CPSC 641 - Assignment 2: Covid Barber Shop [Amirhossein Sefati]

Part 1: Analytical Modeling

1-1- Markov Chain

As I emailed you, I was considering both barbers as one state with an average of $\mu_{avg} = (\mu_F + \mu_G)/2$. But you told me that I should consider different states for each of them. So, this is the result for Markov Chain with K=5. (6 states which one of them has 2 sub-states)

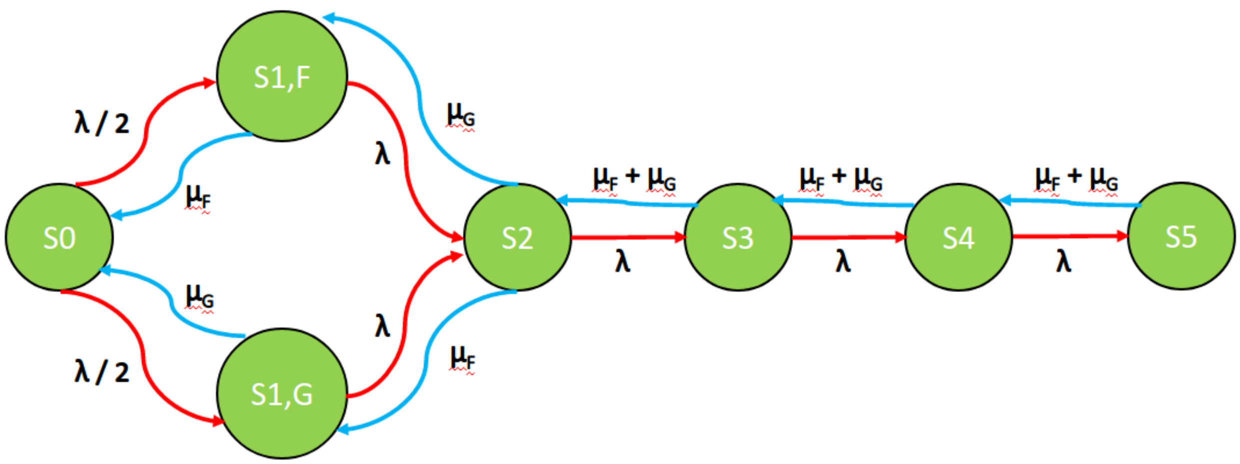

Figure1: Markov Chain of the syste

1-2- Balance Equations

1- P(S0) * $\lambda/2$ = P(S1,F) * μ_F

2- P(S0) * $\lambda/2$ = P(S1,G) * μ_G

3- P(S1,F) * λ = P(S2) * μ_G

4- P(S1,G) * λ = P(S2) * μ_F

5- P(S2) * λ = P(S3) * $(\mu_F + \mu_G)$

6- P(S3) * λ = P(S4) * $(\mu_F + \mu_G)$

7- P(S4) * λ = P(S5) * $(\mu_F + \mu_G)$

8- P(S0) + P(S1,F) + P(S1,G) + P(S2) + P(S3) + P(S4) + P(S5) = 1

1-3- Derive the (steady-state) equilibrium state probabilities

Using balance equations from the previous section, we will get the following probabilities for each state. Figure 2 shows solving the equations in order to find all of the probabilities based on the P(S0) and then from equation number 8, we can find the value for P(S0) and from that we have all the probabilities.

$$P(S1,F) = \frac{P(S0) \times \frac{\lambda}{2}}{\mu_F}$$

$$P(S1,G) = \frac{P(S0) \times \frac{\lambda}{2}}{\mu_G}$$

$$P(S2) = \frac{P(S1,F) \times \lambda}{\mu_G} = \frac{P(S0) \times \frac{\lambda^2}{2}}{\mu_F \times \mu_G}$$

$$P(S3) = \frac{P(S2) \times \lambda}{\mu_F + \mu_G} = \frac{P(S0) \times \frac{\lambda^3}{2}}{(\mu_F + \mu_G) \times (\mu_F \times \mu_G)}$$

$$P(S4) = \frac{P(S3) \times \lambda}{\mu_F + \mu_G} = \frac{P(S0) \times \frac{\lambda^4}{2}}{(\mu_F + \mu_G)^2 (\mu_F \times \mu_G)}$$

$$P(S5) = \frac{P(S4) \times \lambda}{\mu_F + \mu_G} = \frac{P(S0) \times \frac{\lambda^5}{2}}{(\mu_F + \mu_G)^3 (\mu_F \times \mu_G)}$$

$$\text{from 8:} \quad P(0) = \frac{1}{1 + \frac{\lambda}{2\mu_F} + \frac{\lambda}{2\mu_G} + \frac{\lambda^2}{2(\mu_F \mu_G)} + \frac{\lambda^3}{2\mu_F\mu_G(\mu_F+\mu_G)} + \frac{\lambda^4}{2\mu_F\mu_G(\mu_F+\mu_G)^2} + \frac{\lambda^5}{2\mu_F\mu_G(\mu_F+\mu_G)^3}}$$

Figure 5: Probabilities for steady state of the system (Handwritten)

Table 1 shows the probabilities in a better way.

Table 1: Probabilities for steady state of the system (Typed)

P(S0)	$\frac{1}{1+\frac{\lambda}{2\mu F}+\frac{\lambda}{2\mu G}+\frac{\lambda^2}{2(\mu F\mu G)}+\frac{\lambda^3}{2(\mu F\mu G)(\mu F+\mu G)}+\frac{\lambda^4}{2(\mu F\mu G)(\mu F+\mu G)^2}+\frac{\lambda^5}{2(\mu F\mu G)(\mu F +}$
P(S1,F)	$\frac{(P(S0)\cdot\lambda)}{2\mu F}$
P(S1,G)	$\frac{(P(S0)\cdot\lambda)}{2\mu G}$
P(S2)	$\frac{(P(S0)\cdot\lambda^2)}{2\mu G\mu F}$
P(S3)	$\frac{(P(S0)\cdot\lambda^3)}{2\mu G\mu F(\mu G+\mu F)}$
P(S4)	$\frac{(P(S0)\cdot\lambda^4)}{2\mu G\mu F(\mu G+\mu F)^2}$
P(S5)	$\frac{(P(S0)\cdot\lambda^5)}{2\mu G\mu F(\mu G+\mu F)^3}$

1-4- Assume $\lambda = 6$, $\mu F = 5$, and $\mu G = 3$:

I code in python the results of table 1, and with the exact numbers from this section, I got Figure 3.

```
amirh@AHS:~/Desktop/Winter_2022/Courses/CPSC 641/A2$ python test.py
P(S0):  0.170031881
P(S1,F):  0.102019129
P(S1,G):  0.170031881
P(S2):  0.204038257
P(S3):  0.153028693
P(S4):  0.114771520
P(S5):  0.086078640
P(SanityCheck):  1.000000000
```

So, the expected number of customers in the system is: 0*0.17 + 1*0.102 + 1*0.17 + 2*0.204 + 3*0.153 + 4*0.1147 + 5*0.086 = 2.028

Utilization of each barber chair is: $\rho F = 1 – 0.17 - 0.17 = 0.66$, $\rho G = 1 – 0.17 - 0.102 =$ %77.8.

And the probability of losing a customer because of the system being full is as P(5) = %8.6.

Part 2: Simulation

2-1: continuous-time discrete-event simulation model

I write my code in Python (version 3.9) and it is nearly 150 lines long.

2-2: lost customers when λ = 6.0, µ1 = 5.0, and µ2 = 3.0

The screenshot below shows the result of simulation for the given parameters. I use the number of customers as a measurement. The number of iterations is 100 but the number of customers varies from 100 to 100000. It shows the attribute of randomness, when the number of customers is higher, the lost proportion is closer to the mathematical analysis.

```
amirh@AHS:~/Desktop/Winter_2022/Courses/CPSC 641/A2$ python simulation.py
----------- Average of 100 iterations with each time `100` customer requests ------------
Lost Of Customers when Lamda:6, muF:5 and muG:3 is:  8.89
amirh@AHS:~/Desktop/Winter_2022/Courses/CPSC 641/A2$ python simulation.py
----------- Average of 100 iterations with each time `1000` customer requests ------------
Lost Of Customers when Lamda:6, muF:5 and muG:3 is:  8.784
amirh@AHS:~/Desktop/Winter_2022/Courses/CPSC 641/A2$ python simulation.py
----------- Average of 100 iterations with each time `10000` customer requests ------------
Lost Of Customers when Lamda:6, muF:5 and muG:3 is:  8.6519
amirh@AHS:~/Desktop/Winter_2022/Courses/CPSC 641/A2$ python simulation.py
----------- Average of 100 iterations with each time `100000` customer requests ------------
Lost Of Customers when Lamda:6, muF:5 and muG:3 is:  8.63302
```

2-3: lost customers when λ = 6.0, μ1 = 5.0, and μ2 = 3.0

Out of curiosity, I just increase the load balance (with same waiting area) to see if the lost proportion will increase or decrease. (Which should increase because the load balance before was %75 and now is %90), and I see that it increases (Yuuu Huuu!). (From %8.6 to %14.2)

```
amirh@AHS:~/Desktop/Winter_2022/Courses/CPSC 641/A2$ python simulation.py
---------- Average of 100 iterations with each time `10000` customer requests with Load of %90 ------------
Lost Of Customers when Lamda:6, muF:5 and muG:3 is:  14.1983
```

Now, I will change the parameters like the following: λ = 9, μF = 6 and μG = 4. (So, the load is %90). Then I will change the size of the waiting area from 0 to 10 and see the results. Figure 6 shows this.

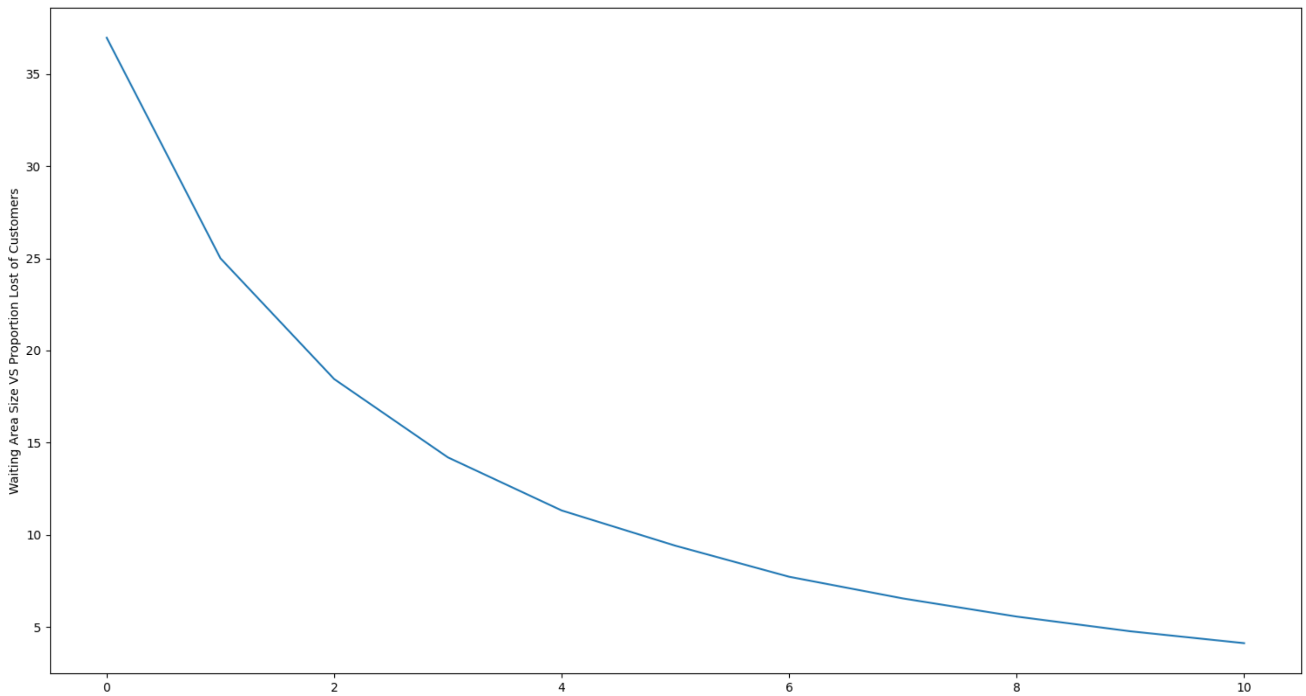

Figure 6: Waiting Area Size VS Proportion Loss of Customers (Percentage)

As we can see, the loss of customers is decreasing with the increasing number of waiting size. The slope is decreasing, that means when the waiting area changes from 0 to 1, the loss of customers changes from ~37 to ~25 which is almost half. But when it changes from 9 to 10, it decreases from 4.8 to 4.2. Please note that here I consider K as the waiting area size, so when K=0, it means we have just servers. But in the next part, I will consider K as the system size, so when K=0, it means we have nothing, even servers so the loss of customer would be 100%!

2-4: Compare M/M/1/K with the simulation μ = μ1 + μ2

I will use k = 0 to 10 as the size of waiting area, then calculate the probability of customer lost. Then I will compare this probability to Figure 6 which is the probability of customer lost for M/M/2/K (heterogeneous). Figure 7 shows the formula for the probability of any of states in a M/M/1/K queue.

$$p_n = p_0 \rho^n, \qquad \text{where } \rho = \frac{\lambda}{\mu}$$

$$p_0 = \left[\sum_{n=0}^{K} \rho^n\right]^{-1} = \begin{cases} \dfrac{1-\rho}{1-\rho^{K+1}} & \rho \neq 1 \\ \dfrac{1}{K+1} & \rho = 1 \end{cases}$$

Figure 7: Probability of each state in M/M/1/K system

So, Table2 shows the probability of losing customer for M/M/1/K is like the following:

Table 2: Probability of losing customers vs size of M/M/1/K system with %90 load

Size of System (K)	Probability of Losing customer
0	1
1	0.473
2	0.298
3	0.211
4	0.160
5	0.126
6	0.101
7	0.083
8	0.070
9	0.059
10	0.050

Figure 8 shows the graphical plot for comparing the results with the previous simulation of M/M/2/K (heterogeneous).

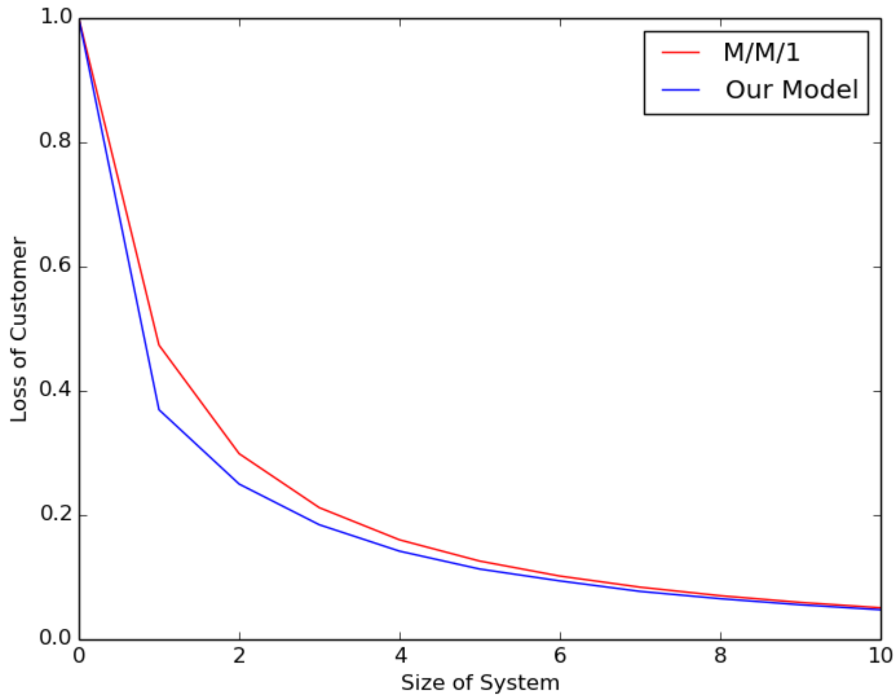

Figure 8: Comparison between our model and M/M/1/K (Loss of Customer vs Size of system)

As we can see from Table 2, when we don't have any queue (K=1), the probability of losing a customer is nearly half. That is because the load of the system is high (0.9). For example, if the load was full (1.0) the probability would be exactly 0.5. Lower the load, lower the probability of losing a customer. From Figure 8 we can see that the M/M/1/K is working worse than the M/M/2/K system. It can be described that with the same load, if we have two weaker servers (compared to one strong server which the power is aggregated from two weaker servers) it will be better in performance. Maybe, that is one of the reasons that having two 4GB of RAM slots is better than having one 8GB RAM! Also, we can see the difference between the performance of these two systems get lower when the K is higher. So, when we increase the size of waiting area, M/M/1/K have same performance as M/M/2/K (of course with same μ). Also, I tried to plot the utilization of servers. Figure 9 shows the servers, the red line indicates the utilization of single server system, blue and green lines show the utilization for M/M/2/K. We can see when the K is not high enough, M/M/2/K utilization for both servers is better than the single server mode. But as we increase the K, the utilization of single server is nearly the average of utilization of two servers of M/M/2/K system.

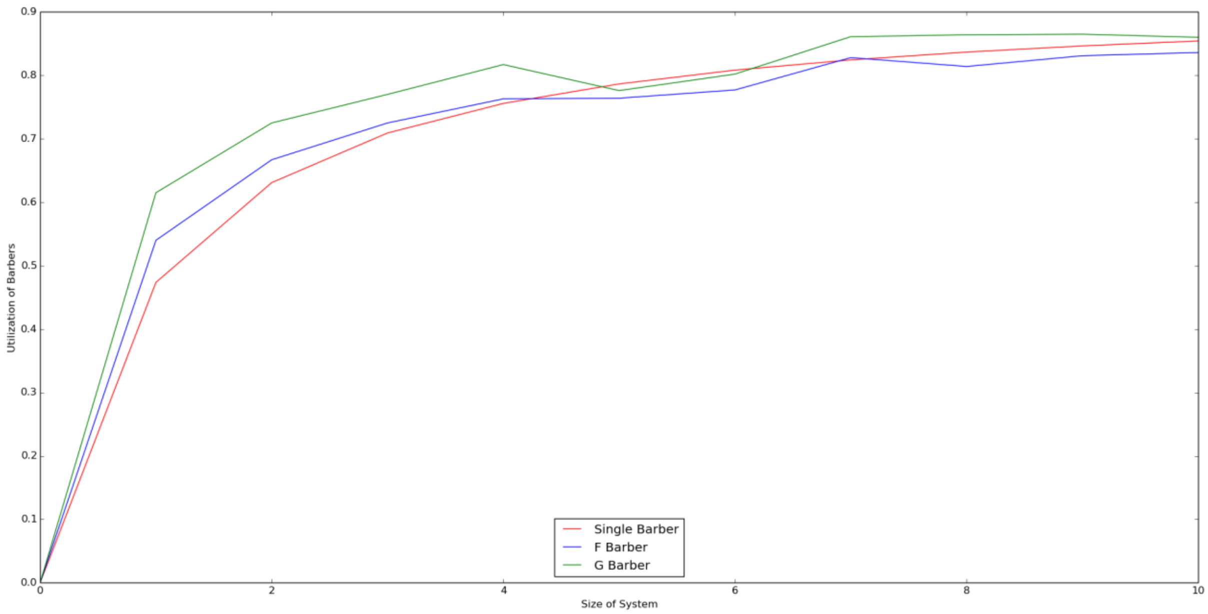

Figure 9: Comparison between our model and M/M/1/K (Utilization of servers - Size of system)

Figure 9 suggests that with increasing the waiting size, utilization of servers is more like each other no matter we are using a single server with $\mu = \mu1 + \mu2$ or two servers with $\mu1$ and $\mu2$. Also, it shows that in this case, with a small waiting area size, the utilization of our model is better than M/M/1.

2-5: Compare M/M/2 with our simulation (μ = μ1 + μ2) (Load=90%)

I will compare our model to the M/M/2 system. First, I will consider the loss of customers in both systems. Since there is no limit in the waiting area, we know the loss of customers for M/M/2 without a finite capacity is zero. Now, for our model, I try to plot the loss of customers for K= 0 to 100 and see what the loss of customers will be. Figure 10 shows that.

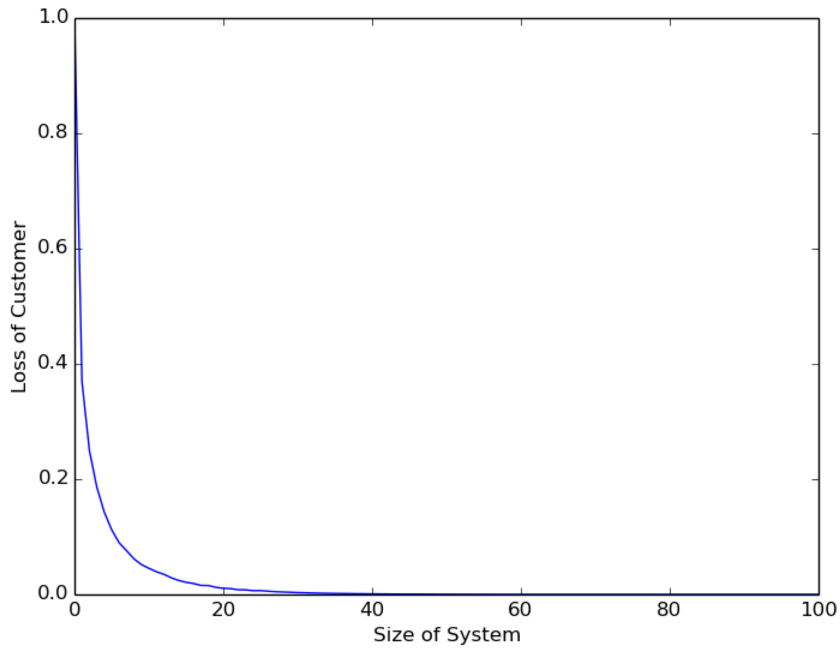

Figure 10: Loss of customer in our model vs system size (K)

As Figure 10 shows, when the waiting area is large enough, the loss of customers is 0. This is like the behavior of M/M/2 with infinite waiting area.

Now, let's compare the utilization of servers. We know that the utilization of servers in M/M/2 with an infinite waiting area is calculated by: $U = \frac{\lambda}{c \cdot \mu}$

So, for λ=9 and μ=5 and 2 servers (c=2), we will have the following: *U = %90.*

Now, I will try to increase the size of system (K) from 0 to 100 and we will see what the utilization of servers will be and whether it will go up to %90 or not. Figure 11 shows that.

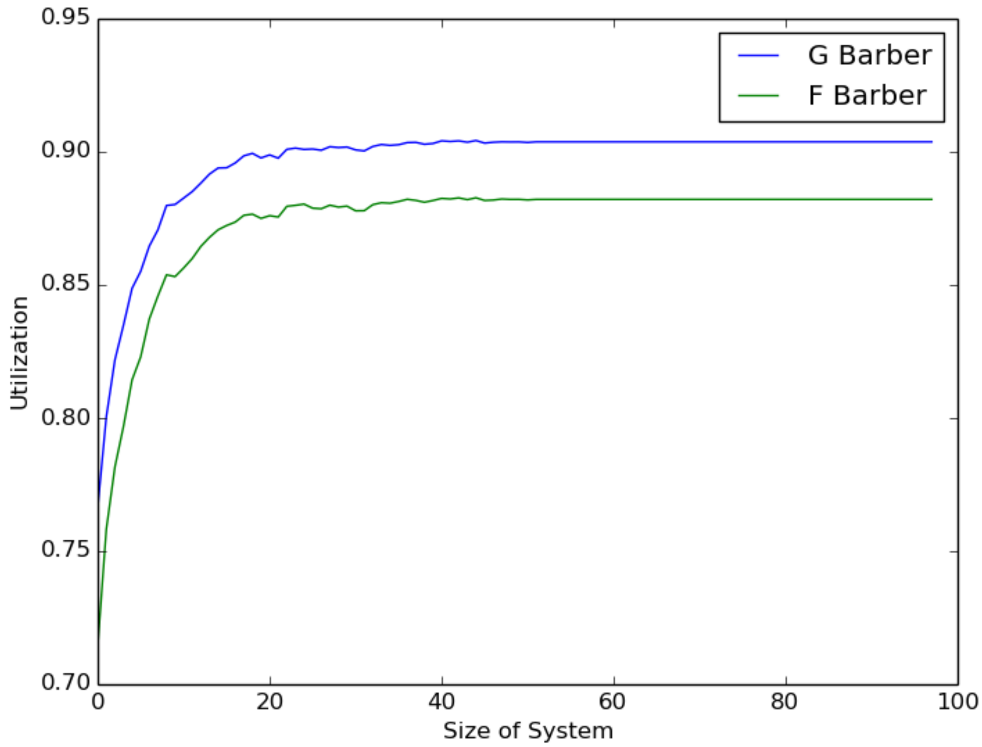

Figure 11: Utilization of Servers in our model

As Figure 11 shows, the utilization of both barbers will increase to nearly %90. It is like our analytics for M/M/2 with infinite waiting area size. So, our model is truly simulated, and it is accurate.