A simulation of a hospital queuing system – Case study, Thika Level Five Hospital

Article *in* International Journal for Engineering Modelling · May 2016

1 author:

Alfred Mutai
Jomo Kenyatta University of Agriculture and Technology

SIMULATION OF A HOSPITAL QUEUING SYSTEM

Case Study – Thika Level Five Hospital

Author

Kipchumba Mutai Alfred

ABSTRACT

Healthcare is essential to the general welfare of society. It provides for the prevention, treatment, and management of illness and the preservation of mental and physical well-being through the services offered by medical and allied health professions. Hospitals crowding causes a series of negative effects, e.g. medical errors, poor patient treatment and general patient dissatisfaction.

In this study the research has compared the existing prediction models and come up with Monte Carlo Simulation model to predict the number of patients in the queue. The model uses Poisson distribution on arrival and exponential distribution on service time. The model is constructed using R program where after running, it generate random numbers. After several experiments the model has proved to be very accurate and efficient. This will assist the hospital to utilize the resources and reduces cost of operations.

Contents

CHAPTER ONE

1.0 INTRODUCTION

1.1 Background

Healthcare is essential to the general welfare of society. It provides for the prevention, treatment, and management of illness and the preservation of mental and physical well-being through the services offered by medical and allied health professions. Today, the issue of healthcare is receiving much attention through the media and politics.

Healthcare is faced with unprecedented challenges, such as staffing shortages an aging population rising costs and inefficient hospital processes. In light of these challenges, a need for review and reform of our healthcare practices has become apparent. Lean is one way in which overcrowding in Emergency department can be addressed.

The basic concept of lean is using less to do more .For healthcare, in particular, one can apply the principles of lean thinking to improve such processes as patient wait time, levels of staffing, and quality of care. Improvements to such processes can greatly impact the health of the community.

The burden on the provision of services can be reduced by model predictive control which will be predicting the number of patients queue. Hospital administrators can also use model predictive control to predict the future resident patient number in each department/ward.

1.2 Problem Statement

Overcrowding in hospitals due to over flow of patients and limited number of doctors to serve the patients.

Overcrowding causes a series of effects such as erroneous diagnosis, poor patient outcomes and patient dissatisfaction and on the extreme: death!!

Over staffing of doctors results in high expenditure in paying for their services, Thus there is need to determine an optimum number of doctors based on projected number of patients.

1.3 Proposed Solution

In this study the research has compared the existing models and come up with Monte Carlo Simulation methods to predict the number of patients in the queue. The simulation model has been chosen in preference to other models because

(i) It allows probabilistic Results - Results show not only what could happen, but how likely each outcome is.

(ii) Sensitivity Analysis - It is easy to see which inputs had the biggest effect on bottom-line results.

(iii) Correlation of Inputs - It is possible to model interdependent relationships between input variables.

(iv) Graphical Results - Because of the data it generates it is easy to create graphs of different outcomes and their chances to occurrence which is important for communication findings to other stakeholders.

1.4 Justification

With Kenya's population growing at a rate of 3 percent annually, the population will continue to place a huge demand for health services. Kenya must continue expanding maternal and child health services while developing the capacity of the health systems to cater for communicable and non-communicable diseases which are on the rise. The Government has committed itself to improving the health Sector infrastructure. Attaining acceptable standards and norms has implications for staffing, equipment, infrastructure, and operating costs.

1.6 Objective

1.6.1 General Objective

The main goal is to develop queuing predictive model by studying and comparing existing predictive models in the hospitals.

1.6.2 Specific Objectives

1. To study predictive models used by hospitals.
2. To identify the short comings associated with the queuing predictive model.
3. To develop a predictive model to enable the hospital predicts the number of patients in the queue.
4. To propose predictive model that will utilize the resources and reduce cost of operations in the hospitals.

1.7 Scope of study

1. The objective is to study the predictive models.
2. The research covers queue models with probabilistic input in a dynamic system.
3. The research uses First-in first-out (FIFO) queuing discipline.

CHAPTER TWO

2.1 Literature Review

2.2.2 Discrete-event simulation and queuing theory

Discrete Event Simulation

In DES, the operation of a system is represented as a chronological sequence of events. Each event occurs at an instance in time and marks a change of state in the system. The modeled system is dynamic and stochastic. DES includes Clock, Events List, Random Number Generators, Statistics and Ending Condition. (K.S. Trivedi, 2001).

For example, in the process that patients wait for a bed in the ward, the system states are queuing length or number of vacant beds. The system events are patients-arrival and patients-departure. The system states, like vacant beds are changed by these events. The random variables that need to be characterized to model this system stochastically are patient arrival time and residence time. To simulate such system, first generate a series of random entities based on the distribution. Let (n, t) be n patients coming into the station at time t. Then all the incoming patients during $(t_1, t_2 = t_1 + dt, t_3 = t_2 + dt, \dots., t_k)$ can be expressed as $\{(n_1, t_1), (n_2, t_2), \dots \dots., (n_k, t_k t_1)\}$. Here $n_1, n_2 \dots .. n_k$ are random numbers. dt is constant. The simulator generates service rate for each patient, $l_1, l_2 \dots .. l_k$ which are random numbers. All the random numbers obey a certain distribution. The patients leave the ward when the residence time is over. The simulator stores all the data. The patient number and other results can be obtained by analyzing the saved data. Such as to compute the resident patient number at time t_i , the simulator find out the patients that time t_i is between this patients' arrival and departure time.

There are several advantages to build such models (J. Banks, & J. Carson, 2005). Detailed system behavior can be modeled; It is possible to model the performance, dependability; Less matrices computing.

2.2.3 Stochastic Simulation and Poisson distribution

Modeled daily bed occupancy variability using stochastic simulation (Harrison et al, 2001). They found that a flexible bed allocation scheme resulted in fewer overflows with the same level of occupancy compared to a fixed bed allocation. Their model also proves that variable discharge rates are more significant than variable admission rates in contributing to overflows. Numerous investigators have also used regression models to analyses needed bed capacity.

A common statistical assumption in modeling count data is that it follows a Poisson distribution. (Dexter et al, 2000) assumed that the number of patients staying in the OB unit is Poisson distributed.

2.2.7 Queuing Theory with Markov Chain (QTMC), and Discrete Event Simulation (DES)

Hidden Markov models (HMMs) have been used in various fields, ranging from Bioinformatics to Storage Workloads (Harrison et al, 2012). HMMs were first used in the late 1960s in statistical papers by Leonard E. Baum for statistical inference of Markov chains (Baum et al, 1966) and also for statistical estimation of Markov process probability functions (Baum et al, 1967). Speech recognition became a field for training HMMs in the 1970s and 1980s (Rabiner et al, 1986), with many such speech models still used today (Ashraf et al, 2010)

A hidden Markov model (HMM) is a probabilistic model (a bivariate Markov chain) which encodes information about the evolution of a time series. The HMM consists of a hidden Markov chain {Ct} (where t is an integer) with states not directly observable and a discrete time stochastic process $\{Ot\}t\geq0$,which is observable. Combining the two, we get the bivariate Markov chain $\{(Ct, Ot)\}\ t\geq0$

The first model (QTMC) is only able to consider limited scenarios that can occur. DES has been well recognized in healthcare and is broadly used for the validation of other models. The DES models offer a valuable tool to study the trade-off between the capacity structure, sources of variability and patient flow times (Arnoud M, 2007)

The Hospital arrivals model was found to train successfully on patient arrivals, collected over months of analysis. The means and standard deviations matched well for raw and HMM-generated traces and both traces exhibited little autocorrelation. HMM parameters, fully converged after training, were used to predict the model's own synthetic traces of patient arrivals, therefore behaving as a fluid input model (with it's own rates). An enhancement could be to assume instead that the arrival process is Poisson, with corresponding rates, and produce a cumulative distribution function for the patient arrivals workload.

2.5 Application of Monte Carlo Simulation

Monte Carlo Simulation has existed before and has been applied in various fields.

(i) In the field of computer engineering and design, (Bhanot et al, 2005) described the use of simulation when optimizing the problem layout of IBM’s Blue Gene ® / L supercomputer.

(ii) In geophysical engineering, Monte Carlo analysis has been used to predict slope stability given a variety of factors (El-Ramly et al, 2002)

(iii) In marine engineering, (Santos et al, 2005) described a probabilistic methodology they have developed to assess damaged ship survivability based on Monte Carlo simulation.

(iv) Monte Carlo simulation in aerospace engineering to geometrically model an entire spacecraft and its payload, using The Integral Mass Model (Lei et al, 1999).

(v) In public health, simulation has been used to estimate the direct costs of preventing Type 1 diabetes using nasal insulin if it was to be used as part of a routine healthcare system (Hahl et al, 2003).

(vi) Monte Carlo simulation should be used by research organizations to determine whether or not future possible research is really worth the cost and effort, by modeling possible outcomes of the research (Phillips, 2001)

(vii) Monte Carlo simulation in personal financial planning, especially when estimating how much money one needs for retirement and how much one can spend annually once retirement has begun (Boinske, 2003).

2.6 Approach of Monte Carlo Simulation

Monte Carlo Simulation has been used in processor performance to predict the Cost Performance Index of in-order architecture and validate it against the Itanium-2 (Ram Srinivasan & Jeanine Cook, 2006). The research will come up with model design to estimate patients demand in the hospitals which will use arrival time, waiting time and service time. It will also use Poisson rule and exponential distribution to facilitate how Monte Carlo Simulation will work.

In order to facilitate Monte Carlo Simulation, the research will consider the simple multi-server queuing model as *M/M/c/∞*. Suppose arrival time fit Poisson distribution and service time to obey exponential distribution. The research has implemented the Monte Carlo Simulation using R programming.

Poisson distribution

$$= P(x;\lambda)\frac{e^{-\lambda}\lambda^{x}}{x!} \qquad \text{for } x = 0, 1,2,\ldots\ldots$$

Exponential distribution

$$= f(x) = \int_0^x \frac{1}{\beta} e^{-t/\beta} dt$$

$$= 1 - e^{-x/\beta}$$

$$\text{For } x \geq 0, t \geq 0$$

CHAPTER THREE

3.0 METHODOLOGY

3.1 Introduction

This chapter presents the description of the methods that were adopted in the study. Here efforts were made to discuss the following:

- Method of Data Collection
- Measure of queue length
- Queuing models Based on the Birth and Death Process

a) Using m/m/s queuing model in measuring system performance in Thika Level 5 Hospital and Kenyatta National Hospital.

b) Formulating Priority – Discipline Queuing Models in Measuring System Performance in Thika Level 5 Hospital and Kenyatta National Hospital.

c) Formulating State – Dependent Service Rate Model in Measuring System Performance in Thika Level 5 Hospital and Kenyatta National Hospital.

3.2 Method of data collection

The basic data used for this study is secondary data consisting of recorded information on the arrival times of the patients and the service time for patient. The instruments used for the data collection are recorded sheets from the hospitals.

The data collection span for a single day i.e Monday from 8am to 2.30 pm. The days were preferred because on Monday the queue is always long. Also most of the times they stop issuing cards at 2.30 pm.

The data collected was based on Thika Level hospital.

3.3 Measures of queue length

Those measures are:

1) The number of patients waiting in line which is collected after an interval to enable us to construct a model. The researcher decided to collect the data after every 30 minutes. The data are collected from arrival recording book from the consultant receptionist.
2) The time patient is served which is considered constant for all patients and that is five minute per patient.

3.4 Queuing models based of birth-death process

Waiting lines (queue) are a direct result of arrival and service variability. They occur because random bunched arrivals and highly variable service patterns cause systems to be temporarily overloaded. In many instances, the variability can be described by theoretical Poisson distribution for arrival time and negative exponential distribution for the service time.

In the context of queueing theory, the term birth refers to the arrival of a new patient into queueing system, and death refers to the departure of a served patient.

3.5 The M/M/S model

The M/M/c Queue

We now illustrate the ideas introduced in this chapter with the use of an example. Consider the M/M/c queue where the arrival and service rates are λ and μ, respectively. Assuming that steady state exists let p_n be the steady state distribution of the number of units in the system. We proceed to derive the equations involving $p_n$by using the rate-equality principle, which states that the rate at which a process enters a state is equal to the rate at which it leaves that state.

Consider state 0, when there are no units in the system. The process can leave this state only when there is an arrival, which causes the system to transition to state 1. The long-run proportion of time the process is in state 0 is p_0, and since λ is the rate of arrival, the rate at which the process leaves state 0 to go to state 1 is λp_0. Moreover, the process can enter state 0 only from state 1 through a departure or service completion. Since the proportion of time the process is in state 1 is p_1 and the rate of leaving state 1 through service completion is μ, the rate at which the process transitions from state 1 to 0 is μp_1. Using the rate-equality principle, we get

$$\lambda p_0 = \mu p_1 \qquad \text{3.5.1}$$

Now consider state 0 < n < c. The process can leave state n in two ways, either through an arrival or through a departure. The proportion of time the process is in state n is p_n and the total rate at which the process leaves state n through arrivals or departures is $\lambda p_n + n\mu p_n$ since there are n servers busy (additive property of the Poisson process). The process can enter state n in two ways, either through arrival from state n - 1 or through a departure from state n + 1. Thus, the rate at which the process enters state n is $\lambda p_{n-1} + \mu p_{n+1}$. By the rate-equality principle

$$\lambda p_n + n\mu p_n = \lambda p_{n-1} + (n+1)\mu p_{n+1} \qquad \text{3.5.2}$$

Similarly, for the case of n ≥ c, we get

$$\lambda p_n + c\mu p_n = \lambda p_{n-1} + c\mu p_{n+1} \qquad \text{3.5.3}$$

Repeated application of (3.5.2) along with (3.5.3) at the last step yields

$$\begin{aligned} \lambda p_n - (n+1)\mu p_{n+1} &= \lambda p_{n-1} - n\mu p_n. \\ &= \lambda p_{n-2} - (n-1)\,\mu p_{n-1}. \\ &\;\vdots \\ &= \lambda p_0 = \mu p_1 \end{aligned}$$

= 0.

By rearranging terms and iterating we obtain that for 0 < n ≤ c

$$p_n = \frac{\lambda/\mu}{n} p_{n-1} = \frac{(\lambda/\mu)^2}{n(n-1)} p_{n-2} = \cdots = \frac{(\lambda/\mu)^n}{n!} p_0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots 3.5.4$$

In a similar fashion, we get that for n > c

$$p_n = \frac{(\lambda/\mu)^n}{c!c^{n-c}} p_0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 3.5.5$$

Now for $\lambda/(c\mu) < 1$, the normalization condition $\sum_{n=0}^{\infty} p_n = 1$ gives

$$p_0 = \left[\sum_{n=0}^{c-1} \frac{(\lambda/\mu)^n}{c!(1-\lambda/c\mu)} + \right] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 3.5.6$$

We now proceed to compute some performance measures. The expected queue length L can be computed as

$$L = \sum_{n=c}^{\infty}(n-c)\, p_n = \frac{\lambda p_c}{\mu(1-p)^2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 3.5.7$$

Where $p = \lambda/c\mu$ is referred to as the server utilization. Applying formula, we also obtain the expected waiting time in the queue

$$W = \frac{L}{\lambda} = \frac{p_c}{\mu(1-p)^2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 3.5.8$$

Knowing the probability distribution, we can now directly compute the pgf of the number in the queue

$$P(z) = \sum_{n=0}^{c-1} p_n + \sum_{n=c}^{\infty} p_n Z^{n-c} = 1 - \frac{p_c}{1-\rho} + \frac{p_c}{1-\rho^z} \ldots\ldots\ldots\ldots\ldots 3.5.9$$

Which allows us to find the LT of the wait time distribution as

$$w^*(s) = P\left(1 - \frac{s}{\lambda}\right) = 1 - \frac{p_c}{1-\rho} + \frac{p_c}{1-\rho+s/c\mu} \ldots\ldots\ldots\ldots\ldots\ldots\ldots 3.6.0$$

The m/m/s queue is a model with parameters inter-arrival time and the service time that is Poisson and exponentially distributed respectively. The queue discipline here is First – Come, First – Served (FCFS). The space for the waiting line is infinite size.

Consequently, this model is just the special case of the birth-and-death process where the queuing systems mean service rate per busy server is constant.

This information is then used by Health Managers to decide on an appropriate level of service for the facility. The basic objective in most queuing models is to achieve a balance between two costs; cost of offering the service and cost of delay in offering the service.

3.6 Priority – discipline queuing models

In priority – discipline queuing models, the queue discipline is based on a priority system. Thus, the order in which patients of the queue are selected for service is based on their assigned priorities. Many real queuing systems fit these priority – discipline models much more closely than other available models. Rush jobs are taken ahead of other jobs, and important patients may be given precedence over others. Therefore, the use of priority-discipline models often provides a very welcome refinement over the more usual queuing models. This model incorporates all of the assumptions of the basic multiple – server model and it uses FCFS.

3.7 A model with state – dependent service rate and/or arrival rate

All the models thus far have assumed that the mean service rate is always constant, regardless of how many patients are in the system. Unfortunately, this rate often is not a constant in real queuing systems, particular when the servers are people. When there is a backlog of work, it is quite likely that such servers will tend to work faster than they do when the backlog is small or none existent. This increase in the service rate may result merely because the servers increase their efforts when they are under the pressure of a long queue. However, it may also result partly because the quality of the service is compromised or because assistance is obtained on certain service phases.

3.8 Description of the patient

The patient arrival at the hospital at random and the queue discipline is first – come, first – served. The hospital under this study operates 24 hours and consultation is open for all patients that are appointment patients, same day appointment patients (walk-ins) and new patients. All patients have to go to the reception desk for submission of their hospital card and if necessary, for initial screening before consultant. Monte Carlo Simulation method was successfully used to describe the complexity and dynamics of patients flow.

3.9 Software in use in modeling

3.9.1 R programming

R is a programming language and software environment for statistical computing, and graphics. The R language is widely used among statisticians and data miners for developing statistical software and data analysis. R is an implementation of the S programming language combined with lexical scoping semantics inspired by Scheme. R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R. R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, …) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.

CHAPTER FOUR

4.0 PRESENTATION AND ANALYSIS OF DATA

4.1 Introduction

In this chapter, I shall present the data collected and use it as illustrative examples on the models.

4.2 Presentation of data

The basis of Monte Carlo simulation is experimentation on change (or probabilistic) elements by means of random sampling. Setting up a probability distribution of important variables.

(a) Building a cumulative distribution for patients in the queue.
(b) Establishing an interval of random numbers for each variable.
(c) Generating random number.
(d) Actually simulating a series of trials.

The distribution of arrivals and accumulated patient in the queue in Thika Level 5 hospital and Kenyatta National Hospital are given below where service time is assumed constant for all patients and thus five minute per patient. The research has used the two hospitals because of the number of patients they handle and the queue they experience.

4.3 Summary of tables

Table 1, 2, and 3 below are secondary data collected from the receptionist arrival patients recording book. The research decided to record at an interval of 30 minutes which is prior to change depending on how to compute the Monte Carlo simulation model. Also there is assumption of cumulative number of patients as doctor increases one by one until the queue diminishes in the last column. The research assumes three doctors is supposed to treat fifteen patients within 30 minutes, four doctors twenty patients, five doctors 25 patients and so on.

4.4 Arrival time and wait time

Table 1: Arrival table for collected on Monday at Thika Level 5 hospital

Inter Arrival Time	Arrival	Cumulative Patient in the queue assuming there are 3, 4, 5 doctors respectively.		
		3	**4**	**5**
8.00	**24**	**9**	**4**	**0**
8.30	**25**	**19**	**9**	**0**
9.00	**19**	**23**	**8**	**0**
9.30	**21**	**29**	**9**	**0**
10.00	**20**	**34**	**9**	**0**
10.30	**27**	**46**	**16**	**0**
11.00	**25**	**56**	**21**	**0**
11.30	**24**	**65**	**25**	**0**
12.00	**26**	**76**	**31**	**0**
12.30	**20**	**81**	**31**	**0**
1.00	**19**	**85**	**30**	**0**
1.30	**24**	**94**	**34**	**0**
2.00	**25**	**104**	**39**	**0**
2.30	**22**	**111**	**41**	**0**

The model has been computed to use arrival time and servers (doctors) which can keep on changing depending on how many doctors are in operation at that hospital.

Monte Carlo simulation is run for several times with arrival time remaining constant and number of doctor changing. The simulated data generated at random after several trials when the model runs for 1000 times with three slots, four slots, five slots and six slots are as follows. Thika Level 5 hospital table 1: 0, 15, 23, 23, 49, 52, 56, 60, 70, 90, 94, 100, 98, 109; 0, 7, 8, 8, 8, 14, 20, 28, 28, 30, 30, 35, 39, 42; 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 respectively. The data will be plotted against data collected from the hospital.

4.5 Performance of different waiting times

Below are results for different graphs for the two hospitals when data collected from hospital is plotted against simulated data from the model.

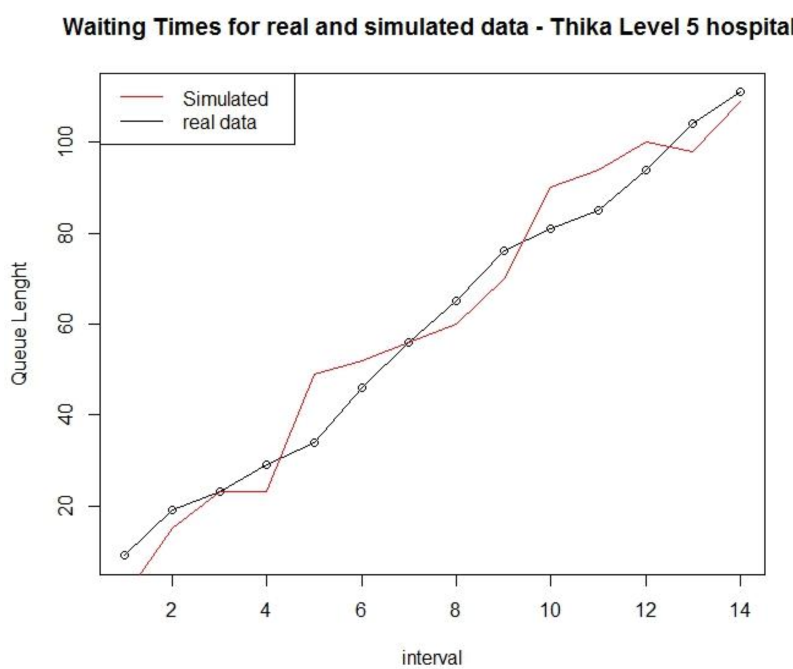


Figure 1: Thika Level 5 hospital on Monday for three slots

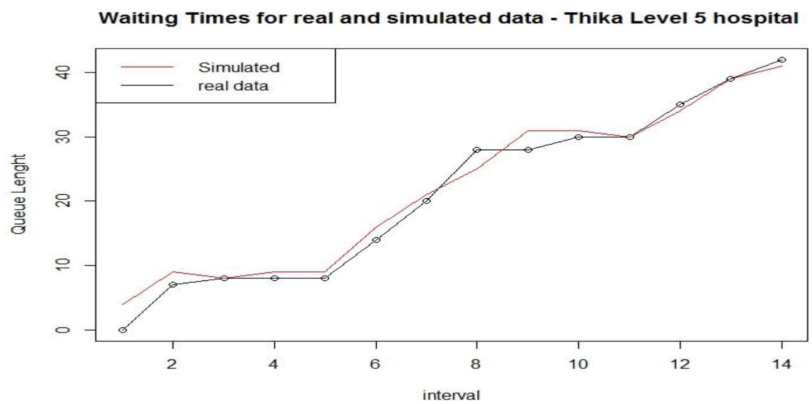


Figure 2: Thika Level 5 hospital on Monday for four slots

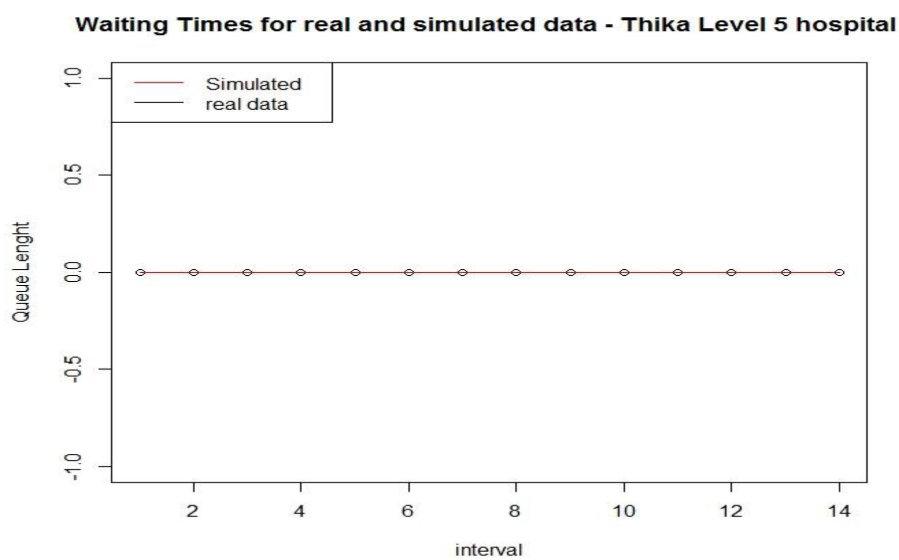
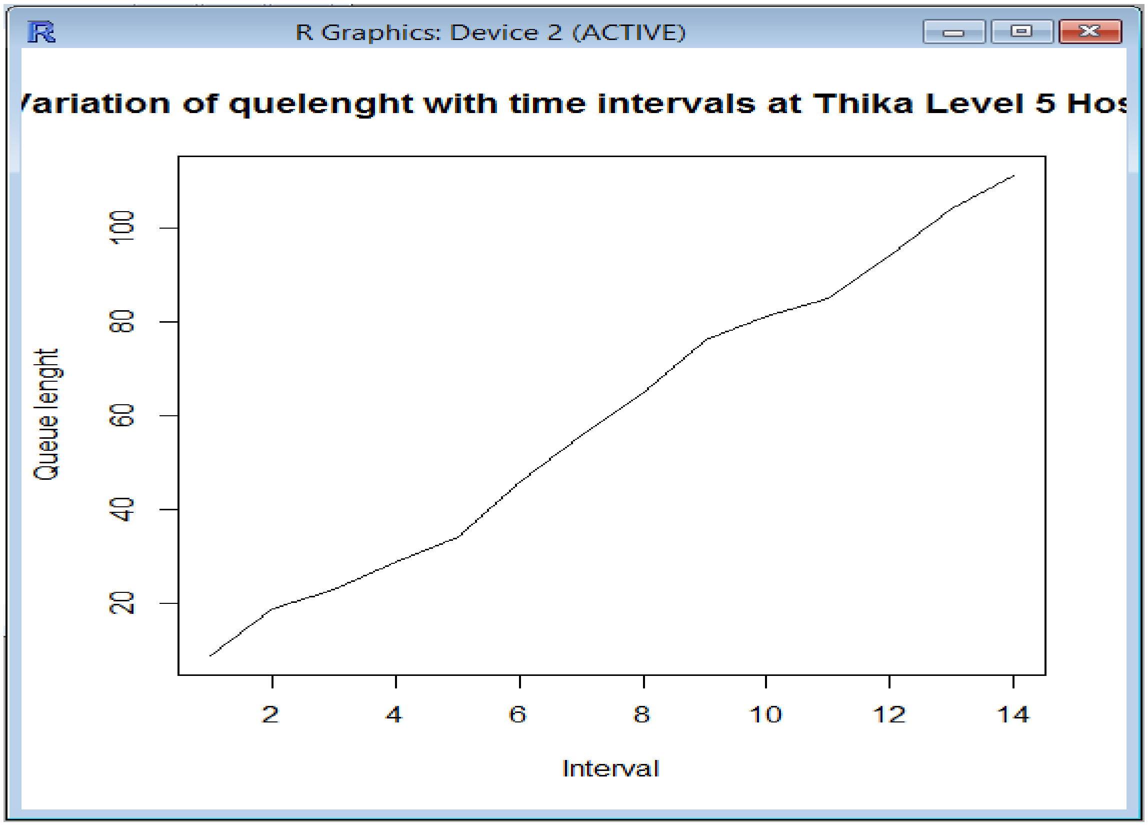
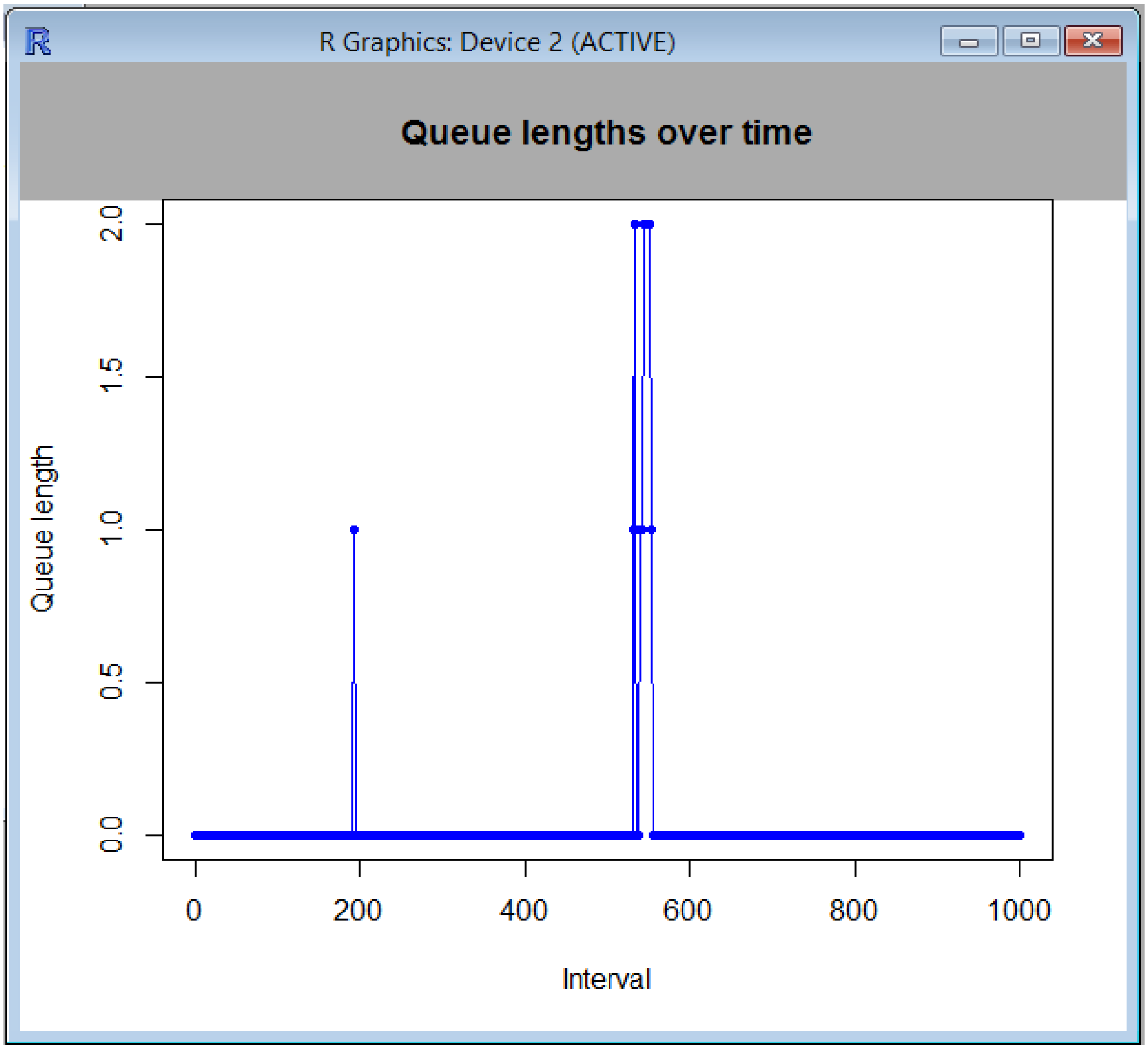


Figure 3: Thika Level 5 hospital on Monday for five slots


R Graphics: Device 2 (ACTIVE)
/ariation of quelenght with time intervals at Thika Level 5 Hos
Queue lenght
20
40
60
80
100
2
4
6
8
10
12
14
Interval


R Graphics: Device 2 (ACTIVE)
Queue lengths over time
Queue length
0.0
0.5
1.0
1.5
2.0
0
200
400
600
800
1000
Interval

CHAPTER FIVE

5.0 DISCUSSION

This chapter is aimed at discussing and summarizing the main findings from the study, drawing relevant conclusions and where necessary making some vital recommendations.

5.1 Discussion

Using computer simulation confirmed the description of service time assumption to be constant for all patients. Patient waiting is undesirable, limiting waiting time is an important objective hence simulation model and first – come, first - served priority model are being used in predicting the queue so as to reduce it.

On the basis of the results obtained from the plotted graph of three doctors from Thika Level 5 Hospital on Monday, the queue length is high as 111 patients waiting in the queue at 2.30pm. Additional of extra one doctor the queue length reduced to 41 patients at the same time 2.30pm. Adding another doctor to total five the queue length reduced to Zero patient all the time. The research concludes its optimal to have four doctors on Monday since five is a waste of resources and most of times the doctors will be inoperative.

According to the study the research has shown Monte Carlo is more accurate than other models since it runs several times which generate more numbers. When this numbers are picked at random and plotted against real or collected data they produce exact results. That is the two lines discrepancy is very small.

This shows and proves Monte Carlo simulation Model is more accurate than other Models which uses probability and generates numbers once.

5.2 Conclusion

As long as increasing the productivity of healthcare organizations remains important, analysts will seek to apply relevant models to improve the performance of healthcare processes. The research shows that many models for estimating waiting time and utilization are available today but not adopted in our country. However, analysts will increasingly need to consider the ways in which distinct queuing systems within an organization interact. Thus, this thesis surveys the contributions and application of queuing models in the field of health care with simulation models.

Experimental results show that Monte Carlo Model provides a high accuracy for the prediction of queue. This is because Monte Caro Model runs for several times and generate numbers randomly. It also uses Poisson distribution for arrival time and exponential distribution for service time. The research used random generated numbers and data collected from different hospital to plot the graphs which results shows the discrepancy is very minimal thus concluded Monte Carlo Simulation Model can be used to predict patient queue in the hospitals.

REFERENCES

American Nurses Association, I. (2010). *Nurse Staffing. Retrieved from;* USA*:* American Nurses Association.

A. M. de Bruin, A. C. van Rossum, M. C. Visser, G. M. Koole (2006) Modelling the emergency cardiac in-patient ow: an application of queueing theory, *Health Care Management Science*, 10(2), 125-137.

Arnoud M, a. G. (2007). *Bottleneck analysis of emergency in-patient flow.*

ARMONY, M. and Zacharias C., (2013)"Panel sizing and appointment scheduling in outpatient medical care".

Ashraf, J., Iqbal, N., Khattak, N. S., Zaidi, A. M. (2010) Speaker Independent Urdu Speech Recognition Using HMM

Averill M. Law (2007) Simulation Modeling and Analysis, Seiten; McGraw-Hill, 226

Asplin, B. R. (2003). A Conceptual Model of Emergency Department Crowding. *Annals of Emergency Medicine* , 173-179.

Asplin, B. R. (2002). Measuring emergency department crowding and hospital capacity. *Academic Emergency Medicine. (9)*, 366-367.

Asplin, B. R. (2000). Measuring emergency department crowding and hospital capacity. *Academic Emergency Medicine (9)*, 366-367.

Babbie, E. (2004). *The practice of social research 10th Ed.* Belmont CA: Wadsworth.

Banks, J. and Carson J. (2005). Discrete-event system simulation - fourth edition. New Yolk: Pearson.

Bazzoli, G. (2003). Does U.S. Hospital Capacity Need to be Expanded? *Health Affairs 22(6)* , 40-54.

Baum, L. E., Petrie, T. (1966) Stastical Inference for Probabilistic Functions of Finite Markov Chains, In The Annals of Mathematical Statistics, UK: University of Washington 37, 1554-63

APPENDIX

Appendix 1 : Monte Carlo Simulation Model

```
#Monte Carlo Simulation Model Code #
# Number of slots to fill assuming the line is long
numbSlots = 5

# Total time to track all patient in line
intervals = 1000

# Probability that a new patient will show up during an interval
# Note, a maximum of one new patient can show up during an interval
p = 0.1
 # Average time each patient takes at the slot, discretized exponential distribution assumed
# Times will be augmented by one, so that everyone takes at least 1 interval to serve
meanServiceTime = 30
#### INITIALIZATION ####
queueLengths = rep(0, intervals)
slots = rep(0, numbSlots)
waitTimes = c()
leavingTimes = c()
queue = list()
arrivalTimes = c()
frontOfLineWaits = c()

#### Libraries ####
# Use the proto library to treat people like objects in traditional oop
library("proto")
```

```
#### Functions ####
# R is missing a nice way to do ++, so we use this
inc <- function(x) {
  eval.parent(substitute(x <- x + 1))
}

# Main object, really a "proto" function
person <- proto(
	intervalArrived = 0,
	intervalAttended = NULL,

	# How much slot time will this person demand?
	intervalsNeeded = floor(rexp(1, 1/meanServiceTime)) + 1,
	intervalsWaited = 0,
	intervalsWaitedAtHeadOfQueue = 0,
)

#### Main loop ####
for(i in 1:intervals) {
	# Check if anyone is leaving the slots
	for(j in 1:numbSlots) {
		if(slots[j] == i) {
			# They are leaving the queue, slot to 0
			slots[j] = 0
			leavingTimes = c(leavingTimes, i)
		}
	}
```

```
	# See if a new patient is to be added to the queue
	if(runif(1) < p) {
		newPerson = as.proto(person$as.list())
		newPerson$intervalArrived = i
		queue = c(queue, newPerson)
		arrivalTimes  = c(arrivalTimes, i)
	}

	# Can we place someone into a slot?
	for(j in 1:numbSlots) {
		# Is this slot free
		if(!slots[j]) {
			if(length(queue) > 0) {
				placedPerson = queue[[1]]
				slots[j] = i + placedPerson$intervalsNeeded
				waitTimes = c(waitTimes, placedPerson$intervalsWaited)
				# Only interested in these if person waited 1 or more
intevals at front of line
				if(placedPerson$intervalsWaitedAtHeadOfQueue) {
					frontOfLineWaits = c(frontOfLineWaits,
placedPerson$intervalsWaitedAtHeadOfQueue)
				}

				# Remove placed person from queue
				queue[[1]] = NULL
			}
		}
	}
```

```
	# Everyone left in the queue has now waited one more interval to be served
	if(length(queue)) {
		for(j in 1:length(queue)) {
			inc(queue[[j]]$intervalsWaited) # = queue[[j]]$intervalsWaited + 1
		}

		# The (possibley new) person at the front of the queue has had to wait there one more interval.
		inc(queue[[1]]$intervalsWaitedAtHeadOfQueue)		# = queue[[1]]$intervalsWaitedAtHeadOfQueue + 1
	}

	# End of the interval, what is the state of things
	queueLengths[i] = length(queue);
}

#### Output ####
plot(queueLengths, type="o", col="blue", pch=20, main="Queue lengths over time", xlab="Interval", ylab="Queue length")
```