Markov Chains & Queuing Theory to Optimize a Vaccine Center System

Mathematics Extended Essay

Contents

1 Introduction:

I came home on the 3rd of May when my boarding school shut classes amidst the height of the 2nd COVID wave in India. As I sat in our car on the drive home, the headline on the Economic Times that lay on my mother's lap screamed "India's vaccination drive in crumbling". Long lines at vaccination centers all around India were resulting in chaos and panic. Ruby Hall, a hospital round the corner from where we live, had a line that snaked for more than a hundred meters outside the facility.

For my EE assignment, I would like to utilize my understanding of mathematics to design a vaccination center such that the chance of overcrowding is minimized. Furthermore, I will explore whether critical resources can be optimized in the vaccination center's design. The mathematics I will deploy as part of my vaccination center design is the theory of Markov Chains. The inputs to my optimal design will be:

a) rate at which people enter the center for vaccinations,

b) the rate at which these people are vaccinated, and

c) the cost of each serving station (a proxy for scarce healthcare resources)

The constraints to the design are:

a) the numbers of seats in the waiting room of the vaccination center, and

b) the confidence level to which overcrowding is to be avoided.

Finally, the output of my design will be the optimal number of serving centers needed and the methodology for deploying these serving stations. The data collection for the inputs is based on a field visit to Ruby Hall Hospital in Koregaon Park, Pune. The analysis of the data is used to design a vaccination center as part of a corporate vaccination drive being conducted by Ruby Hall Hospital.

2 Research Question:

The topic I would like to research for my Mathematics EE is "Can the theory of Markov Chains be utilized to optimize a vaccination center in which the use of scarce healthcare resources is minimized while maximizing the probability of low wait times for patients?"

3 Background:

When I began making this EE back in summer of 2021, vaccines were still not largely available on the market. Now, when I am submitting this paper, the supply of vaccines in the area has increased. The optimal vaccination center design case is a queuing problem where people enter the queue at a random but predictable rate and these people get serviced also at a random yet predictable rate. The length of the queue is defined by the buffer that is available, essentially waiting area seats at the vaccination center. Any overflow of this buffer is termed as overcrowding and this overcrowding is a defect that is to be avoided. To model this process using mathematics, an understanding of Queuing Theory and Markov Chains is needed.

3.1 Queuing Theory:

Queuing theory is the mathematical investigation of waiting in queues or lines. It can be used to model any issues that may arise from coinciding access to a shared resource. In this case, that shared resource is a Covid vaccine center. The mathematics utilized in the theory can model the blocking probability or the average delay in a system, for example, while also keeping in mind dimension resources such as the number of service counters, or the size of waiting buffer.[1]

The characterization of the arrival of customers in this system, in other words, the way customers enter the system, can be modeled using a Poisson process. This means that arrival events occur in a random fashion, are independent of each other, and are exponentially distributed. Clients that arrive must wait in a buffer, or seating area if all the service counters are occupied. We must also specify the scheduling policy which is the order in which customers can access the service counters. As described by Jean Walrand and Pravin Varaiya in their paper about High-Performance Communication Networks, there are various policies that systems can use to manage the input and outflow of 'patients' in a system. The policy used in this system, also the most common one, is known as the FIFO Policy (First in First Out).[2] In this system, λ is the arrival rate which can be denoted as the average number of customer arrivals per unit of time. Similarly, the service rate (μ) can be denoted as the average number of customers that can be serviced by the system per unit of time.

1 2

[1] Mansa, Julius. Queuing Theory, The Investopedia Team, 25 Apr. 2021, www.investopedia.com/terms/q/queuing-theory.asp.

[2] Walrand, Jean, and Pravin Varaiya. "Chapter 9 - Control of Networks: Mathematical Background." High-Performance Communication Networks, Second ed., Morgan Kaufmann, 2000.

3.2 Markov Chains:

According to MIT Professor, John Tsitsiklis' lecture on Markov Chains, Markovian modeling is a branch of mathematics that allows for the representation of probabilistic interactions for statistically random processes[3]. The process to be modeled is set as a series of conditional interactions, hence the use of the word "chain". The transition from one state of the chain to other is determined by some probability distribution. The probability distributions must satisfy the Markov property for the overall process to be classified as a Markov Chain. The unique property of Markov Chains is that the transition from one state to another is memoryless – i.e. the probability of going from one state to next depends only on a time invariant probability distribution. This distribution is not dependent on how the process reached a particular state.

The memory-less property of the exponential distributed can be modeled mathematically[4]:

$$P(X_{n+1} = x \mid X_1 = x_1, X_2 = x_2, .., X_n = x_n)$$

$$\therefore P(X_{n+1} = x \mid X_n = x_n)$$

Markov chains are models used to represent the random flow of a variable through a discrete and defined set of possible outcomes. Although there are two types of Markov chains (discrete-time queues, and continuous-time queues), because the situation of vaccine center requires us to record the state of the system at every unit of time, we will solely be looking at discrete time Markov chains. In DTMCs, the state of the variable in the queue is recorded per unit of time; for example: at time = 5, the system is at state 5 (there are 5 people in the vaccine center), and at time = 6, the system is at state 7 (there are 7 people in the center $\rightarrow$ from minute 5 to 6, 2 people entered the center). The chain will remain in the same state for at least one unit of time before making any transition (a change in state). Again, the essential characteristic that makes this process stochastic and Markovian, is that the future state, given the present state, is completely independent of the past. To begin, I will be discussing Markov chains that are time-homogeneous. This means that all transition probabilities are the same at any time.

3 4

Being a part of queuing theory, Markov chains are utilized from engineers to computer scientists to optimize situations like the progression of calls that a telephone network carries. There is a

[3]Tsitsiklis, John. Probabilistic Systems Analysis and Applied Probability MIT Lecture 16, Markov Chains 1. Youtube, MIT OpenCourseWare, 2011, https://youtu.be/IkbkEtOOC1Y. Accessed 27 Jan. 2022.

[4]Rocca, Joseph. Introduction to Markov Chains, Towards Data Science, 25 Feb. 2019, towardsdatascience.com/brief-introduction-to-markov-chains-2c8cab9c98ab.

randomness in the state of a call and the ability to place one that can be analysed using discrete-time Markov chains.

4 Experiment and Data:

I sat in the waiting room for a COVID-19 vaccine administration center and counted how many people were there. To begin with, there were 7 people in the room. For the next hour (60 minutes), in intervals of 5 minutes, I recorded how many people left the waiting room and how many people entered.

Waiting Area Data			
Time Interval	Enter	Exit	Total People in System
0:00 - 4:59	6	9	4
5:00 - 9:59	10	7	7
10:00 - 14:59	4	5	6
15:00 - 19:59	6	5	7
20:00 - 24:59	6	8	5
25:00 - 29:59	9	6	8
30:00 - 34:59	6	7	7
35:00 - 39:59	7	9	5
40:00 - 44:59	9	4	10
45:00 - 49:59	6	2	14
50:00 - 54:59	6	4	16
55:00 - 60:00	6	8	14

From the table above, we can look at the 'Enter' and 'Time Interval' columns in order to calculate the arrival rate (λ) for this system.

$$Arrival\ Rate: \sum_{0:00-4:59}^{55:00-60:00} \frac{\text{Enter}}{60} = \frac{6+10+4+6+6+9+6+7+9+6+6+6}{60} = \frac{81}{60} = \boxed{1.35}$$

An average arrival rate of 1.35 means that every minute, 1.35 people will enter the vaccine center. Now that we have derived the arrival rate, the rate at which people or cases enter the system, we must now find the service rate, which is the rate at which people leave the system. There are two service counters in the vaccine center. Therefore, the final derived service rate will be based on if there are servers in the system. In order to find μ, we must look at the data collected after the waiting area. The data can be seen in the tables below.

Service Counter 1			
Trial	Min	Seconds	Total Time (sec)
1	2	33.05	153.05
2	2	14.12	134.12
3	2	32.89	152.89
4	2	13.04	133.04
5	2	23.37	143.37
6	2	37.71	157.71
7	2	37.96	157.96
8	1	49.75	167.11
9	2	22.98	142.98
10	2	43.87	163.87

Service Counter 2			
Trial	Min	Seconds	Total Time (sec)
1	2	16.39	136.39
2	2	13.41	133.41
3	2	21.23	141.23
4	2	14.06	134.06
5	2	37.58	157.58
6	2	48.83	168.83
7	2	21.35	141.35
8	2	3.12	123.12
9	2	25.72	145.72
10	2	19.65	139.65

From the two tables above (each representing a service counter), we can look at the 'Trial' and 'Total Time' columns in order to calculate the service rate (μ) for this system. In the below formula, the first parenthesis signifies the total time taken for the 10 trials in service counter 1, and the second parenthesis signifies the total time taken for the 10 trials in service counter 2.

$$\text{Service Rate} : \frac{(\sum_{\text{Trial 1}}^{\text{Trial 10}} \text{ Total Time}) + (\sum_{\text{Trial 1}}^{\text{Trial 10}} \text{ Total Time})}{20} = \frac{2962.96}{20} = 148.148 \text{ seconds/service}$$

This answer, however, is in seconds per service, and needs to be converted to people serviced in the system per minute. It also represents the average for one server, and not the entire current system (which has two servers). Therefore, the following calculations must be made to find the service rate:

(converting to minutes) $\frac{148.148}{60} = 2.47 \longrightarrow$ (for two servers) $\frac{2.47}{2} = 1.234 \longrightarrow$ (converting to people

served per minute) $\frac{1}{1.234} = \boxed{0.81}$ people served per minute

Therefore, 0.81 is the service rate (μ) for this system.

4.1 Data Analysis:

Knowing the arrival rate, the service rate, and their associated distributions are the only two critical parameters required for system design. A steady state Markov Chain has probabilities associated with each state of the system. In a system with a buffer of 10, the states would range from 0 in buffer (no one waiting) to 10 in buffer (10 people waiting) and anything above 10 being called out as overflow (crowding). **The vaccination center is a system with a waiting room of 10 people (buffer capacity) and patient arrival rates and patient service rates as per the data collected in Section 4. The 1^{st} goal of data analysis is to design a system where there is less than 5% chance of overflow in an 8 hour operating period.** To achieve this goal, the steps to be followed are:

1) Calculate jump probabilities from one state to another. The arrival rate, the service rate, and the associated distribution of these processes is used to calculate the jump probability (Section 5.1).

2) The Birth-Death process logic is used to model end state probabilities of any Markov Chain. It is proved that a system where arrival rate is faster than service rate, the system is unstable with an undefined end state. A transition matrix defines the jump state probability from any one state to all other states. (Section 5.3)

3) Analysis from a) and b) is combined to calculate end state probabilities of all possible states. The end state probability from State 0 to State 10 is added to calculate the total probability that the end state does not overflow. (Section 5.4)

4) The number of service stations are increased to the point that the total probability of overflow is less than 5%. (Section 5.6)

5 Investigation

5.1 Jump Probability (Poisson Distribution) Calculation:

A literature study of arrival patterns indicates that such patterns are best modelled as a Poisson process. In a Poisson process, the interarrival time between two consecutive customers is an exponential with parameter λ. The exponential distribution is "memoryless", meaning that the time it takes for the next patient to arrive is independent of how long ago the previous patient arrived. The other requirement for a Poisson process is that only one customer arrives at any one instant (the next customer can be an infinitesimal time interval later). The arrival rate at a vaccination rate meets both these properties and hence can be modelled as a Poisson process. Similarly, the departure rate or service rate can also be modelled as an exponential distribution.

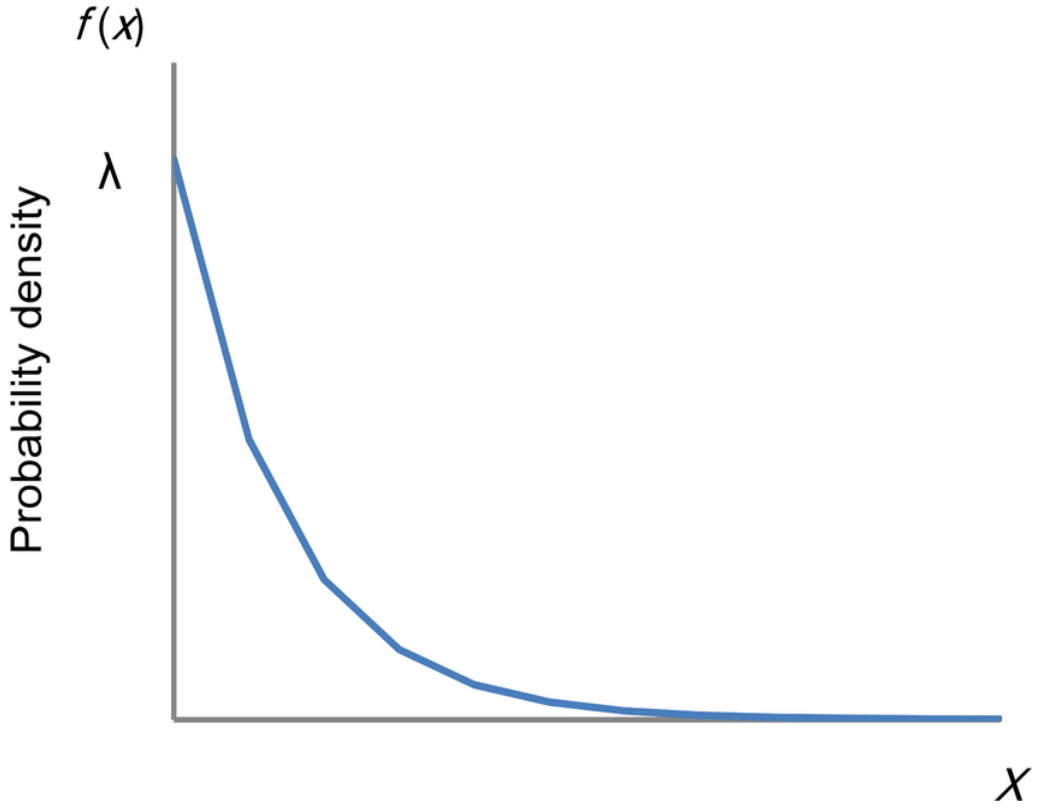

Figure 1: Poisson process distribution graph[4]

[5] From the data collected in Section 4, the values of λ and μ are derived. To convert the arrival

[5]Ide, Kazuki. Distribution of an Exponential Distribution. If a Random Variable, x Is Exponentially Distributed, f (x) = e x for x 0 Where Is the Rate Parameter. The Graph of an Exponential Distribution Starts on the y-Axis at a

and service rates into the jump probability that x patients enter or leave in one minute, the following mathematics is utilized:

A. Set a sufficiently small interval as U and the probability of one change in that interval as λU. And the probability of two or more changes in a system in a sufficiently short interval is essentially 0.

B. Let x denote the number of changes in an interval of 1 minute. If we portion the unit interval into U sub-intervals of length $\frac{1}{U}$, we will be able to find the probability that one change occurs in each of exactly 'x' of these U sub-intervals.

C. The probability of one change occurring in any one sub-interval is $\lambda \cdot \frac{1}{U}$ or $\frac{\lambda}{U}$.

D. Therefore, if we represent each occurrence as a Binomial trial in Bernoulli's Distribution, with 'p' representing $\frac{\lambda}{U}$ and 'q' representing $(1 - \frac{\lambda}{U})$, then $P(X = x) = \binom{n}{x} \cdot (p)^x \cdot (q)^{U-x}$.

E. U must also tend to ∞. This is so that the intervals of $\frac{1}{U}$ can become as small as possible.

$$P(X = x) = \frac{U!}{x! \cdot (U - x)!} \cdot (\frac{\lambda}{U})^x \cdot (1 - \frac{\lambda}{U})^{U-x}, \lim_{U \to \infty}$$

Using algebra, we can take the $x!$ from the denominator of the first expression and place it in the denominator of the second expression, $\frac{\lambda^x}{U^x}$. Then, take the U^x from that second expression, and place it in the denominator of $\frac{U!}{(U-x)!}$. And finally, using the rules of exponents, we can separate $(1 - \frac{\lambda}{U})^{U-x}$ into $(1 - \frac{\lambda}{U})^U \cdot (1 - \frac{\lambda}{U})^{-x}$ because of their like bases. All of these changes gives the following equation:

$$P(X = x) \underset{\text{U} \to \infty}{\lim} = (\frac{U \cdot (U-1) \cdot ... \cdot (U-x+1)}{U^x}) \cdot (\frac{\lambda^x}{x!}) \cdot (1 - \frac{\lambda}{U})^U \cdot (1 - \frac{\lambda}{U})^{-x}$$

If we split this equation up into four parts, we can simplify it easier.

①$\Longrightarrow \frac{U \cdot (U-1) \cdot ... \cdot (U-x+1)}{U^x}$

②$\Longrightarrow \frac{\lambda^x}{x!}$

③$\Longrightarrow (1 - \frac{\lambda}{U})^U$

④$\Longrightarrow (1 - \frac{\lambda}{U})^{-x}$

---

①$\underset{\text{U} \to \infty}{\to lim}$ $(\frac{U \cdot (U-1) \cdot (U-2) \cdot ... \cdot (U-x+1)}{(U) \cdot (U) \cdot (U) \cdot ... \cdot (U) \cdot}) = [1 \cdot (1 - \frac{1}{U}) \cdot (1 - \frac{1}{U}) \cdot (1 - \frac{2}{U}) \cdot (1 - \frac{x-1}{U}) \cdot] = 1$

Positive Value () and Decreases to the Right. . Research Gate, Feb. 2017, www.researchgate.net/figure/Distribution-of-an-exponential-distribution-If-a-random-variable-x-is-exponentially$_f ig1_3$13477966.

$$\text{Now,} \lim_{\text{U} \to \infty} P(X = x) = (1) \cdot (\tfrac{\lambda^x}{x!}) \cdot (1 - \tfrac{\lambda}{U})^U \cdot (1 - \tfrac{\lambda}{U})^{-x}$$

$$\text{③} \to \lim_{\text{U} \to \infty} (1 - \tfrac{\lambda}{U})^U = e^{(log((1-\frac{\lambda}{U})^U))} = e^{((U \cdot log(1-\frac{\lambda}{U})} = e^{\lim_{U \to \infty}(U \cdot log(1-\frac{\lambda}{U}))}$$

$= e^{\lim_{U \to \infty} \frac{log(1-\frac{\lambda}{U})}{1/U}}$ Because this limit gives us an indeterminate expression, we need to use L'Hospital's Rule:

$$\lim_{\text{U} \to \infty} \frac{\frac{d}{dU} log(1-\frac{\lambda}{U})}{\frac{d}{dU}(\frac{1}{U})}$$

$$\lim_{\text{U} \to \infty} = (\frac{\frac{\lambda}{U^2 \cdot (1-\frac{\lambda}{U})}}{-\frac{1}{U^2}}) = -\frac{\lambda \cdot U}{U - \lambda}$$

$$\therefore e^{(\lim_{U \to \infty}(-\frac{\lambda \cdot U}{U-\lambda}))} = e^{(-\lambda \cdot (\lim_{U \to \infty} \frac{U}{U-\lambda}))} = e^{(-\lambda \cdot (\lim_{U \to \infty} \frac{1}{\frac{\lambda}{U}}))}$$

$\therefore \frac{\lambda}{U}$ tends to 0 as U approaches ∞ and $\lim_{U \to \infty}(1 - \frac{\lambda}{U})^U = e^{-\lambda}$

---

④$\Longrightarrow$ And for the final part of the equation, $(1 - \frac{\lambda}{U})^{-x}$, as U tends to ∞, this part of the equations equals 1 as $1^{-x} = 1$

$$\therefore \lim_{\text{U} \to \infty} P(X = x) = (1) \cdot (\tfrac{\lambda^x}{x!}) \cdot (e^{-\lambda}) \cdot (1) = \tfrac{\lambda^x \cdot e^{-\lambda}}{x!}$$

With this final equation, $\boxed{P(X = x) = \dfrac{\lambda^x \cdot e^{-\lambda}}{x!}}$, we can now find the probability of x states entering or x states exiting the system, given an arrival rate λ or a service rate μ. From the preliminary data that I collected at the vaccine center, it can be calculated that $\boxed{\lambda = 1.35 \text{ people/minute}}$ and $\boxed{\mu = 0.81 \text{ people/minute (for 2 serving stations)}}$. $\lambda > \mu$ signifies that the rate of incoming people is larger than the rate of outgoing people. While this will be proven below using mathematics, one can see that if states in a system increase at a rate faster than they decrease in a system, that system will soon overflow.

By utilizing the derived values of arrival rate and service rate, as well as the equation for a Poisson process with given state x, we can find the probability that there are x state entries or x state departures at a given time.

The probability of arrivals in a given state can be denoted as $\boxed{p(x) = \dfrac{\lambda^x \cdot e^{-\lambda}}{x!}}$

The probability of departures from the system at a given state can be denoted as $\boxed{p(x) = \dfrac{\mu^x \cdot e^{-\mu}}{x!}}$

The first 6 iterations of each arrival and each departure has been shown below. The purpose of doing this is to show the probability of x arrivals and x departures in a given time.

For x=0 | p(0) = $\frac{\lambda^0 \cdot e^{-\lambda}}{0!} = e^{-\lambda} = 0.259$

For x=1 | p(1) = $\frac{\lambda^1 \cdot e^{-\lambda}}{1!} = p(0) \cdot \lambda = 0.349$

For x=2 | p(2) = $\frac{\lambda^2 \cdot e^{-\lambda}}{2!} = \frac{p(1) \cdot \lambda}{2} = 0.236$

For x=3 | p(3) = $\frac{\lambda^3 \cdot e^{-\lambda}}{3!} = \frac{p(2) \cdot \lambda}{3} = 0.106$

For x=4 | p(4) = $\frac{\lambda^4 \cdot e^{-\lambda}}{4!} = \frac{p(3) \cdot \lambda}{4} = 0.036$

For x=5 | p(5) = $\frac{\lambda^5 \cdot e^{-\lambda}}{5!} = \frac{p(4) \cdot \lambda}{5} = 0.0097$

For x=0 | p(0) = $\frac{\mu^0 \cdot e^{-\mu}}{0!} = e^{-\mu} = 0.445$

For x=1 | p(1) = $\frac{\mu^1 \cdot e^{-\mu}}{1!} = 0.360$

For x=2 | p(2) = $\frac{\mu^2 \cdot e^{-\mu}}{2!} = 0.146$

For x=3 | p(3) = $\frac{\mu^3 \cdot e^{-\mu}}{3!} = 0.039$

For x=4 | p(4) = $\frac{\mu^4 \cdot e^{-\mu}}{4!} = 0.008$

For x=5 | p(5) = $\frac{\mu^5 \cdot e^{-\mu}}{5!} = 0.0013$

As one can see, the probability that there are 0 departures from the system is higher than that same probability for arrivals. But the probability that there are two arrivals is 0.236 and probability that there are two departures is 0.146. With numbers like this, the system will soon overflow with arrivals. There are some systems in which some amount of overflow is tolerable. In a bank teller service for example, which is responsible for processing routine operations like cashing checks, depositing money and collecting loan payments, it is acceptable for data entries to enter faster than they leave, to some extent. The same is not the case for our situation. A vaccine center must be efficient while also accepting the maximum amount of patients.

Because the vaccine center stays open for 8 hours a day, and these probabilities are represented on a per-minute basis, I ran these permutations in a simulated system for 480 (8 x 60) iterations, 10 times. These 10 iterations will be represented by the 10 colors in the graphs below. I did this for a an unstable system, in which the arrival rate, λ, was higher than the service rate, μ. As well as a system in which the arrival rate was equal to the service rate, and finally, a stable system, in which the arrival rate was lower than the service rate.

In order to graph this for iterations and time, I had to create a code and input the transition probability values for λ and μ. The code works by declaring a variable 'p' equal to 0 signifying the total cases, as well as declaring a variable 'n' equal to 0 signifying the time in minutes. Then, using an in-build java function from the mathematics library called 'Math.random()', a random number between 0 and 1 is generated. I then multiply this number 1000 and see check where it lies in the probability ranges for λ and μ. If the random number lies in the ranges for arrival rate, then the number of states in the system increases and is added to variable p, and is subtracted from the number of cases in the system (p) if done for service rate. At the end of this cycle, n is equated to $n + 1$ and is looped until the condition: $n < 480$, is false. This runs the code for 480 iterations

while printing the total number of cases in the system for each minute. The code and graphs for each are shown below.

```
public class markov
{
    public static void main(String[] args)
    {
        int loop = 0;
        while(loop<10){
            int p=0;
            int n = 0;
            while(n<480){
                int rand = (int)(Math.random() * 1000);
                if (0 < rand && rand <= 258)
                {p=p;}

                if (259 <= rand && rand <= 607)
                {p=p+1;}

                if (608 <= rand && rand <= 845)
                {p=p+2;}

                if (846 <= rand && rand <= 952)
                {p=p+3;}

                if (953 <= rand && rand <= 989)
                {p=p+4;}

                if (990 <= rand && rand <= 1000)
                {p=p+5;}


                rand = (int)(Math.random() * 1000);
                if (0 < rand && rand <= 444)
                {p=p;}

                if (445 <= rand && rand <= 805)
                {p=p-1;}

                if (806 <= rand && rand <= 952)
                {p=p-2;}

                if (953 <= rand && rand <= 992)
                {p=p-3;}

                if (993 <= rand && rand <= 998)
                {p=p-4;}

                if (999 <= rand && rand <= 1000)
                {p=p-5;}

                if(p<0)
                {p=0;}
                System.out.println(p);
                n=n+1;
            }
            loop=loop+1;
        }
    }
```

Figure 2: Current system simulation code using BlueJ

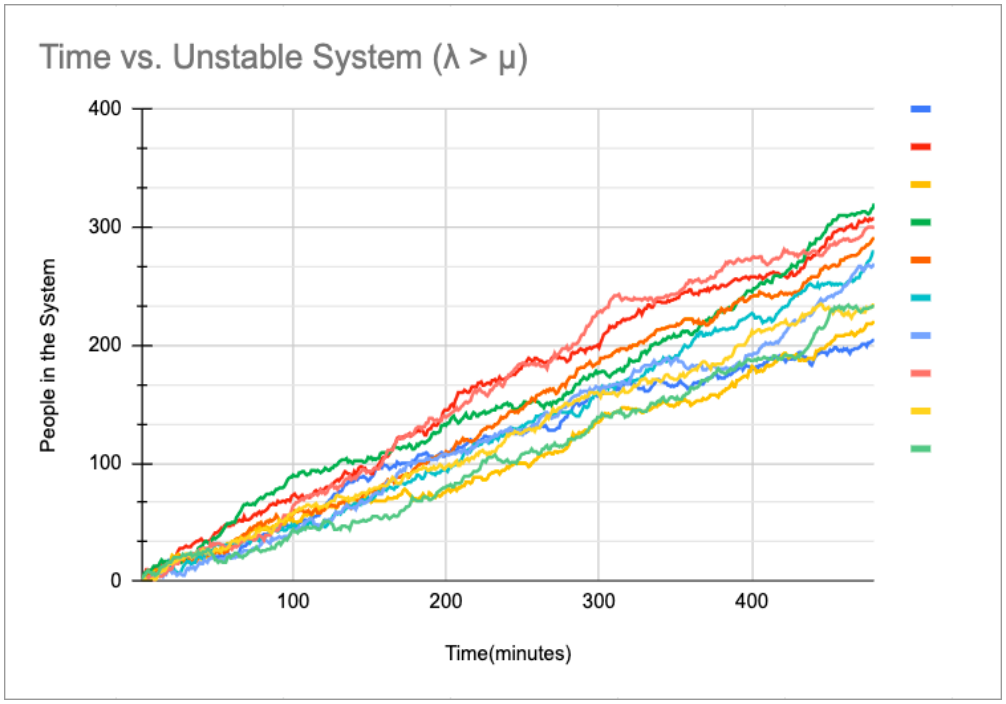

Figure 3: Hospital's current system ($\lambda > \mu$)

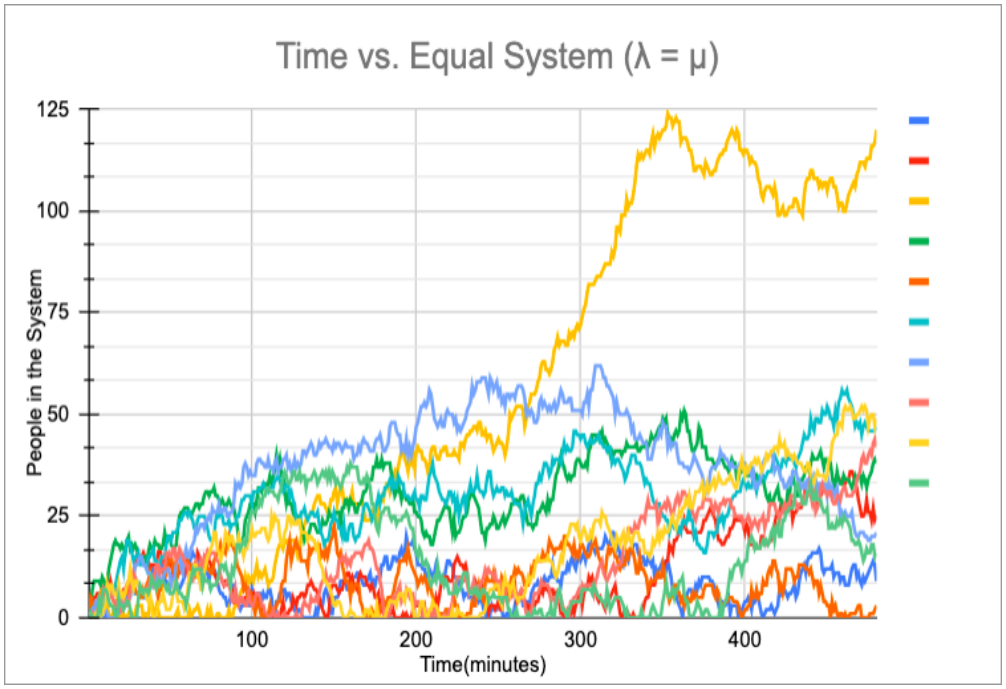

Figure 4: Arrival rate = Service rate ($\lambda = \mu$)

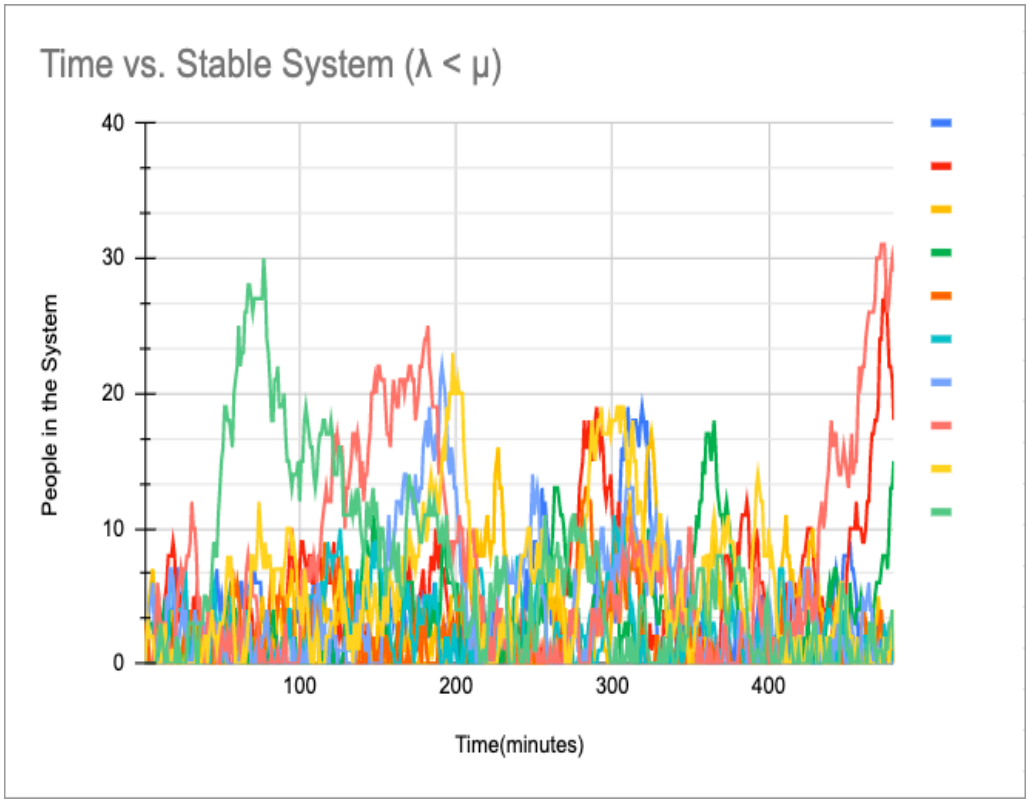

Figure 5: Stable system ($\lambda < \mu$)

5.2 Types of systems:

In the hospital's current system, Figure 3, all ten simulations left at least 200 individuals in the queue waiting to be served by the hospital at the end of the 8 hours of service. This proportion of arrival rate and service rate means that states in the system increase much faster than they decrease. From Figure 4, one can see that even in a system in which the arrival rate is equal to the service rate, the state went as high as 125 individuals in the system for 1 simulation out of the 10, and up to 50 individuals for the other simulations. These numbers are far too high for a hospital to effectively manage. For some days, the hospital maybe be able to work overtime in order to satisfy the number of individuals, but in the long run, this puts pressure on the components of system: arrival counters, service stations, doctors, nurses, etc. Systems where λ is higher than μ, or where they are equal, will always be unstable to some extent, but exactly how high the states can go, can be determined by the proportion between the arrival rate and the service rate ($\frac{\lambda}{\mu}$), also known as ρ. Because the system represents a Coronavirus vaccine center, the only way the arrival rate can be influenced, is if the center limits or reduces the limits for the number of registrations a day. They can also create a limited buffer or waiting area and deny any individuals that attempt to enter after the buffer is full. These decisions will make the system more stable but they will also drastically slow the rate that people in this specific area can be vaccinated at. During a pandemic, where cases can rise exponentially in just a matter of days, it is important to act fast and flatten the curve. That is why it is better for the center to increase their μ value and make their system more efficient, rather than implementing policies that reduce the λ value and slow entries and the rate of vaccination. That is why λ will remain as 1.35, but the value of μ, as well as other characteristics of system, will be modified.

5.3 Steady State Analysis for a Birth Death Process:

In order to determine at what state a system will stabilize at (how many individuals will be in the system when time is ∞) we must analyse the steady state for the birth death process. A birth death process is where the flow of states in a system is measured by a progression of change in time, where that dt or Δt is infinitesimally small. By setting the change in time to an infinitesimally small amount, we can make the assumption that any given time, the only possible transitions in the system are one death, one birth, or no change at all. This means that the only state transitions are from state n to state $n+1$, from state n to state $n-1$, or from state n to state n. For Example, in a birth death process which is currently at state 5, the only possible transitions are to state 4,

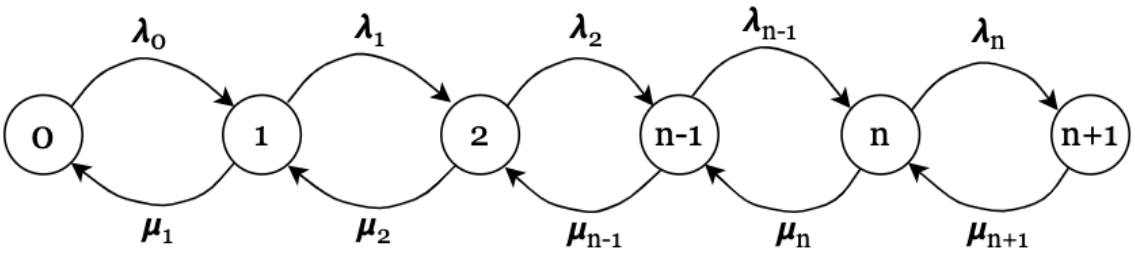

Figure 6: Flow diagram for birth-death process

5, or 6. Because of this assumption, we are also allowed to say that the probability of two or more changes in the system is essentially 0. In order for a system to have a steady state behavior, its characteristics must meet the stability state condition. The condition states that the arrival rate λ, must be less that the service rate μ. The queue will only be stable if the traffic intensity or load, ρ or $(\frac{\lambda}{\mu})$, is less than 1.

The transitions from state to state for the system can be represented below using the state, balanced equations, and the derived relationship between P_n and P_{n-1}, where P_0 and P_1 signify the probabilities that the system is at state 0 and state 1 respectively. λ and μ are rates to go to and from states. For example, λ_1 signifies the arrival rate going out of the first state of the system, and μ_3 means the service rate going out state 3. Using this, the steady state can be derived as shown:

State	Balance Equation	Relation
0	$\mu_1 P_1 = \lambda_0 P_0 \rightarrow$	$P_1 = \frac{\lambda_0}{\mu_1} \cdot P_0$
1	$\lambda_0 P_0 + \mu_2 P_2 = \lambda_1 P_1 + \mu_1 P_1 \rightarrow$	$P_2 = \frac{\lambda_1}{\mu_2} \cdot P_1$
n	$\lambda_{n-1} P_{n-1} + \mu_{n+1} P_{n+1} = (\lambda_n + \mu_n) \cdot P_n \rightarrow$	$P_n = \frac{\lambda_{n-1}}{\mu_n} \cdot P_{n-1}$

Based on the diagram above for the birth-death discrete-time markov chain, the transition rate matrix Q, can be seen below.

$$Q = \begin{array}{c} \\ (0) \\ (1) \\ (2) \\ (i-1) \\ (i) \\ (i+1) \end{array} \begin{array}{c} \begin{array}{cccccc} (0) & (1) & (2) & (i-1) & (i) & (i+1) \end{array} \\ \begin{bmatrix} -\lambda & \lambda & & & & \\ \mu & -(\lambda+\mu) & \lambda & & & \\ & \mu & -(\lambda+\mu) & \lambda & & \\ & & \mu & -(\lambda+\mu) & \lambda & \\ & & & \mu & -(\lambda+\mu) & \lambda \\ & & & & \mu & -(\lambda+\mu) \end{bmatrix} \end{array}$$

Using the relationship between P_n and P_{n-1}: ($P_n = \frac{\lambda_{n-1}}{\mu_n} \cdot P_{n-1}$), a normalization that sums the probabilities from state n to ∞ can be derived.

$$\text{NORMALIZATION} : \sum_{n=0}^{\infty} P_n = P_0 \cdot (1 + \frac{\lambda_0}{\mu_1} + \frac{\lambda_0\lambda_1}{\mu_1\mu_2} + \frac{\lambda_0\lambda_1\lambda_2}{\mu_1\mu_2\mu_3} + ...)$$

However, if the arrival rate between each state stays the same (as it does in this vaccine center's case), and the service rate does not alter with respect to the state (no change in the number of service counters during the period of operation), then λ_0, λ_1, λ_2, and λ_n, can be written as λ, and μ_0, μ_1, μ_2, and μ_n, can be written as μ.

$\frac{\lambda}{\mu} = \rho \rightarrow P_1 = \rho P_0 \rightarrow P_2 = \rho P_1 = \rho^2 P_0 \rightarrow P_3 = \rho P_2 = \rho^2 P_1 = \rho^3 P_0 ... \; P_0 + P_1 + P_2 + P_3 + P_4 + ...$

$= (1 + \rho + \rho^2 + \rho^3 + \rho^4 + ...)P_0$

$\rightarrow 1 = \frac{P_0}{1-\rho} \rightarrow P_0 = 1 - \rho \rightarrow \boxed{P_n = \rho^n(1-\rho)}$

With this equation, we can now calculate the steady state probabilities of each system. Going back to my research question and my case of the vaccine center specifically, this equation can be used to extrapolate the service rate required by the hospital to keep the the number of people within the buffer limit.

5.4 95% Steady State for 10 individuals:

With ρ less than 1, we have determined that each state will have a steady behavior which can be found using the equation: $P_n = \rho^n(1-\rho)$. We can now compute the minimum number of servers to ensure, with a 95% chance, that the number of people in the system at any given time will be 10 or below. This can be done by equating the sum of 0 people to 10 people to 0.95%:

1. $\sum_{n=0}^{n=10} \rho^n(1-\rho)$
2. $(1-\rho) + \rho(1-\rho) + \rho^2(1-\rho) + \rho^3(1-\rho) + \rho^4(1-\rho) + \rho^5(1-\rho) +$

$\rho^6(1-\rho) + \rho^7(1-\rho) + \rho^8(1-\rho) + \rho^9(1-\rho) + \rho^{10}(1-\rho)$

3. $\rightarrow (1+\rho+\rho^2+\rho^3+\rho^4+\rho^5+\rho^6+\rho^7+\rho^8+\rho^9+\rho^{10})(1-\rho) = 0.95$

Using algebra, we can derive the value of ρ.

4. $\therefore \rho = 0.762 = \frac{\lambda}{\mu} = \frac{1.35}{\mu} \rightarrow \mu = 1.78$

We know that when there are two servers in the system, $\mu = 0.81$. So, with one server $\mu = 0.405$. $\frac{1.78}{0.405} = 4.4$ servers. But because the number of servers in a system needs to be an integer value, we can round to 4.4 to where 5 servers gives a μ value of $5 \cdot 0.405 = 2.025$ people/min. Created using the simulation, the graph for 10 iterations in which the system has a $\mu = 2.025$ and a $\lambda = 1.35$ can be seen below.

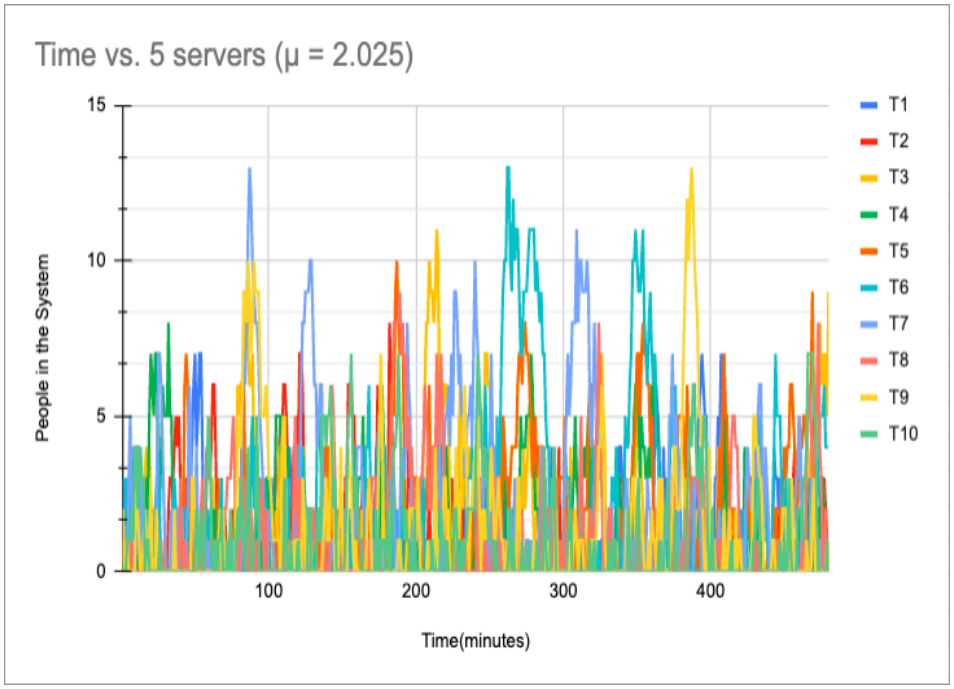

Figure 7: Time vs. 5 servers ($\mu = 2.025$)

For a couple minutes, in 4 of the trials, the number of people in the system went as high as 13, but due to the high service rate, retreated back down to 10, or less than 10, people in the system.

5.5 Cost of Service:

The design chosen above with 5 servers assures the 1^{st} goal of 95% probability of no overcrowding at the vaccination center. The problem with this design is that it also corresponds to high degree of underutilization of servers. In the simulations run in BlueJay, a Java ide, 17865 idle minutes across servers in aggregate were observed over 10 simulations. Although each simulation had an average

of 1787 idle minutes per simulation, the highest utilization rate for a simulation was a mere 35%. The servers are healthcare workers that are in very short supply and hence, a high utilization rate is required. To model this constraint for optimizing utilization of servers, a concept of cost of service is introduced. The server is artificially allocated a cost of Rs. 100₹/minute/server. The 5 server simulation then has a cost of Rs. 240,000₹. The next section explores the opportunity to reduce this cost by adding a constraint to the system developed above.

5.6 Optimizing Servers:

We know that with 5 servers, there is a 95% chance that the number of individuals in the system will not surpass 10. But what if it is not feasible for this specific organization to constantly implement 5 service counters. It would be much more cost-effective to initially have three service counters, and implement three more servers into the system when the state rises to 5 individuals or above. This way, when there are 4 or less people in the system, there are three service counters can be deactivated rather than idle.

Accomplishing this task means that the system's service rate changes from 3μ to 6μ when the state transitions from 4 people to any higher state in the discrete time interval. The earlier processes were all time-homogeneous, which means that all transition probabilities from any state are the same at any time. But for this optimization process, the transition probabilities is not the same for all times and depends on the current and future states. Below, the diagram for the flow of states and the transition rate matrix, Q, are shown.

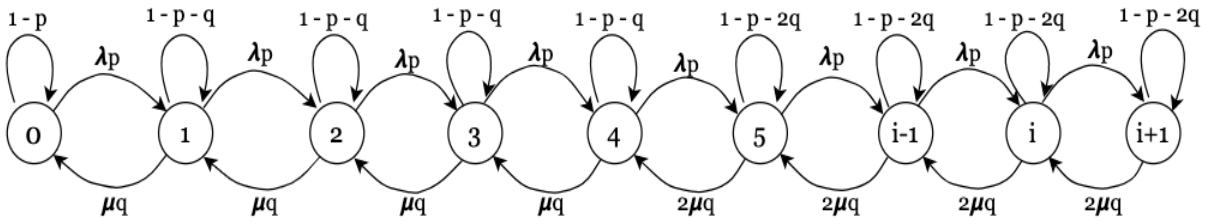

Figure 8: Flow diagram for 3 servers to 6 servers ($\mu \to 2\mu$)

$$
Q = \begin{array}{c} \\ (0) \\ (1) \\ (2) \\ (3) \\ (4) \\ (5) \\ (i-1) \\ (i) \end{array}
\begin{array}{c}
\begin{array}{cccccccc} (0) & (1) & (2) & (3) & (4) & (5) & (i-1) & (i) \end{array} \\
\begin{bmatrix}
-\lambda & \lambda & & & & & & \\
3\mu & -(\lambda+3\mu) & \lambda & & & & & \\
 & 3\mu & -(\lambda+3\mu) & \lambda & & & & \\
 & & 3\mu & -(\lambda+3\mu) & \lambda & & & \\
 & & & 3\mu & -(\lambda+3\mu) & \lambda & & \\
 & & & & 6\mu & -(\lambda+6\mu) & \lambda & \\
 & & & & & 6\mu & -(\lambda+6\mu) & \lambda \\
 & & & & & & 6\mu & -(\lambda+6\mu)
\end{bmatrix}
\end{array}
$$

$\lambda P_0 = 3\mu P_1 \rightarrow \frac{P_1}{P_0} = \frac{\lambda}{3\mu} = \frac{\rho}{3}$ however, for ease of calculation, we will equate $\frac{\rho}{3}$ with ρ and replace it back at the end $\therefore \frac{\lambda}{3\mu} = \rho \rightarrow$ ①

$\lambda P_1 + 3\mu P_1 = \lambda P_0 + 3\mu P_2 \rightarrow \frac{P_2}{P_1} = \frac{\lambda}{3\mu} = \rho \rightarrow$②

$6\mu P_5 + \lambda P_3 = 3\mu P_4 + \lambda P_4 \rightarrow$ ③

$6\mu P_6 + \lambda P_4 = 6\mu P_5 + \lambda P_5 \rightarrow$ ④

③$\rightarrow \frac{P_5}{P_4} = \frac{\lambda}{6\mu} = \frac{\rho}{2}$

④$\rightarrow \frac{P_6}{P_5} = \frac{\lambda}{6\mu} = \frac{\rho}{2}$

Using the Normalizing Condition: $\sum_{P_0}^{\infty} = 1 \rightarrow (P_0 + P_1 + P_2 + P_3 + P_4 + P_5 ... P_\infty) = 1$

With $\sigma = \frac{\rho}{2}$, $P_0 + \rho P_0 + \rho^2 P_0 + \rho^3 P_0 + \rho^4 P_0 + \sigma\rho^4 P_0 + \sigma^2\rho^4 P_0 ... \sigma^\infty \rho^4 P_0 = 1$

$= P_0[1 + \rho + \rho^2 + \rho^3 + \rho^4 + \sigma\rho^4(1 + \sigma + \sigma^2 + \sigma^3 + ...\sigma^\infty)] = 1$

$= P_0[1 + \rho + \rho^2 + \rho^3 + \rho^4 + \frac{\sigma(\rho)^4}{1-\sigma}] = 1$

$= P_0[1 + \rho + \rho^2 + \rho^3 + \rho^4 + \frac{\frac{\rho}{2}(\rho)^4}{1-\frac{\rho}{2}}] = 1$

$= P_0[1 + \rho + \rho^2 + \rho^3 + \rho^4 + \frac{\rho^5}{-\rho+2}] = 1$

By plugging in $\frac{\lambda}{3\mu} = \frac{1.35}{1.215}$ as ρ and by using algebra, we can find the value of P_0 to be 0.122. When the is plugged back into $\frac{P_1}{P_0} = \frac{\lambda}{3\mu} = \frac{\rho}{3}$, the value of P_1 can be derived to be 0.136. Using the same method, the service probabilities of the system being at other states can be derived.

$P_0 = 0.122$, $P_1 = 0.136$, $P_2 = 0.151$, $P_3 = 0.168$, $P_4 = 0.187$, $P_5 = 0.103$, $P_6 = 0.057$, $P_7 = 0.032$, $P_8 = 0.018$, $P_9 = 0.009$, $P_{10} = 0.005$. Using these probabilities, the system can be modeled as shown below.

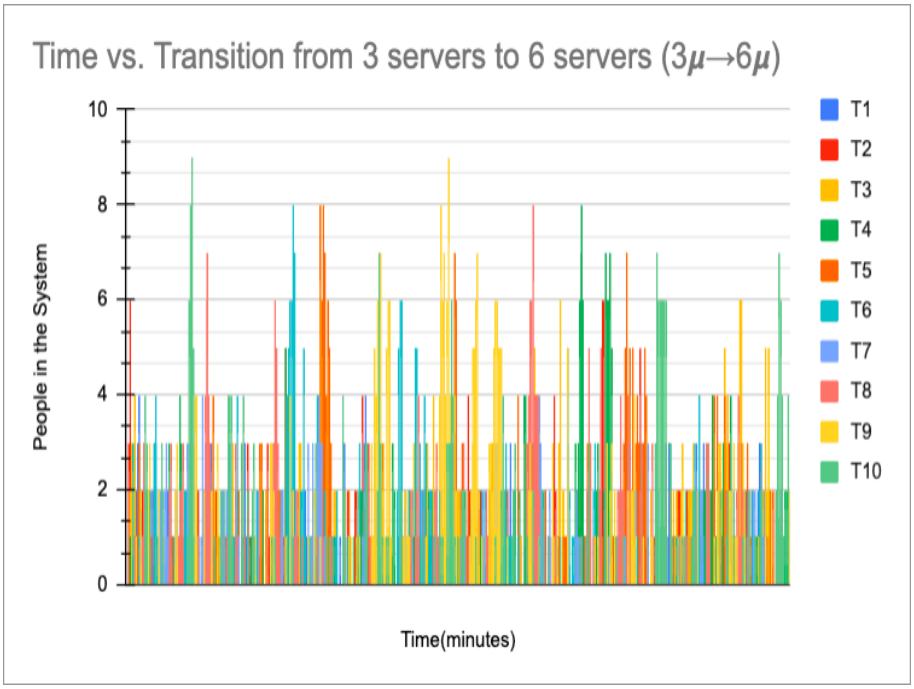

Figure 9: Transition from 3 servers to 6 servers ($3\mu \rightarrow 6\mu$)

This system is much more cost effective than the 5 server system and also does not underutilize servers. Because the 3 extra servers are added when the number of people in the system surpasses 4, in the 10 iterations of the simulation, the system never passes a state of 9 people.

6 Limitations & Improvements:

The study presented above has several areas of improvement that can be categorized as data collection improvement, experiment design improvement, and improvements in modelling.

6.1 Data Collection Improvement

A) Arrival rate data was collected only over a 1-hour period. This may not be sufficient as there may be patterns associated with time of day, day of week, etc. A larger data set will improve analysis.

B) The arrival rate of groups should be separately collected as this data can be relevant for a more complex model.

C) Server data was averaged across the two servers measured. Mapping more servers will provide understanding of performance variance between servers. Also, the impact of training, fatigue, etc. is not measured.

6.2 Experiment Design Improvement

A) I designed my experiment as a simple 1 stage Markov Chain with arrival followed by service. In practice, the chain is more complicated. Arrival is followed by registration followed by vaccination followed by waiting again for safety considerations. In my observation, the registration process was much faster than the vaccination process and hence the vaccination event was the service event chosen for design. A 2 stage Markov chain with a shared buffer would be a more appropriate system design.

B) Also, the system was designed with a fixed buffer that corresponded to the number of seats available in the waiting area. A secondary buffer with people waiting outside, with all safety precautions, may be possible.

6.3 Improvements in Modelling

A) The biggest drawback of my analysis is that in my analysis of Poisson probabilities, I have neglected the probability of greater than one arrivals or departures. This probability, while much smaller than P(0) + P(1), is not insignificant and the model should be modified suitably to take these probabilities into consideration. My computer simulation aligns well with my predicted behaviour and hence, I chose to neglect the probabilities of greater than 1 departures or arrivals. In general, taking these higher probabilities in account will increase the variance in my results and, in turn, increase the probability of overflow.

B) The Markov model I have employed is a discrete model where changes to the system only happen at fixed intervals (e.g. 1 minute). A continuous Markov Chain would be a better model for the system wherein changes to the system can happen at any instant along a continuous time interval. A Poisson process lends itself better to a continuous Markov model. The significantly higher mathematical complexity of a continuous Markov Chain led to the current choice of a discrete Markov chain.

C) The birth death process is modelled as only one arrival or one departure at any one instant. However, group arrivals are possible – e.g. a family coming in a car for vaccinations. Future models can incorporate this probability as well in the transition matrix.

D) The optimization chosen in Section 5.6 is just potential optimization scenario. A mathematical model to find the most optimized solution can also be explored.

7 Conclusion & Extension:

Based on the vast number of improvements listed above, it would greatly interest me to continue studying the application of Markov Chains in different social situations and systems in university. My learning from building this optimization model has several other applications. The direct connection is to other queuing applications such as super market lines. Birth-death processes can be extended to disparate fields ranging from epidemiology, including COVID spread, to evolution. The larger space of Markov Chains has relevance to all fields with "spread" phenomenon including social media and networking. As has been shown, Markov Chains can indeed be utilized to optimize a vaccination center wherein use of scarce healthcare resources is minimized while maximizing probability of low wait times for patients. My analysis indicates that, based on the data collected, that a vaccination center that expects about 750 patients in an 8-hour day, with a waiting room of 10 seats, needs 5 stations to assure a 95% chance of no overcrowding. Alternatively, the vaccination center with the ability to add or remove stations as per demand can choose to work with 3 servers but then add 3 additional servers each time 5 of the seats in the waiting room are occupied. These additional stations can be removed when the waiting room occupancy falls below 5. Several enhancements are possible to improve the accuracy of the solution by implementing improvements in data collection, system design, and modeling techniques.

8 Bibliography and Further Research Links for Citation:

Works Cited

Ide, Kazuki. Distribution of an Exponential Distribution. If a Random Variable, x Is Exponentially Distributed, f (x) = e x for x 0 Where Is the Rate Parameter. The Graph of an Exponential Distribution Starts on the y-Axis at a Positive Value () and Decreases to the Right. . Research Gate, Feb. 2017, www.researchgate.net/figure/Distribution-of-an-exponential-distribution-If-a-random-variable-x-is-exponentially-fig1-313477966.

Tsitsiklis, John. Probabilistic Systems Analysis and Applied Probability MIT Lecture 16, Markov Chains 1. Youtube, MIT OpenCourseWare, 2011, https://youtu.be/IkbkEtOOC1Y. Accessed 27 Jan. 2022.

Mansa, Julius. Queuing Theory, The Investopedia Team, 25 Apr. 2021, www.investopedia.com/terms/q/queuing-theory.asp.

Rocca, Joseph. Introduction to Markov Chains, Towards Data Science, 25 Feb. 2019, towardsdatascience.com/brief-introduction-to-markov-chains-2c8cab9c98ab.

Walrand, Jean, and Pravin Varaiya. “Chapter 9 - Control of Networks: Mathematical Background.” High-Performance Communication Networks, Second ed., Morgan Kaufmann, 2000.

Vaton, Sandrine, et al. Queuing Theory: from Markov Chains to Multi-Server Systems, EdX, 18 Oct. 2017, 11:42am, learning.edx.org/course/course-v1:IMTx+CS101+2T2021/home.