

**A BETTER NUMERICAL APPROACH FOR FINDING THE STEADY-
STATE WAITING TIME AND THE AVERAGE QUEUE LENGTH OF
A SYSTEM FOR THE ARITHMETIC GI/G/1 QUEUE.**

by

BONANE BHATTACHARJA

B.Sc., Mathematics, Khulna University, Bangladesh, 2005

M.Sc., Applied Mathematics, Khulna University, Bangladesh, 2006

A THESIS SUBMITTED IN PARTIAL FULLFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

THE COLLEGE OF GRADUATE STUDIES

(Mathematics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Okanagan)

November, 2011

© Bonane Bhattacharja, 2011

Abstract

In this research, an efficient numerical method is developed to determine the steady-state waiting time distribution of a GI/G/1 queue by solving the discrete-time version of Lindley's equation, when the queue is bounded on a finite interval. Then, by using Little's Formula, we calculate the stationary distribution for the total number of customers in the queue. The derivations are based on the Wiener-Hopf factorization of random walks. The method is carried out using a successive approximation method, by improving the weighted average. Finally, to prove the effectiveness of our method, we apply the algorithm for Uniform, Geometric, and Gamma distributions, to find an approximation. An analytical interpretation is also presented to find the waiting time distribution for the Geom/Geom/1 queue, which is not based on a finite interval, as an example of the GI/G/1 queue. Moreover, compare to the other related methods it has been proven that our method is numerically stable, simple, and robust.

Table of Contents

Abstract	ii
Table of Contents	iii
List of Tables	v
List of Figures	vi
List of Symbols	vii
Acknowledgements	viii
Dedication	ix
Chapter 1 Introduction	1
1.1 Mathematical Model	1
1.2 Objective of the Thesis	5
1.3 Outline of the Thesis.....	7
Chapter 2 Preliminaries	8
Chapter 3 Review of the Literature	17
Chapter 4 Methodology	23
4.1 Overview.....	23
4.2 Wiener-Hopf Factorization Method.....	27
4.3 Our Explanation for using the Wiener-Hopf Factorization Method in 1-U(z)	36
4.4 Wiener-Hopf Factorization by Successive Substitution	42
Chapter 5 Analytical Comparison of GI/G/1 Queue with Geom/Geom/1 Queue ..	51
5.1 Overview.....	51
5.2 The Waiting Time Distribution for the Geom/Geom/1 Queue.....	52
Chapter 6 Numerical Experiment	56
6.1 Numerical Results and Examples	56
6.2 Charts for the Queue Lengths	69
Chapter 7 Conclusion and Recommendations for Future Work	72
7.1 Conclusions.....	72

7.2 Future Works	73
Bibliography	74
Appendix	77

List of Tables

Table 4.1	Results by equating the coefficients of z^j and z^{-j} of Eq. (4.25)	38
Table 6.1	Results of using Uniform Distributions by using our Algorithm and by Algorithm 3.....	58
Table 6.2	Results for the Average Queue Length in queue using our method	59
Table 6.3	Results for Uniform Distributions with Large Bounds by using our Algorithm and by Algorithm 3	60
Table 6.4	Results for the Average Queue Length in queue using our method	61
Table 6.5	Results of using Truncated Geometric Distributions by using our Algorithm and by Algorithm 3	63
Table 6.6	Results for the Average Queue Length in queue using our method	64
Table 6.7	Results of using Gamma distribution for the interarrival time and truncated Geometric distribution for the service time by using our Algorithm and by Algorithm 3	67
Table 6.8	Results for the Average Queue Length in queue using our method	68

List of Figures

Figure 2.1	Queueing process.	12
Figure 2.2	Use of the Erlang for phased service.	15
Figure 6.2.1	The Average Queue Length in queue for the U/U/1 queue	69
Figure 6.2.2	The Average Queue Length of large bound for the U/U/1 queue.....	70
Figure 6.2.3	The Average Queue Length in queue for the Geom/Geom/1 queue	70
Figure 6.2.4	The Average Queue Length for the Gamma/Geom/1 queue in queue.....	71

List of Symbols

(β_1, γ_1)	strict ascending ladder point
$(\bar{\beta}_1, \bar{\gamma}_1)$	weak descending ladder point
c_k	convolution of discrete interarrival time and service time
<i>CDF</i>	cumulative distribution function
C_n	n th customer
<i>GI/G/1</i>	notation with single-server queue with general $Q(t)$ and $P(t)$
<i>U/U/1</i>	notation with single-server queue where interarrival and service times are uniformly distributed
<i>Geom/Geom/1</i>	notation with single-server queue where interarrival and service times are geometrically distributed
<i>Gamma/Geom/1</i>	notation with single-server queue where interarrival times are distributed by gamma distribution and service times are geometrically distributed
$H(x)$	distribution function for first ladder height γ_1
$J(x)$	distribution function for first ladder height $\bar{\gamma}_1$
ρ	traffic intensity
Pr	probability of an event
p_k	discrete distribution of service time
$p(t)$	probability density function of service time
$P(t)$	distribution function of service time
q_k	discrete distribution of interarrival time
$q(t)$	probability density function of interarrival time
$Q(t)$	distribution function of interarrival time
S_n	service time for customer C_n
τ	step size
T_n	T_n is the interarrival time for customer C_n and C_{n+1}
W_n	waiting time (in queue) for customer C_n
$W(t)$	waiting time distribution
$W^{(n)}(t)$	waiting time distribution function for W_n
L_q	mean queue length in the queue

Acknowledgements

First, I would like to express my utmost indebtedness and gratitude to my supervisor, Dr. Javad Tavakoli, for his careful supervision and guidance in completing this thesis, and for providing me with the opportunity to work in my chosen field of study, Mathematics, and in particular, queueing systems.

I would also like to express my sincere appreciation and respect for my honourable committee member, Dr. Winfried K. Grassmann, for his valuable comments and constructive suggestions in preparing this thesis.

Then, I would like to thank all of my colleagues and professors from UBC Okanagan for their support and encouragement.

Finally, I am beholden to my husband, Biddut Bhattacharjee, for his constant support, which helped me to become more confident and enabled me to pursue my education in Mathematics, and to complete my thesis for the M.Sc. degree.

Dedication

This work is dedicated to my parents.

Chapter 1

Introduction

1.1 Mathematical Model

A queueing system consists of “customers” arriving at random times to some facility where they receive service of some kind and depart. We use “customer” as a generic term.

There are varieties of applications of Queueing Theory in our daily life, such as in traffic flow (vehicles, aircraft, people and communications), in scheduling (patients in hospitals, jobs on machines) and also in facility design (banks, post office, fast food restaurants) and so on. Queueing systems are classified according to

1. *The input process*, the probability distribution of the pattern of arrivals of customers in time;
2. *The service distribution*, the probability distribution of the random time to serve a customer (or group of customers in the case of batch service); and
3. *The queue discipline*, the number of servers and the order of customer service.

1. *The input process*: Input is the pattern of customer arrival and conjunction with the system. It is given by the distribution time between successive arrivals which is known as the interarrival time distribution. While a variety of input processes may arise in practice, two simple and frequently occurring types are mathematically tractable. First is the scheduled input where customers arrive at a fixed time $T, 2T, 3T, \dots$ and the second most common model is the “completely random” arrival process where the times of customer arrivals form a Poisson process. Exponentially distributed interarrival times then corresponds to a Poisson process of arrival as a special case. The input can be characterized by the arrival rate which gives the expected number of arrival per time unit. The number of arriving customers and potential customers for a queueing system can be finite or infinite, and they can arrive

individually or in groups. Once customers join in the system, a waiting line is formed. It may also happen that customers cannot join in the system because the system is fully occupied.

2. *The service distribution:* The service pattern describes the way customers are served. With each customer, associate a service time. We will always assume that the duration of service for individual customers is independent and identically distributed which are nonnegative random variables and it is independent of the arrival process. An important measure to describe the service pattern is the service rate, which is the reciprocal of the expected service time. The service rate gives the expected number of services completed per time unit. The situation in which all service times is in the same fixed duration, D , is then a special case. The number of servers in a queueing system is an important parameter. A queueing system may have only a single stage of service (e.g., hair styling salon) and multistage service (physical examination procedure). If the number of servers is infinite, then all customers can enter the service facility immediately, therefore no waiting line will be formed.

3. *The queue discipline:* The queueing discipline is the order according to which customers are served. The most common queue discipline is *first come, first serve (FCFS)* where customers are served in the same order in which they arrive, for example, people are served by the cashier at a supermarket by following FCFS. The model we consider here is of this type; however, this is certainly not the only possible queue discipline. Another one commonly used discipline is the *last come, first serve (LCFS)*, which means that the customer selected for serving is the one who enters the system latest, for example, jobs arriving in a production facility with one or more machining centers are often scheduled according to last-come-first-served basis because of ease of access to stored jobs or in order to meet due date requirements. Customers may also be served according to *service in random order (SRO)*, which means that the service order for customer is decided randomly, for example, the procedure of checking the quality of the commodities in the industry is followed by service in random order. The *priority discipline* is another common process that is employed in queueing systems. In the *priority queue*, customers are allowed to be of different types and both the service discipline and the service time distribution may vary with the customer type.

Queueing models aid the design process by predicting system performance. For example, a queueing model might be used to evaluate the costs and benefits of adding server to an existing system. The models enable us to calculate system performance measures in terms of their basis quantities. Some important measures of the system behaviour are:

- 1) *The probability distribution of the number of customers in the system.* Not only do customers in the system often incur costs, but generally in many systems, physical space for waiting customers must be planned for and provided.
- 2) *The utilization of the server(s).* Idle servers may incur costs without contributing to system performance.
- 3) *System throughput.* The long run number of customers passing through the system in a direct measure of system performance.
- 4) *Customers waiting time.* Long waits for service are annoying in the simplest queueing situations and are directly associated with major costs in many large systems, such as patients awaiting emergency care at a hospital.

The subject of queueing theory originated from the studies of telephone conversations. The Danish mathematician A.K. Erlang was the pioneer researcher in this area in the early 1900s. Later, in the 1930s, F. Pollaczek did further research and expanded much of Erlang's work. Moreover, one of the most powerful relationships in queueing theory was developed by John D.C. Little in the early 1960s. Little related the steady-state mean system sizes to the steady-state average customer waiting times. Let

L = the average number of customers in the system

λ = the rate of arrival of customers to the system and

W = the average time spent by a customer in the system.

The equation $L = \lambda W$, is known as Little's formula, which directly relates to the two of our important measures of system performance, the mean queue size and the mean customer waiting time in steady-state.

The total waiting time in the system is the sum of the waiting time before service, plus the service time. The *busy period* is the length of time during which the server is continuously

busy for serving the customers, and the length of time when the server is not serving or there are no customers in the system, is called the *idle period*. For a single server queue, the probability of the system being idle is the same as the probability of a server being idle.

If we denote the average rate of customers entering the queueing system as λ and the average rate of serving customers as μ , then the ratio of the rate of input and the rate of service is known as *traffic intensity* which is denoted by $\rho = \frac{\lambda}{\mu}$. The ratio of the arrival rate to the service rate plays a significant role in measuring the performance of queueing system. The higher the ratio, the longer the customer will tend to wait. If $\lambda < \mu$ (or, $\rho < 1$), then the queue has a steady-state distribution. In a steady-state situation, ρ is the probability that the server is busy, and therefore it gives the expected number in service. As ρ increases, the length of the busy period also increases. When $\rho > 1$ (or, $\lambda > \mu$), the average number of arrivals in the system exceeds the service rate of the system, and the queue size never settles down, so no steady-state exists. It turns out that for steady-state results to exist, ρ must be less than 1. When $\rho = 1$, unless arrivals and service are deterministic and perfectly scheduled, no steady state exists. The quantity ρ is sometimes referred to as the *offered load*, since, on average, each customer requires $1/\mu$ time units of service and the average number of customers arriving per unit time is λ , so that the product $\lambda(1/\mu)$ is the average amount of work arriving in to the system per unit time. The limiting behaviours, $\rho \rightarrow 0$ and $\rho \rightarrow 1$, are known as light traffic and heavy traffic intensities, respectively.

A standard shorthand is used in much of the queueing literature for identifying simple queueing models. The shorthand assumes that the arrival times form a renewal process, and the format $A/B/c$ uses A to describe the interarrival time distribution, B to specify the individual customer service time distribution, and c to indicate the number of servers. There are some well known queueing models like $M/M/1$, $M/M/\infty$, $M/GI/1$, $GI/M/1$, $E_k/M/1$, $M/D/1$ and so on, which are very common in queueing systems. The symbol M (Markovian or memoryless) is used for exponential distribution, E_k (Erlang) for the gamma distribution order k , and D for a deterministic distribution, a schedule of arrivals or fixed service times. The symbol G represents a general or arbitrary probability distribution.

1.2 Objective of the Thesis

The use of Markov models in queueing theory is very common because they are appropriate for basic systems and they lend themselves to easy applications. But real-world systems are so complex and so general that simple Markov and renewal process models do not represent them well. Thus, the importance of general queueing model has proved useful in handling emerging complex situations.

In this study, we discuss the steady state waiting time distribution and the expected queue length of a discrete GI/G/1 queue and then present a numerical method for calculating these.

The notation GI stands for general independent interarrival time distribution, G stands for general service time distribution and number 1 means that there is one server in the system. More clearly, this symbol G represents a general or arbitrary probability distribution; that is, no assumption is made as to the precise form of the distribution and results in these cases are applicable to any probability distribution. These general-time distributions, however, are required to represent independent and identically distributed random variables. Moreover, this GI/G/1 queue can be used for both the discrete and continuous case, which means that the interarrival and service times could be discrete or continuous.

As we mentioned, we consider discrete-time queues in our work, that is, the queues where the times between arrivals and service are both integers. In general, when the interarrival and service times are discrete random variables in a general queueing model, then it is called discrete GI/G/1 queue. In discrete-time queues, the time axis is segmented into a sequence of equal intervals (of unit duration), called slots, and arrivals and departures of customers are assumed to take place at the slot boundaries.

On the other hand, when the interarrival and service times are continuous random variables, the queue is known as a continuous GI/G/1 queue. The calculation of the waiting time distribution in the continuous GI/G/1 queue is more difficult. In this case, one needs to discretize the distribution function of the interarrival and service times for the numerical analysis. Some authors showed that under certain conditions, the continuous GI/G/1 queue can be approximated quite well by a discrete GI/G/1 queue. In our numerical analysis, we use Gamma/Geom/1 queue, where the interarrival times are continuously distributed.

Approaches in our case, based on Lindley's equation and the derivations are based on a Wiener- Hopf factorization of random walks, using ladder height distribution. This is carried out using a successive approximation method, first suggested by Grassmann [11].

By solving a discrete-time version of Lindley's equation [24], several authors, including Ponstein [28], Konheim [22], Ackroyd [1], Grassmann and Chaudhry [13], Grassmann [11], and Fryer and Winsten [10], also numerically determined the waiting time distribution of the arithmetic GI/G/1 queue, when the queue is bounded on a finite interval. It was mentioned repeatedly [13, 1, 10] that the discrete-time version of Lindley's equation is mathematically equivalent to certain equations arising in bulk queues. We will introduce their methods in the following chapters. A number of alternatives to Lindley's approach have also been suggested. For instance, one can approximate the GI/G/1 queue by the GI/PH/1 queue, an approach originally devised by Neuts [26]. One can also use approximations rather than the exact results to find the important measures of the GI/G/1 queue [37, 34]. This approximation may be satisfactory in a given situation.

At present, there are several approaches available to solve such queueing problems. We consider first an approach where one can set up generating functions, and uses partial fraction expansions to invert these generating functions by using the Wiener-Hopf factorization method. Generally, these expansions require one to find the roots of an equation of degree $g + h$, where g and h are certain parameters that considered as the finite bounds of that equation. It turns out that g of the roots of this expression lie inside the unit circle of the complex plane and that h of them lie outside the unit circle. Moreover, the coefficients of the factor whose roots are the g inner roots are equal to the ratios of the steady state probabilities; that the server idles j units of time between a departure and the next arrival, $j = 0, 1, \dots, g$. The numerical method consists in first finding an approximation of the probabilities from which an approximate factor of outer roots h is obtained to find out the steady state probabilities of the waiting time distribution. However, by following the iterative method given by [12] a numerical method is presented in this study by improving the weighted average to find an approximation for the waiting time distribution of a GI/G/1 queue which eventually converges faster than the method given by [12]. Applying approximating techniques to these general formulae seems to cause problems if the traffic

intensity ρ is close to one. The technique developed here has been applied successfully with traffic intensities of up to 0.99. We also compare this approximation to the other procedures to find the benefit and effectiveness of our method.

1.3 Outline of the Thesis

The thesis consists of seven chapters including an appendix. In the first chapter, a brief summary of GI/G/1 queues is given and the objective of the thesis is stated.

The second chapter states some important definitions on the related topics.

The third chapter includes an overview of the literature of different authors in solving the waiting time distribution of the GI/G/1 queue, and different solution techniques are outlined.

Chapter 4 explains about the Wiener-Hopf factorization method and how to use it to obtain the discrete waiting time distribution. Based on this method and the algorithm given by Grassmann and Jain [12], we also introduce an efficient algorithm by improving the weighted average to solve the discrete waiting time distribution. Then, by using Little's formula, we calculate the average queue length of the system. We also use this algorithm for the numerical experimentation.

Moreover, chapter 5 deals with an analytical interpretation of finding the waiting time for the Geom/Geom/1 queue, which is not based on a finite interval, as an example of the GI/G/1 queue.

We give some numerical examples with different distribution like uniform, gamma and geometric, in Chapter 6.

Chapter 7 summarises the thesis and offers conclusion, and some ideas for the future research.

Finally, MATLAB programs for numerical experimentation are included in the appendix.

Chapter 2

Preliminaries

Stochastic process

Definition 2.1: Stochastic process is a family (sequence) of random variables, X_t , where t is a parameter that runs over a suitable index set T . (Where convenient, we write $X(t)$ instead of X_t) $\{X_t: t \in T\}$.

In a common situation, the index t corresponds to a discrete unit of time and the index set is $T = \{0, 1, 2, \dots\}$.

Examples 2.1:

i) $X_t =$ Temperature at time $t \in [0, \infty)$

ii) $X_t =$ Number of students present in the class on day $t \in \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$.

a. If T is countable (or finite), then the process is called a *discrete-time process*.

b. If it consists of a set of finite or infinite intervals on the time axis, it is a *continuous-time process*.

Markov process

Definition 2.2: A *discrete* parameter stochastic process $\{X(t), t = 0, 1, 2, \dots\}$ or a *continuous* parameter stochastic process $\{X(t), t > 0\}$ is said to be a *Markov process* if for any set of time points $t_1 < t_2 < \dots < t_n$ in the index set or time range of the process, the conditional distribution of $X(t_n)$, given the values of $X(t_1), X(t_2), X(t_3), \dots, X(t_{n-1})$, depends only on the immediately preceding value $X(t_{n-1})$;

More precisely, for any real numbers $x_1, x_2, x_3, \dots, x_n$,

$$\begin{aligned} &Pr\{X(t_n) \leq x_n | X(t_1) = x_1, \dots, X(t_{n-1}) = x_{n-1}\} \\ &= Pr\{X(t_n) \leq x_n | X(t_{n-1}) = x_{n-1}\}. \end{aligned}$$

Thus, the process is *memoryless*.

Markov Chain

Definition 2.3: A Markov chain is a *discrete-time* Markov process with discrete state space which is stationary and satisfies,

$$Pr\{X_{n+1} = j | X_n = i\} = p_{ij}.$$

Here, p_{ij} represents the probability that a system in state i will enter state j at the next transition.

Examples 2.2:

- State may be a certain level of energy for an atom.
- In the public opinion poll, state may be one of the voter's possible state of mind.
- Weather has two states \rightarrow wet and dry.

Theorem 2.1: A Markov process is completely defined by its transition probability matrix and the initial state X_0 .

Steady-state probability

Definition 2.5: Consider a discrete-parameter Markov chain and suppose that $\lim_{n \rightarrow \infty} p_{ij}^{(n)} = \pi_j$ (for all i), which is independent of n . That is, after a long time, the probability that the process is in state j given that it started in state i , is independent of the starting state i . We call $\{\pi_j\}$ the limiting or steady-state probability of Markov chain.

- The *steady-state or equilibrium* distribution of a Markov chain is given by the following equations

$$\pi_j = \sum_{i=0}^{\infty} \pi_i p_{ij}, \quad j \geq 0 \text{ and } \sum_{j=0}^{\infty} \pi_j = 1.$$

- *Transition probability Matrix or Markov matrix:*

$$P = (p_{ij}) = \begin{bmatrix} p_{00} & p_{01} & p_{02} & \dots \\ p_{10} & p_{11} & p_{12} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad p_{ij} \geq 0; \quad i, j = 0, \dots, \infty \quad \text{and} \quad \sum_{j=1}^{\infty} p_{ij} = 1 \quad \forall i \quad [\text{each row's sum} = 1].$$

Irreducible Markov Chain

Definition 2.6: A Markov chain is said to be *irreducible* if there is only one class of states, that is all states communicate with each other.

State j is accessible from state i if $p_{ij}^{(n)} > 0$ for some integer $n \geq 0$. In a regular chain, each state is accessible from each other.

Random Walk

Definition 2.7: When we discuss random walks, it is an aid to intuition to speak about the state of the system as the position of a moving “particle”.

A one-dimensional *random walk* is a Markov chain whose state space is a finite or infinite subset of the integers, in which the particle, if it is in state i , can in a single transition either stay in i or move to one of the neighbouring states $i - 1, i + 1$.

If the state space is taken as the nonnegative integers, the transition matrix of a random walk has the form

$$P = \begin{array}{c|cccccc} & 0 & 1 & 2 \dots & i & i - 1 & i + 1 \\ \hline 0 & r_0 & p_0 & 0 \dots & 0 & 0 & \dots \\ 1 & q_1 & r_1 & p_1 \dots & 0 & 0 & \dots \\ 2 & 0 & q_2 & r_2 \dots & 0 & 0 & \dots \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \dots \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \dots \\ i & \cdot & \cdot & 0 & q_i & r_i & p_i \\ & \cdot & \cdot & \cdot & \cdot & \cdot & \dots \end{array}$$

where $p_i > 0, q_i > 0, r_i \geq 0$, and $q_i + r_i + p_i = 1, i = 1, 2, \dots$,

$$p_0 \geq 0, r_0 \geq 0, r_0 + p_0 = 1.$$

Specifically, if $X_n = i$ then, for $i \geq 1$,

$$\Pr\{X_{n+1} = i + 1 | X_n = i\} = p_i, \Pr\{X_{n+1} = i - 1 | X_n = i\} = q_i,$$

and

$$\Pr\{X_{n+1} = i | X_n = i\} = r_i, \text{ with obvious modifications holding for } i = 0.$$

Example 2.3: i) The path of a person moving randomly one step forward or backward gives us a realization of the random walk.

ii) The fortune of a player engaged in a series of contests is often depicted by a random walk process.

Renewal (counting) process

Definition 2.8: A renewal process $\{N(t), t \geq 0\}$ is a nonnegative integer-valued stochastic process that registers the successive occurrences of an event during the time interval $(0, t]$, where the time durations between consecutive events are *positive, independent, identically distributed* random variables.

Let the successive occurrence times between events be $\{X_k\}_{k=1}^{\infty}$ such that X_i is the elapsed time from the $(i - 1)st$ event until the occurrence of the i th event. We write

$$F(x) = \Pr\{X_k \leq x\}, \quad k = 1, 2, 3, \dots$$

For the common probability distribution of X_1, X_2, \dots , a basic stipulation for renewal processes is $F(0) = 0$, signifying that X_1, X_2, \dots are positive random variables. We refer to

$$S_n = X_1 + X_2 + \dots + X_n, \quad n \geq 1 \quad (S_0 = 0, \text{ by convention})$$

as the *waiting time* until the occurrence of the n th event. Therefore, in this case, S_n is the sum of n random variables.

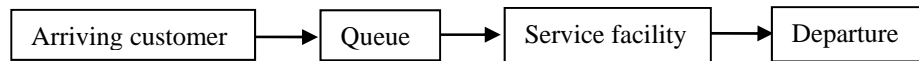
Queueing System

Definition 2.9: A *queueing system* can be described as customers arriving for service, waiting for service if the server is not idle, receive service, then leaving the system after being served.

Queues are common in everyday life. Customers wait to be served in front of service facilities. For example, people wait to use the autoteller in the bank, cars wait to be repaired in an autobody shop, airplanes wait to take off at an airport, machines wait to be repaired in a

factory, people wait to use the phones in phones booths. These are but a few examples among many about the waiting lines.

Figure 2.1 **Queueing process.**



There are six basic *characteristics* of queueing processes provide that an adequate description of a queueing system: (1) arrival pattern of customers, (2) service pattern of servers, (3) queue discipline, (4) system capacity, (5) number of service channels and (6) number of service stages.

Queueing model

Definition 2.10: A queueing process is a process in which customers arrive at some designated place where a service of some kind is being arranged, *for example*, at the teller's window in a bank or beside the cashier at a supermarket. It is assumed that the time between arrivals, or interarrival time, and the time that is spent in providing a service for a given customer, are governed by the probabilistic laws. In short, this procedure is known as *queueing model*.

Queueing models assist the design process by predicting system performance. For example, a queueing model might be used to evaluate the costs and benefits of adding a server to an existing system. A standard shorthand is used in much of the queueing literature for identifying simple queueing models. The shorthand assumes that the arrival times form a renewal process, and the format $A/B/c$ uses A to describe the interarrival distribution, B to specify the individual customer service time distribution, and c to indicate the number of servers.

The following are some definitions of some of the most popular queueing models:

The $M/M/1$ queueing model

Definition 2.10.1: The $M/M/1$ model is known as a single-server exponential queueing system. In this case, arrivals follow a Poisson process, service times are exponentially distributed, and there is a single server.

In general, M refers mainly to three facts:

- (1) Arrival is Poisson and interarrival time distribution is exponential,
- (2) Service time distribution is also exponential,
- (3) Thus, the process is memoryless or Markovian.

Since the successive times for interarrival and service are continuously distributed, they are therefore exponential. The number of customers in the system at time t forms a birth and death process where arrivals can be considered as “births” to the system, and departures can be considered as “deaths”.

The parameters λ and μ represent the rates for the arrivals and service per unit time.

The $M/GI/1$ queueing model

Definition 2.10.2: In this model there are Poisson arrivals with rate λ , but arbitrarily distributed service times which are i.i.d. with mean $1/\mu$ and server is single. The analysis proceeds with the help of an embedded Markov chain.

The $GI/M/1$ queueing model

Definition 2.10.3: In the $GI/M/1$ model, service times have the Markovian property. From the notation it is apparent that the model is a single-server system with an arbitrary or general distribution function of interarrival times which are i.i.d. with mean $1/\lambda$, and the service times are i.i.d. exponential random variables.

The $M/D/c$ queue

Definition 2.10.4: This is a multiserver queueing model where customers arrive in a Poisson process with the rate λ , and service times are constant or deterministic.

The $M^X/M^Y/c$ queue

Definition 2.10.5: The $M^X/M^Y/c$ is also a multiserver Markovian queueing model. In a queueing system, if customer arrivals and /or service occur in groups, then the system is known as a bulk queue. Here,

M^X \longrightarrow means customers arrive in group of size X , where X is a random variable greater than zero and the group of arrival occurs in a Poisson process with rate λ .

M^Y \longrightarrow means that customers are served in a group of size Y and the service time distribution is exponential with rate μ .

Phase-Type distribution

Definition 2.10.6: A *phase-type distribution* is a probability distribution that results from a system of one or more inter-related Poisson process occurring in sequence, or phases. The sequence in which each of the phases occur may itself be a stochastic process. The distribution can be represented by a random variable describing the time until absorption of a Markov process with one absorbing state. Each of the states of the process represents one of the phases.

Some distributions that use the concept of phases are the Erlang, Hyperexponential, hypo exponential or generalized Erlang, and Coxian distribution.

The $E_j/E_k/1$ model

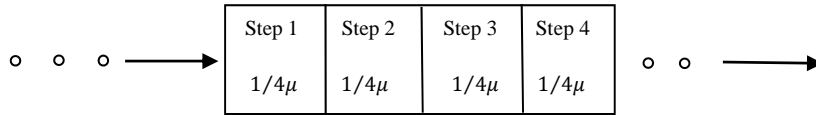
Definition 2.10.7: The $E_j/E_k/1$ is a single server queueing model with arrival rate λ and service rate μ (i.e. each arrival phase has rate $j\lambda$ and each service phase has rate $k\mu$). Specifically, E_k represents the sum of k i.i.d. exponential random variables with mean $1/k\mu$, which is also known as Erlang type- k distribution.

The Erlang family provides more flexibility in modeling than the exponential family, which only has one parameter.

Example 2.4: There are some practical examples of Erlang distribution in our daily life.

- i. In a visa office, people have to follow some steps to enter in the system.
- ii. For the physical examination in a pathology centre, a patient needs to pass several stages to get the service.

Figure 2.2 Use of the Erlang for phased service.



The $GI/G/1$ queueing model

Definition 2.10.8: The symbol G represents a general or arbitrary probability distribution; that is, no assumption is made as to the precise form of the distribution. Results in these cases are applicable to any probability distribution. Thus, by $GI/G/1$, we mean that

- I. The inter-arrival times are i.i.d with a distribution function F_T having mean $1/\lambda$,
- II. The service times independent of arrival times are i.i.d with a distribution function F_S having mean $1/\mu$,
- III. The service is provided by one server,
- IV. Infinite waiting space is available and
- V. The queue discipline is FCFS.

Waiting time

Definition 2.9: The amount of time a new arrival has to wait until its service begins, is known as the *waiting time*.

Queue length

Definition 2.10: The actual number of customers in the system, is called the *queue length*.

Busy period

Definition 2.11: The length of time during which the server is continuously busy with serving their customers is called the *busy period*.

Idle period

Definition 2.12: The length of time when the server is not serving or there are no customers in the system is called the *idle period*.

Probability generating function

Definition 2.13: Suppose,

$$P(z) = \sum_{n=0}^{\infty} p_n z^n = p_0 + p_1 z + p_2 z^2 + \dots$$

If the series converges for some range of z , then $P(z)$ is called a *generating function* of the sequence $\{p_n\}$.

Let, X be a random variable with $Pr\{X = n\} = p_n$, $n = 0, 1, 2, \dots$ and $\sum_{n=0}^{\infty} p_n = 1$. Then $P(z) = \sum_{n=0}^{\infty} p_n z^n$, which is known as the *probability generating function (p.g.f)* of X .

Probability generating function can be used to directly obtain moments of a random variable. If we take the first derivatives of $P(z)$ with respect to z at $z = 1$, then we can calculate the moment. Therefore, we get

$$P'(z) = \sum_{n=1}^{\infty} n p_n z^{n-1} \text{ and then evaluate } P'(z) \text{ at } z = 1, \text{ to get } P'(1) = \sum_{n=1}^{\infty} n p_n = E[X].$$

Moreover, we can determine the variance of a random variable from the probability generating function. In this case, we also need to take the second derivatives of $P(z)$ with respect to z at $z = 1$, since, we know, $\text{Var}[X] = E[X^2] - E^2[X]$. Thus,

$$P'(z) = \sum_{n=1}^{\infty} n p_n z^{n-1},$$

$P''(z) = \sum_{n=2}^{\infty} n(n-1) p_n z^{n-2}$, and then by evaluating $P'(z)$ and $P''(z)$ at $z = 1$, we get $P'(1) = \sum_{n=1}^{\infty} n p_n = E[X]$ and $P''(1) = \sum_{n=2}^{\infty} n(n-1) p_n = E[X(X-1)]$. Now, $\text{Var}[X]$ can be obtained from $P'(1)$ and $P''(1)$:

$$\text{Var}[X] = (E[X^2] - E[X]) + E[X] - E^2[X].$$

Generally, *p.g.f* can be used to determine the probabilities p_0, p_1, \dots and $P(z)$ is written as a power series and p_n is the coefficient in front of z^n in the series.

Chapter 3

Review of the Literature

The queueing model GI/G/1 has been of much interest to queueing theorists and practitioners since the 1950's, and the study of this model is becoming more significant, since it has wide applications in computers, communication systems and networks of queues. The most important and much-studied aspect of this model is the steady-state distribution of the actual queue time of a customer. The analysis of the waiting time of a customer has been given by several authors, including Ponstein [28], Konheim [22], Ackroyd [1], Grassmann and Chaudhry [13], Grassmann [11], Grassmann and Jain [12] and Fryer and Winsten [10]. All of these authors numerically determined the waiting time distribution of the arithmetic GI/G/1 queue by solving a discrete-time version of Lindley's equation (Lindley [24]). Before doing so, one can demand that the random variables $\{T_n\}$ and $\{S_n\}$ be discrete and that their distribution functions are both defined on finite intervals. It was pointed out many times that the discrete-time version of Lindley's equation is mathematically equivalent to certain equations arising in bulk queues (Grassmann and Chaudhry [13]; Fryer and Winsten [10]). In this case, there are various methods of finding the waiting time distribution in a GI/G/1 queue. In this chapter, we present some of these approaches to deal with the waiting time distribution of the GI/G/1 queue.

The method presented by Grassmann and Jain [12] is an efficient numerical method for calculating the waiting time and idle time distribution of the arithmetic GI/G/1 queue. The method is based on Wiener – Hopf factorization, and compared to related methods of others, their method appears to perform very well and is often faster by several orders of magnitude than other methods. There are three algorithms suggested by them which we will introduce in chapter 4. However, based on the same procedure, our method works better than their methods in numerical experimentations.

Here, I should also mention the work of Konheim [22], who gave an elementary method for calculating the stationary distribution of waiting time in a GI/G/1 queue. Konheim based his method on generating function techniques, and his solution required only the ability to factor polynomials, which is part of our method as well. Konheim set a relation $S(z) = (1 - U(z))/(1 - z)$, where $(1 - U(z))$ is the same as in our Eq. (4.25) and $U(z) = u_{-g}z^{-g} + \dots + u_h z^h$ for $-g \leq U \leq h$. Then, he factored $S(z)$ into $S^+(z)$ and $S^-(z)$, where $S^+(z)$ contains all the zeros of $S(z)$ outside the unit circle, and $S^-(z)$ contains all other zeroes. The purpose of this factorization is to find the roots of $S(z)$ which are outside the unit circle. Moreover, $S^+(z)$ is scaled such way that $S^+(1) = 1$. Konheim showed that the generating function of waiting time distribution $W(z)$ can be obtained through the equation $W(z) = 1/S^+(z)$, which is essentially our Eq. (4.30). A similar factorization was also suggested by Ackroyd [1]. Konheim basically used h zeros of $1 - U(z)$ to find $A(z)$ for obtaining the waiting time distribution, where $A(z)$ is a polynomial that has h roots outside the unit circle. In Eq. (4.26), we also mentioned $A(z) = \sum_{i=1}^{\infty} \alpha_i z^i$, is the generating function for the distribution that the server remains busy for serving the customers in the system. Some authors, in particular Fryer and Winsten [9] also followed the same procedure as [22] to find $A(z)$.

A number of methods have been suggested to find $A(z)$ iteratively. The earliest such attempt was done by Ponstein [28], who solved the waiting time distribution in a discrete GI/G/1 queueing model, by a method based upon Markov chains. He also defined $u_k = \Pr(U_n = k)$ for $k = -K_1, \dots, K_2$, where $U_n = S_n - T_n$ is the difference between the service time of customer C_n , and the interarrival time of the C_n and C_{n+1} customers. In our work, we used g and h for K_1 and K_2 . Ponstein then formed a Markov chain with a one-step transition matrix, $Q = [p_{jk}]$, where p_{jk} represents the one-step transition probability that if the queueing system is in state, say F_j , for some time unit, the system will be in state F_k for the next time unit, and p_{jk} is given as $p_{jk} = u_{k-j}$. Here the state is in F_j for some time unit generally stands for a n th customer waits j units of time before being served, $j = 1, 2, \dots$, or the server idles ($-j$) consecutive units of time before a customer arrives, $j = 0, -1, \dots, -K_1$. Then, as a standard procedure in Markov chains, let the steady-state probabilities for the queueing system in state F_j for some time units be $\Sigma = (\pi_{-K_1}', \pi_{-K_1+1}', \dots, \pi_0', \pi_1', \dots)$. When the

system is in equilibrium, it is considered that $\sum Q = \sum$ and $q_1 = \pi'_{-K_1}, \dots, q_{K_1} = \pi'_{-1}$; $\pi_0 = \pi'_{-K_1} + \dots + \pi'_{-1} + \pi'_0$ and $\pi_j = \pi'_j$ for $j = 1, 2, \dots$. Note that according to the definition of π'_j , the $\pi_j, j = 1, 2, \dots$ gives the required waiting time distribution and q_j , for $j < 0$ gives the idle time distribution. Moreover, Ponstein gave an algorithm in order to find the π_j s for $j = 1, 2, \dots$. He also factored $Q(z)z^{K_1}$, where $Q(z) = 1 - U(1/z)$ to locate the roots outside and inside of the unit circle. Here, $U(z) = u_{-K_1}z^{-K_1} + \dots + u_{K_2}z^{K_2}$. It is also noticed that the main difference between Ponstein's method and Konheim's [22] method is: those roots which are inside the unit circle, or inner roots, in Konheim's method corresponds to those roots which are outside the unit circle, or outer roots, in Ponstein's method and those outer roots in Konheim's method corresponds to those inner roots in Ponstein's method.

Powell [29] used a an efficient implementation of Newton's method to find the zeros of $1 - U(z)$ in the case of bulk queues, and in his particular case, Newton's method may be more efficient than our method. However, in the case of bulk queues, $U(z)$ have a very special form, and Powell's method can thus only be used to solve very special GI/G/1 queues.

Another useful method was suggested by Ackroyd [1], who used complex logarithms to separate the zeros inside and outside the unit circle. This can be done very efficiently, using Fast Fourier Transforms.

Grassmann and Chaudhry [13] combined the approach of Neuts, which is known as the matrix-geometric methods, with their method. Neuts' method can also be used to find $A(z)$. In Neuts' approach, a matrix R of dimension $h \times h$ was determined, which satisfied a certain matrix equation. Grassmann and Chaudhry mainly modified the method of Neuts. According to [13], the α_j were given by only the first column of this matrix R , which reduced the number of operations in their algorithm by a factor of h . In order to determine the waiting time in the queue of a GI/G/1 model, they first set up a Markov chain by following the Ponstein method [28]. In their Markov chain, they considered the state spaces as the waiting times rather than the number of customers in the system. However, the number of iterations to obtain R in Neuts' method was often high, it may be 50 or more, and the number of operations per iteration is on the order h^2g . The algorithm that was developed by these

authors, is a faster and efficient algorithm to solve many queueing problems efficiently and precisely.

Grassmann [11] also developed a method to find the equilibrium distribution of the queue length at a certain epoch for such Markov chains which can be applied to the $M^X/M^Y/c$ queue, the $M/GI^X/c$ queue, the discrete $GI/G/1$ queue, and many other related problems in an efficient way. He numerically solved the steady-state distribution of some finite Markov chain models by using the state reduction method. The problem was solved by factorizing $1 - U(z)$, and thus, this method also can be used to find $A(z)$. Moreover, the algorithm proposed in [11] can be used for continuous-time Markov processes without any change. However, the author claimed that the approach in [11] is better than the Neuts' method and he also mentioned that it converged quickly and was numerically stable. He asserted that there was a probabilistic significance in all steps, which made the algorithm easy to understand. For the calculation of $A(z)$, he also used idea of the inside and outside zeros of the polynomial $1 - U(z)$. As we see, the idea of our method for successive approximation algorithm to investigate the waiting time process of the $GI/G/1$ queue, was first used in [11].

Whitt [38] showed a method to develop a closed form approximation for the mean steady-state workload or virtual waiting time in a $GI/G/1$ queue. This approximation can be solved by using the first two moments of service-time distribution and the first three moments plus the density at the origin of the interarrival-time distribution. The approximation was based on light and heavy traffic limiting behaviour. Basically, Whitt used S.L.Brumelle's formula, $E(Z_\rho) = \rho E(W_\rho) + \frac{\rho(c_s^2+1)}{2}$, to relate the mean workload to the mean waiting time, and K.T. Marshall's formula, $E(W_\rho) = \frac{E((\rho^{-1}T-S))^2}{2E((\rho^{-1}T-S))} - \frac{E(I_\rho^2)}{2E(I_\rho)}$; $0 < \rho < 1$, to relate the mean waiting time to the first two moments of the idle period. The interpolation between light and heavy traffic limits was chosen to satisfy the differentiability and monotonicity regularity conditions. Finally, the approximation was compared with the exact values of some queueing models. Moreover, according to Whitt, heavy traffic derivative is more useful than light traffic derivative for determining the normalized mean workload. However, Whitt utilized approximations in his method, but, in our research, we are dealing with exact methods. In particular, we can get as close to the true value as desired by using our method.

A discrete time version of the system $GI/G/1/N$ was analyzed by Hasslinger [15], using a two-component state model at the arrival and departure instants of customers. The equilibrium equations were solved by a polynomial factorization method and the steady state distribution of the queue size was then represented as a linear combination of geometric series, whose parameters were evaluated by closed formulae depending on the roots of a characteristic polynomial. Considering modified boundary constraints, systems with finite waiting room or with an exceptional first service in each busy period, were also included.

Hasslinger [16] also considered a workload-based approach to the single server queue in a discrete time domain with semi-Markov arrivals (SMP/GI/1). He mainly investigated the SMP/GI/1 queue, extending a GI/G/1 algorithm by [12], which performs a Wiener-Hopf factorization with the help of rapidly converging iteration. By dividing the busy period into phases, these authors generalized a computationally attractive algorithm for the discrete-time GI/G/1 queue, and also computed the stationary distributions of the waiting and idle time, as well as the moments of the busy period. Moreover, performance results were given for deterministic servers with autoregressive input, and the output process of a server was modeled by adapting a SMP with a few states to information obtained from busy and idle period analysis.

Rao and Feldman [31] focused on easily computable numerical approximation for the distribution and moments of the steady-state waiting times in a stable GI/G/1 queue and their approximation methodology is based on the theory of Fredholm integral equations. Making comparisons with the competing approaches established by others, they claimed that their methodology was not only more accurate, but also more amenable for obtaining waiting time approximations from the operational data. Approximations were also obtained for the distributions of the steady-state idle times and interdeparture times.

Kim and Chaudhry [20] first established a discrete-time version of what is called the *distributional Little's Law* (DDL); a relation between the stationary distributions of the number of customers in a system (or queue length), and the number of slots a customer spends in that system (or waiting time). Based on this relation, they then presented a simple computational procedure to obtain the queue-length distribution of the discrete-time GI/G/1 queue from its waiting time distribution, which is readily available by various existing

methods: such as, the method based on the zeros of the so-called characteristic equation (Konheim [22], Chaudhry and Gupta [5]), the iterative method based on the Wiener-Hopf factorization (Grassmann and Jain [12], Hasslinger [16]) and the matrix-analytic method. Using the same procedure, they also obtained the queue-length distribution of the discrete-time multi-server GI/D/c queue in a unified manner.

Thus, there is a great deal of literature dealing with the waiting time distribution of GI/G/1 models. There are especially a number of good algorithms available to solve the waiting time of the GI/G/1 queue numerically, for a discrete-time version of Lindley's equation. Of the algorithms that are competitive, none dominates the other consistently, and the best one depends on the situation. The strengths of the algorithm that has been used in this thesis are that it is easily programmed, generally very efficient and very robust. These properties suggest that the algorithm we used in our work is an ideal algorithm compared to others.

Chapter 4

Methodology

4.1 Overview

In this chapter, we present the approach to solve Lindley's equation based on Wiener-Hopf factorization.

We use the same notation as given in [12] in the following discussion:

The aim of this project is to find out the waiting time distribution in equilibrium (i.e., how much time required for $(n + 1)$ th arrival to get into the service if we know the position of the n th customer), and then calculate the average queue length of the system for the discrete-time queue GI/G/1. Thus, the basic assumptions are as follows:

Assuming customers $C_0, C_1, \dots, C_n, \dots$ arrive singly at a service station at epochs $\{\tau_n: 0 \leq n < \infty\}$, ($\tau_0 = 0$) and are served by a single server in the order of their arrivals.

In this case there are four types of random variables:

1. T_n is the interarrival time between the n th and $(n + 1)$ th customer. The interarrival times $\{T_n = \tau_n - \tau_{n-1}: 0 \leq n < \infty\}$ are i.i.d random variables distributed as T with probability distribution

$$Pr(T = k) = q_k.$$

2. S_n is the service time of the n th customer. The service times $\{S_n: 0 \leq n < \infty\}$ are i.i.d random variables distributed as S with probability distribution.

$$Pr(S = k) = p_k .$$

3. $U_n = S_n - T_n$, which could be negative, zero or positive.

4. W_n is the waiting time (in queue) for n th customer.

Customer C_n waits in queue for a time, W_n , and the service time is S_n . Thus, for C_n , the time in the system is his waiting time plus his service time, i.e. $W_n + S_n$, and he leaves at the moment $\tau_n + W_n + S_n$. If C_{n+1} arrives before that time, therefore, if $\tau_{n+1} < \tau_n + W_n + S_n$, C_{n+1} has to wait for $\tau_n + W_n + S_n - \tau_{n+1}$. Since $T_n = \tau_{n+1} - \tau_n$, we have

$$\begin{aligned} W_{n+1} &= \tau_n + W_n + S_n - \tau_{n+1} \\ &= W_n + S_n - T_n \text{ if } W_n + S_n - T_n \geq 0 \end{aligned} \quad (4.1)$$

Otherwise,

$$W_{n+1} = 0. \quad (4.2)$$

Moreover, the idle time for the server becomes

$$I_n = -(W_n + S_n - T_n) > 0, \text{ if } W_n + S_n - T_n < 0. \quad (4.3)$$

$$\text{We denote } U_n = S_n - T_n \quad (4.4)$$

which is the difference between the service time of customer C_n , and interarrival time of the C_n and C_{n+1} customers.

Therefore, (4.1) and (4.2) can be written in terms of U_n :

$$W_{n+1} = \begin{cases} W_n + U_n, & \text{if } W_n + U_n \geq 0 \\ 0, & \text{if } W_n + U_n < 0 \end{cases}. \quad (4.5)$$

This relation was first established by Linley [23].

Using the notation $(x)^+ \equiv \max[0, x]$, (4.5) can be written as

$$W_{n+1} = (W_n + U_n)^+.$$

The sequence of random variables $\{T_n\}$, for $n \geq 0$, is a sequence of i.i.d. random variables, as is $\{S_n\}$, for $n \geq 0$. Consequently, the sequence of random variables $\{U_n\}$, for $n \geq 0$, is also a sequence of identically distributed random variables. Moreover, T_n is independent of T_{n-1}, T_{n-2}, \dots , and independent of S_n, S_{n-1}, \dots , which means that the U_n are also independent

random variables. Let $U(x)$ be the distribution function of the random variables U_n (which obviously can take negative values with a positive probability, according to its definition). From (4.4), it is not difficult to obtain the following expression, which is commonly known as Lindley's equation:

$$W^{(n+1)}(t) = \begin{cases} \int_{-\infty}^t W^{(n)}(t-x)dU^{(n)}(x), & 0 \leq t < \infty \\ 0 & \text{otherwise} \end{cases} \quad (4.6)$$

where $W^{(n)}(t)$ is the waiting time distribution function for W_n , i.e.

$$W^{(n)}(t) = Pr[W_n \leq t]$$

and

$$U(x) = \int_0^{\infty} P(y)Q(y-x) \quad (4.7)$$

where $Q(t)$ and $P(t)$ are the distribution functions of the distributions of T_n and S_n , respectively.

If $Q(t)$ and $P(t)$ have probability density functions, then we define the probability density of $Q(t)$ and $P(t)$ respectively by,

$$q(t) = \frac{dQ(t)}{dt}$$

$$p(t) = \frac{dP(t)}{dt}$$

Moreover, in this case (4.7) becomes

$$u(x) = \int_0^{\infty} p(y)q(y-x)dy = q(-x) * p(u) \quad \text{where } u(x) = \frac{dU(x)}{dx} \text{ and } * \text{ denotes the convolution.}$$

If the random variables T_n , S_n are discrete,

$$q_k = Pr(T_n = k\tau) \text{ and } p_k = Pr(S_n = k\tau), \text{ where } k \text{ is an integer, and } \tau \text{ is a real number.}$$

In this case, $U_n = S_n - T_n$ is also discrete. Moreover, if $c_k = Pr(U_n = k\tau)$,

then the discrete version of (4.7) becomes

$$c_k = \sum_{i=0}^{\infty} q_{-k+i} p_i . \quad (4.8)$$

It has been shown by Lindley [23] that a necessary and sufficient condition for a stable system is $E(U_n) < 0$. If we substitute U_n by $S_n - T_n$ according to (4.4), then

$$E(S_n - T_n) = E(S_n) - E(T_n) < 0,$$

from which we obtain the usual stable condition for a queueing system, namely

the utilization factor $\rho = E(S_n)/E(T_n) < 1$, where ρ is the traffic intensity. Under this stable condition, there exist a $W(t)$ such that

$$\lim_{n \rightarrow \infty} Pr[W_n \leq t] = W(t)$$

and the Lindley's equation in (4.6) becomes

$$W(t) = \begin{cases} \int_{-\infty}^t W(t-x) dU(x) & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (4.9)$$

or

$$W(t) = \begin{cases} -\int_0^{\infty} W(x) dU(t-x) & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (4.10)$$

Note that the waiting time distribution function $W(t)$ depends only on the distribution function $U(x)$, rather than on the individual distribution function $Q(t)$ and $P(t)$, and the (4.9) and (4.10) is an integral equation of the Wiener-Hopf type.

It is further assumed that the U_n s are i.i.d. random variables, which can only assume the values $-g\delta, (-g+1)\delta, \dots, 0, \delta, 2\delta, \dots, h\delta$, where g and h are integers. Without loss of generality, we take $\delta = 1$ by choosing δ as the time unit, these time values become integers. Our aim is to find the waiting-time in equilibrium, which can assume only the values $0, 1, 2, \dots$

4.2 Wiener-Hopf Factorization Method

In this section, we introduce one of the most important factorization methods which is known as the Wiener-Hopf factorization method.

As we mentioned in section 4.1, Lindley's integral equation, as given by (4.9) or (4.10), is of Wiener-Hopf type; therefore, using Wiener-Hopf factorization method becomes the intuitive approach.

The Wiener-Hopf factorization ultimately facilitates us to determine the actual distributions of limiting waiting and idle times. This factorization also involves the difficult task of locating zeros of a complex function. As it will be shown, this method can be used to solve the waiting time distribution for both continuous GI/G/1 queues and discrete GI/G/1 queues. In particular, based on the Wiener-Hopf factorization technique, a method was given in [12] to calculate the waiting time distribution for a discrete GI/G/1 queue and we present their method elaborately in this chapter.

The powerful tool to link queueing theory and Wiener-Hopf factorization is the theory of random walks. Since our derivations are based on the Wiener-Hopf factorization of random walks, at first, it is necessary to briefly discuss how the Wiener-Hopf method factored the discrete version of Lindley's equation.

In our work, the waiting time is shown to be the maximum of a random walk sequence constrained at zero. The distribution of a maximum is related to ladder points and ladder heights of the random walks.

Consider $\{X_n\}$ as a set of random variables. Then the discrete-time process $\{Y_n, n = 0, 1, \dots\}$ where

$$Y_0 = 0,$$

$$\begin{aligned} Y_n &= X_1 + X_2 + \dots + X_n \\ &= Y_{n-1} + X_n \quad n = 1, 2, \dots \end{aligned}$$

is called a random walk on the real line. It may be thought of as a particle moving from a state Y_{n-1} at stage $n - 1$ to Y_n at the next stage by an amount X_n .

Therefore, in our case, W_n is the maximum of partial sums of U_n, U_{n-1}, \dots, U_1 . We also have assumed that the queue begins in an empty state, so that $W_0 = 0$. Thus, W_n is also a random walk. However, in section 4.1, we already defined, $U_n = S_n - T_n$ to be the difference between the service time of customer C_n , and the interarrival time of the C_n and C_{n+1} customers, and W_n to be the waiting time (in queue) for n th customer.

One commonly uses the following definitions from [9]:

Definition 4.2.2: Consider the sequence of the points (n, Y_n) for $n = 1, 2, \dots$; the *first strict ascending ladder point* (β_1, γ_1) is the first term in the above sequence for which $Y_n > 0$, i.e.

$$\{\beta_1 = n\} = \{Y_1 \leq 0, Y_2 \leq 0, \dots, Y_{n-1} \leq 0, Y_n > 0\} \text{ and } \gamma_1 = Y_{\beta_1}.$$

The variable β_1 is called *first ladder epoch*, γ_1 is the *first ladder height*.

Definition 4.2.3: Consider the sequence of the points (n, Y_n) for $n = 1, 2, \dots$; the *first weak descending ladder point* $(\bar{\beta}_1, \bar{\gamma}_1)$ is the first term in the sequence for which $Y_n \leq 0$, i.e.

$$\{\bar{\beta}_1 = n\} = \{Y_1 > 0, Y_2 > 0, \dots, Y_{n-1} > 0, Y_n \leq 0\} \text{ and } \bar{\gamma}_1 = Y_{\bar{\beta}_1}.$$

Definition 4.2.4: A probability distribution function F is *defective* if $F(\infty) < 1$, and its *defect* is defined to be $1 - F(\infty)$.

Let $H_n(x)$ and $J_n(x)$ be the joint distribution function of (β_1, γ_1) and $(\bar{\beta}_1, \bar{\gamma}_1)$, then

$$H_n(x) = Pr\{\beta_1 = n, \gamma_1 \leq x\}; \quad x > 0$$

and

$$J_n(x) = Pr\{\bar{\beta}_1 = n, \bar{\gamma}_1 \leq x\}; \quad x \leq 0.$$

The marginal distributions are given by

$$Pr\{\beta_1 = n\} = H_n(\infty), \quad n = 1, 2, \dots,$$

$$Pr\{\gamma_1 \leq x\} = \sum_{n=1}^{\infty} H_n(x) = H(x) \quad x > 0$$

and

$$Pr\{\overline{\beta}_1 = n\} = J_n(\infty) \quad n = 1, 2, \dots,$$

$$Pr\{\overline{\gamma}_1 \leq x\} = \sum_{n=1}^{\infty} J_n(x) = J(x) \quad x \leq 0.$$

The random variables β_1, γ_1 have the same defect $1 - H(\infty)$, and $\overline{\beta}_1, \overline{\gamma}_1$ have the same defect, $1 - J(\infty)$.

The section of the random walk $(0, Y_{j+1} - Y_j, \dots, Y_k - Y_j)$ $k \geq j$ following the first ladder point is a probabilistic replica of the whole random walk. Its first ladder point is the second point of the whole random walk with the property that $Y_n > Y_0, Y_n > Y_1, \dots, Y_n > Y_{n-1}$, and it is called the *second strict ascending ladder point* of the entire random walk. It has the form $(\beta_1 + \beta_2, \gamma_1 + \gamma_2)$. The pairs (β_1, γ_1) and (β_2, γ_2) are i.i.d.

Similarly, we can define r th ascending ladder point (if it exists), and obviously it is in the form of $(\beta_1 + \beta_2 + \dots + \beta_r, \gamma_1 + \gamma_2 + \dots + \gamma_r)$ and the pairs (β_k, γ_k) $k = 1, 2, \dots, r$ are mutually independent and have the common distribution function $H_n(x)$. The expected number of ladder points in $(0, x]$ ($x \geq 0$) can be found as

$$\Psi(x) = \sum_{n=0}^{\infty} H^{*n}(x)$$

where $H^{*0}(x)$ and $\Psi_0(x)$ are defined as

$$\Psi_0(x) = H^{*0}(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

Since the strict ascending ladder heights form a renewal process, one can obtain by using the renewal equation (Feller [9], page 359),

$$\Psi(x) = \Psi_0(x) + \int_0^x \Psi(x-y) dH(y)$$

or

$$\Psi = \Psi_0 + \Psi * H. \tag{4.11}$$

Furthermore, a relation between Ψ and J exists as given by the following theorem:

Theorem 4.1: *If $K(x) = Pr\{X_k \leq x\}$, $k = 0, 1, 2, \dots$, then*

$$J(x) = \int_{0^+}^{\infty} K(x - y) d\Psi(y) \quad x \leq 0 \quad (4.12)$$

and

$$\Psi(x) = \Psi_0(x) - J(0) + \int_{0^+}^{\infty} K(x - y) d\Psi(y) \quad x > 0. \quad (4.13)$$

Proof: (see [9] chap. XII. Section 3)

Since $J(x) = J(0)$ for $x \geq 0$ and $\Psi(x) = 0$ for $x \leq 0$, then (4.12) and (4.13) can be written as

$$J(x) + \Psi(x) = \Psi_0(x) + \Psi * K(x). \quad (4.14)$$

Equation (4.14) is called the **Wiener-Hopf factorization equation**.

Using the renewal equation (4.11) and convolving (4.14) with $\Psi_0 - H$, we have

$$(J + \Psi) * (\Psi_0 - H) = (\Psi_0 + \Psi * K) * (\Psi_0 - H).$$

Hence,

$$J * \Psi_0 - J * H + \Psi * \Psi_0 - \Psi * H = \Psi_0 * \Psi_0 - \Psi_0 * H + \Psi * K * \Psi_0 - \Psi * K * H.$$

Finally, we have

$$K = H + J - J * H. \quad (4.15)$$

Let $K^*(s)$, $H^*(s)$ and $J^*(s)$ be the two-sided Laplace-Stieltjes transform (**LST**) of $K(x)$, $H(x)$ and $J(x)$ respectively, and take the Laplace-Stieltjes transform on both sides of Eq. (4.15), one has

$$K^*(s) = H^*(s) + J^*(s) - J^*(s)H^*(s). \quad (4.16)$$

This gives

$$(1 - K^*(s)) = (1 - H^*(s))(1 - J^*(s)). \quad (4.17)$$

Definition 4.2.5: Laplace transform: Let $f(t)$ be a real-valued function defined on the interval $0 \leq t < \infty$. The Laplace transform (LT) of f is

$$\mathcal{L}\{f(t)\} \equiv \bar{f}(s) \equiv \int_0^{\infty} e^{-st} f(t) dt, \text{ where } s \text{ is a complex variable.}$$

Definition 4.2.6: Laplace-Stieltjes transform: Let $F(t)$ be a real-valued function defined on the interval $0 \leq t < \infty$. The Laplace-Stieltjes transform (LST) of $F(t)$ is

$$\mathcal{L}^*\{F(t)\} \equiv F^*(s) \equiv \int_0^{\infty} e^{-st} dF(t). \text{ We consider } F(t) \text{ to be the CDF of a nonnegative random variable } X, \text{ so } F^*(s) \equiv E[e^{-sX}] = \int_0^{\infty} e^{-st} dF(t).$$

Laplace transform is useful in solving the differential equations and also used for transforming the continuous function. Two useful properties of Laplace transform are the following:

First, there is a one-to-one correspondence between probability distributions and their transforms. Thus a probability distribution can be uniquely determined from its LST.

Second, the LST of the convolution of independent random variables is the product of the LSTs of the individual random variables. That is, if $F^*(s)$ is the LST of X and $G^*(s)$ is the LST of Y , then $F^*(s)G^*(s)$ is the LST of $X + Y$, provided X and Y are independent. Because the inverse of the LST is unique, this allow one to determine the distribution of a sum of random variables by inverting the LST resulting from the product of individual LSTs.

The LST of a random variable evaluated at s is the same as its moment generating function evaluated at $-s$, where s is a complex variable.

Definition 4.2.7: Moment Generating Function (MGF): If X is a random variable with CDF $F(x)$, then

$$M_X(t) \equiv E[e^{tX}] = \int_0^{\infty} e^{tx} dF(x). \text{ This is similar to the LST, but with } t \text{ replacing } -s. \text{ In other words, } M_X(t) = F^*(-t).$$

Moment generating functions have properties similar to Laplace transform. In particular, there is a one-to-one correspondence between moment generating functions and probability

distribution, and the MGF of the sum of independent random variables is the product of the MGFs of the individual random variables.

However, as shown in [9], because of (4.3), $J(x)$ surprisingly gives the idle time distribution. For the waiting time distribution, we have the following result:

Proposition 4.2.1: The waiting time distribution is the same as the distribution of the maximum of the random walk $\{Y_n\}$. The distribution is given as

$$\begin{aligned} W(t) &= [1 - H(\infty)] \sum_{n=0}^{\infty} H^{*n}(t) \\ &= [1 - H(\infty)] \Psi(t), \end{aligned} \tag{4.18}$$

where $\Psi(t)$ is completely determined by $H(t)$, because it is the renewal measure generated by $H(t)$.

The Wiener-Hopf factorization in Eq. (4.15) can be used for both discrete and continuous cases, but for numerical application purposes, we consider a discrete GI/G/1 queue. Eq.(4.16) is derived by taking the LST on both sides of Eq. (4.15), therefore this relation in (4.16) is Wiener-Hopf factorization for continuous case. On the other hand, the Z -transform plays an important role in discrete analysis. Its role in discrete analysis is same as that of Laplace transform in continuous system. Thus, if we use Z -transform in a function that we get by using Laplace transforms in (4.16), we can get a relation for Wiener-Hopf factorization in discrete case.

Definition 4.2.8: Z-transform: In Mathematics and signal processing, the Z -transform converts a discrete-time domain signal, which is a sequence of real or complex numbers, into a complex frequency domain representation. The Z -transform of a sequence $\{x_n\}$ is denoted as $Z[\{x_n\}]$. It is defined as

$$Z[\{x_n\}] = X(z) = \sum_{n=-\infty}^{\infty} x_n z^{-n}, \text{ where } z \text{ is a complex number and } X(z) \text{ is the } Z\text{-transform of } \{x_n\}.$$

We already defined the generating function and probability generating function in definition 2.15 of chapter 2. The generating function is closely related to the one sided z -transform and defined as $\sum_{n=0}^{\infty} x_n z^{-n}$. On the other hand, the probability generating function for an integer-

valued random variable is similar to the moment generating functions, but with z replacing e^t , i.e. $X(z) \equiv E[z^X]$. In definition 4.2.5, we define $M_X(t) \equiv E[e^{tX}] = \int_0^\infty e^{tx} dF(x)$. If we use Z -transform in this relation, we get $X(e^{-t}) = \sum_{m=0}^\infty x(m)e^{tm}$, which implies

$X(z) = \sum_{m=0}^\infty x(m)z^{-m}$, by replacing e^{-t} with z and we assume m to take integer values from 0 to ∞ .

According to the definition of random walk, the strict ascending ladder heights and the weak descending ladder heights, we construct the following probability distribution functions:

$$u_i = Pr\{U_k = i\} \quad -g \leq i \leq h$$

$$\alpha_i = Pr\{\gamma_k = i\} \quad i = 1, 2, \dots$$

$$b'_i = Pr\{\bar{\gamma}_k = -i\} \text{ and } b_i = b'_{-i} = Pr\{\bar{\gamma}_k = i\} \quad i = 0, 1, \dots$$

and also define the following probability generating function for the random variables U , A , and B , respectively:

$$U(z) = \sum_{i=-\infty}^\infty u_i z^i, \quad A(z) = \sum_{i=1}^\infty \alpha_i z^i, \quad \text{and } B(z) = \sum_{i=0}^\infty b_i z^i.$$

Here A is the strong ascending ladder height and B is the weak descending ladder height, and z is a complex number.

Now, using Z -transform in Eq. (4.16), we get the following relation

$$U(z) = A(z) + B(1/z) - A(z)B(1/z) \tag{4.19}$$

which is equivalent to

$$1 - U(z) = (1 - A(z))(1 - B(1/z)) \tag{4.20}$$

which is known as the Wiener-Hopf factorization of $U(z)$ in discrete case for the GI/G/1 queue. We discuss more about the use of Wiener-Hopf factorization in (4.20), in the following section.

By equating the coefficients of $z^j, j = 0, 1, 2, \dots$ and $z^{-j}, j = 1, 2, \dots$, we get

$$\begin{aligned}
u_j &= \alpha_j - \sum_{i=-\infty}^0 b'_i \alpha_{j-i} \\
&= \alpha_j - \sum_{i=1}^{\infty} b'_{-i} \alpha_{j+i} - b'_0 \alpha_j \\
&= \alpha_j - \sum_{i=1}^{\infty} \alpha_{j+i} b_i - b_0 \alpha_j, \tag{4.21}
\end{aligned}$$

or,

$$\alpha_j = (u_j + \sum_{i=1}^{\infty} \alpha_{j+i} b_i) / (1 - b_0)$$

and

$$u_{-j} = b'_{-j} - \sum_{i=1}^{\infty} b'_{-j-i} \alpha_i = b_j - \sum_{i=1}^{\infty} b_{j+i} \alpha_i,$$

$$\text{or, } b_j = u_{-j} + \sum_{i=1}^{\infty} b_{j+i} \alpha_i. \tag{4.22}$$

Depending on the above derivation of Wiener –Hopf factorization, the following theorem was proposed in [12],

Theorem 4.2: (Wiener –Hopf factorization)

For $\rho < 1$ (where $\rho = \lambda/\mu$) there exists a unique defective distribution given by $\alpha_i = \Pr\{A = i\}, i > 0$, and a unique proper distribution given by $b_i = \Pr\{B = i\}, i \geq 0$ such that,

$$b_j = u_{-j} + \sum_{i=1}^{\infty} \alpha_i b_{i+j}, \quad j = 0, 1, \dots, g \tag{4.23}$$

$$\alpha_j = \frac{u_j + \sum_{i=1}^{\infty} \alpha_{i+j} b_i}{1 - b_0}, \quad j = 1, 2, \dots, h. \tag{4.24}$$

The Wiener-Hopf factorization of $U(z)$ is as follows,

$$1 - U(z) = (1 - A(z))(1 - B(1/z)) \tag{4.25}$$

with

$$A(z) = \sum_{i=1}^{\infty} \alpha_i z^i \text{ and } B(z) = \sum_{i=0}^{\infty} b_i z^i \tag{4.26}$$

Eq. (4.23) and (4.24) can easily be derived similar to Eq. (4.21) and (4.22), which represent the Wiener-Hopf factorization for the discrete case and they also can easily derived from ([9], page 400).

Since we considering the waiting time distribution of the arithmetic GI/G/1 queue, when queue is bounded on a finite interval, for that reason, we assume $U(z) = \sum_{i=-g}^h u_i z^i$ where we set $u_i = 0$ for $i < -g$ and $i > h$. This convention allows us to make the sum from $-\infty$ to $+\infty$.

Moreover, we consider $w_i = \Pr\{W = i\}$ is the equilibrium distribution of W_n and $W(z) = \sum_{i=0}^{\infty} w_i z^i$ denote the probability generating functions (*p.g.f*) of the random variable W .

According to the discrete version of Lindley's equation, w_j is given by

$$w_j = \sum_{i=-g}^h w_{j-i} u_i, \quad j > 0$$

and (4.27)

$w_0 = \sum_{i=-g}^0 \sum_{k=0}^{-i} w_k u_i$ (when there are no wait in queue), where we assume that $w_k = 0$ for $k < 0$.

Theorem 4.2 is helpful to find the solution of (4.27).

Example 4.2.1: To understand clearly the formation of Lindley's equation in our case, which is given by equation (4.27), consider a transition matrix say, M of U where service time and interarrival time are discrete, and also consider, $g = h = 2$,

		S				
		0	1	2	3	4
T	0	u_0	u_1	u_2	0	0
	1	u_{-1}	u_0	u_1	u_2	0
	2	u_{-2}	u_{-1}	u_0	u_1	u_2
	3	0	u_{-2}	u_{-1}	u_0	u_1
	4	0	0	u_{-2}	u_{-1}	u_0

Here, the columns represent the service time and the rows represent the interarrival time. Suppose $wM = w$ in an equilibrium situation, where M denotes the above transition matrix. Then, $[w_0, w_1, \dots]M = [w_0, w_1, \dots]$.

Now, if we use the second column, we get

$w_1 = w_0u_1 + w_1u_0 + w_2u_{-1} + w_3u_{-2} = \sum_{i=-2}^2 w_{1-i}u_i$, which is same as w_j in equation (4.27) when $j = 1$.

4.3 Our Explanation for using the Wiener-Hopf Factorization Method in $1 - U(z)$

Consider an expression in terms of a power series of $U(z)$ for $-g \leq i \leq h$ and expand it after using the Z -transform,

$$U(z) = u_{-g}z^g + \dots + u_{-2}z^2 + u_{-1}z^1 + u_0z^0 + u_1z^{-1} + u_2z^{-2} + \dots + u_hz^{-h} \quad (4.28)$$

If we substitute $z = \frac{1}{z}$ in (4.28), we get the following relation for $U(z)$,

$$U\left(\frac{1}{z}\right) = u_{-g}z^{-g} + \dots + u_{-2}z^{-2} + u_{-1}z^{-1} + u_0z^0 + u_1z^1 + u_2z^2 + \dots + u_hz^h$$

This series in (4.28) is known as Laurent's series that contains negative and positive finite powers of z . The series $U(z)$ converges when $0 < |z| \leq 1$.

Moreover, if we consider $U\left(\frac{1}{z}\right) = R(z)$, then we will get $R(z) = \sum_{i=-g}^h r_i z^i$, where $R(z)$ has the same form of a probability generating function.

We have from (4.19),

$U(z) = A(z) + B(1/z) - A(z)B(1/z)$ which is known as the Wiener-Hopf factorization of $U(z)$ in discrete case for the GI/G/1 queue. Then, we get (4.20), after factoring the above relation which is

$1 - U(z) = (1 - A(z))(1 - B(1/z))$. The main reason of factoring Eq. (4.19) is to obtain two separate expressions for $A(z)$ and $B(1/z)$ in R.H.S. Surprisingly, it is factored in such a way that $(1 - B(1/z))$ contains the constant and positive powers of z and $(1 - A(z))$ contains the negative powers of z .

Clearly, (4.28) is equivalent to

$$1 - U(z) = 1 - u_{-g}z^g - \dots - u_{-2}z^2 - u_{-1}z^1 - u_0z^0 - u_1z^{-1} - u_2z^{-2} - \dots - u_hz^{-h}.$$

and (4.20) can also be expressed in the following way

$$1 - u_{-g}z^g - \dots - u_{-2}z^2 - u_{-1}z^1 - u_0z^0 - u_1z^{-1} - u_2z^{-2} - \dots - u_hz^{-h} = (1 - \alpha_1z^{-1} - \alpha_2z^{-2} - \dots - \alpha_hz^{-h})(1 - b_0 - b_1z^1 - b_2z^2 - \dots - b_gz^g) \quad (4.29)$$

where $U(z) = \sum_{i=-g}^h u_i z^{-i}$, $A(z) = \sum_{i=1}^h \alpha_i z^{-i}$ and $B(z) = \sum_{i=0}^g b_i z^{-i}$.

Previously, we discussed how to convert a moment generating function of the continuous case to the discrete case, by using the Z -transformation. Moreover, we defined $X(z) = \sum_{n=0}^{\infty} x_n z^{-n}$ and clearly $X(1/z) = \sum_{n=0}^{\infty} x_n z^n$. It is also noticed that in moment generating function $M_X(t) \equiv E[e^{tX}] = \int_0^{\infty} e^{tx} dF(x)$, if $t < 0$, then e^t converges inside the unit circle and if $t > 0$, then e^t converges outside the unit circle. Furthermore, we know that a probability generating function converges inside the unit circle and therefore, $X(1/z) = \sum_{n=0}^{\infty} x_n z^n$, will converge inside the unit circle. On the other hand, it is clear that the function we get by using Z -transformation, $X(z) = \sum_{n=0}^{\infty} x_n z^{-n}$, will converge outside the unit circle. Moreover, in the property of the Z -transform, it is assured that the Z -transform of a causal sequence converges outside the unit circle.

The number of degree in $1 - U(z)$ is $g + h$. Among the two factors in R.H.S of (4.29), clearly $A(z)$ has the same formation as the Z -transform, thus it converges outside the unit circle and contains h zeros in $\{z: |z| > 1\}$ that means h zeros lie outside the unit circle. On the other hand, $B(1/z)$ converges inside the unit circle and it contains g zeros where g zeros in $\{z: |z| < 1\}$ which means that g zeros lie inside or on the unit circle. Therefore, by using the Wiener-Hopf Factorization method in (4.29), we also can locate the zeros inside and outside of the unit circle easily.

The Wiener-Hopf factorization given by theorem 4.2 have some probabilistic interpretations; A is the strong ascending ladder height and B is the weak descending ladder height. Moreover, α_i is the probability that the server remains busy (always have a queue) and b_j is the probability that the server remains idle. It is very interesting to notice that α_i and b_j in

Eq. (4.21) and (4.22) also have some probabilistic interpretations as indicated in [11] and [13]. For example, α_i is interpreted as the probability that the system goes from state $j - i$ to state j without visiting any state $k < j$ in between.

Therefore, the Wiener-Hopf factorization method is significant in research on queueing systems.

Moreover, Equation (4.23) and (4.24) represent the solutions of Wiener-Hopf factorization for the discrete case can easily be derived from equation (4.25) by equating the coefficients of $z^j, j = 1, 2, \dots, h$ and $z^{-j}, j = 0, 1, 2, \dots, g$ if we expand (4.25) as the form of a *p.g.f* in the following way:

Table 4.1 Results by equating the coefficients of z^j and z^{-j} of Eq. (4.25)

	L.H.S	R.H.S
Constant power of z	$1 - u_0$	$(1 - b_0) + \sum_{i=1}^{\infty} \alpha_i b_i$
z^{-j}	$-u_{-j}$	$\sum_{i=1}^{\infty} \alpha_i b_{i+j} - b_j$
z^j	$-u_j$	$(1 - b_0)\alpha_j + \sum_{i=1}^{\infty} \alpha_{i+j} b_i$

Therefore, by equating the coefficients of z^{-j} , we get

$$-u_{-j} = \sum_{i=1}^{\infty} \alpha_i b_{i+j} - b_j \text{ which implies } b_j = u_{-j} + \sum_{i=1}^{\infty} \alpha_i b_{i+j}, j = 0, 1, \dots, g,$$

and by equating the coefficients of z^j , we have

$$-u_j = (1 - b_0)\alpha_j + \sum_{i=1}^{\infty} \alpha_{i+j} b_i \text{ which implies } \alpha_j = \frac{u_j + \sum_{i=1}^{\infty} \alpha_{i+j} b_i}{1 - b_0}, j = 1, 2, \dots, h.$$

Example 4.3.1: To clarify the solution of Wiener-Hopf factorization, let us discuss it with an example by representing (4.25) as a matrix notation, where we consider $g = 2$ and $h = 3$:

	$1 - b_0$	b_1	b_2
$-\alpha_1$	$-\alpha_1(1 - b_0)$	$-\alpha_1 b_1$	$-\alpha_1 b_2$
$-\alpha_2$	$-\alpha_2(1 - b_0)$	$-\alpha_2 b_1$	$-\alpha_2 b_2$
$-\alpha_3$	$-\alpha_3(1 - b_0)$	$-\alpha_3 b_1$	$-\alpha_3 b_2$

Here, $\alpha_j, j = 1, 2, 3$ is the coefficients of z^j and $b_j, j = 0, 1, 2$ is the coefficients of z^{-j} . To get u_0 , we have to add all the coefficients of z^0 diagonally, from the above matrix, which are definitely the constant part of (4.25). Moreover, we can get (4.24) for $j = 1$ by equating the coefficients of z^1 and adding them together. If we add all coefficients of z^{-1} , we get (4.23) for $j = 1$.

Once α_i are found, the waiting time distribution can be obtained without difficulties, as the following theorem shows and we give our explanation to proof it:

Theorem 4.3: *The probability generating function (p.g.f) $W(z)$ is given by*

$$W(z) = w_0 / (1 - A(z)), \quad (4.30)$$

or,

$$w_i = 0 \text{ when } i < 0,$$

$$w_0 = 1 - \alpha_1 - \alpha_2 - \dots - \alpha_n, \quad (4.31)$$

$$w_i = \sum_{j=1}^n w_{i-j} \alpha_j \quad i > 0. \quad (4.32)$$

Proof: Eq. (4.32) is the discrete version of Eq. (4.18). Eq. (4.32) can be found directly by applying the Wiener- Hopf method to solve Lindley's equation.

Furthermore, Eq. (4.30) and (4.31) can be derived from (4.32) by using p.g.f $W(z)$ and the fact $\sum_{i=0}^{\infty} w_i = 1$. The derivation of (4.30) is shown as follows,

$$\begin{aligned} W(z) &= \sum_{i=0}^{\infty} w_i z^i \\ &= w_0 + \sum_{i=1}^{\infty} w_i z^i \end{aligned}$$

$$\begin{aligned}
&= w_0 + \sum_{i=1}^{\infty} (\sum_{j=1}^h w_{i-j} \alpha_j) z^i \quad [\text{Since, } w_i = \sum_{j=1}^h w_{i-j} \alpha_j] \\
&= w_0 + \sum_{j=1}^h \alpha_j \sum_{i=1}^{\infty} w_{i-j} z^i \\
&= w_0 + \sum_{j=1}^h \alpha_j \sum_{k=0}^{\infty} w_k z^{k+j} \quad [\text{By replacing } i-j \text{ by } k] \\
&= w_0 + \sum_{j=1}^h \alpha_j z^j (\sum_{k=0}^{\infty} w_k z^k) \\
&= w_0 + A(z)W(z).
\end{aligned}$$

Rearranging the relation, we get

$$W(z)(1 - A(z)) = w_0,$$

Thus, $W(z) = \frac{w_0}{(1-A(z))}$ which is our Eq.(4.30).

We know the *p.g.f* of W is $W(z) = \sum_{i=0}^{\infty} w_i z^i$ and we can write this in the following way

$W(z) = w_0 + w_1 z + \dots + w_h z^h + \dots$, where $W(z)$ is analytic in $|z| < 1$. Thus the series converges when $|z| \leq 1$.

In order to interpret the formation of Eq. (4.30), we consider

$W(z)(1 - U(z)) = V(z)$, where $V(z)$ is such a function that satisfies this equation.

$$\text{Therefore, } (z) = \frac{V(z)}{(1-U(z))} = \frac{V(z)}{(1-A(z))(1-B(1/z))}.$$

As we know, $W(z)$ is analytic in unit circle on L.H.S, so the R.H.S must be analytic in unit circle in the above relation. However, a rational function can be analytic if the denominator is not equal to zero inside the unit circle. We know that the zeros of $(1 - B(1/z))$ lie inside the unit circle in the denominator. On the other hand, we assume $V(z)$ is a polynomial that also has the same number of zeros as $(1 - B(1/z))$ inside the unit circle. Therefore, we can cancel the common zeros from the numerator and denominator. Finally, the rest of the zeros in the numerator are considered together as a constant term, w_0 . Thus, ultimately we get $W(z) = \frac{w_0}{1-A(z)}$ which is clearly analytic in $|z| < 1$. Therefore, one needs to find α_i to calculate the waiting time distribution as it is the probability that the system remains busy.

Definition 4.3.1: Analytic Function: A single valued function $f(z)$ which is differentiable at $z = z_0$ is said to be *Analytic* at the point $z = z_0$.

By taking the first derivative of Eq. (4.30), in terms of z and evaluating at $z = 1$, we can easily find the expectation of the waiting time,

$$W(z) = \frac{w_0}{1 - \sum_{i=1}^{\infty} \alpha_i z^i} = w_0 (1 - \sum \alpha_i z^i)^{-1}.$$

$$\begin{aligned} W'(z) &= w_0 \left[\left\{ -(1 - \sum \alpha_i z^i)^{-2} \right\} \cdot \left\{ -\sum i \alpha_i z^{i-1} \right\} \right], \\ &= \frac{w_0 \cdot \sum i \alpha_i z^{i-1}}{(1 - \sum \alpha_i z^i)^2}. \end{aligned}$$

By putting $z = 1$, we get

$$W'(1) = E(W) = \frac{w_0 \cdot \sum i \alpha_i}{w_0^2} = \frac{\sum i \alpha_i}{w_0} \quad (\text{since } w_0 = 1 - \sum_{i=1}^h \alpha_i).$$

$$\text{Thus, } E(W) = \sum i \alpha_i / w_0.$$

In order to obtain the variance of the waiting time distribution, we are considering the *p.g.f* of W which is $W(z) = \sum_{i=0}^{\infty} w_i z^i$.

By taking the first derivative of $W(z)$ in terms of z and evaluating at $z = 1$, we obtain,

$$W'(z) = \sum_{i=1}^{\infty} i w_i z^{i-1} \text{ and } W'(1) = E[W] = \sum_{i=1}^{\infty} i w_i.$$

We need to take the second derivatives of $W(z)$ in terms of z and need to evaluate at $z = 1$, for calculating the variance of the waiting time; since we know, $Var [X] = E[X^2] - E^2[X]$.

$$W''(z) = \sum_{i=2}^{\infty} i(i-1) w_i z^{i-2} \text{ and } W''(1) = E[W(W-1)] = \sum_{i=2}^{\infty} i(i-1) w_i.$$

Therefore,

$$\begin{aligned} Var(W) &= (E[W^2] - E[W]) + E[W] - E^2(W), \\ &= \sum_{i=2}^{\infty} i(i-1) w_i + \sum_{i=1}^{\infty} i w_i - E^2(W), \end{aligned}$$

$$\begin{aligned}
&= \sum_{i=2}^{\infty} i^2 w_i - \sum_{i=2}^{\infty} i w_i + \sum_{i=1}^{\infty} i w_i - E^2(W), \\
&= \sum_{i=1}^{\infty} i^2 w_i - w_1 - \sum_{i=1}^{\infty} i w_i + w_1 + \sum_{i=1}^{\infty} i w_i - E^2(W), \\
&= \sum_{i=1}^{\infty} i^2 w_i - E^2(W).
\end{aligned}$$

Finally, we note that B represents the idle time, and the moments of the idle time can be found by straightforward calculation. From equation (4.26), it is easy to see that

$B(z) = \sum_{i=0}^{\infty} b_i z^i$. By differentiating with respect to z , we get

$B'(z) = \sum_{i=0}^{\infty} i b_i z^{i-1}$. Now the moment of idle time at $z = 1$ is $E(B) = B'(1) = \sum_{i=0}^{\infty} i b_i$.

4.4 Wiener-Hopf Factorization by Successive Substitution

In this section, we will show a method to determine the α_i and b_j from (4.23) and (4.24). When doing this, one may conceivably obtain the wrong factorization. Hence, criteria to determine whether or not a given factorization is Wiener-Hopf, become important. In [12], the following conditions are sufficient to assure that a given factorization is the correct Wiener –Hopf factorization.

Theorem 4.4: [*Sufficient condition for Wiener –Hopf factorization*]

$$\alpha_i \geq 0, i = 1, 2, \dots \quad (4.33)$$

$$b_j \geq 0, j = 0, 1, 2, \dots \quad (4.34)$$

$$\sum_{j=0}^{\infty} b_j = 1 \quad (4.35)$$

To satisfy (4.34), set $S = 1 - b_0 = \sum_{j=1}^{\infty} b_j$ and notice that $a_i = \alpha_i(1 - b_0)$.

If we consider, $a_0 = b_0$, $a_i = \alpha_i(1 - b_0)$, and $S = 1 - b_0 = 1 - a_0$, and using these assumptions in Eq. (4.21) and (4.22) , we can get the two new solutions of Wiener –Hopf factorization, which are as follows:

$$a_j = (u_j + \sum_{i=1}^{\infty} a_{i+j} b_i) / S \quad j \geq 0 \quad (4.36)$$

and

$$b_j = u_{-j} + \sum_{i=1}^{\infty} a_i b_{i+j} / S \quad j \geq 0 \quad . \quad (4.37)$$

The recursive equation (4.36) and (4.37) are the starting point of the algorithm 1 in [12]. In [12], they combine (4.21), (4.22) to give the following three algorithms in order to obtain α_i and b_j . Here we give a proof for Algorithm 1, where the solution obtained in this way always converges to the Wiener-Hopf factorization.

Algorithm 1 by [12]:

1. $a_i^0 = 0, \quad i = 1, 2, \dots, h$

$$b_j^0 = 0, \quad j = 0, 1, 2, \dots, g \quad .$$

2. For $m = 0, 1, 2, \dots$ do the following until $\max(|a_i^m - a_i^{m+1}|) < \epsilon$.

- 2.1. $b_j^{m+1} = u_{-j} + \sum_{i=1}^{\infty} a_i^m b_{i+j}^m / (1 - b_0^m), \quad j = 0, 1, \dots, g.$

- 2.2. $a_j^{m+1} = u_j + \sum_{i=1}^{\infty} a_{i+j}^m b_i^m / (1 - b_0^m), \quad j = 1, 2, \dots, h.$

Proof: We want to prove that the values of a_j^m and b_j^m of Algorithm 1 converge to the Wiener-Hopf factorization.

To see this, set up two sequences $\{a_i^m\}$ and $\{b_j^m\}$ as follows:

1. For $j = 0, 1, \dots, g$ and $i = 1, 2, \dots, h$; it is given that $b_j^0 = 0$ and $a_i^0 = 0$.

Define

- 2.1. $b_j^{m+1} = u_{-j} + \sum_{i=1}^{\infty} a_i^m b_{i+j}^m / (1 - b_0^m)$

- 2.2. $a_i^{m+1} = u_i + \sum_{k=1}^{\infty} a_{k+i}^m b_k^m / (1 - b_0^m)$ and they work recursively in terms of m .

Let a_i and b_j be the solutions of the Wiener-Hopf factorization, and let $a_i^m \leq a_i$ for all m .

We have,

$$a_i^{m+1} = u_i + \sum_{k=1}^{\infty} a_{k+i}^m b_k^m / (1 - b_0^m)$$

$$\leq u_i + \sum_{k=1}^{\infty} a_{k+i} b_k / (1 - b_0) = a_i \text{ [from Eq. (4.36)]}$$

$$\text{Thus, } a_i^{m+1} \leq a_i \text{ for all } m \text{ and clearly } a_i^{m+1} \geq a_i^m. \quad (4.38)$$

Similarly, for $\{b_j^m\}$ if $b_j^m \leq b_j$, for all m , then

$$\begin{aligned} b_j^{m+1} &= u_{-j} + \sum_{i=1}^{\infty} a_i^m b_{i+j}^m / (1 - b_0^m) \\ &\leq u_{-j} + \sum_{i=1}^{\infty} \frac{a_i b_{i+j}}{1 - b_0} = b_j \end{aligned}$$

Correspondingly for a_i , we also have

$$a_i^{(m+1)+1} = u_i + \sum_{k=1}^{\infty} a_{k+i}^{m+1} b_k^{m+1} / (1 - b_0^{m+1}) \text{ which implies } a_i^{m+2} \geq a_i^{m+1}. \quad (4.39)$$

Note: Let, $b_0^m \leq b_0$ or, $-b_0^m \geq -b_0$

$$\text{i.e. } 1 - b_0^m \geq 1 - b_0 \text{ or, } \frac{1}{1 - b_0^m} \leq \frac{1}{1 - b_0}.$$

We now prove that $a_i^{m+1} \geq a_i^m$.

This is clearly true for $m = 0$, because we get $a_i^1 \geq a_i^0$ [This is also clear from step 1 and the 1st iteration].

We then use induction to prove it for all m . Suppose, the statement $S(m)$ in our case is:
 $S(m) = a_i^{m+1} \geq a_i^m$.

When $m = 1$, $a_i^2 \geq a_i^1$ and consequently, the statement is also true for $m = 1$.

Now, if $a_i^{m+1} \geq a_i^m$, then clearly, $a_i^{m+2} \geq a_i^{m+1}$ and similarly, we can show that $b_j^{m+1} \geq b_j^m$ and $b_j^{m+2} \geq b_j^{m+1}$.

Thus, the statement is true by the principle of induction, which follows that a_i^m is bounded and monotonic with respect to m . In a similar way, we can show that b_j^m is also bounded and monotonic.

So, the sequences $\{a_i^m\}$ and $\{b_j^m\}$ converge to the Wiener-Hopf solution.

Algorithm 2 by [12]:

1. $\alpha_i^0 = 0, i = 1, 2, \dots, h.$

2. For $m = 0, 1, 2, \dots$ do the following until $\max(|\alpha_i^m - \alpha_i^{m+1}|) < \epsilon.$

2.1. $b_j^m = u_{-j} + \sum_{i=1}^{\infty} \alpha_i^{m-1} b_{j+i}^m, \quad j = g, g-1, \dots, 0.$

2.2. $S^m = \sum_{j=1}^{\infty} b_j^m.$

2.3. $\alpha_j^m = \frac{(u_j + \sum_{i=1}^{\infty} \alpha_{i+j}^m b_i^m)}{S^m}, \quad j = h, h-1, \dots, 1.$

In Algorithm 2 in [12], the sequence $\{\alpha_j^m\}$ is considered instead of $\{a_i^m\}$ and used $S^m = \sum_{j=1}^{\infty} b_j^m$ in step 2.2 to get α_j^m . For this algorithm [12], it is assumed that α_j^m and b_j^m converges to the Wiener-Hopf factorization, and it is also verified that Algorithm 2 will always satisfy (4.32), (4.33) and (4.34). Hence, they did not show it theoretically that α_j^m remains bounded, we derive a simple proof to show that α_j^m is indeed bounded, as it follows:

Step 2.1 assures that $b_j^m > u_{-j},$

which implies that $S^m > \sum_{i=1}^{\infty} u_{-i}.$

From step 2.3, in [12] is already shown that

$$\begin{aligned} \alpha_j^m &= \frac{u_j}{S^m} + \sum_{i=1}^{\infty} \alpha_{i+j}^m \frac{b_i^m}{\sum_{k=1}^{\infty} b_k^m} \\ &< \frac{u_j}{\sum_{i=1}^{\infty} u_{-i}} + \max_{k>j} \{\alpha_k^m\}. \end{aligned}$$

To see that the above inequality is true, let $L = \max_{k>j} \{\alpha_k^m\}.$ Thus, $\alpha_{i+j}^m \leq L$ [Since $i + j >$

$$\begin{aligned} j] \text{ and clearly, } \sum_{i=1}^{\infty} \alpha_{i+j}^m \frac{b_i^m}{\sum_{k=1}^{\infty} b_k^m} &\leq \sum_{i=1}^{\infty} L \cdot \frac{b_i^m}{\sum_{k=1}^{\infty} b_k^m} \\ &= L \cdot \sum_{i=1}^{\infty} \frac{b_i^m}{\sum_{k=1}^{\infty} b_k^m} \\ &= L \cdot \frac{\sum_{i=1}^{\infty} b_i^m}{\sum_{k=1}^{\infty} b_k^m} \end{aligned}$$

$$= L = \max_{k>j} \{\alpha_k^m\}.$$

The following algorithm (Algorithm 3 in [12]) is a combined form of some of the features of Algorithm 1 and 2. In [12], it was also not proved theoretically that the given sequences converge, nor it obtains the Wiener-Hopf factorization.

Algorithm 3 by [12]:

1. $\alpha_i^0 = 0, \quad i = 1, 2, \dots, h.$

2. For $m = 0, 1, 2, \dots$ do the following until $\max(|\alpha_i^m - \alpha_i^{m+1}|) < \epsilon$ for given $\epsilon > 0$.

- 2.1. $b_j^m = u_{-j} + \sum_{i=1}^{\infty} \alpha_i^{m-1} b_{j+i}^m, \quad j = g, g-1, \dots, 0.$

- 2.2. $S^m = \frac{(1 - b_0^m + \sum_{j=1}^g b_j^m)}{2}.$

- 2.3. $\alpha_j^m = \frac{(u_j + \sum_{i=1}^{\infty} \alpha_{i+j}^m b_i^m)}{S^m}, \quad j = h, h-1, \dots, 1.$

Like [12], we couldn't prove mathematically that Algorithm 3 converges to the Wiener-Hopf factorization.

Moreover, we have the following important theorem from [12] and we add our explanation to make the proof clear:

Theorem 4.5:

There are exactly two solutions satisfying (4.23), (4.24), (4.33) and (4.34). Of these two solutions, one satisfies (4.35), whereas the other one satisfies

$$\sum_{i=1}^{\infty} \alpha_i = A(1) = 1. \tag{4.40}$$

Proof: At first, it is showed that there are two factorizations satisfying (4.23), (4.24), (4.33) and (4.34):

Let $\alpha_i^{(1)}$ and $b_j^{(1)}$ be the Wiener-Hopf solution, and let $A^1(z)$ and $B^1(z)$ be the corresponding p.g.f.s. We now generate a second solution as follows: $\alpha_i^{(2)}$, $b_j^{(2)}$, $A^2(z)$ and $B^2(z)$. Because $A^1(z)$ is the p.g.f of a defective distribution, $A^1(z) < 1$. For $z > 0$, $A^1(z)$ is monotonously increasing with $A^1(\infty) = \infty$.

Note: We know, $A^1(z) = \sum_{i=1}^{\infty} \alpha_i^1 z^i$ and $A^1(1) = \sum_{i=1}^{\infty} \alpha_i < 1$. Since, $A^1(0) = 0$, thus $A^1(z)$ increases monotonously when $z > 0$. Moreover, $1 - A^1(z)$ is decreasing, but $1 - A^1(0) = 1$, hence $1 - A^1(z)$ has one root.

Consequently, $1 - A^1(z)$ has exactly one positive zero $z = z_0$, where $z_0 > 1$. Using this zero and also the zero $z = 1$ of $1 - B^1(1/z)$, one can write (4.25) as

$$1 - U(z) = ((1 - z/z_0)A^*(z))((1 - 1/z)B^*(1/z)), \quad (4.41)$$

$$\begin{aligned} \text{Since, } 1 - A^1(z) &= (z_0 - z)\bar{A}(z) = (1 - z/z_0)(z_0\bar{A}(z)), \\ &= (1 - z/z_0)A^*(z) \text{ and} \end{aligned}$$

$$\begin{aligned} 1 - B^1(1/z) &= (z - 1)\bar{B}(z) = (1 - 1/z)(z\bar{B}(z)) \\ &= (1 - 1/z)B^*(1/z), \end{aligned}$$

where, $A^*(z)$ and $B^*(z)$ are polynomials of the form

$$\begin{aligned} A^*(z) &= 1 + \alpha_1^* z + \alpha_2^* z^2 + \dots + \alpha_{h-1}^* z^{h-1} \\ B^*(z) &= (1 - b_0) + b_1^* z + b_2^* z^2 + \dots + b_{g-1}^* z^{g-1}. \end{aligned}$$

Here, 1 and $(1 - b_0)$ of $A^*(z)$ and $B^*(z)$ are considered to be the constant part, respectively, where 1 comes in both cases, by factoring the polynomials $1 - A^1(z)$ and $1 - B^1(1/z)$.

Since $1 - A^1(z) = (1 - z/z_0)A^*(z)$, one has

$$1 - \sum_{i=1}^{\infty} \alpha_i^1 z^i = (1 - z/z_0) \sum_{i=1}^{h-1} \alpha_i^* z^i.$$

By expanding $\sum_{i=1}^h \alpha_i^1 z^i$ and $\sum_{i=1}^{h-1} \alpha_i^* z^i$, we get

$$1 - \alpha_1^1 z^1 - \alpha_2^1 z^2 - \dots - \alpha_h^1 z^h = \sum_{i=1}^{h-1} \alpha_i^* z^i - z/z_0 (\alpha_1^* z^1 + \alpha_2^* z^2 + \dots + \alpha_{h-1}^* z^{h-1})$$

By rearranging the above, one yields

$$\begin{aligned} 1 - \alpha_1^1 z^1 - \alpha_2^1 z^2 - \dots - \alpha_h^1 z^h \\ = (\alpha_1^* z^1 + \alpha_2^* z^2 + \dots + \alpha_{h-1}^* z^{h-1}) - 1/z_0 (\alpha_1^* z^2 + \alpha_2^* z^3 + \dots + \alpha_{h-1}^* z^h) \end{aligned}$$

Now, equating the coefficients of z^1, z^2, \dots, z^h , we finally get

$$-\alpha_i^{(1)} = \alpha_i^* - \alpha_{i-1}^*/z_0, \quad i = 1, 2, \dots \quad (4.42)$$

Equation (4.42), together with $\alpha_h^* = 0$, implies that $\alpha_i^*, i > 0$ is nonnegative, nonincreasing in i , as we subtract α_{i-1}^*/z_0 from α_i^* . This can easily be shown by complete induction.

Similarly, b_j^* satisfies

$$-b_j^{(1)} = b_j^* - b_{j-1}^*. \quad (4.43)$$

b_j^* are also nonnegative. In fact, one has

$$b_j^* = 1 - b_0^{(1)} - b_1^{(1)} - \dots - b_j^{(1)} \geq 0 \quad (\text{Because of equation (4.33) and (4.34)}).$$

Since,

$$(1 - z/z_0)(1 - 1/z) = 1/z_0 (1 - z)(1 - z_0/z).$$

Equation (4.25) can also be written as

$$\begin{aligned} 1 - U(z) &= z_0^{-1} (1 - z) A^*(z) (1 - z_0/z) B^*(1/z), \\ &= (1 - A^2(z))(1 - B^2(1/z)). \end{aligned}$$

where,

$$A^2(z) = (1 - z) A^*(z) \quad (4.44)$$

$$B^2(z) = (1 - z_0/z) B^*(z) / z_0. \quad (4.45)$$

Hence, the $A^2(z)$ and $B^2(z)$ define a factorization of $1 - U(z)$, and it remains to show that all the coefficients $\alpha_i^{(2)}$ and $b_j^{(2)}$ are nonnegative. Using (4.42) and (4.44), one has

$$\begin{aligned} -\alpha_i^{(2)} &= \alpha_i^* - \alpha_{i-1}^* \\ &< \alpha_i^* - \alpha_{i-1}^*/z_0 = -\alpha_i^{(1)} \leq 0. \end{aligned}$$

Similarly, from (4.45), one finds

$$\begin{aligned} -b_j^{(2)} &= b_j^* - z_0 b_{j-1}^* \\ &< b_j^* - b_{j-1}^* = -b_j^{(1)}. \end{aligned}$$

Now, we can conclude that

$$\alpha_i^{(2)} > \alpha_i^{(1)} \geq 0, \text{ and } b_j^{(2)} > b_j^{(1)} \geq 0. \quad (4.46)$$

Thus, it is clear from the above discussion that there are two factorizations satisfying equation (4.23), (4.24), (4.33) and (4.34).

Now we show that there are no other factorization satisfying (4.23), (4.24), (4.33) and (4.34).

To do this, we classify the zeros of $1 - U(z)$, and by using equation (4.41), one can divide these zeros into 3 groups:

The first group consists of the two positive zeros $z = 1$ and $z = z_0$, the second group includes all zeros of $A^*(z)$, and the third group has all zeros of $B^*(1/z)$.

All zeros of $A^*(z)$ must satisfy the condition $0 \leq |z| < z_0$. This is because all zeros of $A^*(z)$ are also zeros of $1 - A^1(z)$, which is true from equation (4.41). Since $A^1(z)$ increases from $A^1(0)$, reaching 1 at $z = z_0$, clearly

$$1 - A^1(z) = (1 - z_0/z_0)A^*(z) = 0 \text{ which implies, } A^1(z) = 1.$$

$$\text{But, } A^1(z) < 1, \quad 0 \leq z < z_0 \quad (4.47)$$

Since $A^1(z)$ has only positive coefficients, $A^1(z) < A^1(|z|)$, and because of (4.47), $A^1(|z|) < 1$ for $|z| < z_0$. For example, if we consider $1 - A^1(z) > 0$ for $z > z_0$, then $1 - A^1(z)$ have zeros inside a circle of radius z_0 . Because,

$$\begin{aligned} 1 - A^1(z) &= (1 - z/z_0)A^*(z) \\ &= - \cdot \delta < 0 \text{ (which satisfies).} \end{aligned}$$

Note: δ indicate any number.

and if $z < z_0$, $1 - A^1(z) = +\delta > 0$.

Hence, $1 - A^1(z)$ cannot have any zeros inside a circle of radius z_0 , and as a consequence, all zeros of $A^*(z)$ must lie outside this circle.

By using a similar argument shows that all zeros of $B^*(z)$ must lie outside the unit circle, which means that the zeros of $B^*(1/z)$ must lie inside the unit circle.

If $1 - A^1(z)$ is a factor of $1 - U(z)$, the only two zeros available of this factor are $z = z_0$ and $z = 1$. Suppose, we let $z = z_0$ be the unique positive zero of $A(z)$. If this is the case, one can show by applying essentially the same arguments as before, that all other zeros of $A(z)$ have a modulus greater than z_0 i.e. $|z| > z_0$.

Moreover, for complex z , $A(z) \leq A(|z|) < 1$, which implies that there are no zeros, $|z| < z_0$ where z_0 is real. Hence, $A(z)$ can only include zeros of $A^*(z)$. Since a similar condition can be shown to hold for $B(z)$. Thus, it follows that $A(z)$ must include all zeros of $A^*(z)$, that is, $A(z) = A^1(z)$. In a similar fashion, it can be shown that $A(z)$ must be equal to $A^2(z)$, if $z = 1$ is chosen as the unique positive zero of $A(z)$.

Therefore, depending on the two available zeros are $z = z_0$ and $z = 1$ for $A(z)$, and similarly for $B(z)$ respectively, it is proved that there are no other factorizations that can satisfy (4.23), (4.24), (4.33) and (4.34). This completes the proof of *theorem 5*.

Chapter 5

Analytical Comparison of GI/G/1 Queue with Geom/Geom/1 Queue

5.1 Overview

As we know, the notation GI stands for general independent interarrival time distribution, G stands for general service time distribution and number 1 means that there is one server in the system. More clearly, this symbol G represents a general or arbitrary probability distribution; that is, no assumption is made as to the precise form of the distribution and results in these cases are applicable to any probability distribution. But, in our work, we are interested about the arithmetic GI/G/1 queue where the interarrival and service times are discrete random variables and their distribution functions are both defined on finite intervals.

So far, we have discussed several methods to determine the waiting time distribution in a discrete GI/G/1 queue in chapters 3 and 4. Now we are interested to use Geom/Geom/1 queue as an example of a GI/G/1 queue to explore what happens when the interval is infinite to obtain the waiting time distribution. This is the topic we are going to discuss in this chapter.

To do so, let the Geom/Geom/1 queue be defined first. Geom/Geom/1 is a single-server queue where interarrival and service times are geometrically distributed. In this situation, the time axis is segmented into a sequence of time intervals (slots) of unit duration, but the times are unbounded like exponential distribution. Moreover, it is known that the Geom/Geom/1 queue is the discrete analogue of the M/M/1 queue. First of all, we need to make ourselves clear about the difference between the M/M/1 queue and the Geom/Geom/1 queue. M/M/1 queue means exponential interarrival time, exponential service time and one single server queueing system. In general, M refers mainly to three facts: (1) The customers

arrive according to a Poisson process and the interarrival time distribution is exponential, (2) the service time distribution is also exponential, (3) Thus, the process is memoryless or Markovian. Since the successive times for interarrival and service are continuously distributed, they are therefore exponential.

However, we already came to know, in the case of arithmetic GI/G/1 queue which is based on a finite interval, that g of the roots of the polynomial $1 - U(z) = (1 - A(z))(1 - B(1/z))$ lie inside or on the unit circle of the complex plane and that h of them lie outside the unit circle. On the other hand, if we use the above polynomial for Geom/Geom/1 queue having the infinite intervals for the interarrival times and service times, clearly the degree of this polynomial goes to infinity and the number of roots must be infinite.

In order to verify our idea, at first we assume that the waiting time of Eq.(4.29) for the Geom/Geom/1 queue is geometrically distributed.

5.2 The Waiting Time Distribution for the Geom/Geom/1 Queue

In this section, we discuss if the Geom/Geom/1 queue has strictly geometric distribution or not.

We considered the same notation for the Geom/Geom/1 queue by following the chapter 4:

A_n is the time of n th arrival C_n , T_n is the interarrival time, i.e. $A_{n+1} = A_n + T_n$ and S_n is the service time of C_n .

W_n is the waiting time of C_n . Then we have, $W_{n+1} = D_n - A_{n+1} = D_n - A_n - T_n$, where D_n is the departure time of C_n and $D_n = A_n + W_n + S_n$.

$$\text{Hence, } W_{n+1} = A_n + W_n + S_n - A_n - T_n = W_n + S_n - T_n. \quad (5.1)$$

Since we are dealing with the Geom/Geom/1 queue, S_n and T_n are both geometrically distributed. We assume that $S_n > 0$ and $T_n > 0$. Thus, we know the probability density functions of S_n and T_n are $Pr(S_n = i) = p^i(1 - p), i = 1, 2, \dots$; and $Pr(T_n = i) = q^i(1 - q), i = 1, 2, \dots$; respectively, where p and q are the corresponding parameters.

We know, the difference between the service time of customer C_n , and interarrival time of the C_n and C_{n+1} customers, is $U_n = S_n - T_n$. We now have to find $Pr(U_n = i)$ for the Geom/Geom/1 queue. To do so, set $U_n = i$ if $S_n = k + i$ and $T_n = k$.

$$\begin{aligned}
\text{Thus, } Pr(U_n = i) &= \sum_{k=0}^{\infty} Pr(S_n = k + i)Pr(T_n = k) \\
&= \sum_{k=0}^{\infty} p^{k+i}(1-p)q^k(1-q) \\
&= p^i(1-p)(1-q) \sum_{k=0}^{\infty} p^k q^k \\
&= p^i(1-p)(1-q) / (1-pq). \tag{5.2}
\end{aligned}$$

For $i < 0$, we can easily exchange p and q and get the following:

$$Pr(U_n = -i) = q^i(1-p)(1-q) / (1-pq). \tag{5.3}$$

Now, Eq.(5.1) allow us to find a relation for the distribution of W_n for the Geom/Geom/1 queue in the following way

$$Pr(W_{n+1} = j) = \sum_{k=0}^{\infty} Pr(W_n = k)Pr(U_n = j - k) \quad j > 0. \tag{5.4}$$

Then we consider, the distribution of W_n is geometric with unknown parameter x

$$Pr(W_{n+1} = j) = Pr(W_n = 0) \cdot x^j = \pi_0 x^j \tag{5.5}$$

$= Pr(W_n = j)$; because in equilibrium W_n and

W_{n+1} eventually have the same distribution.

Now, using (5.5) in (5.4), we get

$$\pi_0 x^j = \sum_{k=0}^{\infty} \pi_0 x^k Pr(U_n = j - k)$$

By cancelling π_0 from both sides, one yeilds

$$x^j = \sum_{k=0}^j x^k Pr(U_n = j - k) + \sum_{k=j+1}^{\infty} x^k Pr(U_n = j - k)$$

$$= \left[\sum_{k=0}^j x^k p^{j-k} + \sum_{k=j+1}^{\infty} x^k q^{k-j} \right] \cdot \frac{(1-p)(1-q)}{1-pq}$$

Therefore, $x^j = \frac{(1-p)(1-q)}{1-pq} \left[p^j \cdot \frac{1-(x/p)^{j+1}}{1-(x/p)} + \frac{1}{q^j} \cdot \frac{x^{j+1}q^{j+1}}{1-xq} \right]$

$$= \frac{(1-p)(1-q)}{1-pq} \left[p^j \cdot \left(\frac{1-(x/p)^{j+1}}{1-(x/p)} \right) + \frac{x^{j+1}q}{1-xq} \right]. \quad (5.6)$$

We then need to find the value for x to know the waiting time for Geom/Geom/1 queue. In order to simplify the above relation, we get from (5.6)

$$x^j = \frac{(1-p)(1-q)}{1-pq} \left[\frac{p^j - x^{j+1}/p}{1-(x/p)} + \frac{qx^{j+1}}{1-xq} \right]$$

$$= \frac{(1-p)(1-q)}{1-pq} \left[\frac{p^{j+1} - x^{j+1}}{p-x} + \frac{qx^{j+1}}{1-xq} \right]$$

i.e. $x^j = \frac{(1-p)(1-q)}{1-pq} \left[x^{j+1} \left[\frac{q}{1-xq} - \frac{1}{p-x} \right] + \frac{p^{j+1}}{p-x} \right]$

$$= \frac{(1-p)(1-q)}{1-pq} \left[x^{j+1} \left[\frac{(p-x)q - 1 + xq}{(1-xq)(p-x)} \right] + \frac{p^{j+1}}{p-x} \right]$$

$$= \frac{(1-p)(1-q)}{1-pq} \left[x^{j+1} \cdot \frac{pq-1}{(1-xq)(p-x)} + \frac{p^{j+1}}{p-x} \right].$$

Now by multiplying $\frac{(1-p)(1-q)}{1-pq}$ with $x^{j+1} \cdot \frac{pq-1}{(1-xq)(p-x)} + \frac{p^{j+1}}{p-x}$, we get

$$x^j = -(1-p)(1-q) \cdot x^{j+1} \cdot \frac{1}{(1-xq)(p-x)} + \frac{(1-p)(1-q)}{1-pq} \cdot \frac{p^{j+1}}{p-x}$$

i.e. $(p-x)x^j = p^{j+1} \cdot \frac{(1-p)(1-q)}{1-pq} - x^{j+1} \cdot \frac{(1-p)(1-q)}{(1-xq)}$

i.e. $(p-x)x^j + (1-p)(1-q) \cdot \frac{x^{j+1}}{(1-xq)} = (1-p)(1-q) \cdot \frac{p^{j+1}}{1-pq}$

i.e. $x^j \left[(p-x) + (1-p)(1-q) \cdot \frac{x}{(1-xq)} \right] = (1-p)(1-q) \cdot \frac{p^{j+1}}{1-pq}$

i.e. $x^j \left[\frac{(p-x)(1-xq) + (1-p)(1-q)x}{1-xq} \right] = (1-p)(1-q) \cdot \frac{p^{j+1}}{1-pq}$

$$\text{i.e. } x^j \left[\frac{p+x^2q-x(p+q)}{1-xq} \right] = (1-p)(1-q) \cdot \frac{p^{j+1}}{1-pq} . \quad (5.7)$$

If we let $x = 1$ in (5.7), it works out as

$$\frac{p+q-p-q}{1-q} = 0 = (1-p)(1-q) \cdot \frac{p^{j+1}}{1-pq} .$$

and if we let $x = 0$ in (5.7), this does not work to give a solution.

Now, if we consider the first case where $x^2q - x(p+q) + p = 0$, we can solve it for x .

Therefore,

$$x = \frac{p+q \pm \sqrt{(p+q)^2 - 4pq}}{2q} = \frac{p+q \pm (p-q)}{2q} = p/q, \text{ or } 1. \quad (5.8)$$

Thus, the waiting time distribution in the Geom/Geom/1 queue which is the discrete analogue of the M/M/1 queue, is asymptotically geometric with $x = p/q$.

Chapter 6

Numerical Experiment

6.1 Numerical Results and Examples

A GI/G/1 queue is frequently used in the performance evaluation of manufacturing, telecommunications and computer systems. The steady-state waiting time is an important measure of the effectiveness of this model. The closed form expressions for the distribution and moments of the steady-state waiting times are available for particular cases of this model. However, the exact analysis, in many cases, has been difficult. The numerical approximations presented in this chapter are intended to bridge this gap.

In this section, we use the various techniques discussed in previous chapters to solve the steady-state waiting time and the average queue length for the discrete time GI/G/1 queue. We use Uniform distribution, Geometric distribution and Gamma distribution to calculate the computed waiting time distribution, and then we compare them with the closed form approximation formula in order to determine the effectiveness of our method.

We now find the waiting time distribution in a discrete GI/G/1 queue with the method described in chapter 4. An algorithm is developed by us for the purpose of figuring this out and is given below:

Our Algorithm:

1. $\alpha_i^0 = 0, i = 1, 2, \dots, h.$
2. For $m = 0, 1, 2, \dots$ do the following until $\max(|\alpha_i^m - \alpha_i^{m+1}|) < \epsilon$
 - 2.1. $b_j^m = u_{-j} + \sum_{i=1}^{\infty} \alpha_i^{m-1} b_{j+i}^m, j = g, g-1, \dots, 0$
 - 2.2. For $0 \leq t \leq 1,$

$$S^m = (\{(1-t) * (1 - b_0^m)\} + \{t * \sum_{j=1}^g b_j^m\})$$

$$2.3. \alpha_j^m = \frac{(u_j + \sum_{i=1}^{\infty} \alpha_{i+j}^m b_i^m)}{S^m}, \quad j = h, h-1, \dots, 1.$$

Our Algorithm is identical to Algorithm 3 in [12]. In step 2.2, we use the fact that $S^m = (\{(1-t) * (1 - b_0^m)\} + \{t * \sum_{j=1}^g b_j^m\})$ for $0 \leq t \leq 1$. In Algorithm 3, Grassmann and Jain [12] considered $t = \frac{1}{2}$ as the weight in step 2.2, but we are considering $(1-t)$ and t as the weights for $(1 - b_0^m)$ and $\sum_{j=1}^g b_j^m$, respectively, instead of $\frac{1}{2}$. We utilize 100 different values of t between 0 and 1, where t increases by 0.01 and thus, we can justify at which values of t the convergence is faster, apart from $t = \frac{1}{2}$. On the other hand, we increase the value of m from 0 to a maximum value for which the condition $\max(|\alpha_i^m - \alpha_i^{m+1}|) < \epsilon$ is fulfilled. For each value of m , we do the calculation repeatedly by using our Algorithm. Furthermore, the approximations for the expected waiting time and the average queue length can be obtained at each value of t and the number of iteration can also be easily investigated. Obviously, we are considering the minimum number of iteration for a particular value of t , which is always less than or equal to the number of iterations in Algorithm 3 in [12]. In addition, it is possible to calculate the total time required to converge for each value of t by using our method and thus we can find the time required for those values of t , where we get the minimum number of iteration. In this way, the best value of t can be determined such that we reach to an optimal situation. We can also find the actual number of iteration to get α_j and b_j . In all the numerical experiments, our Algorithm always converges to the correct solution, and most of the time, its convergence is faster than the convergence of Algorithm 3. However, analytically so far there is no proof that our Algorithm will converge, and even if it does, there is no proof that it obtains the Wiener-Hopf factorization.

For our numerical calculations, the precision ϵ of our Algorithm is set to 0.000001, and it is possible to solve problems with h and g above 2000 by using our Algorithm. Several values of ρ are used and the number of iteration are also provided.

Then by using Little's formula ($L_q = \lambda \cdot W_q$), we can easily calculate the approximate average queue length, which we call L_{q_1} . Thus, $L_{q_1} = \lambda \cdot W_{q_1}$.

Uniform Distributions: A random variable, X , is said to have a discrete uniform distribution on the integers $1, 2, \dots, N$, if the probability density function (pdf) has the form

$$f(x) = 1/N; \quad x = 1, 2, \dots, N, \quad E(X) = \frac{N+1}{2} \quad \text{and} \quad Var(X) = \frac{(N+1)(N-1)}{12}.$$

In our case,

$$Pr(T_n = i) = \frac{1}{N}; \quad i = 1, 2, \dots, N \quad \text{and} \quad (S_n = i) = \frac{1}{N}; \quad i = 1, 2, \dots, N.$$

The following tables give results of some queues with uniform interarrival and service times. In all cases, the minimal service time is 1, as is the minimal interarrival time. The maximum interarrival and service times vary, as shown in Table 6.1-6.4.

Table 6.1 contains the results obtained by Algorithm 3 of [12], as well as the results obtained by our method. In all cases, we get better approximation for $E(W)$ and in most of the cases; the number of iteration in our method is less than in theirs.

Table 6.1 **Results of using Uniform Distributions by using our Algorithm and by Algorithm 3 [12]**

Case	Max (T)	Max (S)	ρ	Algorithm 3	Iter.	Our method	Iter.
				$E(W)$		$E(W)$	
1	50	25	0.510	3.720	5	3.7199	4
2	50	40	0.804	24.9	5	24.8725	5
3	49	48	0.980	381.0	6	380.9675	5
4	66	65	0.985	700.1	6	700.0980	5
5	99	98	0.990	1596.6	6	1594.4171	5

Note: Iter. is representing the number of iteration, in all the Tables of chapter 6.

Table 6.2 represents the results of the approximation for average Queue Length in queue which is calculated by using our method. We also include the optimal values of t and required time (sec) for minimum iteration.

Table 6.2 Results for the Average Queue Length in queue using our method

Case	Max (T)	Max (S)	Our method	Our method	Our method
			Queue Length L_q	Optimum values of t	Best value of t & required time (sec) for minimum iter.
1	50	25	0.1459	0.22 to 0.42	0.42 <i>(time=0.0014)</i>
2	50	40	0.9754	0.43 to 0.62	0.60 <i>(time=0.0023)</i>
3	49	48	15.2387	0.57 to 0.67	0.60 <i>(time=0.0025)</i>
4	66	65	20.8984	0.53, 0.54, 0.56 to 0.67	0.60 <i>(time=0.0034)</i>
5	99	98	31.8883	0.53 to 0.67	0.55 <i>(time=0.0053)</i>

In table 6.1 and 6.2, we get more than 10 values of t between 0 and 1 where we have the optimal situation. But, in each case, we choose the best approximation among the results at the optimal points depending on the minimum number of iterations, where the required time for convergence is also shortest. For example, in our first case, the minimum number of iteration is 4 for $t = 0.22$ to $t = 0.42$ and the shortest time that required for this iteration among these values of t is 0.0014 sec. at $t = 0.22, 0.25, 0.26, 0.27, 0.30, 0.31, 0.32, 0.35, 0.36, 0.37, 0.41,$ and 0.42 . Therefore, we can choose any of these values as the best value of t and we have chosen 0.42 as the best value of t for the first case in table 6.2. However, we noticed that in some cases, we have the same approximation in all 100 points of t , but the number of iterations are not same.

Table 6.3 & 6.4 shows that with our algorithm, we can easily compute the parameters with h and g exceeding 2000. In the last two rows we worked with h and g , which exceeding 2000

and mentioned by new data and therefore, we do not have results for Algorithm 3. However, in most of cases, the numbers of iterations are similar to Algorithm 3.

Table 6.3 Results for Uniform Distributions with Large Bounds by using our Algorithm and by Algorithm 3 [12]

Max (<i>T</i>)	Max (<i>S</i>)	ρ	Algorithm 3	Algorithm 3	Our method	Our method
			<i>E(W)</i>	Iter.	<i>E(W)</i>	Iter.
200	180	0.9	260.4	5	260.4448 (at t=0.50)	4
500	450	0.9	651.1	4	651.0762 (at t=0.55)	4
1000	900	0.9	1302.2	4	1302.097 (at t=0.63)	4
2000	1800	0.9	2604.5	4	2602.44 (at t=0.65)	4
2700	2400	0.9	New data	New data	3072.3 (at t=0.65)	4
3000	2700	0.9	New data	New data	3901.3 (at t=0.66)	4

Table 6.4 Results for the Average Queue Length in queue using our method

Case	Max (T)	Max (S)	Our method	Our method	Our method
			Queue Length L_q	Optimum values of t	Best value of t and required time (sec) for minimum iter.
1	200	180	2.5915	0.50	0.50 <i>(time=0.0099)</i>
2	500	450	2.5991 <i>(at $t=0.55$)</i>	0.49 to 0.51 & 0.55 to 0.62	0.55 <i>(time=0.0355)</i>
3	1000	900	2.6016 <i>(at $t=0.63$)</i>	0.48 to 0.63	0.63 <i>(time=0.0882)</i>
4	2000	1800	2.6011 <i>(at $t=0.65$)</i>	0.47 to 0.65	0.65 <i>(time=0.2801)</i>
5	2700	2400	2.2749 <i>(at $t=0.65$)</i>	0.45 to 0.65	0.65 <i>(time=0.5037)</i>
6	3000	2700	2.6000 <i>(at $t=0.66$)</i>	0.46 to 0.66	0.66 <i>(time=0.5848)</i>

In the tables, 6.3 and 6.4, we have mentioned the optimal values of t where we have minimum number of iteration for convergence. Therefore, we selected the better approximation of $E(W)$ and L_q from the set of optimal points of t where the total number of iterations also takes less time. It is apparent that, if we start going to the right from the point 0.50, we usually get the optimum situation. But, in the first case, we see that 0.50 is the only point where we have the minimum number of iteration and can get the required approximation.

Geometric Distributions: The following tables (Table 6.5, Table 6.6) contain the results for the GI/G/1 Queue, where we used the geometric distribution for both the interarrival and service time. In this case, we have

$$Pr(T_n = i) = (1 - q)q^{i-1}, \quad i = 1, 2, \dots; \quad E(T_n) = \frac{1}{1-q}; \quad Var(T_n) = \frac{q}{(1-q)^2}.$$

$$Pr(S_n = i) = (1 - p)p^{i-1}, \quad i = 1, 2, \dots; \quad E(S_n) = \frac{1}{1-p}; \quad Var(S_n) = \frac{p}{(1-p)^2}.$$

But, our Algorithm does not work in this case because the service times and the interarrival times are both unbounded in the above situation. Hence, we need to truncate them. For the purpose of computation, one has to truncate the distribution function to avoid a possible overflow and to define on a finite range. Thus, the truncated geometric distribution for both the interarrival time and the service time is

$$Pr(T_n = i) = (1 - q)q^{i-1}/(1 - q^m); \quad i = 1, 2, \dots, m.$$

$$Pr(S_n = i) = (1 - p)p^{i-1}/(1 - p^m); \quad i = 1, 2, \dots, m.$$

$$E(T_n) = \frac{1-(m+1)q^m+mq^{m+1}}{(1-q)(1-q^m)}; \quad E(S_n) = \frac{1-(m+1)p^m+mp^{m+1}}{(1-p)(1-p^m)};$$

$$Var(T_n) = \frac{1-2q-(m+1)^2q^m+(m^2+2m+2)q^{m+1}}{(1-q)(1-q^m)};$$

$$Var(S_n) = \frac{1-2p-(m+1)^2p^m+(m^2+2m+2)p^{m+1}}{(1-p)(1-p^m)}.$$

We consider different values for the parameter q and p . For example, if we let $q = 0.6$ for the interarrival time, and $p = 0.2$ for the service time, we get approximately $\rho = 0.5$.

Table 6.5 gives us a comparison of some queues with the geometric interarrival and the service times. In all cases the minimal service time is 1, as are the minimal interarrival times. The maximum service and interarrival times vary, as shown in the table below. The first row of Table 6.5, compares the results obtained in [12] with ours, where the approximations obtained in our case are better than their results. For the rest of the cases, we used the Algorithm 3 of [12] to compare the results with ours, which were not shown in [12]. In all

cases, the number of iteration is less than Algorithm 3, which is still considered as a fast convergence.

Table 6.5 Results of using Truncated Geometric Distributions by using our Algorithm and by Algorithm 3 [12]

Max (T)	Max (S)	ρ	Algorithm 3	Iter.	Our method	Iter.
			$E(W)$		$E(W)$	
10	10	0.51	0.2636	4	0.2537	3
20	20	0.61	0.3952	4	0.3952	3
50	50	0.75	0.7500	6	0.7500	3
65	65	0.88	1.7500	9	1.7500	4
99	99	0.99	10.8764	15	10.8764	4

Table 6.6 contains the results of the same cases which we have used in Table 6.5 above, and gives the average queue length in queue by using our method. We also include the optimum values of t and required time (sec) for minimum iteration.

Table 6.6 Results for the Average Queue Length in queue using our method

Case	Max (T)	Max (S)	ρ	Our method	Our method	Our method
				Queue Length L_q	Optimum values of t	Best value of t & required time (sec) for minimum iter.
1	10	10	0.512 $q = 0.6, p = 0.2$ $\lambda = 0.4, \mu = 0.8$	0.1040	0.46 to 0.48	0.46 <i>(time = 0.0004)</i>
2	20	20	0.612 $q = 0.51, p = 0.2$ $\lambda = 0.49, \mu = 0.8$	0.1936	0.55 & 0.56	0.55 <i>(time = 0.0045)</i>
3	50	50	0.75 $q = 0.4, p = 0.2$ $\lambda = 0.6, \mu = 0.8$	0.4500	Only at 0.66	0.66 <i>(time = 0.0015)</i>
4	65	65	0.88 $q = 0.3, p = 0.2$ $\lambda = 0.7, \mu = 0.8$	1.2250	0.74 to 0.77	0.76 <i>(time = 0.0026)</i>
5	99	99	0.99 $q = 0.12, p = 0.11$ $\lambda = 0.88, \mu = 0.89$	9.5712	0.90 & 0.91	0.90 <i>(time = 0.0043)</i>

In the tables, 6.5 and 6.6, we get the same approximations at all optimum points for the Geom/Geom/1 queue. For the first four cases we obtained the same results of $E(W)$ and L_q for all 100 values of t , but for case 5, the results vary at the different points of t . We also included the values of the parameters for the interarrival time distribution, q , and the service time distribution, p , in Table 6.6 which eventually help us to determine λ and μ from the truncation formula of $E(T_n)$ and $E(S_n)$. In addition, in Table 6.6, we mentioned the result where t is optimum and also mentioned the best value of t where the minimum iteration takes shorter time. However, in case 3, we have an optimal situation only at $t = 0.66$, which is apparently the best value of t .

Note: It makes a huge difference whether one starts the interarrival and service time at 0 or at 1 when p and q not close to 1 and this does not affect the waiting-time distribution. The waiting time depends on the difference between p and q , but it does affect λ , μ and ρ significantly. We can check this easily because starting with 1 rather than with 0, increases the expected interarrival time and service time by 1.

Gamma distributions: A continuous variable X is said to have a gamma distribution with parameter $k > 0$ and $\theta > 0$ if it has a pdf of the form

$$f(x; \theta, k) = \begin{cases} \frac{1}{\theta^k \Gamma(k)} x^{k-1} e^{-x/\theta}; & x > 0 \\ 0; & x \leq 0 \end{cases}.$$

CDF: If $X \sim \Gamma(\theta, n)$, where n is a positive integer, then the CDF can be written as

$$F(x; \theta, n) = 1 - \sum_{j=0}^{n-1} \frac{(x/\theta)^j}{j!} e^{-x/\theta}; \quad E(X) = \theta k, \quad Var(X) = k\theta^2.$$

Gamma/Geom/1 queue: The following table (Table 6.7) contains the results for the GI/G/1 queue, where we used the *gamma* distribution for the interarrival time and the *truncated geometric* distribution for the service time. So far, we discussed several methods to solve the waiting time distribution in a discrete GI/G/1 queue. But, can we obtain the waiting time distribution for a GI/G/1 queue if the given interarrival time distribution and service time distribution are continuous distribution function? Since all computers are digital machines, the logical way to obtain the waiting time distribution in a continuous GI/G/1 queue is to first discretize the given distribution function, and we can do this by replacing it with a sequence of discrete distribution function.

It has been suggested by Kimura [18] that to discretize the distribution functions of the interarrival time $Q(t)$ and the service time $P(t)$ in a GI/G/1 queue, we let the probability distribution of interarrival time $\{T_n\}$ be $q_i = Q_i - Q_{i-1}$ and the probability distribution of service time $\{S_n\}$ be $p_i = P_i - P_{i-1}$.

e.g. $p_1 = Pr(0 \leq X \leq 0.1) = F(0.1) - F(0)$ and

$$p_2 = Pr(0.1 \leq X \leq 0.2) = F(0.2) - F(0.1), \text{ and so on.}$$

In this case, we compare the results of our method versus Algorithm 3 [12], though this was not done in [12]. This is our new idea, to show how well Gamma distribution works for the interarrival times where the service times are considered as the truncated geometric distribution. Since the service time has the truncated geometric distribution, we use the same formula as we used in the previous discussion in order to find the probability density function, mean and variance. The minimal time for both service and interarrival is 1 and the maximal time is varying which is the same as shown in Tables, 6.5 and 6.6. In the case of Gamma distribution for the interarrival time, we have not truncated it, even though it is not compactly supported (not bounded). But in each case, we have considered the maximum time for interarrival as its upper bound. Moreover, we considered different values for the parameters k and θ to get λ as well different values for p to get μ . In all cases, the results of $E(W)$ obtained by Algorithm 3 coincide with ours. We also calculated the average queue length in queue in Table 6.8.

Note: we assume $k = A$ and $\theta = B$ in our computer program.

Table 6.7 Results of using Gamma distribution for the interarrival time and truncated Geometric distribution for the service time by using our Algorithm and by Algorithm 3 [12]

Case	Max(<i>T</i>)	Max (<i>S</i>)	ρ	Algorithm 3	Iter.	Our method	Iter.
				<i>E(W)</i>		<i>E(W)</i>	
1	10	10	0.5 <i>p</i> = 0.2, <i>A</i> = 2, <i>B</i> = 1.25	0.1611	4	0.1601	3
2	20	20	0.61 <i>p</i> = 0.2, <i>A</i> = 2, <i>B</i> = 1.02	0.2397	4	0.2429	3
3	50	50	0.75 <i>p</i> = 0.2, <i>A</i> = 2, <i>B</i> = 0.83	0.3621	5	0.3821	3
4	65	65	0.88 <i>p</i> = 0.2, <i>A</i> = 2, <i>B</i> = 0.71	0.4896	6	0.5653	3
5	99	99	0.99 <i>p</i> = 0.2, <i>A</i> = 2, <i>B</i> = 0.63	0.600	6	0.7970	3

Table 6.8 Results for the Average Queue Length in queue using our method

Max (T)	Max (S)	ρ	Our method	Our method	Our method
			Queue Length L_q	Optimum values for t	Best value of t and required time (sec) for minimum iter.
10	10	0.5	0.0640	from 0.45 to 0.50	0.45 <i>(time = 0.0003)</i>
20	20	0.61	0.1191	0.55 to 0.58	0.58 <i>(time = 0.0005)</i>
50	50	0.75	0.2302	0.65, 0.66	0.65 <i>(time = 0.0037)</i>
65	65	0.88	0.3981	0.72	0.72 <i>(time = 0.0051)</i>
99	99	0.99	0.6325	0.77	0.77 <i>(time = 0.0086)</i>

In Table 6.7 & 6.8, we see that there are many optimal values of t between 0 and 1. Except in the fourth case, we get only one optimal point of t . We choose the best approximation among those points of t where we get the lowest value for $E(W)$ and L_q for the minimum number of iteration. We also added the required time for the minimum number of iteration. Moreover, in all cases, the number of iteration in our method is less than Algorithm 3, which is still considered as a fast convergence.

The above output shows that most of the cases the best values for t of the U/U/1, the Geom/Geom/1 and the Gamma/Geom/1 queue are found when we start going to the right

from the point 0.50, where we usually get the optimum situation by using our method. Moreover, we also showed that we are always able to get a better approximation for $E(W)$ and our execution times are shorter compare to Algorithm 3. Thus, between the two algorithms (ours and Algorithm 3), our result always proves its superiority.

6.2 Charts for the Queue Lengths

In figures 6.2.1 to 6.2.4, we added some charts to show the results of the average queue length in queue, L_q , with different values of traffic intensity (ρ) for the U/U/1 queue (Figure 6.2.1 and Figure 6.2.2), the Geom/Geom/1 queue (Figure 6.2.3), and the Gamma/Geom/1 queue (Figure 6.2.4).

Figure 6.2.1 The Average Queue Length in queue for the U/U/1 queue based on the data in Table 6.1 and 6.2.

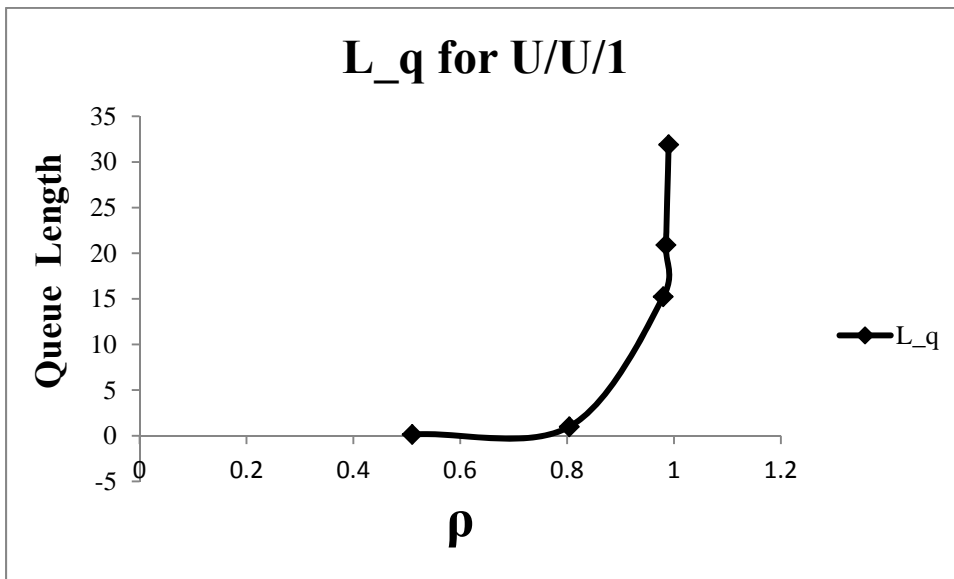


Figure 6.2.2 The Average Queue Length of large bound for the U/U/1 queue based on the data in Table 6.3.

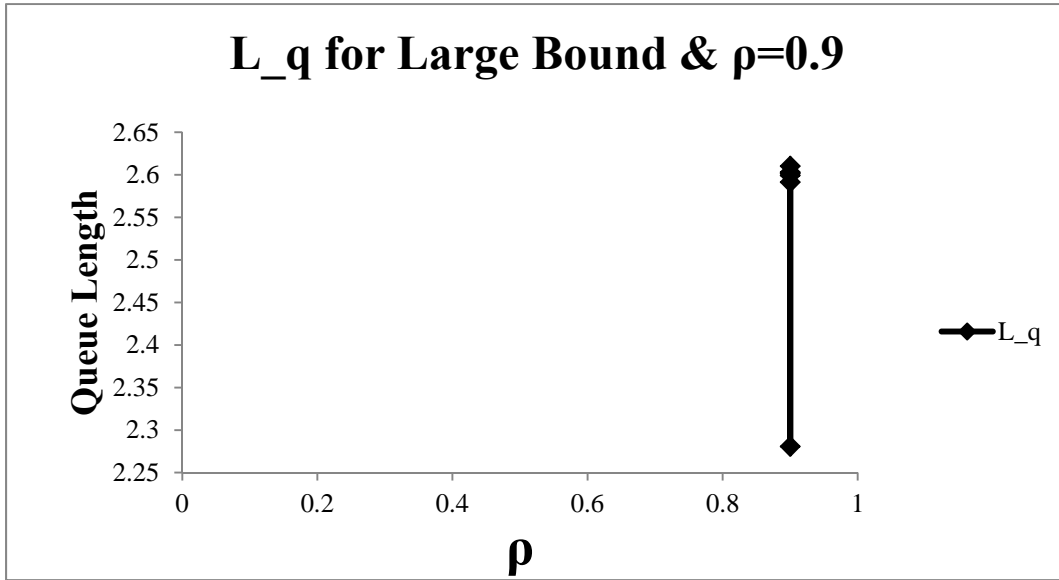


Figure 6.2.3 The Average Queue Length in queue for the Geom/Geom/1 queue based on the data in Table 6.4.

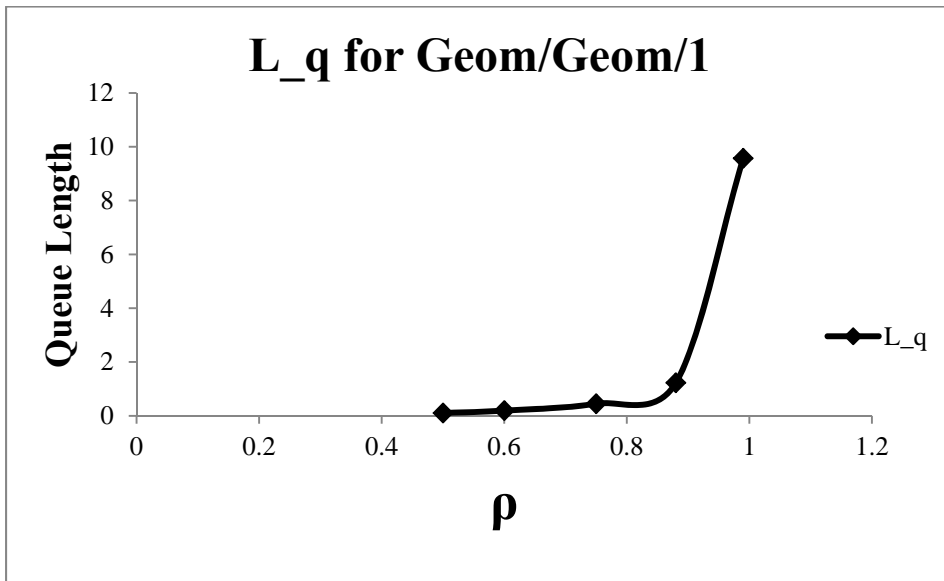
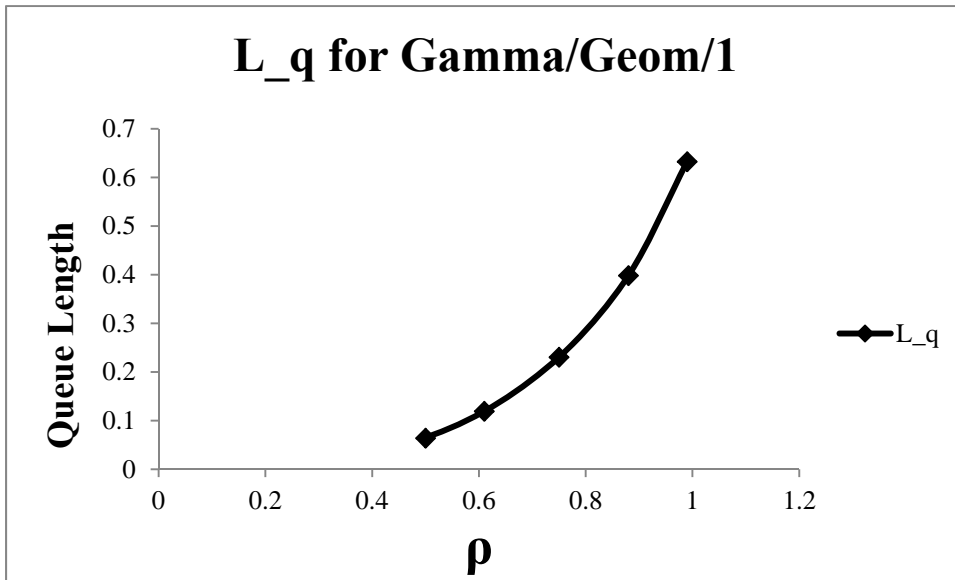


Figure 6.2.4 The Average Queue Length for the Gamma/Geom/1 queue in queue based on the data in Table 6.7.



Chapter 7

Conclusion and Recommendations for Future Work

7.1 Conclusions

In this thesis, we have discussed some of the popular methods of finding the waiting time distribution of the arithmetic GI/G/1 queue. In particular, the methods given by Grassmann [11], Grassmann and Jain [12], Konheim [22] and Ponstein [28], respectively could be the best ways of obtaining the discrete waiting time distribution. Other techniques for solving the problem of finding the waiting time distribution were also mentioned. Furthermore, based on the theory of random walk and the Wiener-Hopf factorization method, Grassmann and Jain [12] gave an efficient algorithm which we described in some detail. We also explained the reasons for using the Wiener-Hopf factorization method for $1 - U(z)$. Moreover, we added our proofs for Algorithm 1 of [12], where we showed why it is monotonic and converges to the Wiener-Hopf factorization. Furthermore, we added our explanation for Algorithm 2 of [12] to show α_i^m to be bounded. In addition, we thoroughly discussed the preliminary ideas related to our topic. We also indicated the various probabilistic interpretations in the above-mentioned methods and more importantly, we demonstrated that all of these techniques [11, 12, 22, 28] are fully equivalent.

Moreover, we discussed an analytical interpretation to investigate the waiting time distribution of the Geom/Geom/1 queue, which is based on an infinite interval, as an example of the arithmetic GI/G/1 queue. Thereby, we proved that the waiting time distribution of the Geom/Geom/1 queue is asymptotically geometric.

The most important part we have done in this thesis is, based on Grassmann and Jain's [12] method, we developed a new algorithm by improving the weighted average and nicely represented how one can implement this technique to find the waiting time distribution in a

discrete GI/G/1 queue. We also calculated the average queue length in queue by using Little's formula.

Finally, we gave some numerical examples to conclude this thesis. At first, we implement this method in numerical test for the U/U/1 queue and it worked out nicely. We also applied this in the Geom/Geom/1 queue where we truncated the distribution, and we briefly discussed the approach of truncation and showed the convergence of the waiting time distribution for the truncation case. Furthermore, our method could be used not only for the discrete type input distribution functions, but also for the continuous type input distribution function. Discretizing a continuous distribution function into a discrete sequence is a critical step in practical computation in queueing models. However, for the Gamma/Geom/1 queue, we used Gamma distribution for the interarrival time and we also discussed the technique of discretizing the gamma distribution function. Moreover, we numerically experimented with it and we found that very precise results are achievable. It can be concluded from the above discussion that the method shown in this research is numerically stable, effective, simple and robust.

7.2 Future Works

We numerically experimented with our method in the Uniform distribution and Geometric distribution for the interarrival time and service time, and also with gamma distribution for the interarrival time. In future, we could implement our method in various ways to develop the concept of our work. For example, we could try to develop our method to work for the Geom/Gamma/1 queue, which is not possible by using our method. We could also compare the different t with the other distributions, like Poisson, Normal, Log Normal, etc., and then make a comparison with the above distributions in such a way that one easily figures out which distribution works better in order to find the best value for t by our method. Furthermore, we could also try to find a better t for both the discrete and continuous cases by using the hyperbola concept. Additionally, we could explore the roots of α_i and b_j of $1 - U(z)$ by using the Geom/Geom/1 queue, which is based on an infinite interval, as an example of the arithmetic GI/G/1 queue

Bibliography

- [1] ACKROYD, M. H., 1980. Computing the Waiting Time Distribution for the G/G/1 Queue by Signal processing Methods. *IEEE Trans. Commun .COM-28* ,52-58.
- [2] ACKROYD, M. H., 1984. Stationary and Cyclostationary Finite Buffer Behaviour Computation via Levinson's Method. *AT &T Bell Lab. Tech. J.* **63**, 2159-2170.
- [3] ALFA, A. S., and LI, W., 2001. Matrix-Geometric Analysis of the Discrete Time Gi/G/1 System. *Stochastic Model*, 17, 541-554.
- [4] ALFA, A. S., and XUE, J., 2007. Efficient Computations for the Discrete GI/G/1 System. *Inform J. on computing*, **19**, 480-484.
- [5] CHAUDHRY, M. L., and GUPTA, U. C., 2001. Computing waiting-time probabilities in the discrete-time queue: $GI^X/G/1$, *perform. Eval.* **43**, 123-131.
- [6] CHAUDHRY, M. L., and TEMPLETON, J. G. C., 1983. *A First Course on Bulk queues*. Wiley, New York.
- [7] CHAUDHRY, M. L., JAIN, J. L., and TEMPLETON, J. G. C., 1987. Numerical Analysis for Bulk-Arrival Queueing System: Root finding and Steady-State Probabilities in $GI^rM/1$ Queues. *Ann. Opns. Res.***8**, 307-320.
- [8] CHAUDHRY, M. L., AGARWAL, M., and TEMPLETON, J. G. C., 1992. Exact and Approximate Numerical solutions of Steady-State distributions arising in the Queue GI/G/1. *Queueing Systems*, **10**, 105-152.
- [9] FELLER, W., 1971. *An Introduction to probability Theory and its Applications*, Vol. 2 Ed.2 . John Wiley & Sons, New York.
- [10] FRYER, M. J., and WINSTEN, C. B., 1986. An Algorithm to compute the Equilibrium Distribution of a one Dimensional Bounded Random Walk. *Opns. Res.* **34**, 449-454.
- [11] GRASSMANN, W. K., 1985. The Factorization of Queueing Equations and Their Interpretation. *J. Opnl. Res. Soc.* **36**, 1041-1050.
- [12] GRASSMANN, W. K., and JAIN, J. L., 1989. Numerical Solutions of the Waiting Time Distribution and Idle Time Distribution of the Arithmetic GI/G/1 Queue. *Opns. Res.Vol.* **37**, 141-150.

- [13] GRASSMANN, W. K., and CHAUDHURY, M. L., 1982. A New Method to Solve Steady State Queueing Equations. *Naval Res. Logist. Quart.* **29**, 461-473.
- [14] GROSS, D., HARRIS, C. M., SHORTLE, J. F. and THOMPSON, J. M., *Fundamental of Queueing Theory*, Ed.4. John Wiley & Sons, New Jersey, 2008.
- [15] HASSLINGER, G., 1995. A Polynomial Factorization Approach to the Discrete time GI/G/1/N Queue Size Distribution, *performance Evaluation*, **23**, 217-240.
- [16] HASSLINGER, G., 2000. Waiting Time, Busy Periods and Output Models of a Server Analyzed via Wiener-Hopf Factorization. *performance Evaluation*, **40**, 3-26.
- [17] HWANG, G. U., CHOI, B. D., and KIM, J. K., 2002. The Waiting Time Analysis of a Discrete-Time Queue with Arrivals as a Discrete Autoregressive Process of Order 1. *J. of Appl. Prob.* **39**, 619-629.
- [18] JAIN, J. L., MOHANTY, S. G., and BOHM, W., *A Course on Queueing Models*, Ed.1. Chapman & Hall/CRC, New York, 2007.
- [19] KIMURA, T., 1985. An Analytical Interpretation of the Discrete Approximation for the GI/GI/1 Queue, *Optimization*, Vol. **2**, 285-296.
- [20] KIM, N. K., and CHAUDHRY, M. L., 2008. The Use of the Distributional Little's Law in the Computational Analysis of the Discrete-Time GI/G/1 and DI/D/c queues. *performance Evaluation*, **65**, 3-9.
- [21] KOBAYASHI, H., 1983. Discrete-Time Queueing Systems. In *probability Theory and Queueing Models*. G. Louchard and G. Latouche (eds.). Academic Press, New York.
- [22] KONHEIM, A. G., 1975. An Elementary Solution of the Queueing System G/G/1. *SIAM J. Comput.* **4**, 540-545.
- [23] LI, S. Q., 1991. A General Solution Technique for Discrete Queueing Analysis of Multimedia Traffic on ATM, *IEEE Trans. Commun.* Vol **39**, 1115-1132.
- [24] LINDLEY, D. V., 1952. The Theory of Queues with a Single Server. *Proc. Cambridge Philos. Soc.* **48**, 277-289.
- [25] LINWONG, P., KATO, N., and NEMOTO, Y., 2004. A Polynomial Factorization Approach for the Discrete Time $GI^X/G/1/k$ Queue. *Methodology & Computing in Appl. Prob.* **6**, 277-291.
- [26] NEUTS, M. F., 1981. *Matrix-Geometric Solutions in Stochastic Models*. Johns Hopkins, University Press, Baltimore.

- [27] OTT, T. J., 1987. On the Stationary Waiting-Time Distribution in The GI/G/1 Queue I: Transform Method & Almost-Phase-Type Distributions. *Advanced in Appl. Prob.* **19**, 240-265.
- [28] PONSTEIN, J., 1974. Theory and Solution of a Discrete Queueing Problem. *Statist. Neerland.* **20**, 139-152.
- [29] POWEL, W. B., 1986. Analysis of Vehicle Holding and Cancellation Strategies in Bulk Arrival, Bulk Service Queues. *Trans. Sci.* **20**, 352-377.
- [30] PRABHU, N. U., 1980. *Stochastic Storage Processes*. Springer, New York.
- [31] RAO, B. V., and FELDMAN, R.M., 1999. Numerical Approximation for the Steady State Waiting Time in a GI/G/1 Queue, *Queueing Systems*, **31**, 25-42.
- [32] RIEGER, E. S., and HASSLINGER, G., 1994. An Analytical Solution for The Discrete Time Single Server System with Semi-Markovian Arrivals, *Queueing Systems*, **18**, 69-105.
- [33] ROSS, S. M., Introduction to Probability Models, Ed. 6. HARCOURT ASIA PTE LTD, India, 2000.
- [34] SHANTHIKUMAR, J. G., and BUZACOTT, J. A., 1980. On the Approximations of the Single Server Queue. *Int. J. Prod. Res.* **18**, 761-733.
- [35] SHENG, P., The Waiting Time Distribution in Queueing Systems, MSc dissertation, Graduate School of Mathematics, University of Saskatchewan, Saskatoon, Saskatchewan, 1988.
- [36] TAYLOR, H. M., and KARLIN, S., *An Introduction to Stochastic Modeling*, Academic Press, Florida.
- [37] WHITT, W., 1982. The Marshall and Stoyan Bounds for IMRL/G/1 Queues are Tight. *Opns. Res. Lett.* **1**, 209-213.
- [38] WHITT, W., 1988. An Interpolation Approximation for the Mean Workload in a GI/G/1 Queue. *Opns. Res. Vol.* **37**, 936-952.
- [39] ZHOU, W. H., 2005. Performance Analysis of Discrete-time Queue GI/G/1 with Negative Arrivals. *Appl. Maths. & computation*, **170**, 1349-1355.

Appendix

Program Listing

1. % THIS IS THE PROGRAM TO SOLVE THE U/U/1 WAITING TIME DISTRIBUTION PROBLEM

```
clear;
Max_S=25;Max_T=50;
P_S(1:Max_S)=1/(Max_S);P_T(1:Max_T)=1/(Max_T);UJ(1:Max_S-
1)=0;U_J(1:Max_T)=0;

for k=1:Max_S-1
    for i=1+k:Max_S
        PST=P_S(i)*P_T(i-k);UJ(k)=UJ(k)+PST;
    end
end
for k=0:Max_T-1
    if k<=Max_T-Max_S
        for i=k:Max_S+k-1
            P_ST=P_S(i+1-k)*P_T(i+1);U_J(k+1)=U_J(k+1)+P_ST;
        end
    else
        for i=k:Max_T-1
            P_ST=P_S(i+1-k)*P_T(i+1);U_J(k+1)=U_J(k+1)+P_ST;
        end
    end
end
end

g=Max_T-1; h=Max_S-1; Eps1=1e-6;t=0;Count_t=1; TIME=[];

for t=0:0.01:1
    tic;alfa(1:h,1)=0; m=1;
    b(1:g+1,m)=U_J(1:g+1)';S(1)=0;
    m=2;b(1:g+1,m)=U_J(1:g+1)';
    S(m)=((1-t)*(1-b(1,m))+t*sum(b(2:g+1,m)));
    alfa(1:h+h,m)=0;
    for j=h:-1:1
        alfa(j,m)=(UJ(j)+alfa(j+1:h+j,m) '*b(2:h+1,m))/S(m);
    end
    Max_dif=max(abs(alfa(1:h,m-1)-alfa(1:h,m)));
    if Max_dif<Eps1
        disp(':(=)');
    else
        while Max_dif>Eps1
            m=m+1;M(Count_t)=m;
            b(1:g+h+1,m)=0;
        end
    end
end
```

```

        for j=g:-1:0
            b(j+1,m)=U_J(j+1)+alfa(1:h,m-1)'*b(j+2:j+1+h,m);
        end
        S(m)=((1-t)*(1-b(1,m))+t*sum(b(2:g+1,m)));
        alfa(1:h+h,m)=0;
        for j=h:-1:1
            alfa(j,m)=(UJ(j)+alfa(j+1:h+j,m)'*b(2:h+1,m))/S(m);
        end
        Max_dif=max(abs(alfa(1:h,m-1)-alfa(1:h,m)));
    end
end

A123=1:h;w0=1-sum(alfa(1:h,m));Exp_w(Count_t)=(A123*alfa(1:h,m))/w0;
Time_exp=toc;TIME(Count_t)=Time_exp;

A321=fliplr(alfa(1:h,m)');
W=[w0 zeros(1,h)];
for i=1:h
    W(i+1)=[A321(h-i+1:h) zeros(1,h-i+1)]*W';
end
EXP_w(Count_t)=A123*W(2:h+1)';
variance(Count_t)=((A123.^2)*W(2:h+1)')- EXP_w(Count_t)^2;
%variance(Count_t)=(A123.^2)*alfa(1:h,m)/w0+(Exp_w(Count_t))^2;
Count_t=Count_t+1;
end

Opt_m=min(M);
Exp_w(find(M<=Opt_m));
%variance=(A123.^2)*alfa(1:h,m)/w0-(Exp_w).^2;

rho=(Max_S+1)/(Max_T+1);
lamda=1/((1+Max_T)/2);
L=lamda*Exp_w;
Miu=1/((1+Max_S)/2);
Var_T=((Max_T)^2-1)/12;
Var_S=((Max_S)^2-1)/12;
SCV_T=(lamda)^2*Var_T;
SCV_S=(Miu)^2*Var_S;

%COMPARING WITH THE FOLLOWING APPROXIMATE FORMULA
W_q=((SCV_T+SCV_S)/2)*(rho/(1-rho)*(1/Miu));
L_q=lamda*W_q;

```

2. % THIS IS THE PROGRAM TO SOLVE THE Geom/Geom/1 WAITING TIME DISTRIBUTION PROBLEM

```

clear;
Max_S=65;Max_T=65;p=0.2;q=0.3 ;
for ai=1:Max_S
    P_S(ai)=((1-p)*p^(ai-1))/(1-p^(Max_S));
end
for ai=1:Max_T
    P_T(ai)=((1-q)*q^(ai-1))/(1-q^(Max_T));
end

UJ(1:Max_S-1)=0;U_J(1:Max_T)=0;

```

```

for k=1:Max_S-1
    for i=1+k:Max_S
        PST=P_S(i)*P_T(i-k);UJ(k)=UJ(k)+PST;
    end
end
for k=0:Max_T-1
    if k<=Max_T-Max_S
        for i=k:Max_S+k-1
            P_ST=P_S(i+1-k)*P_T(i+1);U_J(k+1)=U_J(k+1)+P_ST;
        end
    else
        for i=k:Max_T-1
            P_ST=P_S(i+1-k)*P_T(i+1);U_J(k+1)=U_J(k+1)+P_ST;
        end
    end
end
end

g=Max_T-1; h=Max_S-1; Eps1=1e-6;t=0;Count_t=1; TIME=[];

for t=0:0.01:1
    tic;alfa(1:h,1)=0; m=1;
    b(1:g+1,m)=U_J(1:g+1)';S(1)=0;
    m=2;b(1:g+1,m)=U_J(1:g+1)';
    S(m)=((1-t)*(1-b(1,m))+t*sum(b(2:g+1,m)));
    alfa(1:h+h,m)=0;
    for j=h:-1:1
        alfa(j,m)=(UJ(j)+alfa(j+1:h+j,m) '*b(2:h+1,m))/S(m);
    end
    Max_dif=max(abs(alfa(1:h,m-1)-alfa(1:h,m)));
    if Max_dif<Eps1
        disp(':(=)');
    else
        while Max_dif>Eps1
            m=m+1;M(Count_t)=m;
            b(1:g+h+1,m)=0;
            for j=g:-1:0
                b(j+1,m)=U_J(j+1)+alfa(1:h,m-1) '*b(j+2:j+1+h,m);
            end
            S(m)=((1-t)*(1-b(1,m))+t*sum(b(2:g+1,m)));
            alfa(1:h+h,m)=0;
            for j=h:-1:1
                alfa(j,m)=(UJ(j)+alfa(j+1:h+j,m) '*b(2:h+1,m))/S(m);
            end
            Max_dif=max(abs(alfa(1:h,m-1)-alfa(1:h,m)));
        end
    end
end

A123=1:h;w0=1-sum(alfa(1:h,m));Exp_w(Count_t)=(A123*alfa(1:h,m))/w0;
Time_exp=toc;TIME(Count_t)=Time_exp;

A321=fliplr(alfa(1:h,m)');
W=[w0 zeros(1,h)];
for i=1:h
    W(i+1)=[A321(h-i+1:h) zeros(1,h-i+1)]*W';
end

```



```

EXP_w(Count_t)=A123*W(2:h+1)';
variance(Count_t)=((A123.^2)*W(2:h+1)')-EXP_w(Count_t)^2;
%variance(Count_t)=(A123.^2)*alfa(1:h,m)/w0+(Exp_w(Count_t))^2;
Count_t=Count_t+1;
end

Opt_m=min(M)
Exp_w(find(M<=Opt_m))
%variance=(Exp_w).^2 + (A123.^2)*alfa(1:h,m)/w0;

lamda=((1-q^(Max_T))*(1-q))/(1-(Max_T+1)*q^Max_T+Max_T*q^(Max_T+1));
L=lamda*Exp_w;
Miu=((1-p^Max_S)*(1-p))/(1-(Max_S+1)*p^Max_S+Max_S*p^Max_S+1);
rho=lamda/Miu;
Var_T=(1-2*q-
(Max_T+1)^2*(q^(Max_T))+((Max_T)^2+2*Max_T+2)*q^(Max_T+1))/((1-
q^Max_T)*(1-q));
Var_S=(1-2*p-
(Max_S+1)^2*(p^Max_S))+((Max_S)^2+2*Max_S+2)*p^Max_S+1)/((1-
p^Max_S)*(1-p));
SCV_T=(lamda)^2*Var_T;
SCV_S=(Miu)^2*Var_S;

% COMPARING WITH THE FOLLOWING APPROXIMATE FORMULA
W_q=((SCV_T+SCV_S)/2)*(rho/(1-rho)*(1/Miu));
L_q=lamda*W_q;

```

3. % THIS IS THE PROGRAM TO SOLVE THE Gamma/Geom/1 WAITING TIME DISTRIBUTION PROBLEM

```

clear;
Max_S=10;Max_T=10;p=0.2;A=2;B=1.25;
for ai=1:Max_S
    P_S(ai)=((1-p)*p^(ai-1))/(1-p^(Max_S));
end
% for ai=1:Max_T
%     if ai==1
%         P_T(1)=gamcdf(1,A,B);
%     else P_T(ai)=gamcdf(ai,A,B)-P_T(ai-1);
%     end
%
% end

for ai=1:Max_T
%   GMCDF=0;
%   for N=0:1:A-1
%   GCDF=((ai/B)^N)*(exp(-ai/B))/factorial(N);
%   GMCDF=GMCDF+GCDF;
%   end
%   Gama_CDF=1-GMCDF;
%   P_T(ai)=Gama_CDF-F0;
%   F0=Gama_CDF;
%   N=0:A-1;

```

```

% Secndprt=((ai/B).^N)./factorial(N);
% P_T(ai+1)=1-exp(-ai/B)*sum(Secndprt)-P_T(ai);

P_T(ai)=((B^A)*(ai^(A-1))*exp(-ai/B))/factorial(A-1);
end
UJ(1:Max_S-1)=0;U_J(1:Max_T)=0;

for k=1:Max_S-1
    for i=1+k:Max_S
        PST=P_S(i)*P_T(i-k);UJ(k)=UJ(k)+PST;
    end
end
for k=0:Max_T-1
    if k<=Max_T-Max_S
        for i=k:Max_S+k-1
            P_ST=P_S(i+1-k)*P_T(i+1);U_J(k+1)=U_J(k+1)+P_ST;
        end
    else
        for i=k:Max_T-1
            P_ST=P_S(i+1-k)*P_T(i+1);U_J(k+1)=U_J(k+1)+P_ST;
        end
    end
end
end

g=Max_T-1; h=Max_S-1; Epsl=1e-6;t=0;Count_t=1;TIME=[];
for t=0:0.01:1
    tic;alfa(1:h,1)=0; m=1;
    b(1:g+1,m)=U_J(1:g+1)';S(1)=0;
    m=2;b(1:g+1,m)=U_J(1:g+1)';
    S(m)=((1-t)*(1-b(1,m))+t*sum(b(2:g+1,m)));
    alfa(1:h+h,m)=0;
    for j=h:-1:1
        alfa(j,m)=(UJ(j)+alfa(j+1:h+j,m) '*b(2:h+1,m))/S(m);
    end
    Max_dif=max(abs(alfa(1:h,m-1)-alfa(1:h,m)));
    if Max_dif<Epsl
        disp(':(=)');
    else
        while Max_dif>Epsl
            m=m+1;M(Count_t)=m;
            b(1:g+h+1,m)=0;
            for j=g:-1:0
                b(j+1,m)=U_J(j+1)+alfa(1:h,m-1) '*b(j+2:j+1+h,m);
            end
            S(m)=((1-t)*(1-b(1,m))+t*sum(b(2:g+1,m)));
            alfa(1:h+h,m)=0;
            for j=h:-1:1
                alfa(j,m)=(UJ(j)+alfa(j+1:h+j,m) '*b(2:h+1,m))/S(m);
            end
            Max_dif=max(abs(alfa(1:h,m-1)-alfa(1:h,m)));
        end
    end
end
A123=1:h;w0=1-
sum(alfa(1:h,m));Exp_w(Count_t)=(A123*alfa(1:h,m))/w0;Time_exp=toc;TIME(Co
unt_t)=Time_exp;
A321=fliplr(alfa(1:h,m)');

```

```

W=[w0 zeros(1,h)];
for i=1:h
    W(i+1)=[A321(h-i+1:h) zeros(1,h-i+1)]*W';
end
EXP_w(Count_t)=A123*W(2:h+1)';
variance(Count_t)=(A123.^2)*W(2:h+1)')- EXP_w(Count_t)^2;
%variance(Count_t)=(A123.^2)*alfa(1:h,m)/w0+(Exp_w(Count_t))^2;
Count_t=Count_t+1;
end
Opt_m=min(M)
Exp_w(find(M<=Opt_m));
%variance=(Exp_w).^2 + (A123.^2)*alfa(1:h,m)/w0;
lamda=1/(A*B);
L=(lamda)*(Exp_w);
Miu=((1-p^Max_S)*(1-p))/(1-(Max_S+1)*p^(Max_S)+Max_S*p^(Max_S+1));
rho=lamda/Miu
Var_T=A*B^2;
Var_S=(1-2*p-
(Max_S+1)^2*(p^(Max_S))+((Max_S)^2+2*Max_S+2)*p^(Max_S+1))/(1-
p^Max_S)*(1-p));
SCV_T=1/A;
SCV_S=(Miu)^2*(Var_S);
% COMPARING WITH THE FOLLOWING APPROXIMATE FORMULA
W_q=((SCV_T+SCV_S)/2)*(rho/(1-rho)*(1/Miu));
L_q=lamda*W_q;

```