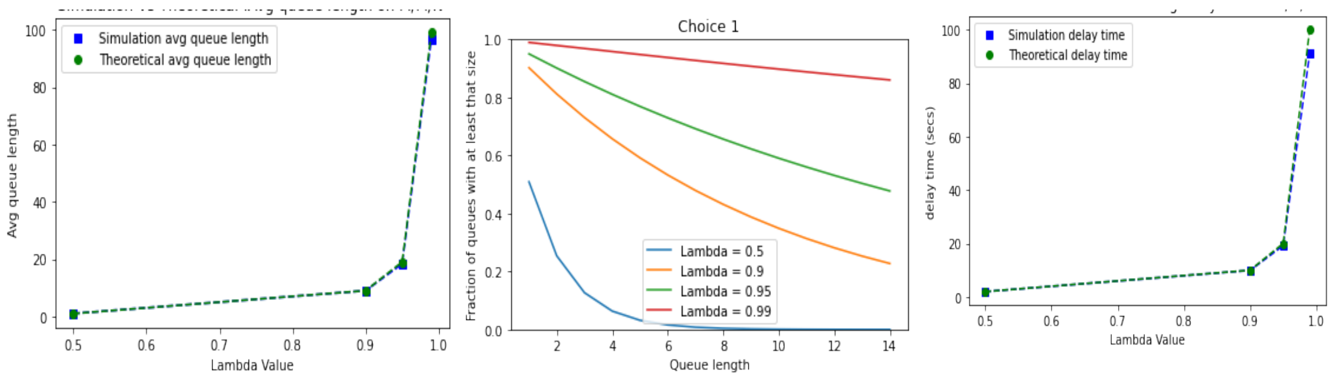


Chart 4 lambda with spent time

Shortest Job First (SJF) is an algorithm in which the process having the shortest execution time is chosen for the next execution.

The duration time of each job is calculated, which is the difference between arrival time and service time.

Random Selection – SJF

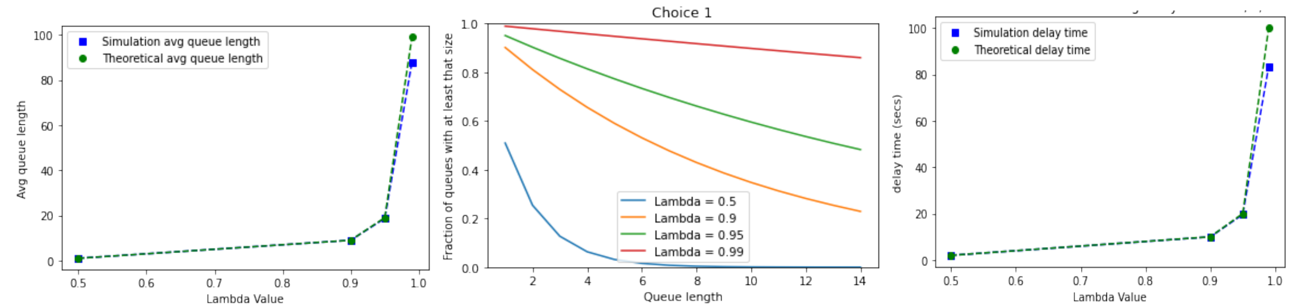


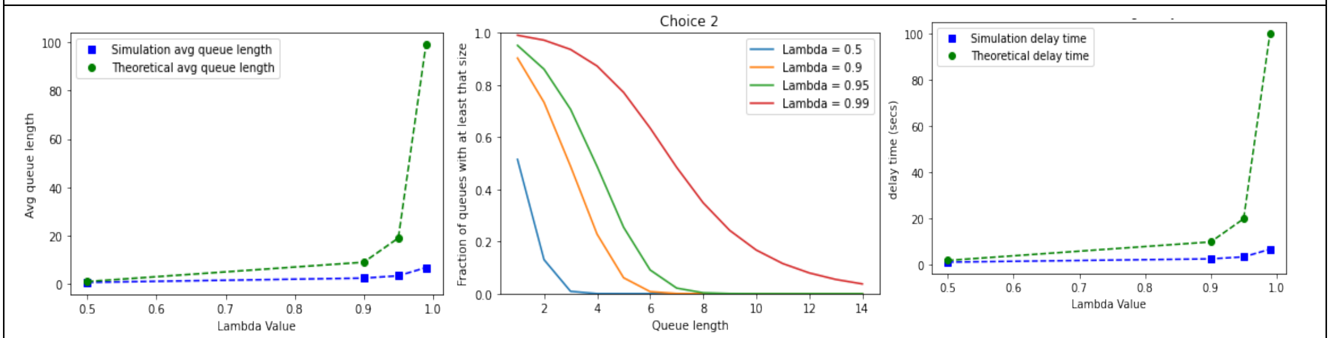
Super Market Strategy – SJF

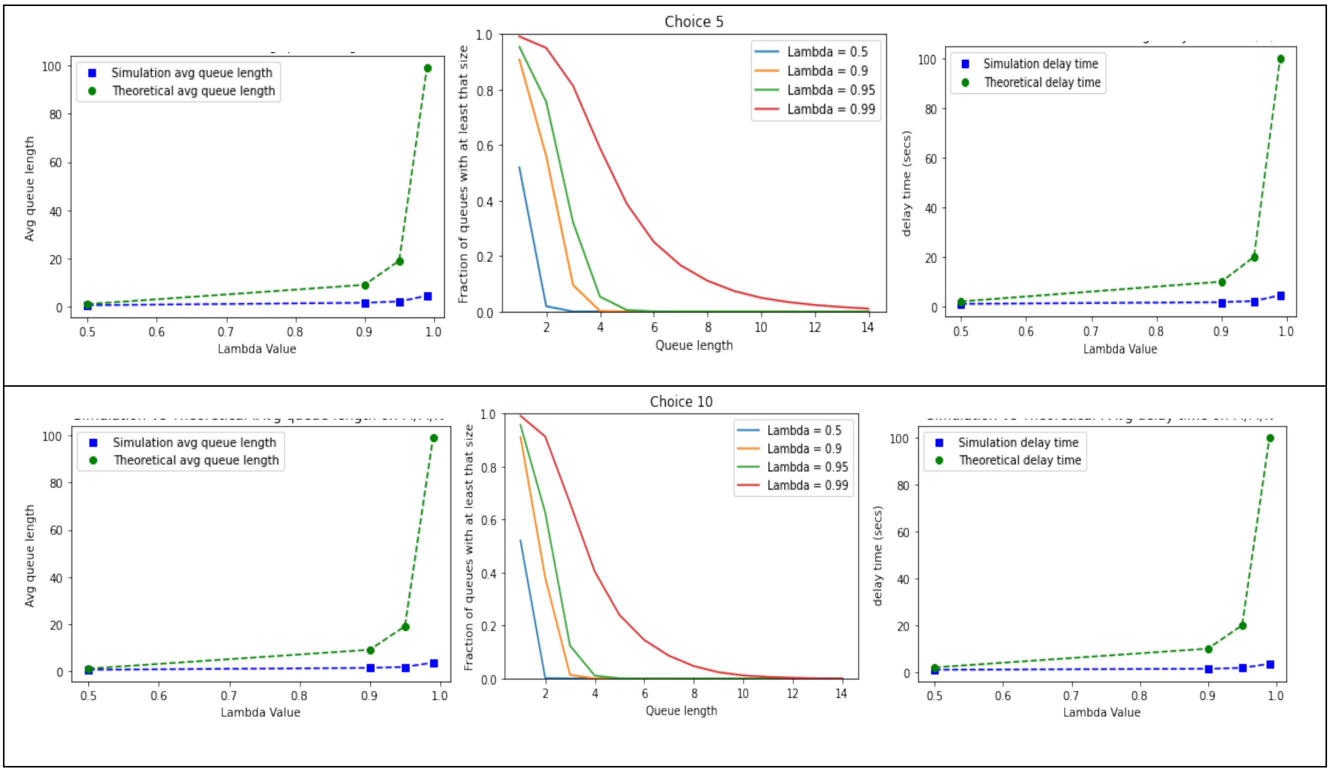
In this implementation, A sample of queues d is chosen from the full queues, the job is assigned to the shortest queue, and the jobs are further sorted in the queue based on the job size (duration of each job).

The figures below represent different d values 1,2,5,10 out of 50 servers.

M/M/N Super Market with SJF







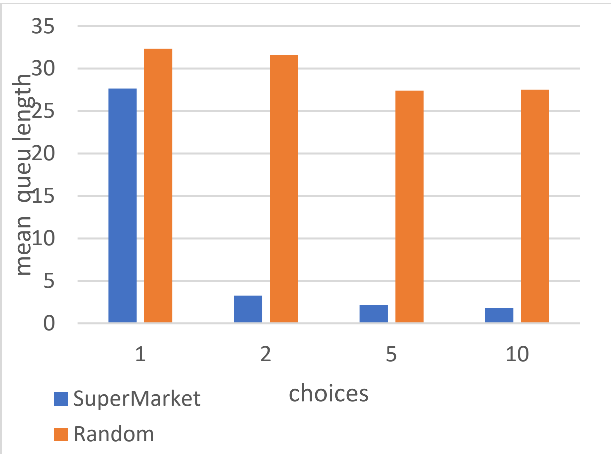


Chart 5 choices with mean queue length

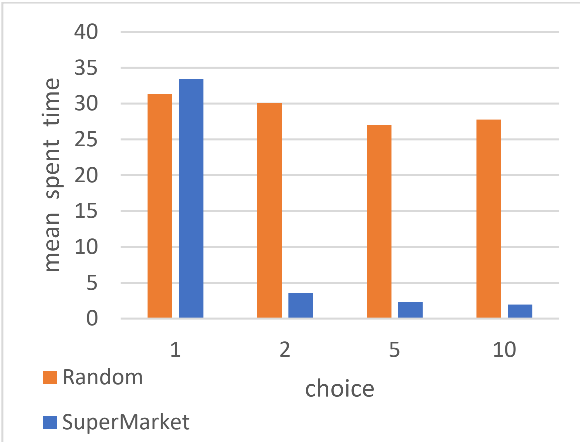


Chart 6 choices with mean spent time

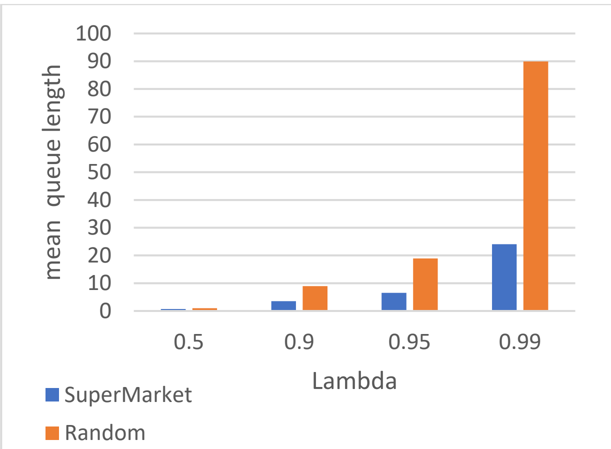


Chart 7 lambda with mean queue length

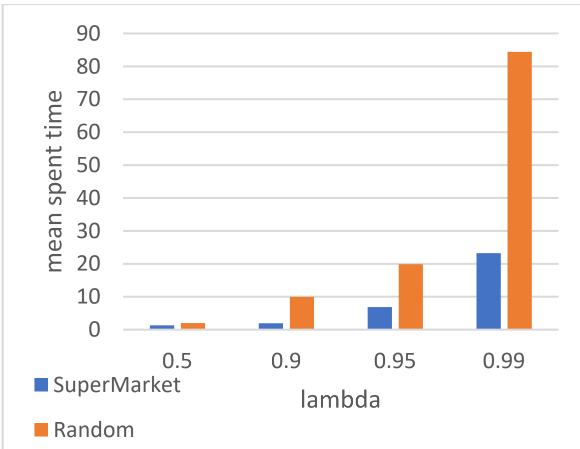


Chart 8 lambda with mean spent time

The major advantage of this algorithm is that it gives a minimum waiting time for a given set of processes and thus reduces the average waiting time. However, looking at the results, SJF seems to perform very similar to FIFO maybe with a slight improvement in the waiting time(but not always). This scheduling algorithm can be beneficial, if the job time is known ahead of processing the job(bulk processing).

The disadvantage of this algorithm is that long processes may never be processed by the system and may remain in the queue for very long time leading to starvation of processes.

Real Data:

For a real scenario, the Mustang dataset is used from ATLAS. This dataset stores a list of jobs with all information, such as start time and end time.

Figure 4 shows the average queue length on the real data sing M/M/1 ,while Figure 5 shows the average queue length on the real data using M/M/N. Since M/M/1 has a single queue, the average queue length seems to be a lot higher than M/M/N where we have several queues.

For the given dataset, the lines in figure 5 seem to fall very close to each other. On observing figure 5 closely, it can be seen that the queue length also decreases with the number of choices in the server.

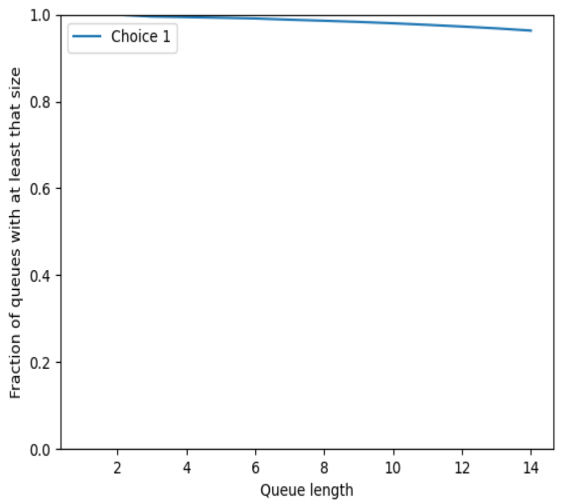


Figure 3 MM1

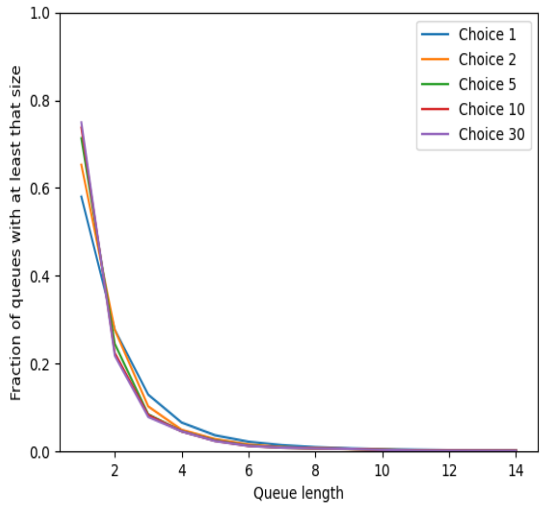


Figure 5 MMN