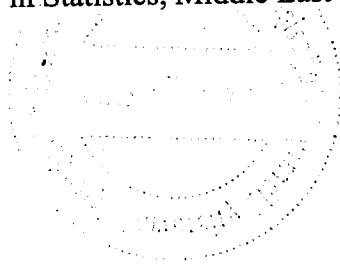


**SPREADSHEET SIMULATION FOR QUEUING SYSTEMS AND REAL
IMPLEMENTATION**

by

Zeynep Başkurt

B.S. in Statistics, Middle East Technical University, 1999



Bogazici University Library



39001102132571

14

**Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science**

Graduate Program in Industrial Engineering

Boğaziçi University

2004

ACKNOWLEDGEMENTS

I would like to express my gratitude to my thesis supervisor, Assoc. Prof. Wolfgang Hörmann for his invaluable guidance, advice and contributions throughout this study.

I wish to thank Prof. Yaman Barlas, for serving on my thesis committee and especially for his contributions about this study. I am thankful to Assist. Prof. Aslı Erdem, for her comments and suggestions about this study as a member of my thesis committee.

I wish to thank all my friends for their encouragement and moral supports. I would especially like to thank Arzu Sunam Kayk1 for her great concern and support. I want to thank Işıl Ünlü and Vildan Yüksel for their moral support. And I want to thank much to Mehmet Güzel for always supporting about technical problems with effort.

I want to mention my grateful to my directors in the office, Nilsu Güner and Neş e Kuzu for their high tolerance and support.

Finally, I would like to thank to my mother, my father, my brothers and my husband Fatih Tandoğan for their moral support and patience not only during this study, but also throughout my life.

ABSTRACT

SPREADSHEET SIMULATION FOR QUEUING SYSTEMS AND REAL IMPLEMENTATION

Spreadsheets have become popular during recent years. Spreadsheet simulation involves the use of a spreadsheet to represent the model and conduct sampling. Spreadsheets such as Lotus 1-2-3, Microsoft Excel and Quattro Pro are popular software products, and the skills to program and use them are widely distributed. The aim of this study is to use Excel spreadsheet as a tool for real world queuing simulation modeling. Instead of commercial add-in packages such as @RISK, CrystalBall and INSIGHT.xla, visual basic macros are used as needed in simulations. The process-interaction approach is implemented. Spreadsheet implementations of simple network M/G/c queue at serial servers model with balk and renege properties of customers and break hours of servers are constructed. Moreover, the distribution of the queue length is estimated.

Spreadsheet simulation is applied to a real world queuing problem. And, it is concluded that the spreadsheet simulation model is a good representation of the real queuing problem. Despite of some limitations, spreadsheet simulation can be used for real world queuing problems. Moreover, it can also be a useful tool for teaching queuing simulation.

ÖZET

KUYRUK SİSTEMLERİNDE ELEKTRONİK TABLOLAMA BENZETİMİ VE GERÇEK UYGULAMASI

Elektronik tablolama benzetimine olan ilgi son yıllarda artmaktadır. Elektronik tablolama benzetimi bir modeli temsil etmek ve örneklemeyi yönlendirmek üzere bir elektronik tablolama programı kullanmayı içerir. Lotus 1-2-3, Microsoft Excel ve Quattro Pro gibi elektronik tablolama programları en popüler yazılımlardır. Bu yazılımları programlama ve kullanma becerikliliği adamakıllı yaygınlaşmaktadır. Bu çalışmanın amacı, Excel'de uygulanan elektronik tablolama benzetimini gerçek bir kuyruk probleminin benzetiminde kullanmaktır. Benzetim için, @RISK, CrystalBall ve INSIGHT.xla gibi ticari paket programlar yerine, visual basic makroları kullanılmıştır. Elektronik tablolama uygulamasında süreç etkileşim yaklaşımı kullanılmıştır. Basit seri halinde olan M/G/c modellerinin, uzun kuyruk sırası olması halinde geri dönen (balk) ve bir süre bekledikten sonra geri dönen (renege) müşteri tiplerini içeren elektronik tablolama uygulamaları gösterilmiş, ayrıca servis verenlerin öğle tatillerini dikkate alan kuyruk modellerinin elektronik tablolama uygulaması da denenmiştir. Bunu yanında, sıra uzunluğunun dağılımı hesaplanmıştır.

Çalışmanın son aşamasında, elektronik tablolama benzetimi gerçek bir banka şubesi kuyruk problemine uygulanmıştır. Bunun sonucunda, kuyruk problemlerinin çözümünde, elektronik tablolama benzetiminin iyi bir yaklaşım olabileceği sonucuna varılmıştır. Bazı kısıtlamalara rağmen, elektronik tablolama benzetimi gerçek kuyruk problemlerinin çözümünde kullanılabilir. Bundan başka, okullarda kuyruk benzetimi konusunu tartışmada yararlı bir araç olabilir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
ÖZET	v
LIST OF FIGURES	x
LIST OF TABLES	xii
LIST OF SYMBOLS / ABBREVIATIONS.....	xviii
1. INTRODUCTION	1
2. SPREADSHEET SIMULATION.....	3
2.1. What is Spreadsheet Simulation?	3
2.2. Spreadsheet Modeling	4
2.3. When is simulation Using a Spreadsheet Appropriate?	5
2.4. Building Spreadsheet Simulation Models	6
2.5. Limitations of Using Spreadsheet Simulation	6
2.6. Spreadsheets as a Tool for Teaching Queuing Models	7
3. QUEUING SYSTEMS.....	9
3.1. Characteristics of Queuing Systems	10
3.1.1. The Calling Population	10
3.1.2. System Capacity	10
3.1.3. The Arrival Process	10
3.1.4. Queue Behavior and Queue Discipline	11
3.1.5. Service Times and the Service Mechanism.....	11
3.2. Queuing Notation	12
3.3. Long-Run Measures of Performance of Queuing Systems.....	13
3.3.1. Time-Average Number in System, L , in the Queue, L_Q	13
3.3.2. Average Time Spent in System per Customer, w	14
3.3.3. The Conservation Equation: $L=\lambda w$ (Little's Equation).....	14
3.3.4. Server Utilization in $G/G/1$ and $G/G/c$	14
3.4. Steady-State Behavior of Infinite Population Markovian Models.....	15
3.4.1. Single-Server Queues with Poisson Arrivals and Unlimited Capacity: $M/G/1$	16

3.4.2. An Approximation for the M/G/c Queue.	17
4. THE GENERATION OF RANDOM VARIATES	19
4.1. Random Number Generation	19
4.1.1. A Technique for Generating Random Numbers.	20
4.2. Random Variate Generation	22
4.2.1. Inverse Transform Method	23
4.2.1.1. Generating Interarrival Times from Exponential Distribution	24
4.2.1.2. Generating Arrival Times from Nonhomogeneous Poisson Process	25
4.2.1.3. Generating Service Times from the Weibull Distribution.	26
4.2.1.4. Generating Service Times from the Triangular Distribution.	27
5. IMPLEMENTATION OF QUEUING SIMULATION IN SPREADSHEETS.	30
5.1. Event-scheduling Spreadsheet Simulation.	31
5.1.1. One Server Queuing Systems.	32
5.1.2. Two Server Queuing Systems	32
5.2. Process-Interaction Spreadsheet Simulation.	33
5.2.1. Basic Queuing Models	34
5.2.1.1. The Spreadsheet Implementation of M/G/1 Model.	36
5.2.2. More Complicated Queuing Models	40
5.2.2.1. M/G/c Models with the Distributions of No. Customers in the System and Queue Length	44
5.2.2.2. M/G/c Models with Balking Property	52
5.2.2.3. M/G/c Models with Balking and Reneging Properties. ...	53
5.2.2.4. Simple Network Queuing Models with M/G/c Queue at Serial Servers with Balking and Reneging Properties. ...	55
5.2.2.5. Simple Network Queuing Models with M/G/c queue at Serial Servers with Balking and Reneging Properties and Servers Break Time.	62
5.2.2.6. M/G/c Models with Changing Service Times in case of Long Queue Length	63
6. AN ACTUAL APPLICATION OF SPREADSHEET SIMULATION OF	

QUEUEING SYSTEMS.	66
6.1. Input Data Modeling.	68
6.1.1. The Distribution of Interarrival Times.	68
6.1.2. The Distribution of Service Times.	71
6.1.3. The Distribution of Types of Transactions.	73
6.1.3.1. Point Estimation of π 's	73
6.2. Spreadsheet Implementation	74
6.2.1. Spreadsheet Implementation on Excel Worksheet.	74
6.2.2. Calculating Performance Measures.	89
6.3. Simulation by Using Visual Basic Application.	94
6.4. Verification and Validation of Simulation Model.	101
6.4.1. Verification of the Model	102
6.4.2. Validation of the Model	103
6.4.2.1. Taking Averages of Transaction Times..	104
6.4.2.2. Taking Transaction Times with True Probabilities	106
6.4.2.3. Taking Transaction Times with Different Probabilities. .	107
6.5. Output Analysis.	109
6.6. What-if Analysis.	113
6.6.1. Decreasing Server Number by one at the 1st floor: Case 1.	113
6.6.1.1. Independent Sampling with Equal Variances..	113
6.6.1.2. Correlated Sampling.	114
6.6.2. Changing Periods of Break Hours: Case 2	115
6.6.2.1. Independent Sampling with Equal Variances..	115
6.6.2.2. Correlated Sampling.	117
6.6.3. Increasing Server Number by one at the 1st floor: Case 3.	119
6.6.3.1. Independent Sampling with Equal Variances..	119
6.6.3.2. Correlated Sampling.	120
6.7. Discussion.	121
7. CONCLUSION	122
APPENDIX A: EXCEL WORKSHEETS	124
APPENDIX B: FIGURES	174
APPENDIX C: PROGRAMS CODED IN VBA.	192
APPENDIX D: FORMULAS.	201

REFERENCES..... 203

LIST OF FIGURES

Figure 5.1.	Queuing system with balk property number	35
Figure 5.2.	Queuing system with balk property number	52
Figure 5.3.	Queuing system with balk and renege property	54
Figure 5.4.	A simple network queuing system with balk and renege properties	55
Figure 6.1.	Flowchart of queuing system in the Central Bank	67
Figure 6.2.	Distribution of number of arriving customers during summer months.....	69
Figure 6.3.	Distribution of mean number of arriving customer in one day, in August .	70
Figure 6.4.	The graph of comparison of server's utilization	110
Figure B.1.	Event-driven M/M/1, approximate time plot of average number in queue, Evans	175
Figure B.2.	Basic queuing model, customer graph, Grossman	176
Figure B.3.	Basic queuing model, server graph, Grossman	177
Figure B.4.	The graph of queue length at customer arrivals, Hora	178
Figure B.5.	The graph of number of server utilized with respect to customer arrivals, Hora.....	179
Figure B.6.	The graph of waiting time and total time that customers spend in the system with respect to customer arrivals, Hora	180

Figure B.7. Figure B.7. Empirical state of the system frequency, Hora.....	181
Figure B.8. M/G/c queueing model, the graph of distribution of average time spent in the system.....	182
Figure B.9. M/G/c queueing model, the graph of distribution of average waiting time in queue	183
Figure B.10. M/G/c queueing model, the graph of distribution of servers utilization	184
Figure B.11. M/G/c queueing model, the graph of distribution of customer number in the system	185
Figure B.12. M/G/c queueing model, the graph of distribution of queue length	186
Figure B.13. The graph of distribution of average time spent at floor 1	187
Figure B.14. The graph of distribution of average time spent at floor 2.....	188
Figure B.15. The graph of distribution of average time spent at stage 2	189
Figure B.16. The graph of probability distribution of customers having X minutes service times.....	190
Figure B.17. The graph of probability distribution of customers waiting Y minutes in queue	191

LIST OF TABLES

Table 3.1.	Summary of the most common symbols	12
Table 4.1.	Parameters of LCM	21
Table 4.2.	Random number generation in Excel	22
Table 4.3.	Parameters for nonhomogeneous Poisson arrivals in “data” worksheet	25
Table 5.1.	Spreadsheet implementation of M/G/1 model	36
Table 5.2.	Spreadsheet implementation formulas for M/G/1 model	37
Table 5.3.	Performance measure formulas for M/G/1 model	38
Table 5.4.	Spreadsheet implementation of M/G/c model	45
Table 5.5.	Performance measure formulas for M/G/c model	47
Table 5.6.	Probability distribution of no. of customers in the system for M/G/c model	50
Table 5.7.	Probability distribution of no. of customers in the queue for M/G/c model	51
Table 5.8.	Spreadsheet implementation of the simple network queue model	57
Table 5.9.	Simple network queuing system with balking and reneging properties, performance measure formulas	61
Table 5.10.	Spreadsheet implementation of servers’ break time	62

Table 5.11.	M/G/c model with changing service time	64
Table 6.1.	Q-MATIC print output of the summary data	68
Table 6.2.	Triangular distribution parameters for generating transaction times	72
Table 6.3.	The distribution of no. of arriving customers according to the transaction types	73
Table 6.4.	The transaction probability array.....	74
Table 6.5.	The “data” Excel worksheet for spreadsheet implementation of queuing system in the Bank	75
Table 6.6.	The “Spreadsheet” Excel worksheet for spreadsheet implementation of queuing system in the Bank	77
Table 6.7.	Performance measures for the first floor.....	89
Table 6.8.	Performance measures for the second floor	90
Table 6.9.	Performance measures for the second stage.....	91
Table 6.10.	Performance measures for the whole system	92
Table 6.11.	Probability distribution of customers having X minutes service	93
Table 6.12.	Probability distribution of customers waiting Y minutes.....	93
Table 6.13.	Finding number of arriving customers between 09:00 and 12:00.....	94
Table 6.14.	Simulation output at “Simulation” Excel worksheet	99

Table 6.15. Simulation output at “Validation” Excel worksheet	100
Table 6.16. Simulation output at “Service_Time” Excel worksheet.....	100
Table 6.17. Simulation output at “Waiting_Time” Excel worksheet.....	101
Table 6.18. Triangular distribution parameters	104
Table 6.19. Triangular distribution parameters, 50% probabilities.....	107
Table 6.20. Probability distribution of customers having X minutes service time	111
Table 6.21. Probability distribution of customers waiting Y minutes.....	111
Table 6.22. 95% c.i. for the difference of the means of performance measures, independent sampling for Case 1	113
Table 6.23. 95% c.i. for the mean difference, correlated sampling for Case 1	114
Table 6.24. 95% c.i. for the difference of the means of performance measures, independent sampling for Case 2	116
Table 6.25. 95% c.i. for the mean difference, correlated sampling for Case 2	118
Table 6.26. 95% c.i. for the difference of the means of performance measures, independent sampling for Case 3	119
Table 6.27. 95% c.i. for the mean difference, correlated sampling for Case 3	120
Table A.1. Spreadsheet implementation of M/M/1 queuing model, Evans and Olson.	125
Table A.2. Spreadsheet implementation of M/M/1 queuing model formulas, Evans and Olson	126

Table A.3.	Spreadsheet implementation of M/M/1 queuing model, Evans	127
Table A.4.	Spreadsheet implementation of M/M/1 queuing model formulas, Evans...	128
Table A.5.	Crystal Ball simulation results for single-server queuing model	129
Table A.6.	Basic queuing model, "Data" sheet, Grossman.....	130
Table A.7.	Basic queuing model, "Simulation" and "Computations" sheet, Grossman	131
Table A.8.	Basic queuing model, "Simulation" sheet formulas, Grossman	132
Table A.9.	Basic queuing model, "Computation" sheet formulas, Grossman	133
Table A.10.	Two server queuing model, Evans and Olson	134
Table A.11.	Three server queuing model, "Simulation" and "Computations" sheets, Grossman.....	135
Table A.12.	Three server queuing model, "Simulation" sheet formulas, Grossman	136
Table A.13.	Three server queuing model, "Calculation" sheet formulas, Grossman	137
Table A.14.	G/G/c queuing model, Hora	138
Table A.15.	G/G/c queuing model formulas, Hora	139
Table A.16.	M/G/c queuing model, "data" sheet	141
Table A.17.	M/G/c queuing model, "Spreadsheet" sheet	142
Table A.18.	M/G/c queuing model, "Spreadsheet" sheet, data for performance	

measures	143
Table A.19. M/G/c queuing model, "Spreadsheet" sheet, probability distribution of no. of customers in the system and in the queue	144
Table A.20. M/G/c queuing model, "Spreadsheet" sheet	145
Table A.21. M/G/c queuing model, "Simulation_Summary" worksheet	146
Table A.22. M/G/c queuing model, "Simulation_cust" worksheet	147
Table A.23. M/G/c queuing model, "Simulation_que" worksheet	148
Table A.24. M/G/c queuing model with balking property, "Spreadsheet" sheet	149
Table A.25. M/G/c queuing model with balking and reneging properties, "Spreadsheet" sheet	150
Table A.26. Simple_network queuing models with M/G/c queue at serial servers with balking and reneging properties, "Spreadsheet" sheet	151
Table A.27. Simple_network queuing models with M/G/c queue at serial servers with balking and reneging properties and servers break time, "data" sheet	153
Table A.28. Simple_network queuing models with M/G/c queue at serial servers with balking and reneging properties and servers break time, "Spreadsheet" sheet	154
Table A.29. Number of arriving customers day by day at June, July and August	155
Table A.30. Mean number of arriving customers in time periods at August	158

Table A.31. Average number of arriving customers in time periods at August.....	159
Table A.32. Average number of arriving customers in time period 09:00-12:00	160
Table A.33. Probability distribution of customers having X minutes service	161
Table A.34. Probability distribution of customers waiting Y minutes in queue.....	162
Table A.35. Data for average waiting time of customers in the queue	163
Table A.36. Simulation results after 1000 runs at “Simulation_Summary” Excel worksheet	164
Table A.37. Simulation results after 1000 runs at “Simulation_Summary” Excel worksheet formulas	165
Table A.38. Probability distributions of some performance measures at “Simulation_Summary” Excel worksheet.....	167
Table A.39. Simulation results after 1000 runs at “Simulation_Summary” Excel worksheet, Case 1.....	168
Table A.40. Probability distribution of customers waiting Y minutes in Case 1.....	169
Table A.41. Simulation results after 1000 runs at “Simulation_Summary” Excel worksheet, Case 2.....	170
Table A.42. Probability distribution of customers waiting Y minutes in Case 2.....	171
Table A.43. Simulation results after 1000 runs at “Simulation_Summary” Excel worksheet, Case 3.....	172
Table A.44. Probability distribution of customers waiting Y minutes in Case 3.....	173

LIST OF SYMBOLS / ABBREVIATIONS

L	The long-run time average number of customers in the system
L_Q	The long-run time average number of customers in the queue
S^2	Sample variance
s.e.	Standard error
$U(0,1)$	Uniform distribution between 0 and 1
w	The long-run average time spent in the system
w_Q	The long-run average time spent in the queue
λ	Arrival rate
μ	Service rate
ρ	The server utilization
ν	Degrees of freedom
c.i.	Confidence interval
cdf	Cumulative density function
cv	Correction factor
FIFO	First-in, first-out
i.i.d	Independent and identically distributed
LCM	Linear Congruential Method
LIFO	Last-in, first-out
p.e.	Point estimation
pdf	Probability density function
PR	Service according to the priority
SIRO	Service in random order
SPT	Shortest processing time first
VBA	Visual Basic Application

1. INTRODUCTION

The aim of this study is to use excel spreadsheet as a tool for real world queuing simulation modeling. When we investigated the spreadsheet simulation of queuing models in the literature, we saw that interest in spreadsheet simulation has grown during recent years. There are challenging studies on web sides of INFORMS Transactions on Education, which is a peer-reviewed electronic journal, published by the Institute for Operations Research and the Management Sciences, and Winter Simulation Conference. Grossman had built spreadsheet implementation of queuing models with up to 12 servers with balk and renege properties of customers [1]. But, each server is represented by one column, thus increasing server number means increasing the program load. Recently, Hora achieved spreadsheet implementation with the number of servers being an adjustable parameter, i.e. there is only one column for all servers, and the server number can be changed without changing the structure of the spreadsheet [2]. Of course, this makes the program execution faster when compared to Grossman's 12-server engines. Through Hora's approach, we implement spreadsheet simulation of a simple network M/G/c queue at serial servers with balk and renege properties of customers and break hours of the servers. Moreover, simulation with commercial add-in packages such as @RISK, Crystal Ball and INSIGHT.xla are recommended in the literature. We seek to show that we can use spreadsheets to implement simple realistic queuing models without buying commercial add-ins. So we use simple Visual Basic macros to perform simulation in excel.

In Chapter 2, literature of spreadsheet simulation is investigated. In chapter 3, basic queuing theory is described. Characteristics of queuing system is summarized, i.e. calling population, system capacity, arrival process, queue behavior and queue discipline, service times and the service mechanism.

In Chapter 4, random variate generation that will be used to generate arrival and service times in queuing models is presented. The Linear Congruential Method, for generating uniform random numbers and the Inverse Transform Method for generating

non-uniform random variates is described. Probability distributions for generating interarrival times and service times are listed.

In Chapter 5, we discuss the implementation of queuing models using excel. We concentrate on the process-interaction approach. Only using excel functions, it is possible to calculate the distribution of the queue length. The models that are presented in Chapter 5 include basic queuing models with single server and multiple servers. We also consider balk and renege properties of customers and break hours of the servers. Moreover, a simple network queuing model is constructed with multiple servers in each step. An approach to spreadsheet implementation of multiple server queuing models, where service times are affected by the number of customers in the queue is studied.

Finally, in Chapter 6, spreadsheet simulation is applied to a real world queuing problem: The customer service department called "Laborer Foreign Exchange Department" in the Central Bank of the Republic of Turkey. The queuing system here is a simple network with four kinds of transactions, A, B, C and D and two stages. Validation and verification of the model is performed. Confidence intervals for the means of performance measures are found. Also, what-if analysis is conducted as well.

2. SPREADSHEET SIMULATION

2.1. What is Spreadsheet Simulation?

Simulation is the imitation of the operation of a real world process or system over time. Simulation involves the generation of an artificial history of a system, and the observation of that artificial history to draw inferences concerning the operating characteristics of the real system [3]. The most significant advantage of simulation is that it can provide answers for the problems that are difficult, or even impossible, to solve in a purely mathematical way. Simulation thoroughly evaluates each alternative by generating a series of values for each random variable at the frequencies indicated by their probability distributions. And, it is a procedure that tries out each alternative over and over again; in fact it can also be seen as a sample from the future [4].

Simulation has been an important tool in management science throughout its history. Pritsker *et al*, mentioned that most applications of simulation have revolved around dynamic systems, i.e. inventory, project planning, queuing and scheduling, production, material handling and logistics, and service and communication systems [5].

The behavior of the system as it evolves over time is studied by developing a simulation model [3]. Simulation models are logical descriptions of the interrelationships among elements of a decision problem, or the sequence of events that occur in a system over time [6]. Spreadsheet simulation involves the use of a spreadsheet to represent the model and conduct the sampling [7].

The idea of an electronic spreadsheet for storing information and performing calculations dates back to Mattesich [8] when computers were mainframes, but spreadsheets as we know them are a product of the microcomputer age. The earliest spreadsheet, VisiCalc, had limited functionality but introduced to the concept of the electronic spreadsheet, i.e., a program that stores data and computations in a rectangular array of cells. Later spreadsheets such as Lotus 1-2-3, Microsoft Excel and Quattro Pro

greatly expanded the features and developed the spreadsheet into an effective modeling, prototyping, analysis and presentation tool [7]. By the late 1990s the use of spreadsheet has become routine in all areas of business and accounting [9]. Spreadsheet software such as Microsoft Excel and Lotus 1-2-3 has dramatically changed the way management science is applied in business. They have provided managers and analysts alike with powerful means to model and analyze complex business decisions and to incorporate management science tools into these analyses [10].

2.2. Spreadsheet Modeling

A software environment that will be used to develop, test and run a simulation experiment must have some capabilities, i.e. it must be able to represent mathematical and logical relationships between variables in the form of computations and assignment of values, and has algorithms that describe how to do a series of computations. Also, it has a way to generate random numbers and random variates. Moreover, implementing replications must be available. These are the must requirements that a software environment must have. All popular spreadsheets have these features. In addition, some features must be available in most spreadsheets to make the process quick and reliable which are; availability of functions to do mathematical, statistical, database, financial and other calculations; database representations and database access; charting and graphing, display and documentation features such as fonts, colors and geometric shapes to improve presentation; and automation through scripting languages such as Visual Basic Application (in case of Excel) [7].

Moreover, the table structure of spreadsheets allows the developer to organize the computations and results in a natural and intuitive manner. Spreadsheets are ubiquitous and files written by one spreadsheet can usually pass simulation models from one to another [7]. Spreadsheets provide a natural interface for model building, are easy to use in terms of inputs, solutions and report generation, and allow users to perform what-if analysis [9]. Also, the entry level, in terms of the cost and experience necessary is low [11]. The flexibility of spreadsheets and their statistical capabilities make them a natural framework

for simulation modeling [10]. For these reasons, the spreadsheet is an attractive platform for simulation [7].

Today, spreadsheets are available for all of the major operating systems: Windows, Unix/Linux and Mac OS. The most prevalent spreadsheet today is Microsoft Excel [7]. Excel has extensive data analysis capabilities and on-line help and good manuals and training guides are available [10].

2.3. When is Simulation Using a Spreadsheet Appropriate?

Certain modeling situations can be implemented in a spreadsheet. Indeed, every set of calculations can be considered a model [7].

In some cases, the unknown parameters are actually random variables whose value can not be predicted, i.e. stochastic models. Many stochastic models in finance, logistic and engineering can be conveniently set up in a spreadsheet for simulation [7]. For example, suppose a queuing system in which the arrival rates of the customers are stochastic. The arrival rates must be samples when the simulation experiment is run. An experiment could consist of sampling arrival times and running the queuing system over a long period of time to compute observations of the waiting time of the customers. These observations could then be used to estimate the average waiting times of the customers in the queue. The experiment could be repeated for several scenarios to find the number of servers required in order to minimize the customers waiting time and adding servers cost at the same time. This is a typical stochastic model that can be analyzed using simulation.

Spreadsheet simulation is also useful when doing “what-if” or sensitivity analysis for models having unknown parameters that are not necessarily random. Modelers frequently want to determine how sensitive the performance measure is to variations in these parameters, which can easily be done in spreadsheet models [7]. For example, in a model that concerns whether the leasing or purchasing of a real estate, the mortgage interest rate at the time the contract is signed is a parameter that is unknown. The present value of each decision (lease vs. buy) will depend upon this parameter value.

2.4. Building Spreadsheet Simulation Models

Generally, each cell in a spreadsheet model can be classified as containing one of three types of quantities [7]:

- Inputs to the model. These can be parameters that are part of the model.
- Intermediate computations. These are calculations that are involved in the model.
- Outputs of the model. These are the observations on quantities of interest one seeks from the model.

In general, building a simulation model involves the following steps [6];

- Formulating the problem to determine the objectives of the simulation study, the system performance measures to be computed, and the scenarios to be evaluated.
- Developing a logical model of the system under study that describes key events and actions that take place. Visual models provide the basis for implementing the simulation model on a spreadsheet.
- Specifying probabilistic assumptions needed to drive the simulation, i.e. the probability distributions of input variables.
- Implementing the model on a computer to perform the necessary calculations required to evaluate the distribution of model outputs.

2.5. Limitations of Using Spreadsheet Simulation

Spreadsheets are powerful, convenient tools for simulation modeling. But, they also have some limitations;

- Spreadsheets can not be used for more complex problems [6]. They do not have convenient facilities to implement a while-loop or a for-loop [7].
- Spreadsheets are slower than some alternatives [7]. Formulas are stored in “source code”. That is, the spreadsheet must interpret the formula before it can be executed.

This interpretation action normally takes much longer than the execution. A model that is very large and/or requires long simulation runs would need to be programmed in a compiled language in order to execute in a feasible length of time or use a reasonable amount of main memory.

- Data storage is limited [7]. Since the output data must be stored in the spreadsheet, usually in a column, the length of the output series is limited.
- Only simple data structures are available [7]. The spreadsheet consists of a group of pages, each of which has a table consisting of rows and columns of cells. One can treat a column or row of cells as a vector, and a two-dimensional range of cells can be treated as a two-dimensional array, or matrix. In some simulation models, more elaborate data structures such as lists and trees are needed.

2.6. Spreadsheets as a Tool for Teaching Queuing Simulation

There are three approaches to teach about queuing systems: analytical queuing theory, dedicated simulation software, and spreadsheet simulation. Queuing theory is widely taught in business school OR/MS survey courses, and students can use it quickly compute a handful of steady-state performance measures, anyway queuing theory is not to particularly useful for understanding what actually transpires in queues. Furthermore, simple performance measures, such as the 95th percentile of waiting time or the fraction of customers who wait more than Y minutes often can not be computed using queuing theory [1]. Simulation using dedicated software packages is flexible, powerful, and it is often the best technology for computing quantitative results. Many simulation packages are available, such as SLAM, GPSS, Extent+ and Arena [12]. However, simulation packages hide the mechanics of queues, and are expensive and time-consuming to learn [13].

Spreadsheet queuing simulation can be used to give students insight into queue behavior -- why queues form, why they persist, the role of variability, and the non-linear effect of increasing utilization on queue performance [13]. Spreadsheet shows the experiences of individual customers and servers explicitly [1].

Thus, spreadsheets provide an ideal platform on which to learn the basics of simulation methodology [6].

3. QUEUING SYSTEMS

Waiting lines occur in many business operations. Customers arrive from time to time and join a queue or waiting line, are eventually served and finally leave the system. Many service systems; banks, fast-food restaurants, gasoline stations, production systems can be viewed as queuing systems [3].

As a field of study, queuing theory is one of the richest areas of operations research methodology. The number of models representing specific situations has grown steadily over the years, and new ones are being reported even now. Queuing theory is one of the earliest quantitative methods. Its origin is the 1909 paper of the Danish telephone engineering, Erlang, whose name is associated with a large class of probability distributions used in conjunction with mathematical queuing models. Since then, thousands of articles and numerous books have been written on the subject [4].

Queuing models, whether solved mathematically or analyzed through simulation, provide the analyst with a powerful tool for designing and evaluating the performance of queuing system. Typical measures of the system performance include server utilization (percentage of time a server is busy), length of waiting lines, and delays of customers (waiting time of customers in queue). Queuing theory and simulation analysis are used to predict these measures of system performance as a function of the input data. The input parameters include the arrival rate of customer, the service demands of customers, the rate at which the server works, and the number and arrangement of servers. To a certain degree, some of the input parameters are under their indirect control, provided that the relationship between the performance measures and the input parameters is adequately understood for the given system.

For relatively simple systems these performance measures can be computed mathematically at great savings in time and expense compared to the use of simulation model. Anyway, for realistic models of complex systems, simulation is required. Nevertheless, analytically tractable models, although usually requiring many simplifying assumptions, are valuable for rough-cut estimates of system performance [3].

3.1. Characteristics of Queuing Systems

The key elements of a queuing system are customers and servers. The term “customer” can refer to people, machines, trucks, mechanics, patients, etc., anything that arrives at a facility and requires service. The term “server” might refer to receptionist, repairpersons, mechanics, etc., any resource (people, machine, etc.) which provides the requested service [3].

3.1.1. The Calling Population

The population of potential customers, referred to as the calling population, may be assumed to be finite or infinite. In systems with a large population of potential customers, the calling population is usually assumed to be infinite. The main difference between finite and infinite population models is how the arrival rate is defined. In an infinite population model, the arrival rate (i.e. the average number of arrivals per unit of time) is not affected by the number of customers who have left the calling population and joined the queuing system [3].

In our study, we assume the calling population to be infinite.

3.1.2. System Capacity

In many queuing systems, there is a limit to the number of customers that may be in the waiting line or system. Some systems, such as concert ticket sales for students, may be considered as having unlimited capacity [3].

In our study, we mostly assume unlimited capacity.

3.1.3. The Arrival Process

The arrival process for infinite-population is usually characterized in terms of interarrival times of successive customers. Arrivals may occur at scheduled times or at

random times. The most important model for random arrivals is the Poisson arrival process [3].

In our study, at Chapter 5, we generate random interarrival times from Poisson process. And in Chapter 6, we generate interarrival times from non-homogeneous Poisson process.

3.1.4. Queue Behavior and Queue Discipline

Queue behavior refers to customer actions while in a queue waiting for service to begin. In some situations, there is a possibility that incoming customers may balk, renege, or jockey (move from one line to another if they think they have chosen a slow line).

Queue discipline refers to the logical ordering of customers in a queue and determines which customer will be chosen for service when a server becomes free. Common queue disciplines include first-in, first-out (FIFO); last-in, first-out (LIFO); service in random order (SIRO); shortest processing time first (SPT); and service according to the priority (PR) [3].

In our study, we assume different queue behaviors (balk, renege and both), and FIFO queue discipline.

3.1.5. Service Times and the Service Mechanism

The service times of the successive arrivals may be constant or of random duration. The exponential, Weibull, gamma, lognormal and truncated normal distributions have all been used as models of service times in different situations [3].

Sometimes, services may be identically distributed for all customers of a given type of priority, while customers of different types may have completely different service-time distributions. In addition, in some systems, service times depend upon the time of the day or the length of the waiting line [3].

In our study, at Chapter 5, we generate service times from weibull distribution. And at Chapter 6, we generate service times from triangular distribution.

3.2. Queuing Notation

Kendall (1953) proposed a notational system for parallel server systems which has been widely adopted. It is based on the format $A/B/c/N/K$. These letters represent the following system characteristics:

- A represents the interarrival-time distribution.
- B represents the service-time distribution.
- c represents the number of parallel servers.
- N represents the system capacity.
- K represents the size of the calling population.

Table 3.1. Summary of the most common symbols

Characteristic	Symbol	Explanation
Interarrival-time distribution (A) and Service-time distribution (B)	M	Exponential
	D	Deterministic
	E_k	Erlang type k ($k=1, 2, \dots$)
	H_k	Mixture of k exponential
	G	General distribution
Number of parallel servers (c)	1, 2... ∞	
System capacity (N)	1, 2... ∞	
Calling population (K)	1, 2... ∞	
Queue discipline	FIFO	First-in, first-out
	LIFO	Last-in, first-out
	SIRO	Service in random order
	SPT	Shortest processing time first
	PR	Service according to the priority

For example, $M/M/1/\infty/\infty$ indicates a single-server system that has unlimited queue capacity and an infinite population of potential arrivals. The interarrival times and the server times are exponentially distributed [1].

3.3. Long-Run Measures of Performance of Queuing Systems

The primary long-run measures of performance of queuing systems are the long-run time average number of customers in the system (L), and in the queue (L_Q), the long-run average time spent in the system (w), and in the queue (w_Q) per customer, and the server utilization (ρ). The term “system” usually refers to the waiting line plus the service mechanism, but, in general, it can refer to any subsystem of the queuing system; whereas the term “queue” refers to the waiting line alone. Other performance measures of interest include the long-run proportion of customers who are delayed in queue longer than t_0 time units and the long-run proportion of time the waiting line contains more than k_0 customers. This following equations defines the measures of performance for general $G/G/c/N/K$ queuing systems

3.3.1. Time-Average Number in System, L , in the Queue, L_Q

$$\hat{L} = \frac{1}{T} \sum_0^{\infty} iT_i = \frac{1}{T} \int_0^T L(t)dt \rightarrow L \text{ as } T \rightarrow \infty \quad (3.1)$$

where T_i denotes the total time during $[0, T]$ in which the system contained exactly i customers.

$$\hat{L}_Q = \frac{1}{T} \sum_0^{\infty} iT_i^Q = \frac{1}{T} \int_0^T L_Q(t)dt \rightarrow L_Q \text{ as } T \rightarrow \infty \quad (3.2)$$

where T_i^Q denotes the total time during $[0, T]$ in which the system contained exactly i customers waiting in queue.

3.3.2. Average Time Spent in System per Customer, w

$$\hat{w} = \frac{1}{N} \sum_1^N W_i \rightarrow w \text{ as } N \rightarrow \infty \quad (3.3)$$

$$\hat{w}_Q = \frac{1}{N} \sum_1^N W_i^Q \rightarrow w_Q \text{ as } N \rightarrow \infty \quad (3.4)$$

3.3.3. The Conservation Equation: $L = \lambda w$ (Little's Equation)

$$\hat{L} = \hat{\lambda} \hat{w} \rightarrow L = \lambda w \text{ as } T \rightarrow \infty \text{ and } N \rightarrow \infty \quad (3.5)$$

The above equation is called conservation equation and is usually attributed to Little [14]. It says that the average number of customers in the system at an arbitrary point in time is equal to the average number of arrivals per unit time, times the average time spent in the system.

3.3.4. Server Utilization in G/G/1 and G/G/c

$$\rho = \frac{\lambda}{\mu} \quad (3.6)$$

For a single server queue to be stable, the arrival rate λ must be less than the service rate μ ; that is; $\lambda < \mu$ or $\rho = \lambda / \mu < 1$.

If the arrival rate is greater than the service rate ($\lambda > \mu$), the server will eventually get further and further behind. After a time the server will always be busy and the waiting will tend to grow in length at an average rate of $(\lambda - \mu)$ customers per time unit, because the departures will be occurring at rate μ per unit time. The long-run measures of performance are meaningless under such unstable queues.

If there are c identical servers in parallel;

$$\rho = \frac{\lambda}{c\mu} \quad (3.7)$$

3.4. Steady-State Behavior of Infinite Population Markovian Models

The following equations are the steady-state solution of a number of queuing models that can be solved mathematically. The arrivals are assumed to follow a Poisson process with rate λ arrivals per time unit; that is the interarrival times are assumed to be exponentially distributed with mean $1/\lambda$. Service times may be exponentially distributed (M) or arbitrary (G). The queue discipline is FIFO.

A queuing system is said to be in statistical equilibrium, or steady state, if the probability that the system is in a given state is not time dependent; that is;

$$P(L(t)=n)=P_n(t)=P_n$$

is independent of time.

For simple models, the steady-state parameter L , the time average number of customers in the system, can be computed as

$$L = \sum_{n=0}^{\infty} nP_n \quad (3.8)$$

Once L is given, the other steady state parameters can be computed by Little's Equation (3.5) applied to the whole system and the queue alone:

$$w = \frac{L}{\lambda} \quad (3.9)$$

$$w_Q = w - \frac{1}{\lambda} \quad (3.10)$$

$$L_Q = \lambda w_Q \quad (3.11)$$

A necessary and sufficient condition for the G/G/c queues to have a statistical equilibrium is that $\lambda/c\mu < 1$, where c is the number of parallel servers. For these unlimited-capacity, infinite calling population models, it shall be assumed that the theoretical server utilization, $\rho = \lambda/c\mu$ satisfies $\rho < 1$.

3.4.1. Single-Server Queues with Poisson Arrivals and Unlimited Capacity: M/G/1

$$\rho = \frac{\lambda}{\mu}$$

$$L = \rho + \frac{\lambda^2(1/\mu^2 + \sigma^2)}{2(1-\rho)} = \rho + \frac{\rho^2(1 + \sigma^2\mu^2)}{2(1-\rho)} \quad (3.12)$$

$$w = \frac{1}{\mu} + \frac{\lambda(1/\mu^2 + \sigma^2)}{2(1-\rho)} \quad (3.13)$$

$$w_Q = \frac{\lambda(1/\mu^2 + \sigma^2)}{2(1-\rho)} \quad (3.14)$$

$$L_Q = \frac{\lambda^2(1/\mu^2 + \sigma^2)}{2(1-\rho)} = \frac{\rho^2(1 + \sigma^2\mu^2)}{2(1-\rho)} \quad (3.15)$$

$$P_0 = 1 - \rho \quad (3.16)$$

3.4.2. An Approximation for the M/G/c Queue

Approximate formulas for the M/G/c queue can be obtained by applying the correction factor;

$$\frac{1 + (cv)^2}{2} \quad \text{where} \quad cv^2 = \frac{V(X)}{[E(X)]^2} \quad (3.17)$$

to the M/M/c formulas for L_Q and w_Q (no exact formula exist for $1 < c < \infty$). The nearer cv is to 1, the better the approximation. Little's formula can then be used to calculate L and w (3.9 and 3.10). There is no general method for correcting the steady-state probabilities, P_n .

Steady State formulas for the M/M/c Queue are;

$$\rho = \frac{\lambda}{c\mu}$$

$$P_0 = \left\{ \left[\sum_{n=0}^{c-1} \frac{(c\rho)^n}{n!} \right] + \left[(c\rho)^c \left(\frac{1}{c!} \right) \frac{1}{1-\rho} \right] \right\}^{-1} \quad (3.18)$$

$$P(L(\infty) \geq c) = \frac{(c\rho)^c P_0}{c!(1-\rho)} \quad (3.19)$$

$$L = c\rho + \frac{\rho P(L(\infty) \geq c)}{1-\rho} \quad (3.20)$$

$$w = \frac{L}{\mu} \quad (3.21)$$

$$w_Q = w - \frac{1}{\lambda} \quad (3.22)$$

$$L_Q = \frac{\rho P(L(\infty) \geq c)}{1 - \rho} \quad (3.23)$$

$$L - L_Q = \frac{\lambda}{\mu} = c\rho \quad (3.24)$$

So, approximate values of w_Q and L_Q can be found by multiplying Equation 3.22 and 3.23 by correction factor (3.17), respectively. Then w and L can be found through Little's formula in equations 3.9 and 3.10.

4. THE GENERATION OF RANDOM VARIATES

4.1. Random Number Generation

Random numbers are the basic ingredient in the simulation of all stochastic systems. Most computer languages have their own random number generation program inside. Similarly, simulation languages generate random numbers that are used to generate event times and other random variables [3].

The sequence of random numbers $R_1, R_2, \dots$ must have two important statistical properties; uniformity and independence. Each random number R_i is an independent sample drawn from a continuous Uniform Distribution between 0 and 1, $(U(0,1))$. The probability density function (pdf) is given by;

$$f(x) = \begin{cases} 1, & 0 \leq x \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

$$\text{with } E(R) = \int_0^1 x dx = \frac{1}{2} \text{ and } V(R) = \int_0^1 x^2 dx - [E(R)]^2 = \frac{1}{12}$$

Random number generation was developed in the '60s together with the development of the computers. However, computer is not designed to perform random behaviors. Computers perform deterministic calculations and give predictable results. Nevertheless, numerous methods can be used to generate the values. In selecting among these methods, or routines, there are some important considerations.

- The routine should be fast. Simulation may require many hundreds of thousands of random numbers

- The routine should be portable to different computers, and ideally to different programming languages. Then the simulation program produces the same result wherever it is executed.
- The routine should have a sufficiently long cycle. The cycle length, or period, represents the length of random number sequence before previous numbers begin to repeat themselves.
- The random numbers should be replicable. Given the starting point, it should be possible to generate the same set of random numbers. This is helpful for debugging purposes and is a means of facilitating comparisons between systems. Also, it should be possible to easily specify different starting points within the sequence.
- The generated random numbers should closely approximate the ideal statistical properties of uniformity and independence.

Excel also has its own RAND() function to produce random numbers between (0,1). However, the sequence that is produced by RAND() in Excel does not hold some important properties that are discussed above. The routine is very fast, but, the user can not specify the starting point and can not replicate the sequence.

4.1.1. A Technique for Generating Random Numbers

The Linear Congruential Method (LCM) is the most widely used technique for generating random numbers. This method was initially proposed by Lehmer [15]. The method produces a sequence of integers, $X_1, X_2, \dots$ between zero and $m-1$ according to the following recursive equation:

$$X_{i+1} = (a X_i - c) \bmod m, \quad i=0, 1, 2, \dots$$

There are three parameters; a , c and m , where,

Table 4.1. Parameters of LCM

Parameter	Name	Range
M	the modulo	$\{1,2,\dots\}$
A	the multiplier	$\{0,1,\dots,m-1\}$
C	the increment	$\{0,1,\dots,m-1\}$
X_0	the seed	$\{0,1,\dots,m-1\}$

The quality of the generator strongly depends upon the choice of these constants.

Each generator needs to be given an initial value of seed, X_0 , to start with. Initializing the generator with the same seed will give the same sequence of random numbers. If a different sequence is desired, then a different seed should be initialized. It is crucial to use a random number generator that has been thoroughly tested and recommended.

The LCM has the advantage of being very fast, requiring only a few operations per call. Generator has a cycling behavior. It is a cycle of m different states.

A good generator is;

$$m=2^{32}$$

$$a=69069$$

$$c=1$$

There are 2^{32} different states of X_i : $0,1,\dots, 2^{32}-1$.

We use LCM while generating random variate numbers in all spreadsheet implementation in Excel. In worksheets called "data" in all spreadsheets files in this study, the first column, A, is dedicated to generated Random Numbers. At cell A1, the seed number is written as 0 at the beginning of the simulation. And the formulas in column A can be seen at Table 4.2.

Table 4.2. Random number generation in Excel

	A
1	0
2	=MOD(A1*69069+1;2^32)
3	=MOD(A2*69069+1;2^32)
4	=MOD(A3*69069+1;2^32)
5	=MOD(A4*69069+1;2^32)
6	=MOD(A5*69069+1;2^32)

The random numbers are generated in column A, and through a VBA function, called random(), uniform random numbers are generated;

```

Const CON As Double = 2 ^ 32
Function random() As Double
Dim number As Currency
Set data = Sheet1
r=r + 1
number = data.Cells(r + 1, 1)
random = number / CON
End Function

```

This function takes the number in the column A of the “data” worksheet at every call. Then the function returns U(0,1) by dividing the number by 2^{32} . While the simulation is running, the program uses this random() function when needed. In the simulation program at the beginning of every new simulation run the seed cell A1 is updated with the number after the last used random number. The new number will be the seed for the following run.

4.2. Random Variate Generation

“Random Variate Generation” refers to getting an observation of a random variable from a desired distribution, which is specified by fitting any appropriate distributional form to the observed data, e.g. Exponential, Poisson or Gamma.

Generation of Random Variates includes two steps; the first step is generating independent continuous $U(0,1)$. All methods of generating random variates from any distribution require independent identically distributed $U(0,1)$ random variates. Therefore, it is most necessary that a statistically reliable $U(0,1)$ random number generator be available as described in section 4.1. Without an acceptable random number generator, it is impossible to generate random variates correctly from any distribution. The second step is to transform these uniform variates into the desired distribution. There are several alternative algorithms that can be used for this purpose.

4.2.1. Inverse Transform Method

Let X be the continuous random variate that should be generated. X has distribution function F that is continuous and strictly increasing between $(0,1)$. Let F^{-1} denote the inverse of the function F . Then the algorithm for generating a random variate X having distribution function F is as follows:

- Generate $U \sim U(0,1)$.
- Return $X = F^{-1}(U)$.

F^{-1} will always be defined, since $0 \leq U \leq 1$ and the range of F is $[0,1]$.

To prove that the value X returned by the above algorithm has the desired distribution F , it must be shown that for any real number x , $P(X \leq x) = F(x)$. Since F is invertible;

$$P(X \leq x) = P(F^{-1}(U) \leq x) = P(U \leq F(x)) = F(x)$$

The last equality follows since $U \sim U(0,1)$ and $0 \leq F(x) \leq 1$.

4.2.1.1. Generating Interarrival Times from Exponential Distribution. The continuous random variable X is said to be have an exponential distribution with parameter $\lambda, \lambda > 0$, if its pdf is given by

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

The exponential distribution has mean and variance given by

$$E(X) = \frac{1}{\lambda} \quad \text{and} \quad V(X) = \frac{1}{\lambda^2}$$

The cdf is;

$$F(x) = \begin{cases} 1 - e^{-\lambda x}, & x \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

We generate exponential interarrival times by using inverse transform method.

$$F(x) = U, \text{ where } U \sim U(0,1)$$

$$1 - e^{-\lambda x} = U$$

$$e^{-\lambda x} = 1 - U$$

$$-\lambda x = \ln(1 - U)$$

$$x = -\left(\frac{1}{\lambda}\right) \ln(1 - U) \quad (4.1)$$

The equation 4.1 is called random-variate generator for the exponential distribution. We generate arrival times from exponential distribution in the models at Chapter 5. The VBA function, called expo() is used to generate exponential random variates (App C.1).

4.2.1.2. Generating Arrival Times from Nonhomogeneous Poisson Process. Let $N(t)$ represent the total number of events that have occurred up to time t . Homogeneous Poisson process follows that the interarrival times of the events are exponentially distributed with constant arrival rate, λ . The Nonhomogeneous, also called nonstationary, Poisson process is obtained by allowing the arrival rate, $\lambda(t)$ at time t to be a function of time [16].

That is, $N(s+t)-N(s)=n$ is a Poisson random variable with mean $m(s+t)-m(s)$. Since this implies that $N(t)$ is Poisson with mean $m(t)$, we call $m(t)$ the mean value function of the nonhomogeneous Poisson process.

Generating random variates from nonhomogeneous Poisson process over the interval (t_{i-1}, t_i) is to generate exponential random variates with rate λ_i . If we are at $t \in [t_{i-1}, t_i)$ and generate X , an exponential with rate λ_i , which is such that $t + X > t_i$ then we can use

$$\lambda_i [X - (t_i - t)] / \lambda_{i+1}$$

as the next exponential with rate λ_{i+1} .

We generate arrival times from Nonhomogeneous Poisson Process at the model in Chapter 6. The rate is constant and has 4 jumps. The specified t times and corresponding mean arrival times are entered at “data” worksheet (Table 4.3).

Table 4.3. Parameters for nonhomogeneous Poisson arrivals in “data” worksheet

	C	D	E
23	Interarrival means for nonhomogeneous Poisson arrivals		
24	0	180	1,849315068
25	180	240	7,5
26	240	360	2,126582278
27	360	600	4,405594406

Thus, the mean $(= 1 / \lambda_i(t), i=1, 2, 3, 4)$ function follows;

$$\text{Mean}(t) = \begin{cases} 1,849315068 & 0 \leq t < 180 \\ 7,5 & 180 \leq t < 240 \\ 2,126582278 & 240 \leq t < 360 \\ 4,405594406 & 360 \leq t < 600 \end{cases}$$

We use VBA function, called Nonhexpo(), to generate arrival times (App C.2).

4.2.1.3. Generating Service Times from the Weibull Distribution. The continuous random variable X is said to have a weibull distribution, with parameters, α , β and ν , if its pdf is given by

$$f(x) = \begin{cases} \frac{\beta}{\alpha} \left(\frac{x - \nu}{\alpha} \right)^{\beta-1} \exp \left[- \left(\frac{x - \nu}{\alpha} \right)^{\beta} \right], & x \geq \nu \\ 0, & \text{otherwise} \end{cases}$$

The three parameters for the Weibull distribution are ν ($-\infty < \nu < \infty$), which is the location parameter; α ($\alpha > 0$), which is the scale parameter; and β ($\beta > 0$), which is the shape parameter. When $\nu = 0$, the Weibull pdf becomes;

$$f(x) = \begin{cases} \frac{\alpha}{\beta} \left(\frac{x}{\alpha} \right)^{\beta-1} \exp \left[- \left(\frac{x}{\alpha} \right)^{\beta} \right], & x \geq 0 \\ 0, & \text{otherwise} \end{cases} \quad (4.2)$$

The weibull distribution has mean and variance given by

$$E(X) = \nu + \alpha \Gamma \left(\frac{1}{\beta} + 1 \right)$$

$$V(X) = \alpha^2 \left[\Gamma\left(\frac{2}{\beta} + 1\right) - \left[\Gamma\left(\frac{1}{\beta} + 1\right) \right]^2 \right]$$

where $\Gamma(\cdot)$ denotes the gamma function.

The cdf is;

$$F(x) = \begin{cases} 1 - \exp\left[-\left(\frac{x-\nu}{\alpha}\right)^\beta\right] & x \geq \nu \\ 0 & x < \nu \end{cases}$$

We generate weibull service times by using the inverse transform method.

When the location parameter ν is set 0, its pdf is given by equation 4.2, thus the cdf becomes $F(x) = 1 - e^{-(x/\alpha)^\beta}$, $x \geq 0$

$$\begin{aligned} 1 - e^{-(x/\alpha)^\beta} &= U \\ e^{-(x/\alpha)^\beta} &= 1 - U \\ X &= \alpha [-\ln(1 - U)]^{1/\beta} \end{aligned} \tag{4.3}$$

The equation (4.3) is called random-variate generator for the weibull distribution. We generate service times from Weibull distribution in the models of Chapter 5. The VBA function called zweibull(alfa,beta) is used to generate weibull random variates (App C.3).

4.2.1.4. Generating Service Times from the Triangular Distribution. The continuous random variable X is said to be have a triangular distribution, with parameters a , m and b , if its pdf is given by

$$f(x) = \begin{cases} \frac{2(x-a)}{(m-a)(b-a)}, & a \leq x \leq m \\ \frac{2(b-x)}{(b-m)(b-a)}, & m < x \leq b \\ 0, & \text{otherwise} \end{cases}$$

where $a \leq m \leq b$, the mode occurs at $x=m$.

The triangular distribution has mean given by

$$E(X) = \frac{a + m + b}{3}$$

The cdf is;

$$F(X) = \begin{cases} 0, & x \leq a \\ \frac{(x-a)^2}{(m-a)(b-a)}, & a < x \leq m \\ 1 - \frac{(b-x)^2}{(b-m)(b-a)}, & m < x \leq b \\ 1 & x > b \end{cases}$$

We generate triangular service times by using the inverse transform method.

For $a < x \leq m$;

$$\frac{(x-a)^2}{(m-a)(b-a)} = U \quad (4.4)$$

Solving Equation (4.4) For X,

$$X = \begin{cases} a + \sqrt{U(m-a)(b-a)}, & U < \frac{m-a}{b-a} \\ b - \sqrt{(1-U)(b-m)(b-a)}, & \text{otherwise} \end{cases}$$

We generate service times from the triangular distribution in the model in Chapter 6. The VBA function that is used to generate service times is called `Tri_Pay(a,m,b)` (App. C.4). In the model in Chapter 6, in another column, we generate triangular distribution again, but this function includes a parameter “kind”. The parameters a, m and b are determined according to this “kind”. This VBA function is called `Tri(kind)` (App. C.5)

5. IMPLEMENTATION OF QUEUING SIMULATION IN SPREADSHEETS

In this chapter, several spreadsheet simulation implementations of queuing system are presented. The following assumptions are valid for all examples.

- Interarrival times are independent and identically distributed (i.i.d.)
- Service times are i.i.d.
- Interarrival and Service times are independent

The execution of a simulation or program has to ensure that events occur at the correct time and in the correct sequence. The method of achieving this is dependent on the world view adopted. A world view is a way of looking at the system [11]. When using a simulation package or even doing a manual simulation, a modeler adopts a world view or orientation for developing a model. Those most prevalent are the event-scheduling world view, the process interaction world view and the activity-scanning world view [3].

When using the event-scheduling approach, a simulation analyst concentrates on events and their effect on the system state [3]. An event is an instantaneous occurrence that changes the state of the system. An event calendar is created that includes the type of event that occurs and the timing of the event [11]. The state of the system is only changed by events. The event-scheduling approach is suitable for systems which change in a series of events, where the time of the occurrence is random. In a typical queuing model, the events are arrival and departure of the customers.

When using the process-interaction approach, a simulation analyst thinks in terms of processes [3]. The analyst defines the simulation model in terms entities or objects and their life cycle as they flow through the system, demanding resources and queuing to wait for resources. More precisely, a process is the life cycle of one entity (customer, car, order etc.). When an entity enters the system, all necessary variables (e.g. arrival time, service time) are assigned to the entity using variables of past entities. Most commercial

simulation packages adopt this approach. The process interaction approach is easy to understand and is good for queuing type applications, where an entity is moving between activities. Since most discrete-event applications involve queuing, this is an appropriate method. [11]

Both the event-scheduling and the process-interaction approaches use a variable time advance; that is, when all events and the system state changes have occurred at one instant of simulated time, the simulation clock is advanced to the time of the next event in the event list; the event list or calendar is a list of event notices for future events, ordered by time of occurrence. The activity-scanning approach, in contrast, uses a fixed time increments and a rule-based approach to decide whether an activity begins at each point in simulated time [3].

With the activity-scanning approach, the analyst concentrates on the activities of the model and those conditions that allow each activity to begin. At each clock advance, the conditions for each activity are checked, and if the conditions are true, then the corresponding activity begins [3]. Its disadvantage is that it is slower to operate, as all activities have to be scanned and time intervals do not always correspond to an event [11]. A typical application is on inventory models.

Grossman and Ingolfsson mentioned that the activity-scanning approach oversimplifies most queuing systems, unless the simulation time intervals are made very small, which makes the simulation model very slow [13]. There is no example of spreadsheet simulation of queuing system with activity-scanning approach in the literature. We constructed simulation of queuing models with event-scheduling approach and process-interaction approach in Sections 5.1 and 5.2 respectively.

5.1. Event-scheduling Spreadsheet Simulation

There are many spreadsheet implementations of queuing system by using event-scheduling approach in the literature. The applied queuing models are all single queuing

models, i.e. one-server, no bulk or renege property, etc. Below are two examples of this approach to one server and two server queuing systems.

5.1.1. One Server Queuing Systems

This approach was apparently first used for queuing by Winston [17]. Evans and Olson constructed the one server queuing system in an Excel spreadsheet [10]. They assumed that the interarrival and service times are exponentially distributed with means λ_1 and λ_2 . They used the inverse transform method for generating exponential random variates. For the Random Number, R, they used Excel Function RAND(). We redo by using $\text{expo}(\lambda)$ and $\text{zweibull}()$ in VBA functions for generating arrival and service times. The notation is M/G/1. The interarrival times are assumed to be exponentially distributed with mean (λ) , and the service times are assumed to be Weibull(α, β) distributed. Moreover, the start time and the closing time of the system can be set. The mean of interarrival times $(1/\lambda)$, the mean of service times $(1/\mu)$, α & β values and the random numbers are also set in Excel spreadsheet. There are two events in the system; *Arrival* and *Departure*. The spreadsheet implementation can be seen at the Excel file, 5.1.1.xls, on the attached CD. The execution time for $n=1000$ runs is 5 minutes.

5.1.2. Two Server Queuing Systems

We construct a spreadsheet implementation of the two server queuing system, M/G/2. This example is the extended version of the model, M/G/1. There is no example of an event-scheduling spreadsheet implementation of two server queuing simulations in the literature. The number of event types increases to three here; Arrival, Departure from Server 1 and Departure from Server 2. In the spreadsheet implementation, the model M/G/1 is revised by adding three columns. Two columns are added after the columns belong to Server 1. These columns determine the second server's Status (Idle or Busy) and second server's idle time. And the third column is added as Next Departure from server 2. The spreadsheet implementation can be seen at the Excel file, 5.1.2.xls, on the attached CD. The execution time for $n=1000$ runs is 6 minutes.

The maximum number of customers in the system, the average number of customers in the system, the average queue length, the maximum queue length and the idle time of the servers are the performance measures of the simulation. Moreover, the distribution of queue length can be found. And, it is not clear how to compute the waiting time of customers. This is the major limitation of using the event-scheduling approach with spreadsheets, but no problem in process-interaction approach.

In the event-scheduling approach, the changes in the system occur at the events, but the customer experience of queues is obscured. Moreover, analyzing characteristics of the customers (i.e. selecting different kinds of service) in event-driven implementation is not flexible. It increases the event types. It is seen that handling event-scheduling simulation on spreadsheet gets more complex and cumbersome when more events are necessary in the system, even just increasing the server number to two. It is not flexible to generalize this implementation for more complex problems (i.e. adding bulk or renege property of customers). Thus, it is not reasonable to go forward in event-driven spreadsheet implementation of queuing models. Event-scheduling approach is less intuitive than the process-interaction approach.

5.2. Process-Interaction Spreadsheet Simulation

In this approach, each row corresponds to an entity. Process-interaction approach models the logical sequence of events for each customer, using one row of the spreadsheet for each customer. Therefore, the behavior of the customer is easily observed. This approach was apparently first used for queuing by Chase and Aquilano [18]. Process-interaction simulation was first described in the management science literature by Plane for a model of single-server queue [19]. Grossman *et al*, developed prototypes of multi-server process-interaction models, including balking and reneging [20]. Although Evans and Olson state that "...Another serious limitation of spreadsheets for waiting-line models is that it is not possible to include behavior such as balking ..." [10], adding such a behavior is not as difficult as expected. Process-interaction models are particularly suitable for spreadsheet simulation.

5.2.1. Basic Queuing Models

In a process-interaction simulation model, the definitions of the columns are as follows;

Interarrival(I) = Interarrival time between customer I-1 and customer I

ArrTime(I) = Arrival time of customer I

StartService(I) = Start service time of customer I

Service(I) = Service time for customer I

EndTime(I) = End service time of customer I

WaitTime(I) = Waiting time of customer I

Time = The system is closed at "Time", when the arrival time of the customers reaches "Time", simulation ends.

The simulation logic and the flowchart for M/G/1 are illustrated in Figure 5.1.

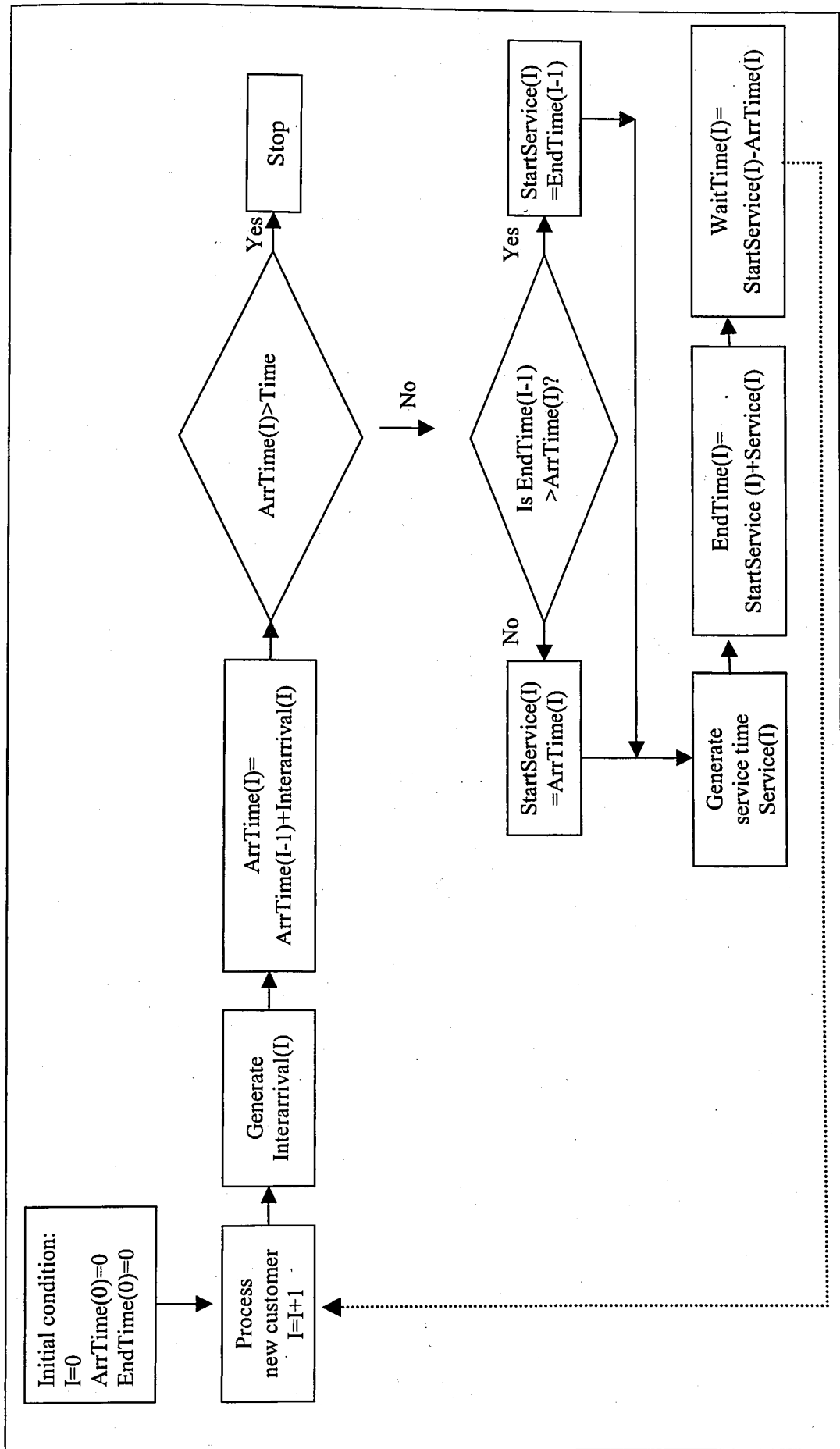


Figure 5.1. The simulation logic and the flowchart for M/G/1

If we develop a process-interaction simulation model on a spreadsheet, we consider the sequence of activities that each customer undergoes;

- 1. Customer arrives.
- 2. Customer waits for service if the server is busy.
- 3. Customer receives service.
- 4. Customer leaves the system.

Putting this sequence of activities into a spreadsheet row for every customer, we arrive at an example spreadsheet implementation of M/G/1 model (Table 5.1).

Table 5.1. Spreadsheet implementation of M/G/1 model

	A	B	C	D	E	F	G	H
1	PROCESS INTERACTION QUEUEING SYSTEM SPREADSHEET, M/G/1							
2								
3	Customer	Interarrival	Arrival	Start	Service	End	Wait	Total
4		Time	Time	Service	Time	Service	Time	Time
5	Number	(min)	(min)	(min)	(min)	(min)	(min)	(min)
6	1	89,4324	89,4324	89,4324	7,2583	96,6907	0,0000	7,2583
7	2	20,7783	110,2107	110,2107	6,2545	116,4652	0,0000	6,2545
8	3	4,0367	114,2473	116,4652	5,1398	121,6050	2,2178	7,3577

The critical point is the interaction of the customers. In this simple model, the interaction only has an influence on the waiting time. To compute the waiting time, we need to know the time that the customer arrived and the time service began; the waiting time is the difference. If the customer arrives and the server is idle, service can begin immediately upon arrival. If the server is busy when the customer arrives, the customer can not begin until the previous customer has completed service. The time that a customer completes service equals the time service begins plus the actual service time.

The generation of new customers is simple. We just sample interarrival times and calculate the arrival of the next customer. Thus we get a sequence of customers, where each customer is represented by one row.

5.2.1.1. The Spreadsheet Implementation of M/G/1 Model. The simulation is initialized by setting the arrival and end service times to zero. As each new customer arrives, interarrival

times are generated. Then the arrival time of the customer is calculated by summing the interarrival time and the previous customer arrival (initially 0). And it is checked whether the time passes the system close time, Time. If it is the case, the simulation ends, else, the simulation goes on. In column D, start service time of the customer is found. Here, we have to consider the interaction with the other customers. If the end service time of the previous customer is greater than the arrival time of the current customer, then the customer has to wait until the previous customer completes service. If not, the customer can start service immediately. The next steps are easy again. It is enough to consider the "current customer". Then in column E, service time is generated. In column F, end service time is calculated, and in column G, waiting time of the customers is found as shown as the difference between start service and arrival time. In column H, total time of the customers in the system is calculated as the difference of end service and the arrival time.

In our example, the required data are entered in the "data" sheet. We assume exponential interarrival times, and weibull service times. The spreadsheet implementation of the M/G/1 model is placed in "Spreadsheet" sheet. In Table 5.2, the spreadsheet formulas can be seen;

Table 5.2. Spreadsheet implementation formulas for M/G/1 model

	A	B	C	D
1	PROCESS INTERACTION QUEUEING SYSTEM SPREADSHEET, M/G/c			
2				
3	Customer	Interarrival	Arrival	Start
4	Number	Time	Time	Service
5		(min)	(min)	(min)
6	1	=expo(Interarrival)	=IF(B6>time;"closed";B6)	=IF(ISTEXT(C6);"";C6)
7	=IF(ISTEXT(C7);"";A6+1)	=IF(A6="";"",expo(Interarrival))	=IF(ISTEXT(C6);"";IF(B7+C6>time;"closed";B7+C6))	=IF(ISTEXT(C7);"";MAX(C7;F6))
8	=IF(ISTEXT(C8);"";A7+1)	=IF(A7="";"",expo(Interarrival))	=IF(ISTEXT(C7);"";IF(B8+C7>time;"closed";B8+C7))	=IF(ISTEXT(C8);"";MAX(C8;F7))
9	=IF(ISTEXT(C9);"";A8+1)	=IF(A8="";"",expo(Interarrival))	=IF(ISTEXT(C8);"";IF(B9+C8>time;"closed";B9+C8))	=IF(ISTEXT(C9);"";MAX(C9;F8))

	E	F	G	H
1				
2				
3	Service	End	Wait	Total
4	Time	Service	Time	Time
5	(min)	(min)	(min)	(min)
6	=zweibull(alfa;beta)	=IF(ISTEXT(C6);"";E6+D6)	=IF(ISTEXT(C6);"";D6-C6)	=IF(ISTEXT(C6);"";F6-C6)
7	=IF(ISTEXT(C7);"";zweibull(alfa;beta))	=IF(ISTEXT(C7);"";E7+D7)	=IF(ISTEXT(C7);"";D7-C7)	=IF(ISTEXT(C7);"";F7-C7)
8	=IF(ISTEXT(C8);"";zweibull(alfa;beta))	=IF(ISTEXT(C8);"";E8+D8)	=IF(ISTEXT(C8);"";D8-C8)	=IF(ISTEXT(C8);"";F8-C8)
9	=IF(ISTEXT(C9);"";zweibull(alfa;beta))	=IF(ISTEXT(C9);"";E9+D9)	=IF(ISTEXT(C9);"";D9-C9)	=IF(ISTEXT(C9);"";F9-C9)

The Interarrival times are generated by using exponential(Interarrival) VBA function in column B. And in column E, the service times are generated by using zweibull(alfa,beta) VBA function. The performance measures are shown in Table 5.3.

Table 5.3. Performance measure formulas for M/G/1 model

	I	J
1		
2		
3	PERFORMANS MEASURES	
4		
5		
6	Number of Customers enter in the System	=MAX(A:A)
7		
8	Max time spent in the system	=MAX(H:H)
9		
10	Average time spent in the system, W	=AVERAGE(H:H)
11		
12	Max waiting time in queue	=MAX(G:G)
13		
14	Average waiting time in queue, Wq	=AVERAGE(G:G)
15		
16	Server Utilization, ρ	=SUM(E:E)/(MAX(F:F,time))
17		

When the arrival mean and service mean are determined as 25 and 20, respectively, and the system is open between 9:00 and 17:00, the 1000 run simulation output follows;

Average time spent in the system, $w = 75,85$

Average waiting time in queue, $w_Q = 56,63$

Servers Utilization, $\rho = 58,85\%$

The steady state values of these variables are as follows;

By Equation 3.13; $w = 260$

By Equation 3.14; $w_Q = 240$

By Equation 3.7; $\rho = 80\%$

The simulated w , w_Q and ρ values are smaller than the steady state values. This is due to the fact that we always start with an empty system.

And the VBA function of Run_Sim can be seen in App C.6. The execution time for $n=1000$ runs is about 3 minutes.

There are several studies on process-interaction spreadsheet simulation. Evans and Olson presented entity-based spreadsheet modeling of queuing systems for one server and two servers [10]. A portion of the M/M/1 queuing system is shown in Table A.1. For generating arrival and service times, they used the inverse transformation method and random numbers were generated by the Excel RAND() function. The formulas are shown in Table A.2. The performance measures are the average waiting time and the server's percentage idle time. The average waiting time in cell G3 is calculated by averaging the sum of column G. And the percent idle time, in cell H3 is found by summing the idle times in column H and dividing by the completion time of the last customer (cell H106). They mentioned that the average number of customers in the queue or in the system is not estimated when using the process-interaction simulation. In 2000, Evans presented single-server queuing model using process-interaction (Table A.3) [6]. He revised his previous model at Table A.1, by adding one column containing the "number in queue". This column shows the number in queue at the arrival of each customer. He mentioned that this could be used to approximate the average number in queue as a function of time. He assumed exponential interarrival times and service times. The formulas are shown in Table A.4. Moreover, he performed approximate time plot of the average number in the queue (Figure B.1). He used "Crystal Ball" to automatically replicate the entire queuing model to show the distribution of the statistics (Table A.5).

The number of columns and the complexity of the formulas used are important factors that affect the simulation speed. Grossman consulted more detailed functions while implementing spreadsheet modeling [1]. The models that he performed look like complex and cumbersome when compared to Evans. But, there is a reason why he presented such a spreadsheet implementation. Because he wanted to compose a standard model in order to generalize the model easily, i.e. adding more servers, or adding balking or reneging property of customers become flexible. He built spreadsheet templates with up to 12 servers, and showed that there is no limitation to the number of servers that can be programmed, i.e. adding more servers is not difficult. Moreover, he provided 1 through 12

servers not only for basic queues, but also for queues showing with balking and reneging properties. These issues will be investigated at the next sections.

The basic model of Grossman includes three sheets. In the first sheet, called “Data”, the data that are needed for the model are provided (Table A.6). There are two random variates tables for the arrival and service times. They are generated according to the predefined probability distributions. By using Excel function, VLOOKUP, the arrival and service times are generated. In the “Simulation” sheet, the spreadsheet implementation is shown. And in the “Computations” sheet, the background calculations take place (Table A.7). The formulas of the “Simulation” sheet and the “Computation” sheet can be seen in Table A.8 and Table A.9, respectively.

Moreover, Grossman and Ingolfsson performed two graphics, the customer graph and the server graph (Figure B.2 and Figure B.3) [13]. Figure B.2 shows each customer’s time periods of waiting and being served for one day, while Figure B.3 shows the performance of the server, i.e. the time periods of being idle or busy. They coded a VBA module, for the customer graph and for the server graph.

5.2.2. More Complicated Queuing Models

Evans and Olson performed two server queuing model through process-interaction approach [10]. The spreadsheet implementation and the formulas are shown in Table A.10. The spreadsheet is quite similar to the single-server model. Column D identifies which of the servers is assigned to the customer. This is selected by using an IF statement, selecting for each server the smallest of the finished times of the prior customer. Another difference from the single server implementation is the identification of the completion times of the servers in columns H and I. Initially, customer 1 is assigned to server 1, and customer 2 is assigned to server 2. The following customer is assigned to the server who finishes serving the previous customer earlier.

Grossman’s implementation of a three server queuing system is shown in Table A.11 [1]. The whole calculations are quite similar to the single server case. The “Data” sheet is

exactly same. In the "Simulation" sheet, the spreadsheet implementation is seen. And in the "Computations" sheet, the background calculations take place. In the "Simulation" sheet, the formulas of the first 4 columns are exactly the same. The other column formulas are shown in Table A.12. In the "Computations" sheet, the formulas of the first 6 columns are exactly the same. The other column formulas are shown in Table A.13. Grossman generalize this model up to 12 servers. As it is seen, adding server increase the column number, which makes the program slow down.

Finally, Hora performs a spreadsheet implementation of the G/G/c queue. (Table A.14) [2]. He uses Excel array functions to create a spreadsheet model with the number of servers being an adjustable parameter. So, there is no need to use multiple templates, where each template works for a different number of servers. The formulas are shown in Table A.15. Hora assumes exponential arrival times, and he uses inverse transformation method in order to generate arrival times in column G. For uniform random numbers, he uses Excel RAND() function in column F. After generating interarrival times in column G, the arrival times are found by summing them in column H. And in column I, the number of customers in the system is found by counting the previous customers' end times which are bigger than the current customer's arriving time. It will give the number of customers that are still in the system. And in column J, the actual start time of the service to the according customers is found. Hore uses array formulas. An array formula can perform multiple calculations and then returns either a single result or multiple results. Array formulas act on two or more sets of values known as array arguments. Each array argument must have the same number of rows and columns. One creates an array formula in the same way that one creates other formulas, except the keys CTRL+SHIFT+ENTER are pressed simultaneously to enter the formula [2].

In J2, the start service time of the first customer is equal to his/her arrival time, since there is no one in the system.

$$J2=H2$$

Hora mentioned that the needed insight is that one must look backward in time (or upward in the table) to determine how many units are currently in the system. Now, suppose that the n th unit arrives at time T_n and that the preceding $n - 1$ units were scheduled to end processing at times $P_1, \dots, P_{n-1}$. The numbers of P_i , which are greater than T_n are counted, using an array formula to find how many units are currently in the system. Call this number x . If this x is less than c , then a server is free and the current unit can seize a server and begin processing immediately. If not, the unit must be scheduled to seize a server at a time dictated by the first-come first-serve queuing discipline. In order to find this time, $x - c$ is computed which gives the rank in ascending order of the x end of processing times greater than T_n . Then, the value of this particular end of processing time is retrieved and placed in the cell which shows the beginning processing time for the current observation. Retrieving the value of the P_i having the rank $x - c$ is accomplished using the PERCENTILE function in Excel. The PERCENTILE function is used in this implementation because it provides a convenient way to find the value in a set of service completion times that has a given rank

The formula in J3 follows;

$$J3 = \text{IF}(I3 \geq \text{Servers}; \text{PERCENTILE}(\text{IF}(\$M\$2:M2 > H3; \$M\$2:M2); (I3 - \text{Servers}) / (I3 - 1)); H3)$$

The outer IF statement checks if all servers are busy. If this is not the case, the current customer (at H3) immediately begins service. If all servers are busy, the PERCENTILE function is used to calculate the time when a server will become available for this customer.

Then the PERCENTILE function will return the value of n values which has the rank k by using $\text{PERCENTILE}(\text{range}, (n - k) / (n - 1))$, where range is $\text{IF}(\$M\$2:M2 > H3; \$M\$2:M2)$ which is evaluated in the array mode and returns the subset of the end of service times later than H3. It is the second time this range is investigated. The first time is in column I in order to find how many customers are in the system at the arrival of the current customer. The second argument determines the rank of the end of service times that will be the

arriving customer's start of service time. For example, suppose that there are $c = 3$ servers and 8 customers in the system, 3 in service and 5 in the queue at the time the next customer arrives. The arriving customer will not begin service until the time when the customer with the end of service time ranked 3rd (largest numerically or latest) among the 8 end of service times [2].

A drawback to the formulation given in J column is that as more rows are added, the formula in each successive row must look back to the first arrival to determine the number of units in the system at the time of arrival and thus the number of end of service times to be considered for the n th arrival is $n-1$. When the number of arrivals simulated becomes large, such as one-thousand, there is a noticeable increase in the recalculation time. One way to work around this problem is to restrict the number of end of service times to be examined. For example, one might determine that it is not necessary to look further back than the last fifty units arriving before the current unit. For the 51st unit, the number of units in the system would be calculated as;

$$\{=IF(I52 \geq \text{servers}, PERCENTILE(IF(M2:M51 > H11, M2:M51), (I52 - \$A\$8) / (I52 - 1)), H52)\}$$

This formula can be copied downward so that all beginning of service times after the 50th are calculated using the end of service times of the most recent 50 arrivals [2].

At column K, the RAND() function is used for generating service times. And in column L, the service times are generated according to the probability table given at matrix \$A\$11:\$C\$14. At column M, the end times of the service is calculated, by summing up start time and service time. In columns N and O, the waiting time and the total time of the customers being in the system is found. In column P, the number in the queue is found at the moment of a new arrival; by subtracting the number of servers from the number of customers in the system in case the subtraction is larger than 0, otherwise 0 is output indicating that there is no queue. And in column Q, the number of busy servers is found by taking the minimum of the number of servers and the number of customers in the system.

Moreover, Hora presents four graphs, showing that “Queue length at customer arrivals”, “Server utilized at customer arrivals”, “total and wait time of customers” and “Empirical State of System Frequencies” (Figures B.4, B.5, B.6 and B.7). For the first three graphs, the x-axis shows the arrival times of the customers. But, the events between these times are missed, i.e. the first graph does not really reflect the motion of the queue length with respect to time. Since the x-axis does not show all the times that the queue length changes. And in figure B.14, the frequency of the number of customers in system is calculated and the related graph is drawn. Again, this does not reflect the real frequency distribution of the number of customers in the system. These statistics must be functions of time rather than the number of customers. The information between two customers can not be captured in this approach.

Although Evans and Olson stated that “...one critical limitation of this process-interaction simulation is that the average number of customers in the queue or in the system is not estimated” [10], we can find the distribution of the number of customers in the system, and in the queue with the process-interaction approach.

5.2.2.1. M/G/c Models with the Distributions of No. Customers in the System and Queue Length. The sequence of activities that each customer undergoes is as follows;

1. Customer arrives.
2. Customer waits for service if all the servers are busy.
3. Customer receives service.
4. Customer leaves the system.

Putting this sequence of activities into a spreadsheet row for every customer, we arrive at an example spreadsheet implementation of M/G/c model (Table 5.4).

Table 5.4. Spreadsheet implementation of M/G/c model

	A	B	C	D	E	F	G	H	I
1	PROCESS INTERACTION QUEUEING SYSTEM SPREADSHEET, M/G/c								
2									
3	Customer	Interarrival	Arrival	Number of	Start	Service	End	Wait	Total
4	Number	Time	Time	customers	Service	Time	Service	Time	Time
5		(min)	(min)	in the system	(min)	(min)	(min)	(min)	(min)
6	1	13,7575	13,7575	0	13,7575	37,8693	51,6268	0,0000	37,8693
7	2	21,1570	34,9144	1	34,9144	30,7251	65,6395	0,0000	30,7251
8	3	39,7325	74,6470	0	74,6470	0,0046	74,6516	0,0000	0,0046

In column D, the number of customers at the arrival time of the corresponding customer is found by counting the end service times that are greater than the arrival time of the corresponding customer. Here, again we have to consider the interaction with the other customers. If the number of customers in the system is greater than the server number, then there exists a queue. And in column E, the start time of the service is found. To compute the start service time, we need to know whether there are any customers waiting in queue at the moment. If the customer arrives and there is no one in the system, service can begin immediately upon arrival. If there are more customers in the system than the server number (all servers are busy), the customer enter the queue and the service can begin when any server becomes idle.

We revised Hora's model with the assumption of exponential interarrival times, and weibull service times. Moreover, as an alternative to the usage of the PERCENTILE Excel function at the start service column (Column E), we use the SMALL Excel function.

In the "data" sheet, the required data are entered. In D5, the number of servers is seen (Table A.16). The spreadsheet implementation is shown in Table A.17.

In column A, the customer numbers are shown. And, in column B and C, the interarrival and arrival times are generated from exponential distribution.

Column D:

D6=IF(ISTEXT(C6);"";0)

D7=IF(ISTEXT(C7);"";COUNTIF(\$G\$6:G6;">"&C7))

In column D, number of customers in the system is found, by counting the end service times that are greater than the arrival time of the corresponding customer.

Column E:

$E6=IF(ISTEXT(C6);"";C6)$

$E7=IF(ISTEXT(C7);"";IF(D7 \geq Servers;SMALL(IF(\$G\$6:G6 > C7; \$G\$6:G6); (D7-Servers+1));C7))$

And in column E, the start time of the service is found. Here, we use the SMALL function instead of the PERCENTILE function.

At Hora's model, c must be greater than 1, since $c=1$ causes the denominator to be zero inside the PERCENTILE function $(I3-Servers)/(I3-1)$. By using the SMALL Excel function, it does not matter what the Server number is, moreover, the formula simplifies when compared to the formula using the PERCENTILE function.

The SMALL function includes two parameters; i.e. $SMALL(array,k)$. The first part is the array $IF(\$G\$6:G6 > C7; \$G\$6:G6)$, that refers to the array greater than the arrival time of the current customer. The second part, k , is the position in the array. The function returns the k^{th} smallest value in the data set, where k is represented by $(D7-Servers+1)$ here, e.g. suppose there are 2 servers. If there are 3 customers in the system when customer i enters, then customer i can start being served after these three customers leave the system. This time refers to the second smallest time in the array of the end time of these three customers $(D7-Servers+1=2)$. Because at the smallest time in the array, the third customer (1^{st} in the queue) will start being service.

Column F:

$F6=zweibull(alfa;beta)$

$F7=IF(ISTEXT(C7);"";zweibull(alfa;beta))$

Service times are generated from weibull(α , β) in column F.

Columns G, H and I:

G6=IF(ISTEXT(C6);"";F6+E6)

H6=IF(ISTEXT(C6);"";E6-C6)

I6=IF(ISTEXT(C6);"";G6-C6)

At columns G, H and I, end service time, waiting time and total time spent in the system are calculated.

In columns J & K, performance measures are calculated as in Table 5.5.

Table 5.5. Performance measure formulas for M/G/c model

	J	K
2		
3	PERFORMANS MEASURES	
4		
5		
6	Number of Customers enter in the System	=MAX(A:A)
7		
8	Max no. of Customers in the System	=MAX(T:T)
9		
10	Average no. of Customers in the System, L	=SUM(U:U)/SUM(S:S)
11		
12	Max no. of Customers in the Queue	=MAX(V:V)
13		
14	Average no. of Customers in the Queue, Lq	=SUM(W:W)/SUM(S:S)
15		
16	Max time spent in the system	=MAX(I:I)
17		
18	Average time spent in the system, W	=AVERAGE(I:I)
19		
20	Max waiting time in queue	=MAX(H:H)
21		
22	Average waiting time in queue, Wq	=AVERAGE(H:H)
23		
24	Server Utilization, p	=SUM(F:F)/(Servers*(MAX(G:G,time)))
25		

The columns, Q to W, include the formulas to find the distribution of number of the customers in the system and in the queue (A numerical view can be seen in Table A.18).

Column Q:

Q6=IF(ISNUMBER(C6);C6;INDEX(G\$6:G\$1000;ROW()-rowno-5;1))

In column Q, the arrival times and the end service times are listed. The outer IF function first lists the arrival times row by row. The IF function returns the first part (=C6) until the arriving times in column C terminates. Whole arrival times are written, then the second part of the IF function starts working. The INDEX(array,row_num,column_num) Excel function returns the value from G\$6:G\$1000, selected by the row and column number indexes. The row_num selects the row in array G\$6:G\$1000 from which to return a value (ROW()-rowno-5), and column_num selects the column in array from which to return a value (1).

- ROW(reference) Excel function return the row number of a reference. If the reference is omitted, the function returns the reference of the cell in which the ROW function appears. "rowno" is the number of arrivals. It was calculated at cell R1. (R1= =COUNT(C:C)). And since the calculations of the spreadsheet start after row 5, "ROW()-rowno-5" is the true row number at which the end of service time starts. The column number is "1" because array contains only one column. It can be also omitted since there is only one column.

Column R:

The R column contains the sorted of arrival times together with the end service times. These are the event times. When we write them in single column and sort them, we get the exact information about the number of customers in the system. The formula at column R follows;

$$R6=IF(Q6="";"";SMALL(Q:Q;ROW()-5))$$

The SMALL function requires two parameters. The first part is the array (Q:Q), that refers to the event times. The second part is represented by row()-5. The SMALL function returns the (ROW()-5) smallest value in Q:Q.

Thus, the event times at column Q are sorted in column R.

Column S:

In column S, the time periods between events are written.

$S6=IF(Q6="";";R6)$

$S7=IF(Q7="";";R7-R6)$

Column T:

And in column T, the number of customers during that period is started.

$T6=IF(Q6="";";0)$

$T7=IF(Q7="";";1)$

$T8=IF(Q8="";";IF(ISNA(MATCH(R8;C:C;0));T7-1;T7+1))$

Cells T6 and T7 are set to 0 and 1 respectively. It is certain that there is no one in the system until the first event, the first event is an arrival, and until the second event, there is one customer in the system. The function at cell T8 follows; the second IF function checks whether the number in column R is an arrival or an end of service time. The MATCH (lookup_value,lookup_array,match_type) Excel function returns the relative position of the lookup_value in the lookup_array that matches a specified value in a specified order. Here, the MATCH function searches column R in column C, where column C includes arrival times. Match_type 0 indicates exact match. If the MATCH function fails to find a match, it returns the #N/A (value not available) error value, which means, in our example, the value in the corresponding column R is not an arrival time, it is an end of service time. The ISNA function checks this #N/A error. If $ISNA(MATCH(R8;C:C;0))=TRUE$, then the number in the above cell (T7) is decreased by one, else the number in cell T7 is increased by one, indicating an arrival.

Column U:

In column U the weighted no. of customers in the system is calculated by multiplying the number in the system (T column) with time period (S column).

$U6=IF(T6="";";S6*T6)$

Column V:

In column V, the queue length is calculated by taking the max of 0 and the number of customers in the system minus the number of servers.

V6=IF(T6="";"";MAX(0;T6-Servers))

Column W:

And finally, in column W, the weighted queue length is calculated by multiplying the queue length by the time period.

W6=IF(T6="";"";S6*V6)

In columns J, K and L, the distribution of the number of customers in the system are summarized (Table A.19). The formulas can be seen in Tables 5.6 and 5.7.

Table 5.6. Probability distribution of no. of customers in the system for M/G/c model

	J	K	L
26			
27	Probability distribution of number of customers in the system		
28	Number in the system	Total Time	probability
29	0	=IF(J29="","",SUMIF(T:T,J29;S:S))	=IF(J29="","",K29/SUM(S:S))
30	=IF(J29>=\$K\$8;"",J29+1)	=IF(J30="","",SUMIF(T:T,J30;S:S))	=IF(J30="","",K30/SUM(S:S))
31	=IF(J30>=\$K\$8;"",J30+1)	=IF(J31="","",SUMIF(T:T,J31;S:S))	=IF(J31="","",K31/SUM(S:S))

In column J, number of customers from 0 to max number of customers in that run (\$K\$8) are listed. In column K, these numbers are searched from the T column, and summed with the values of the corresponding time period in column R. Finally in column L, the calculated values in column K are divided by the total time period (SUM(S:S)) in order to find the probability. The same is done in columns M, N and O for the distribution of the queue length. These formulas can be seen in Table 5.7.

Table 5.7. Probability distribution of no. of customers in the queue for M/G/c model

26	M	N	O
27	Probability distribution of queue length		
28	Number in queue	Total Time	probability
29	0	=IF(M29="","",SUMIF(V:V,M29;S:S))	=IF(M29="","",N29/SUM(S:S))
30	=IF(M29>=\$K\$12;"",M29+1)	=IF(M30="","",SUMIF(V:V,M30;S:S))	=IF(M30="","",N30/SUM(S:S))
31	=IF(M30>=\$K\$12;"",M30+1)	=IF(M31="","",SUMIF(V:V,M31;S:S))	=IF(M31="","",N31/SUM(S:S))

After the simulation run, we have n outputs of each performance measures. They are stored in the “Simulation” Excel worksheet (Table A.20). And the summary statistics are combined in the “Simulation_Summary” Excel worksheet (Table A.21). The probability distributions of the waiting time in the system, w and in the queue, w_Q and the server utilization are also found in this sheet. The related graphs are given in Figures B.8, B.9 and B.10, respectively. Also, the data for finding the probability distributions of customer number in the system and queue length are collected at sheets “Simulation_cust” and “Simulation_queue” respectively (Tables 5.22 and 5.23). And the probability graphs are shown in Figures B.11 and B.12, respectively.

When the arrival mean and service mean are determined like in Table A.16, the 1000 run simulation output follows;

Average number of customers in queue, L_Q= 0,2172

Average waiting time in queue, w_Q =6,1130

Servers Utilization, ρ = 32,8874%

The steady state values of these variables are as follows;

By Equation 3.23 multiplied by Equation 3.17; L_Q = 0,457143

By Equation 3.22 multiplied by Equation 3.17; w_Q = 11,42857

By Equation 3.7; ρ = 40%

The simulated results of L_Q, w_Q and ρ are smaller than the steady state values, since we always start with an empty system.

The VBA function of Run_Sim can be seen in App C.7. The Run_Sim program has to be updated when applied on a new simulation model. The execution time for n=1000 runs is about 10 minutes.

5.2.2.2. M/G/c Models with Balking Property.

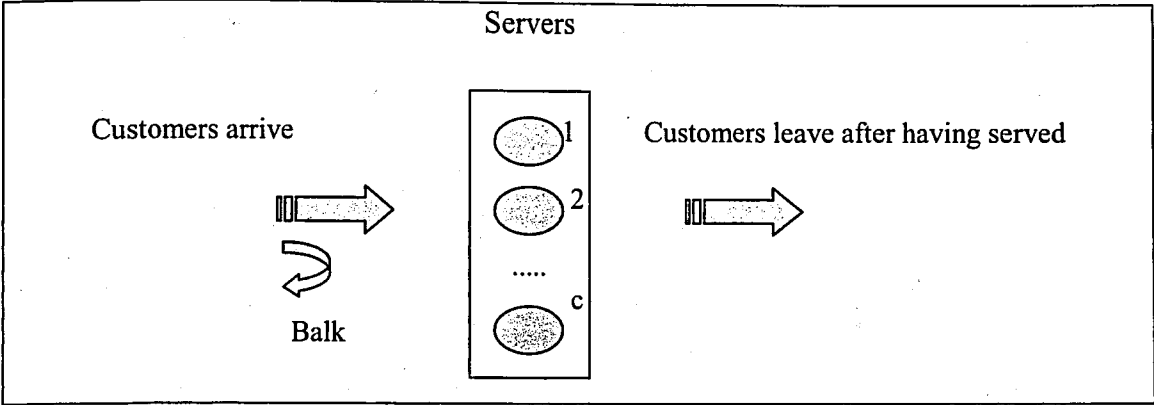


Figure 5.2. Queuing system with balking property

The sequence of activities that each customer undergoes is as follows;

1. Customer arrives.
2. If the queue length is greater than or equal to the balk number, customer leaves the system without being served.
3. Customer waits for service if all the servers are busy.
4. Customer receives service.
5. Customer leaves the system.

This model includes the possibility of balking, i.e. customers enter the system, if they see that there are more than “n” customers waiting at the queue, they leave the system without being served. Two more columns are added to the previous model. (Table A.24).

Column E:

E7 =IF(ISTEXT(C7);"";MAX(0;D7-Servers))

Column E finds the number of customers in the queue, just by taking the maximum of column D (no. of customers in the system at the moment) minus Server number and 0.

Column F:

$F7=IF(E7="";"";IF(E7\geq \text{Balk};\text{"Balk"};""))$

And column F investigates whether the current customer decides to enter the queue, or leave the system because it is too long, i.e. the queue length is larger than the “Balk” number. The “Balk” number is entered at the “data” worksheet at cell D6.

Column G:

$G7=IF(OR(ISTEXT(C7);F7=\text{"Balk"});"";IF(D7\geq \text{Servers};SMALL(IF(\$I\$6:I6>C7; \$I\$6:I6);(D7-\text{Servers}+1));C7))$

In column G, the start times of service are found. This time one extra condition ($F7=\text{Balk}$ or not) is added to the first part of the IF condition.

Since the customer will leave the system without entering the queue, we do not need to compute the start time of his/her service.

Moreover, there is one performance measures added, the percentage of customers who balk. The cell M8 includes;

$M8=COUNTIF(F:F;\text{"Balk"})/M6$

where M6 indicates the number of arriving customers ($M6=MAX(A:A)$).

The execution time for $n=1000$ runs is about 12 minutes.

5.2.2.3. M/G/c Models with Balking and Reneging Properties.

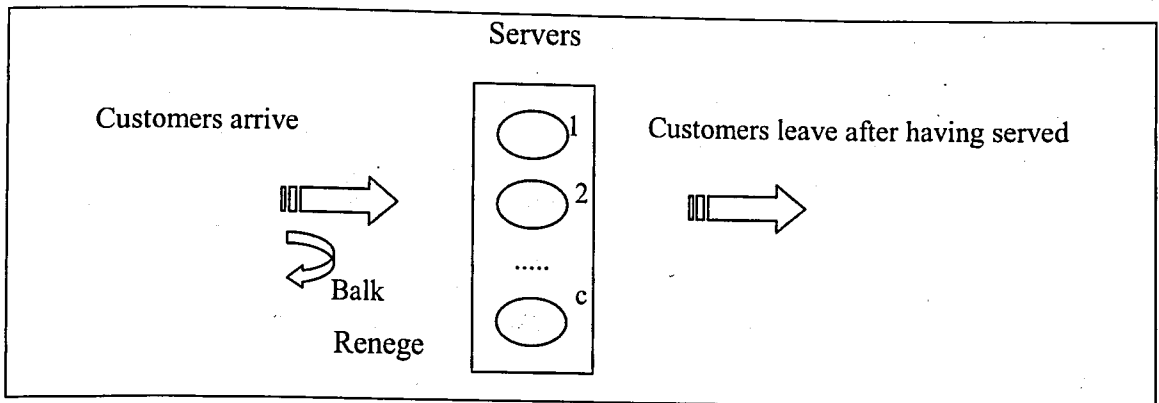


Figure 5.3. Queuing system with balking and reneging properties

The sequence of activities that each customer undergoes is as follows;

1. Customer arrives.
2. If the queue length is greater than or equal to the balk number, customer leaves the system without being served.
3. Customer waits for service, if waiting time is greater than Renege time, customer leaves the system without being served.
4. Customer receives service.
5. Customer leaves the system.

This model includes balking and reneging of customers together. i.e. customers enter the system, if they see that there are more than “n” customers waiting at the queue, they leave the system immediately. Moreover, customers enter the system, enter the queue, anyway, if they wait more than “Renege” time in the queue, they get bored about waiting, and they leave the service without being served. One more column is added at the previous model. (Table A.25).

Column G:

$G7 = \text{IF}(E7 = ""; "", \text{IF}(F7 = \text{"Balk"}; "", \text{IF}(H7 - C7 \geq \text{Renege}; \text{"Renege"}; "")))$

Column G investigates whether the waiting time of the current customer is greater than “Renege” time or not. The “Renege” number is entered at the “data” worksheet at cell D7 as minutes. The first IF function inside checks whether the corresponding customer

shows balked. If he/she balked, then the formula ignores that customer. And the innermost IF function checks whether the waiting time of the corresponding customer is greater than “Reneged” time or not. If it is greater than the “Reneged” time, the corresponding customer is ignored from the rest of the calculations.

Although the customer will leave the system without entering the queue, we need to compute the start time of his/her service in order to find his/her potential waiting time until his/her service begins. Therefore, in column H, the potential start time of that customer is computed. Since he/she will leave the system, no service time is generated for him/her in column I.

Thus, there is another performance measures investigated, the percentage of customers who renege. The cell N10 includes

$$N10=COUNTIF(G:G;"Reneged")/N6$$

where N6 indicates the number of arriving customers ($N6=MAX(A:A)$).

The execution time for n=1000 runs is about 15 minutes.

5.2.2.4. Simple Network Queuing Models with M/G/c Queue at Serial Servers with Balking and Reneging Properties.

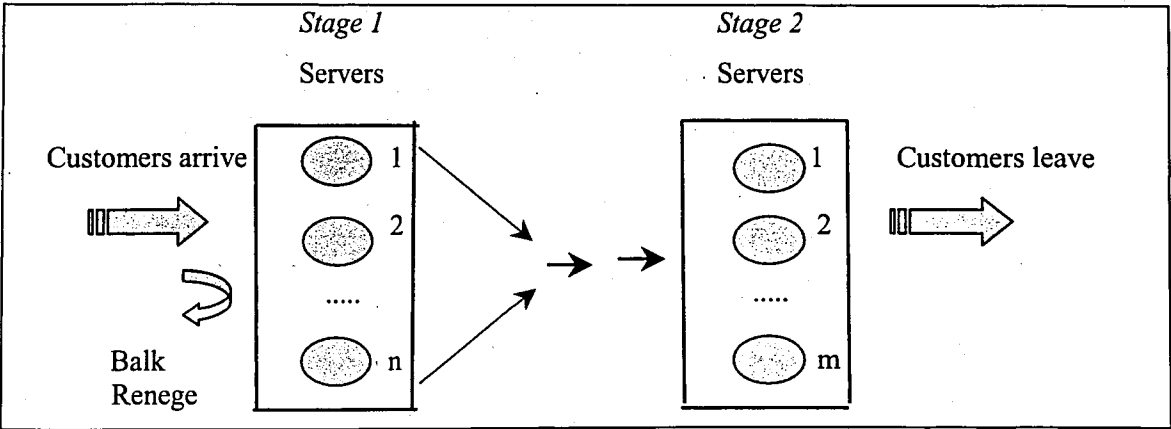


Figure 5.5. A simple network queuing system with balking and reneging properties

In this model, there are two stages that the customers have to be served. Customers have to continue to stage 2, in order to complete their work. There are n servers at stage 1 and m servers at stage 2. The customers show Balk and Renege properties at the first stage. Once, they have received service at the first stage, they continue to second stage without balking and renegeing. The spreadsheet implementation of the first stage is exactly same as the model in section 5.2.2.3. The rest can be seen at Table A.26.

The sequence of activities that each customer undergoes is as follows;

1. Customer arrives at stage 1.
2. If the queue length is greater than or equal to the balk number, customer leaves the system without being served.
3. Customer waits for service, if waiting time is greater than Renege time, customer leaves the system without being served.
4. Customer receives service.
5. Customer leaves the stage 1 and enters stage 2.
6. Customer waits for service if all the servers are busy.
7. Customer receives service.
8. Customer leaves the system.

Until step 4, the sequence is same as in model M/G/c. Putting this sequence of activities after step 4 into the spreadsheet, we arrive the following spreadsheet implementation of the model (Table 5.8).

Table 5.8. Spreadsheet implementation of the simple network queue model

	J	K	L	M	N	O	P	Q
1								
2								
3	End	Customer	Arrival Time	Which	Number of	Start	Service	End
4	Service	Number	at the second stage	customer?	customers	Service	Time	Service
5	(min)				in the system	(min)	(min)	(min)
6	20,15960804	1	20,15960804	1	0	20,15960804	106,1281724	126,2877805
7	24,13816296	2	24,13816296	2	1	126,2877805	5,739485216	132,0272657
8	61,79316483	3	61,79316483	3	2	132,0272657	73,901363	205,9286287
9	73,58163367	4	73,58163367	4	3	205,9286287	0,383324277	206,3119529
10	86,05809332	5	86,05809332	5	4	206,3119529	1,542907978	207,8548609
11	193,6967994	6	193,6967994	6	3	207,8548609	0,028430216	207,8832911
12	205,0126252	7	205,0126252	7	4	207,8832911	1,254572992	209,1378641
13	208,2952586	8	208,2952586	8	1	209,1378641	41,36009773	250,4979619
14	218,5995966	9	218,5995966	9	1	250,4979619	3,639146792	254,1371086
15	221,6045465	10	221,6045465	10	2	254,1371086	0,844077334	254,981186
16	334,6698895	11	334,6698895	11	0	334,6698895	7,218446571	341,888336
17			585,3936454	13	0	585,3936454	4,614262414	590,0079078
18	585,3936454	13						

In column J, the customers' end service times at the first stage are seen. The critical point is to determine the arrival sequence at stage 2. In column L, the end service times are sorted in order to determine the arrival sequence.

Column J:

$$J6=IF(I6="";"";I6+H6)$$

At column J, the end service times of the first stage is performed.

Column K:

$$K6=IF(J6="";"";A6)$$

At column K, the customers who will enter the second stage, after balking and reneging, are listed.

Column L:

$$L6=IF(ISERR(SMALL((J:J);ROW()-5));"";SMALL((J:J); ROW()-5))$$

And at column L, the arrival times for at the second stage are calculated, by sorting the end service times of stage 1. The first part of SMALL function is the array (J:J), that

refers to the end service times at stage 1. The second part is represented by the row number of the function that is placed minus 5. The SMALL function returns the k^{th} smallest value in J:J, where k is 1 for the sixth row and increases by one as the row number increases. The first part of the IF functions include ISERR function. This function returns “TRUE” if the value returns inside the parenthesis returns error. The formula first checks whether SMALL function returns error. If error is returned, then cell L6 returns empty cell, else the next smaller value is returned. We have to include this error check, because, if there are no values left in column, the SMALL function begins to give #NUM! error. This error check prevents this kind of error. The sorted column L now represents the arrival times at the second stage.

Column M:

$M6=VLOOKUP(L6;J:K;2;0)$

And in column M, the sequence of customers is found by using the VLOOKUP Excel function.

The VLOOKUP (lookup_value, table_array, col_index_num, range_lookup) Excel function has four parameters. The Lookup_value is the value to be found in the first column of the array. Table_array is the table of information in which the data are looked up. Col_index_num is the column number in table_array from which the matching value must be returned. And Range_lookup is a logical value that specifies whether we want VLOOKUP to find an exact match or an approximate match. If TRUE is 1 or omitted, an approximate match is returned. In other words, if an exact match is not found, the next largest value that is less than lookup_value is returned. If FALSE (0), VLOOKUP will find an exact match. If one is not found, the error value #N/A is returned.

This function searches the value in L column inside J and K columns (the end times from the first stage and corresponding customers). When the value is matched some value in the J column, the formula returns the customers number, written in the K column (2nd column).

Columns N, O, P and Q:

$N6=IF(M6="";"";0)$

$N7=IF(M7="";"";COUNTIF(\$Q\$6:Q6;">"\&L7))$

Column N lists the customer numbers in the second stage.

$O6=IF(M6="";"";L6)$

$O7=\{IF(M7="";"";IF(N7\geq Servers_2;SMALL(IF(\$Q\$6:Q6>L7;\$Q\$6:Q6);(N7-Servers_2+1));L7))\}$

Column O calculates the start service times of the corresponding customer.

$P6=IF(O6="";"";zweibull(alfa;beta))$

$Q6=IF(M6="";"";O6+P6)$

Columns P and Q contain the generated service times and end service times respectively.

Column R and S:

In columns R and S, the waiting times and total time of customers in the second stage are given.

$R6=IF(M6="";"";O6-L6)$

$S6=IF(M6="";"";Q6-L6)$

Columns T and U:

In columns T and U, these spent times at stage 2 are ordered in order to match to the sequence in the beginning. This is done by VLOOKUP Excel function.

$T6=VLOOKUP(K6;M:R;6;0)$

Since it was found in which sequence customer enter the second stage in customer sequence which is in column J. In column M, now it is easy to assign the waiting times to the corresponding customer sequence which is in column K. The VLOOKUP function searches the K column in M:R columns, when it matches the number in column M, the waiting time in column R is returned.

$U6=VLOOKUP(K6;M:S;7;0)$

The same work is done in column U. Customers total time spent in the second stage is calculated in this column.

Columns V and W:

$V6=IF(OR(ISTEXT(H6);G6="Renega");"";H6-C6+T6)$

$W6=IF(OR(ISTEXT(H6);G6="Renega");"";J6-C6+U6)$

Let V and W columns, the waiting time and total times at both stages are found by summing the waiting time and total time at stages of stage 1 and stage 2.

The table with the performance measures is given in Table 5.9.

Table 5.9. Simple network queuing system with balking and renegeing properties,
performance measure formulas

	X	Y
3	PERFORMANS MEASURES	
4		
5		
6	Number of Customers enter in the System	=MAX(A:A)
7		
8	Percentage of customers who show Balk property	=COUNTIF(F:F;"Balk")/Y6
9		
10	Percentage of customers who show renege property	=COUNTIF(G:G;"Renege")/Y6
11		
12	STAGE 1	
13	Max no. of Customers in the System	=MAX(AO:AO)
14		
15	Average no. of Customers in the System, L	=SUM(AP:AP)/SUM(AN:AN)
16		
17	Max no. of Customers in the Queue	=MAX(AQ:AQ)
18		
19	Average no. of Customers in the Queue, Lq	=SUM(AR:AR)/SUM(AN:AN)
20		
21	Servers Utilization at stage 1, ρ	=SUM(I:I)/(Servers_1*(MAX(J:J;time)))
22		
23	STAGE 2	
24	Max no. of Customers in the System	=MAX(AW:AW)
25		
26	Average no. of Customers in the System, L	=SUM(AX:AX)/SUM(AV:AV)
27		
28	Max no. of Customers in the Queue	=MAX(AY:AY)
29		
30	Average no. of Customers in the Queue, Lq	=SUM(AZ:AZ)/SUM(AV:AV)
31		
32	Servers Utilization at stage 2, ρ	=SUM(P:P)/(Servers_2*(MAX(Q:Q;time)))
33		
34	TOTAL STAGES	
35		
36	Max time spent in the system	=MAX(W:W)
37		
38	Average time spent in the system, W	=AVERAGE(W:W)
39		
40	Max waiting time in queue	=MAX(V:V)
41		
42	Average waiting time in queue, Wq	=AVERAGE(V:V)

In the columns from AL to AR, the necessary calculations needed to find the probability distribution of number of customers in the system and in the queue at first stage are placed, while columns AT to AZ, the calculations needed to find the probability

distributions of number of customers in the system and in the queue at second stage are performed by the method described in section 5.2.2.1.

The execution time for n=1000 runs is about 20 minutes.

5.2.2.5. Simple Network Queuing Models with M/G/c Queue at Serial Servers with Balking and Reneging Properties and Servers Break Time. This model introduces break times of servers. Servers have break times at specified hours. If the customers' service start time is in break hours, they have to wait until the server returns to his job. The break hours are entered in the "data" sheet as in table A.27, a part of the spreadsheet implementation is shown in table A.28. A column is added next to column H (start time of the service) called Actual Start Time (Table 5.10).

Table 5.10. Spreadsheet implementation of servers' break time

	H	I
1		
2		
3	Start	Actual
4	Service	Start Service
5	(min)	(min)
6	105,5991734	105,5991734
7	105,5991734	105,5991734
8	145,6683832	145,6683832
9	181,4120367	240

The formulas in the column H, I, J and K follow;

```
H7=IF(OR(ISTEXT(C7);F7="Balk");"";IF(D7>=Servers_1;  
SMALL(IF($K$6:K6>C7;$K$6:K6);(D7-Servers_1+1));C7))  
I7=IF(AND(H7>=Begin;H7<=End);IF(D7>=Servers_1;MAX(H7;End);End);H7)  
J7=IF(OR(ISTEXT(H7);G7="Reneg");"";zweibull(alfa;beta))  
And  
K7=IF(J7="";"";J7+H7)
```

In column H, the customers start service times are calculated as before. In column I, this start time is checked whether it is between the break hours or not. If the start time of the service, calculated in column H, is in the break time, and if there are more customers than number of servers in the system, the start time is set to maximum of end break time and H7. We take the maximum, because the service of any customer may continue during break time, and can finish after break time. In that case, the server can not have a break hour. If there are fewer customers than the server number in the system, the actual start time is set to end break time. If the start time of the service in column H is not in break hours, then the actual start time is set to the time in column H. In column J, the service times are generated for the customers except the ones who balk or renege. And in column K, the end times of the service are found, by adding actual start times and service times. The rest of the calculations are the same as in the previous example.

The execution time for $n=1000$ runs is about 25 minutes.

5.2.2.6. M/G/c Models with Changing Service Times in case of Long Queue Length.

Suppose the service times is changing according to the number of customers in the queue, i.e. the servers start to serve quicker when they see that there are “ n ” customers in the system, or the service times are proportional to the number of customers in the system.

We insert a column next to “Start Service” column (E), and call it “Number of customers in the queue at start service time (Table 5.11).

Table 5.11. M/G/c model with changing service time

	E	F	G	H
1	SHEET, M/G/c			
2				
3	Start	Number of	Service	End
4	Service	cust. in the system	Time	Service
5	(min)	at start service time	(min)	(min)
6	6,2727	0	25,2588	31,5315
7	66,6352	0	4,2033	70,8385
8	79,3335	0	27,9244	107,2579
9	93,2459	1	16,9939	110,2397
10	120,6553	0	0,7874	121,4427
11	138,5548	0	4,8793	143,4342
12	140,7025	1	11,9945	152,6970
13	174,2896	0	0,2601	174,5496
14	189,8280	0	7,6255	197,4535
15	229,9191	0	0,2259	230,1450
16	289,4565	0	6,6979	296,1544
17	294,7390	1	1,7276	296,4667
18	317,2660	0	5,0994	322,3654
19	377,2538	0	7,5431	384,7970
20	381,4228	1	1,9356	383,3584
21	413,6293	0	30,9284	444,5577
22	419,9814	1	75,7372	495,7186
	data Spreadsheet Simulation Simula			

And the formula at column F follows;

Column F:

F6=0

F7=IF(ISTEXT(E7);"";COUNTIF(C8:C\$1000;"<"&E7))

The function COUNTIF(C8:C\$1000;"<"&E7)) counts how many customers have come up to the time corresponding customer's serving start, i.e. the number of people waiting behind the current customer.

In column G, the service times were generated through VBA function zweibull(alfa;beta). Now, this VBA function includes another parameter, which is the number of customers waiting in the system at start service time of the corresponding customer.

Anyway, we must keep in mind that it is not possible to change the generated service time, in the middle of the service. Once the service time is generated at the moment in time

in column E, even the queue length decreases, the service time generated at the time in column E is taken into account. This is the basic limitation in Process-interaction approach. The spreadsheet implementation is not so dynamic.

Column G:

G6=zweibull(alfa;beta;F6)

G7=IF(ISTEXT(C7);"";zweibull(alfa;beta;F7))

The function becomes;

Function zweibull(alfa, beta As Double, no As Integer) As Double

If no <> 0 Then

zweibull = alfa * (-1 * Log(1 - random())) ^ (1 / beta) / no

Else

zweibull = alfa * (-1 * Log(1 - random())) ^ (1 / beta)

End If

End Function

The third parameter is called "no". The function first control whether this number is zero, or not. If the "no" is zero, then a weibull variable is generated. If "no" indicates a number greater than 0, then the generated weibull variable is divided by this number.

The service time that will be generated can be a function of number of the current customers in the queue.

The execution time for n=1000 runs is about 12 minutes.

6. AN ACTUAL APPLIATION OF SPREADSHEET SIMULATION OF QUEUING SYSTEMS

There is a customer service department at the Central Bank of the Republic of Turkey, called "Laborer Foreign Exchange Department". The customers enter the Bank, get service and depart from the Bank which can be formulated as a queuing system. Especially in the summer months Turkish Workers working abroad come to Turkey and invest their money to the Central Bank. About 3500-4000 workers come to the Bank to invest their money, or to make different transactions in their accounts in one summer month. There are basically four kind of transactions called; Open an account (A), Express transaction desk (B), Payment & Other Operations (C) and Inheritance Operations (D). There are two floors in the department reserved for these transactions. And the transactions are divided into these two floors. Transactions (A) and (B) are processed in the first floor. The other transactions (C) and (D) are processed in the second floor. After the transactions (A) and (C), the customers have to go to the "pay-office" and the "pay-office" is also in the first floor. There are 4 clerks at the first floor, 6 clerks in the second floor and 2 clerks at the "pay-office".

The Bank is opened at 09:00 and closed at 16:30. The customers arriving after 16:30 are not allowed to enter the system, but if there are already customers in the system, then their service are completed even after 16:30. The Bank is also open during the Lunch Break (12:00-13:00), but the server number is restricted during these hours. The clerks have Lunch Break for a half hour and periodically. We learnt from the workers that the customers generally do not show Balk or Renege behavior, because they do not want to leave the Bank until their process is completed. The customers enter the Bank and first they go to the ticket-machine, where they take their number. They push the related button about what they want to process. The machine gives out a short print-out with a number. The customers are divided into two groups according to their transaction. Customers having transactions A and B go to the first floor. Customers having transactions C and D go to the second floor. Then customers start to wait until their number is called by one of the transaction-desks. After that, customers having transactions A and C join to wait for the next transaction in the pay-office. So there can again be a queue in the pay-office

transaction desk. The queuing discipline is first-come-first-out. And it is assumed that there is infinite waiting space. Customers having transactions B and D leave the system directly after being served.

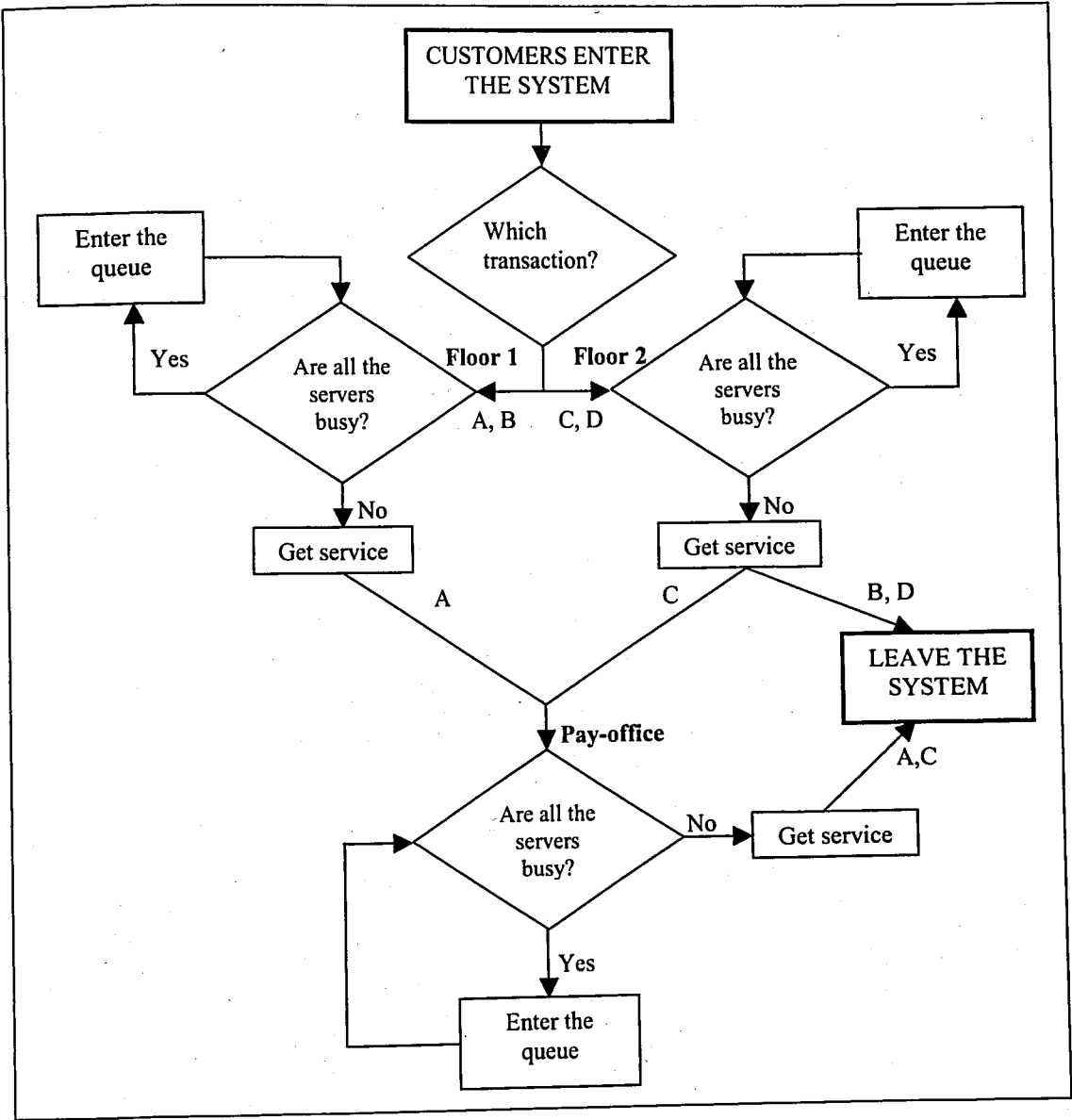


Figure 6.1. Flowchart of queuing system in the Central Bank

The directors wonder whether the server's performance is satisfactory, and the numbers of servers are sufficient for the rush hours in summer months. Therefore, we investigate each server separately for one summer month.

The distributions of customers' interarrival times and transactions service times are to be found.

6.1. Input Data Modeling

6.1.1. The Distribution of Interarrival Times

A program, called Q-MATIC, controls the system. It is installed on a PC. The program stored the past data. Anyway, the past statistics is not fully stored; instead, the average values are available. The system collects the data, and presents a statistic from them. It examines the system in periods of half an hour. An example can be seen in Table 6.1.

Table 6.1. Q-MATIC Print output of the summary data

Period Time	...	Tickets Taken	Customer Served	...	No Shows	Average Values		Customer Waiting	Max Values	
						Waiting Time	Transaction Time		Waiting Time	Transaction Time
08:30	..									
09:00	..									
09:30	...	3	3			0:00:22	0:13:52	1	0:01:06	0:19:30
10:00	...	3	-			0:00:00	-	-	0:00:02	-
10:30	..	1	4			-	0:12:58	-	-	0:19:02
11:00	..	4	3			-	0:17:12	-	-	0:19:56
11:30	..	2	1			-	0:19:30	-	-	0:19:30
12:00	..	2	4			-	0:18:56	-	-	0:19:42
12:30						-	-	-		
13:00						-	-	-		
13:30		3	1			0:00:00	0:19:10	-	0:00:02	0:19:10
14:00		6	5		1	0:00:46	0:16:32	2	0:03:50	0:19:52
14:30		5	6			0:00:22	0:11:26	1	0:01:30	0:19:44
15:00		4	4			0:01:06	0:15:04	1	0:04:26	0:19:56
15:30			1				0:19:12			0:19:12
16:00										
...										
Total		33	32							
Average		3	3		1	0:00:26	0:16:18			
Max value		6	6		1	0:01:06	0:19:30	2	0:04:26	0:19:56

The Q-MATIC only controls the four transactions, A, B, C and D in the system. Therefore the output includes the transactions at the first and second floor. There are no data stored about the pay-office. So the required information is taken from the authorities at the pay-office.

The “no-show” column shows the number of customers pressing the button twice by mistake. Therefore, the system ignores that one, and does not include the service time when calculating the average. The system detects the no-show, when the transaction time is less than 10 second.

The Q-MATIC system does not give the exact arrival times of the customers, it gives the number of arriving customers in half hour periods. Since the Poisson distribution describes many random processes quite well and is mathematically quite simple, it is reasonable to assume Poisson Process with rate λ per time unit. We have the number of customers per time unit. And the number of arrivals in unit intervals is independent. Thus the interarrival time of the customers is exponentially distributed with λ .

The bank opens at 08:30, but the system opens at 09:00. Therefore the customers coming before 09:00 are not allowed to enter the system until 09:00. We have the data that belong to June, July and August 2002. The data are seen at Table A.29. The distribution graph of arriving customers during summer months day by day can be seen in Figure 6.2.

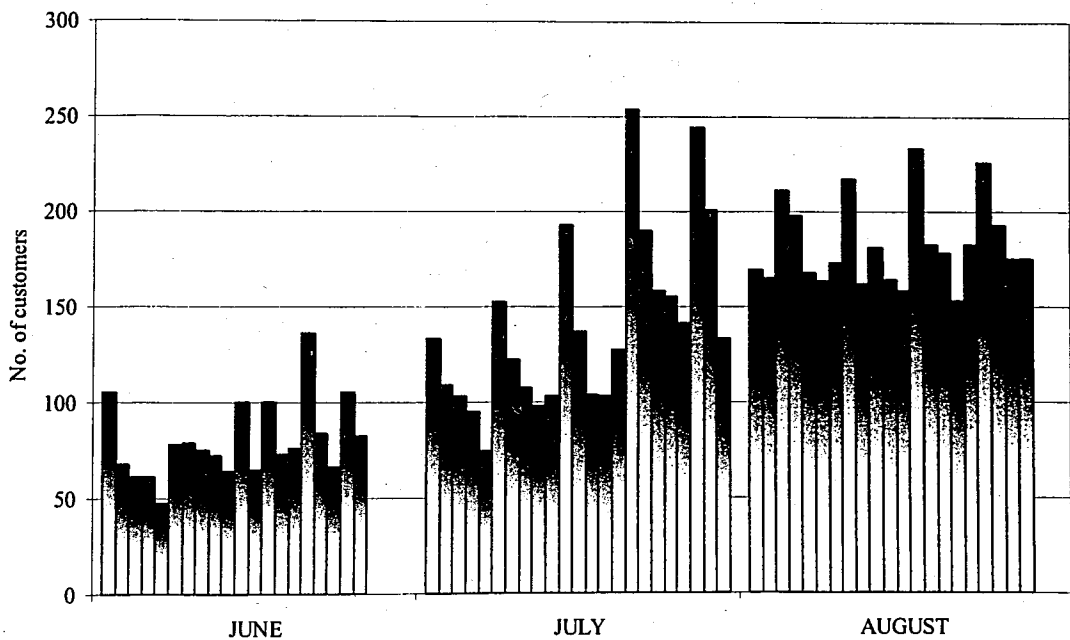


Figure 6.2. Distribution of number of arriving customers during summer months

Referring to Figure 6.2, it is seen that the density of the customers varies during summer, and although the peak of the arrivals occur in July, August is the month with the highest number of arriving customers. Therefore, we investigate the system for August.

For August

The data of August are entered in an Excel worksheet day by day (Table A.29). Then the data belonging to the same time period (half hours period) are summed and averaged (Table A.30). The graph of the mean data is shown in Figure 6.3.

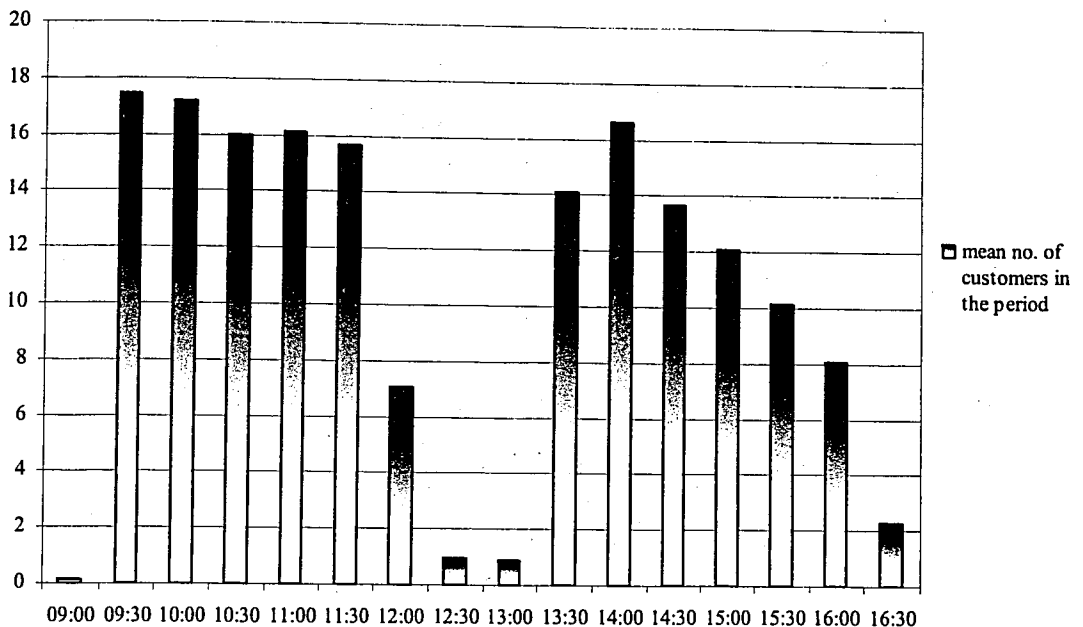


Figure 6.3. Distribution of mean number of arriving customer in one day, in August

By examining the histogram, it is seen that arrivals depends on time. The customers are likely to come until 12:00. The customer arrivals decrease dramatically between 12:00 and 13:00. And after 13:00, the customer arrivals again increases, and decreases as the time process. We assume Nonhomogeneous Poisson Process for generating the arrival times. The time intervals are taken as 08:30-12:00, 12:00-13:00, 13:00-15:00, and 15:00-16:30. We found mean number of customers in these periods and also arrival rates λ for that period (Table A.31).

- The average number of customer is found as 97,333 for 3 hour (09:00-12:00). $\lambda = 97,333$, $180 \text{ minutes} / 97,333 = 1,849315068$ minutes as mean of the first time interval.
- The average number of customer is found as 8 for one hour (12:00-13:00). $\lambda = 8$, $60 \text{ minutes} / 8 = 7,5$ minutes as mean of second time interval.
- The average number of customer is found as 56,42857 for 2 hours (13:00-15:00). $\lambda = 56,42857$, $120 \text{ minutes} / 56,42857 = 2,126582278$ minutes as mean of third time interval.
- The average number of customer is found as 20,42857 for 1,5 hour (15:00-16:30). $\lambda = 20,42857$, $90 \text{ minutes} / 20,42857 = 4,405594406$ minutes as mean of third time interval.

Customers arrival times are generated from Nonhomogeneous Poisson Process by using VBA function, called Nonhexpo().

To check whether we generate consistent numbers for arriving customer times, we examine the number of arriving customers between 09:00 and 12:00, after the simulation runs n times at "Spreadsheet" Excel sheet, we collect the number of arriving customers up to 12:00 at every run at "ArrivalControl" Excel sheet (Table A.32). We see the average number of arriving customers up to 12:00 is 97,3 which is consistent with the real data (97,333 - see Table A.31).

6.1.2. The Distribution of Service Times

It is seen from Q-MATIC output that, the average and maximum values of Transaction times are available (Table 6.1). We examine each print output for three months daily. We notice that the average transaction times and maximum transaction times do never exceed 20 minutes which looks very strange. When we discussed this matter with the officers working there, they also said it is impossible, they said there are many transaction times that exceed 20 minutes. When we talked to the technical service, they said there is a technical problem in the software. They examined the program and discovered that when

first installing the Q-MATIC on the PC, the maximum value of transaction time was set to 20 minutes.

Thus, we did not take into account the wrong transaction times. Instead, we talked to the officers about the transaction times of each transaction type. We learnt that the transaction times differ according to the type of transaction. They mentioned the minimum, most-likely, and maximum transaction times for each transaction type, A, B, C and D. they also gave the detailed information about what each transaction type involves. Transaction A can be “Open a single account” and “Open a multiple account”. Their occurrence proportions are %60 and %40 respectively. Transaction C can be “Payment from account super currency” and “Payment to agent”, where the occurrence proportions are %96 and %4, respectively. Finally, transaction D can be “Transaction made incase of agents’ recourse” and “Payment of transfer currency to possessor”, where the occurrence proportions are %55 and %45, respectively. It was said that the transaction time at the pay-office is about 5 minutes. We assume Triangular Distribution for the transaction times since we have the information of the minimum, maximum and most likely values. The parameters of the Triangular Distribution are entered at “data” sheet (Table 6.2).

Table 6.2. Triangular distribution parameters for generating transaction times

	C	D	E	F
28				
29	Parameters for Triangular Distribution			
30	Transactions	a	m	b
31	A0,60,4	10	15	20
32		15	20	30
33	B	10	15	20
34	C0,960,04	15	20	35
35		17	21	25
36	D0,550,45	60	90	120
37		10	14	15
38	Pay-Office Desk	3	5	7
39				
data / Spreadsheet / Simulation / Simulation Summary / ArrivingCor				

6.1.3. The Distribution of Types of Transactions

As mentioned before, customers can have 4 types of transactions.

A-Open an account,

B-Express transaction desk,

C-Payment & Other Operations,

D-Inheritance Operations.

We have the data of how many customers choose which transaction type.

As;

- There are four possible outcomes with probabilities p_i , $i=A, B, C, D$ and $\sum p_i = 1$.
- If n of these outcomes are performed, any one of the n does not affect the outcome of the other $n-1$ experiments.

We assume multinomial distribution.

6.1.3.1. Point Estimation of p_i 's. We have data for 21 days at August. The frequencies and the probabilities of the customers having transaction A, B, C and D are as follows (Table 6.3).

Table 6.3. The distribution of no. of arriving customers according to the transaction types

Transaction	August,2002	Probability	Cumulative Probability
A	1494	0,3903	0,3903
B	246	0,0643	0,4545
C	2039	0,5327	0,9872
D	49	0,0128	1,0000
TOTAL	3828	1	1

By VLOOKUP() Excel Function, The transaction type is generated as A, B, C or D.

The function looks like;

=VLOOKUP(Random();Transaction;3))

- In the first part of the function Random(), a uniform random number is generated. This is the look-up value to be found in the first column of the array, where “Transaction” is the array. This array indicates the probability distribution of the transaction types (Table 6.4).

Table 6.4. The transaction probability array

Lower Probability	Cumulative Probability	Transactions
0,0000	0,3903	A
0,3903	0,4545	B
0,4545	0,9872	C
0,9872	1,0000	D

- VLOOKUP tries to find the look-up value in the first column of the array, if it can not find the exact value, then it finds the largest value that is less than or equal to lookup_value.
- After finding the value, the function returns the value in the third column of the array at the same row.

6.2. Spreadsheet Implementation

6.2.1. Spreadsheet Implementation on Excel Worksheet

All the required data are entered in the “data” Excel worksheet. The input data in worksheet is seen in Table 6.5.

Column A lists the Random Numbers that will be used to generate uniform random variates (0,1). At cell B2, the desired Run Time is entered. In D1 and D2, the Start Time and Close Time of the Bank are entered, and at cell E2, the time period in minutes is calculated. In cells (C4:E12), servers start and end times of break times are entered as minutes, i.e. start=180, end=210, means, the break of the server starts three hours later than the Start Time (12:00), and continuous for half an hour (12:30). In cells (C15:H20), the transactions and the related probability tables are seen. And finally, in cells (C23:E27), the interarrival means for the intervals are listed, i.e. C24=0 and D24=180 means that the first three hours after the system start. The interarrival times of the customers are exponentially distributed with mean E24=1,899315068.

The spreadsheet implementation is done in the “spreadsheet” Excel worksheet; a part of the spreadsheet can be seen in Table 6.6.

In this example, we assign separate columns for each server, different from the last examples in Chapter 5. By doing this, we can make inferences about each server. As the break times of the servers are different, this approach is the most convenient way to analyze this matter.

Let us call the service at floor 1 and 2 as service at stage 1; and service at the pay-office desk as service at stage 2.

The formulas in the spreadsheet follow;

In column A, the customer numbers are shown. And in column B, the arrival times for the coming customers are generated by using the Nonhexpo() function.

Column C:

After the customers enter the system, they go to the ticket machine and choose their transaction type as A, B, C or D. In column C, the generated transaction types are seen.

`C6=IF(ISTEXT(A6);"";VLOOKUP(random();Transaction;3))`

Column D:

In column D, the service times are generated according to the transaction type.

`D6=IF(ISTEXT(A6);"";Tri(C6))`

Columns E, F, G and H:

The E, F, G and H columns indicate the potential start time of the servers 1,2,3 and 4 at the first floor, respectively.

`E6=IF(ISTEXT(A6);"";IF(AND(B6>=Begin_1;B6<=End_1);End_1;0))`

`F6=IF(ISTEXT(A6);"";IF(AND(B6>=Begin_1;B6<=End_1);End_1;0))`

`G6=IF(ISTEXT(A6);"";IF(AND(B6>=Begin_2;B6<=End_2);End_2;0))`

`H6=IF(ISTEXT(A6);"";IF(AND(B6>=Begin_2;B6<=End_2);End_2;0))`

Since the lunch break hours differs according to the servers, the function checks if the arrival time of that customer is between break hours of that server. If this is the case, then the potential start time is set to the end of the break time of the related server, else, the potential start time is set to 0.

```
E7=IF(ISTEXT(A7);"";IF(AND(K6>=Begin_1;K6<=End_1);End_1;K6))
F7=IF(ISTEXT(A7);"";IF(AND(L6>=Begin_1;L6<=End_1);End_1;L6))
G7=IF(ISTEXT(A7);"";IF(AND(M6>=Begin_2;M6<=End_2);End_2;M6))
H7=IF(ISTEXT(A7);"";IF(AND(N6>=Begin_2;N6<=End_2);End_2;N6))
```

The seventh row and the rest of the rows contain the formula above. In columns K, L, M and N, the completion times of the service at Servers 1, 2, 3 and 4 in the first floor are found, respectively. Again the lunch break hours are checked whether the end service time for the previous work of the server (potential start time of service) is between break hours. If it is, then the potential start time is set to end of the break time of the related server, else, the potential start time is set to time when that server gets free, by looking at the completion time of that server from the previous work, i.e. for the first server; the completion times of transactions of that server are written in K column, and the formula returns the value of the last completion time of the first server (at the same time the earliest time of that server getting free).

Column I:

```
I6=IF(OR(C6="A";C6="B");MATCH(MIN(E6:H6);E6:H6;0);"")
```

At the column I, the server who will serve that customer is selected. Since these are the first floor servers, the function searches for the type of the transaction from column C. If the transactions are A or B, then the function returns the server number who is first available for serving that customer. The function finds this value first by taking the minimum of column E:H, which show the potential start times of the server. Then the MATCH function is used in order to find the position of the smallest potential time. The MATCH function returns the relative position of min (E6:H6) in the array E6:H6. If the function detects that the transactions are not A or B, then the empty cell is returned.

Column J:

$J6 = \text{IF}(I6="", "", \text{MAX}(B6; \text{MIN}(E6:H6)))$

The actual start time of the service for the corresponding customer in column A is found. in column J. The IF function first detects if the type of transaction will be done in that floor or not, by looking at column I. If the transaction is not belong to that floor, then the function returns empty cell, else if the column I indicates the server number, then the function returns the actual start time of the service. It first takes the minimum of potential start times of the servers, and then takes the maximum of this number and the arrival time of the customer.

Columns K, L, M and N:

At column K, L, M and N, the completion times of the service for Servers 1, 2, 3 and 4 are calculated, respectively.

$K6 = \text{IF}(\text{ISTEXT}(A6), "", \text{IF}(I6=1; AC6; 0))$

$L6 = \text{IF}(\text{ISTEXT}(A6), "", \text{IF}(I6=2; AC6; 0))$

$M6 = \text{IF}(\text{ISTEXT}(A6), "", \text{IF}(I6=3; AC6; 0))$

$N6 = \text{IF}(\text{ISTEXT}(A6), "", \text{IF}(I6=4; AC6; 0))$

In the sixth row of column K, L, M, and N, the function searches column I. Column I indicates which server will process that customer. i.e. if the server 1 is selected for service, K6 returns the value in cell AC6, if not, 0 is returned in order to indicate that that server has not started service yet. The value in cell AC indicates the completion time of that transaction.

$K7 = \text{IF}(\text{ISTEXT}(A7), "", \text{IF}(I7=1; AC7; K6))$

$L7 = \text{IF}(\text{ISTEXT}(A7), "", \text{IF}(I7=2; AC7; L6))$

$M7 = \text{IF}(\text{ISTEXT}(A7), "", \text{IF}(I7=3; AC7; M6))$

$N7 = \text{IF}(\text{ISTEXT}(A7), "", \text{IF}(I7=4; AC7; N6))$

In cell AC, the completion time of that service is found. In the seventh row and the rest of column K, L, M, and N, the function again searches column I, i.e. if the server 1 is selected for service, the function returns the value in cell AC7 (completion time of the transaction), if not, the previous value of that column is selected (K6).

Column O, P, Q, R, S and T:

O6=IF(ISTEXT(A6);"";IF(AND(B6>=Begin_3;B6<=End_3);End_3;0))

P6=IF(ISTEXT(A6);"";IF(AND(B6>=Begin_3;B6<=End_3);End_3;0))

Q6=IF(ISTEXT(A6);"";IF(AND(B6>=Begin_3;B6<=End_3);End_3;0))

R6=IF(ISTEXT(A6);"";IF(AND(B6>=Begin_4;B6<=End_4);End_4;0))

S6=IF(ISTEXT(A6);"";IF(AND(B6>=Begin_4;B6<=End_4);End_4;0))

T6=IF(ISTEXT(A6);"";IF(AND(B6>=Begin_4;B6<=End_4);End_4;0))

Again like column E, F, G and H columns, these columns indicate potential start time of the servers 1, 2, 3, 4, 5 and 6 at the second floor, respectively.

In the sixth row of these columns, the function checks if the arrival time of that customer is in the break hours of that server, if it is, then the potential start time is set to the end of the break time else, the potential start time is set to the start time of the System.

O7=IF(ISTEXT(A7);"";IF(AND(W6>=End_3;W6<=End_3);End_3;W6))

P7=IF(ISTEXT(A7);"";IF(AND(X6>=End_3;X6<=End_3);End_3;X6))

Q7=IF(ISTEXT(A7);"";IF(AND(Y6>=End_3;Y6<=End_3);End_3;Y6))

R7=IF(ISTEXT(A7);"";IF(AND(Z6>=End_4;Z6<=End_4);End_4;Z6))

S7=IF(ISTEXT(A7);"";IF(AND(AA6>=End_4;AA6<=End_4);End_4;AA6))

T7=IF(ISTEXT(A7);"";IF(AND(AB6>=End_4;AB6<=End_4);End_4;AB6))

The seventh row and the rest of the rows contain the formula above. In columns W, X, Y, Z, AA and AB, the completion times of the service at Servers 1, 2, 3, 4, 5 and 6 in the second floor are found, respectively. The lunch break hours are checked whether the end of service time for the previous work of the server (potential start time of service) is between break hours. If it is the case then the potential start time is set to the end of the

break time of the related server, else, the potential start time is set to the time when that server gets free, by looking at the completion time of that server from the previous work. (At the same time this is the earliest time that server getting free)

Column U:

$$U6=IF(OR(C6="C";C6="D");MATCH(MIN(O6:T6);O6:T6;0);"")$$

In the column U, like in column I, the server who will serve that customer is selected. Since this is the second floor server, the function searches for the type of the transaction from column C. If the transactions are C or D, then the function returns the server number who is first available for service of that customer. The function finds this value first by taking the minimum of column O:T, which shows the potential start times of the servers, and then the MATCH function is used in order to find the position of the smallest possible time. The MATCH function returns the relative position of min(O6:T6) in the array O6:T6. If the function detects that the transactions are not C or D, then the empty cell is returned.

Column V:

$$V6 =IF(U6="";"";MAX(B6;MIN(O6:T6)))$$

Like in column J, the actual start time of the service for the corresponding customer in column A is found. The IF function first detects the type of transaction will be done in that floor or not, by looking at column U. If the transactions do not belong to that floor, then the function returns the empty cell, else if U column indicates the server number, then the function returns the actual start time of the service. It first takes the minimum of potential start time of the servers, and then takes the maximum of this number and the arrival time of the customer.

Columns W, X, Y, Z, AA and AB:

In columns W, X, Y, Z, AA and AB, the completion times of the service for Servers 1, 2, 3, 4, 5 and 6 are calculated, respectively.

$$W6=IF(ISTEXT(A6);"";IF(U6=1;AC6;0))$$

```

X6=IF(ISTEXT(A6);"";IF(U6=2;AC6;0))
Y6=IF(ISTEXT(A6);"";IF(U6=3;AC6;0))
Z6=IF(ISTEXT(A6);"";IF(U6=4;AC6;0))
AA6=IF(ISTEXT(A6);"";IF(U6=5;AC6;0))
AB6=IF(ISTEXT(A6);"";IF(U6=6;AC6;0))

```

In the sixth row of column W, X, Y, Z, AA and AB, the function searches column U. Column U indicated which server will process for that customer. i.e. if the server 1 is selected for servicing, W6 returns the value in cell AC6, if not, 0 is returned in order to indicate that the server has not started the service yet.

```

W7=IF(ISTEXT(A7);"";IF(U7=1;AC7;W6))
X7=IF(ISTEXT(A7);"";IF(U7=2;AC7;X6))
Y7=IF(ISTEXT(A7);"";IF(U7=3;AC7;Y6))
Z7=IF(ISTEXT(A7);"";IF(U7=4;AC7;Z6))
AA7=IF(ISTEXT(A7);"";IF(U7=5;AC7;AA6))
AB7=IF(ISTEXT(A7);"";IF(U7=6;AC7;AB6))

```

In the seventh row and the rest of column W, X, Y, Z, AA and AB, the function again searches the value in column U, i.e. if the server 1 is selected for service, W6 returns the value in cell AC7, if not, the previous value of that column is selected (W6)

Column AC:

In this column the completion times of the transactions of the customers is calculated. Whether in the first floor or the second floor, all the completion times can be seen in this column.

```
AC6 =IF(D6="" ;"";SUM(D6;J6;V6))
```

The column AC sums the values of D6, J6, V6, which are service time, start time in floor one, and start time on floor 2, respectively. Notice that, only one of the start times, J6 or V6 exist.

Column AD:

$AD6=IF(OR(C6="A";C6="C");AC6;"")$

This column lists the end service time of the customers who have transactions A or C. Customers having transactions B and D leave the system here. Customers with transactions A and C join here and enter the queue for the pay-office process.

The IF function selects the ones having transaction A and C, and returns the end service times, where the other ones are removed from the system after now.

Column AE:

$AE6=IF(ISTEXT(AD6);"";A6)$

Which customers are still in the system is listed here. The IF function searches column AD, where the end service times are listed, if there is number in AD, meaning that customer will continue processing, then the customer number in column A is returned.

Column AF:

$AF6=IF(ISERR(SMALL((AD:AD);A6)));"";SMALL((AD:AD);A6))$

In this column, the arrival times of the customers having finished their first-part process are sorted from the smallest to largest. The SMALL function does this sorting. It returns the k^{th} smallest value in AD:AD, where k^{th} is represented by the number in column A. The customer sequence changes. In the first part of the IF statement, there is an error check. Since not all the costumers pass through the second stage (ones having transaction B and D), the small function gives error when it fails to find the number. By adding ISERR(SMALL((AD:AD);A6), when Excel encounters such error, it returns an empty cell, if this is the case. That means, the customer number in the A column does not continue for service at stage 2.

Column AG:

$AG6=IF(AF6="";"";expo(5))$

In this column the service times for the pay-office transaction is generated. The function generates the service times only for the customers passing to the pay-office desk by checking column AF.

Columns AH and AI:

AH6 =IF(AF6="";"";IF(AND(AF6>=Begin_5;AF6<=End_5);End_5;0))

AI6 =IF(AF6="";"";IF(AND(AF6>=Begin_6;AF6<=End_6);End_6;0))

These columns indicate the potential start time of the servers 1 and 2 at the pay-office, respectively. The IF function checks if the arrival time of that customer is in the break hours of that server. If this is the case, then the potential start time is set to the end of the break time else, the potential start time is set to the start time of the System.

AH7=IF(AF7="";"";IF(AND(AL6>=Begin_5;AL6<=End_5);End_5;AL6))

AI7=IF(AF7="";"";IF(AND(AM6>=Begin_6;AM6<=End_6);End_6;AM6))

The seventh row, and the rest of the rows include the formula above. The IF function checks whether the potential start time of the customer is in break hours. If this is the case, then the potential start time is set to the end of the break time else, the potential start time is set to time when that server gets free by looking the service completion time of that server.

Column AJ:

AJ6=IF(AF6="";"";MATCH(MIN(AH6:AI6);AH6:AI6;0))

In the column AJ, like in the columns I and U, the server who will serve that customer is found.

Column AK:

AK6=IF(ISTEXT(AF6);"";MAX(AF6;MIN(AH6:AI6)))

Column AK determines the actual start time of the service for the customer, like in columns J and V.

Columns AL and AM:

AL6 =IF(AF6="";"";IF(AJ6=1;AK6;0))

AM6 =IF(AF6="";"";IF(AJ6=2;AK6;0))

AM7 =IF(ISTEXT(AF7);"";IF(AJ7=1;AN7;AL6))

AL7 =IF(ISTEXT(AF7);"";IF(AJ7=2;AN7;AM6))

These columns calculate the completion times of the service for each server like columns K to L and W to AB, where AK indicates the completion time of that transaction.

Column AN:

AN6 =IF(ISTEXT(AF6);"";AG6+AK6)

In this column, the completion time of the service is calculated, simply by adding the service time in column AG and actual start time in column AK.

Column AO:

AO6 =VLOOKUP(AF6;AD:AE;2;0)

At column AD, the completion times of the service at floor 1 or floor 2 are written, and the AE column shows which customers have these completion times. In cell AF6, the completion times are sorted, where the sequence of the customers changes is no longer the same as before.

Then, in column AO, by the help of VLOOKUP function, the right sequence of the customers are found by matching the completion time of the first stage and the arrival time at the second stage. The function searches value AF6 inside column AD:AE, when this function is able to match, then it returns the customer number from column AE.

Column AP:

AP6 =IF(AF6="";"";AK6-AF6)

This column calculates the waiting time of the customers in queue at the pay-office desk.

Column AQ:

AQ6=IF(AF6="";"";AN6-AF6)

This column calculates the total waiting time of the customers at the pay-office desk.

Column AR:

AR6=VLOOKUP(AE6;AO:AP;2;0)

In column AP, the waiting time at the pay-office desk is calculated, where AE includes the number of the customer who enter stage 2. This column rearranges the column AP according to the sequence like in the beginning. The customer number after the first stage is matched with the one at the end of the second stage, when the match occurs, the waiting time at stage 2 is returned.

Column AS:

AS6=VLOOKUP(AE6;AO:AQ;3;0)

This column rearranges the column AQ according to the sequence of the beginning. The right customer number after the first stage is matched with the one at the end of the second stage. When the match occurs, the total waiting time at stage 2 is returned.

Columns AT, AU, AV and AW:

AT6=IF(I6="";"";J6-B6)

AU6=IF(I6="";"";AC6-B6)

AV6=IF(U6="";"";V6-B6)

AW6=IF(U6="";"";AC6-B6)

In columns AT, AU, AV and AW, customers waiting time in queue, total waiting time in the system at stage 1 and customers waiting time in queue, total waiting time in the system at stage 2 are found, respectively.

Columns AX and AY:

AX6=IF(ISTEXT(A6);"";SUM(AR6;AT6;AV6))

AY6=IF(ISTEXT(B6);"";SUM(AS6;AU6;AW6))

In columns AX and AY, customers waiting time in the queue in the whole system (both in the first stage and in the second stage) and customers total time spent in the system are found respectively, i.e. AX columns sum the AR, AT and AV, which represent customers waiting time in queue at the second stage, customers waiting time in queue at the 1st floor and customers waiting time in queue at the 2nd floor, respectively. Of course, only one of the columns, AT or AV includes values (1st floor or 2nd floor). And if the corresponding customer did not select transactions A or C, column AR is empty as well.

Columns AZ to BK:

In these columns, the server's busy time is calculated.

At the first floor;

AZ6=IF(I6=1;D6;"")

BA6=IF(I6=2;D6;"")

BB6=IF(I6=3;D6;"")

BC6=IF(I6=4;D6;"")

At the second floor;

BD6=IF(U6=1;D6;"")

BE6=IF(U6=2;D6;"")

BF6=IF(U6=3;D6;"")

BG6=IF(U6=4;D6;"")

BH6=IF(U6=5;D6;"")

BI6=IF(U6=6;D6;"")

At pay-office desk;

BJ6=IF(AJ6=1;AG6;"")

BK6=IF(AJ6=2;AG6;"")

Since each row corresponds to a customer, each customer is served by any server. The IF function searches whether the related server served the customer. In this case, the service time generated in columns D and AG is returned.

6.2.2. Calculating Performance Measures

The performance measures are calculated in the “spreadsheet” Excel worksheet, in the right part of the spreadsheet implementation. The performance measures for the first floor can be seen in Table 6.7.

Table 6.7. Performance measures for the first floor

	BL	BM
1		
2		
3	PERFORMANS MEASURES	
4		
5		
6	Number of Customers enter in the System	=MAX(A:A)
7		
8	First Floor, Transactions A and B	
9		
10	Max time spent at floor 1	=MAX(AU:AU)
11		
12	Average time spent at floor 1	=AVERAGE(AU:AU)
13		
14	Max waiting time in queue at floor 1	=MAX(AT:AT)
15		
16	Average waiting time in queue at floor 1	=AVERAGE(AT:AT)
17		
18	Server 1 Utilization	=SUM(AZ:AZ)/MAX(K:K;time)
19		
20	Server 2 Utilization	=SUM(BA:BA)/MAX(L:L;time)
21		
22	Server 3 Utilization	=SUM(BB:BB)/MAX(M:M;time)
23		
24	Server 4 Utilization	=SUM(BC:BC)/MAX(N:N;time)
25		

data Spreadsheet Simulation Simulation_Summary Arr

The number of customers who entered the system can be seen in cell BM6. In cells BM8 to BM24, the first floor performances are shown. Customers maximum and average time spent in the system are found in Cells BM10 and BM12, respectively where AU represents the total waiting time of customers in the first floor. Customers maximum and average waiting time in the queue are found in Cell BM14 and BM16, respectively, where AT represents the waiting time of customers in queue (Table 6.7).

Moreover, the fraction of time that the servers 1, 2, 3 and 4 being busy at the first floor are found in Cells BM18, BM20, BM22 and BM24, respectively i.e. for the first server; the formula is “BM18=SUM(AZ:AZ)/MAX(K:K;time)”. The sum of server 1’s service times (SUM(AZ:AZ)) is divided by the total time the server is in the system (MAX(K:K;time)) (Table 6.7).

The performance measures for the second floor can be seen at Table 6.8.

Table 6.8. Performance measures for the second floor

	BL	BM
25		
26	Second Floor, Transactions C and D	
27		
28	Max time spent at floor 2	=MAX(AW:AW)
29		
30	Average time spent at floor 2	=AVERAGE(AW:AW)
31		
32	Max waiting time in queue at floor 2	=MAX(AV:AV)
33		
34	Average waiting time in queue at floor 2	=AVERAGE(AV:AV)
35		
36	Server 1 Utilization	=SUM(BD:BD)/MAX(W:W;time)
37		
38	Server 2 Utilization	=SUM(BE:BE)/MAX(X:X;time)
39		
40	Server 3 Utilization	=SUM(BF:BF)/MAX(Y:Y;time)
41		
42	Server 4 Utilization	=SUM(BG:BG)/MAX(Z:Z;time)
43		
44	Server 5 Utilization	=SUM(BH:BH)/MAX(AA:AA;time)
45		
46	Server 6 Utilization	=SUM(BI:BI)/MAX(AB:AB;time)
47		

« ‹ › » data Spreadsheet Simulation Simulation Summary Arrivi »

In cells BM26 to BM46, the second floor performance is shown. Customers maximum and average time spent in the system are shown in Cells BM28 and BM30, respectively where AW represents the total waiting time of customers in the second floor. Customers' maximum and average waiting time in the queue are shown in cells BM32 and BM34, respectively, where AV represents the waiting time of customers in queue (Table 6.8).

Moreover, the fraction of time that the servers 1, 2, 3, 4, 5 and 6 are busy at the second floor are found in Cells BM36, BM38, BM40, BM42, BM44 and BM46, respectively i.e. for the first server; the formula is "BK36=SUM(BD:BD)/MAX(W:W;time)". The sum of his service time (SUM(BD:BD)) is divided by the total time he is in the system (MAX(W:W;time)) (Table 6.8).

The performance measures for the second stage can be seen at Table 6.9.

Table 6.9. Performance measures for the second stage

	BL	BM
47		
48	Second Stage, Transactions A and C	
49		
50	Max time spent at second stage	=MAX(AS:AS)
51		
52	Average time spent at second 2	=AVERAGE(AS:AS)
53		
54	Max waiting time in queue at second 2	=MAX(AR:AR)
55		
56	Average waiting time in queue at second	=AVERAGE(AR:AR)
57		
58	Server 1 Utilization	=SUM(BJ:BJ)/MAX(AL:AL;time)
59		
60	Server 2 Utilization	=SUM(BK:BK)/MAX(AM:AM;time)
61		

In cells BM48 to BM60, the second stage performance can be found. Customers maximum and average time spent in the system are found in Cells BM50 and BM52, respectively where AS represents the total waiting time of customers in the second stage. Customers maximum and average waiting time in the queue are found in Cell BM54 and

BM56, respectively, where AR represents the waiting time of customers in the queue (Table 6.9).

The fractions of time that the servers 1 and 2 are busy at the second stage are found in Cells BM58 and BM60, respectively (Table 6.9).

The performance measures for the whole system can be seen in Table 6.10.

Table 6.10. Performance measures for the whole system

	BL	BM
61		
62	The whole system	
63		
64	Max waiting time queue in whole system	=MAX(AX:AX)
65		
66	Average time spent in whole system	=AVERAGE(AY:AY)
67		
68	Average waiting time in queue in the whole system	=AVERAGE(AZ:AZ)
69		
70		

data / Spreadsheet / Simulation / Simulation Summary / Arrivir

We, also, calculate the average time spent in the whole system (not only at the 1st floor or 2nd floor, but together with first stage and the second stage), and the average waiting time in the queue in the whole system (Table 6.10).

Moreover, the probability distribution of customers having X minutes service and probability distribution of customers waiting Y minutes are found. The formulas can be seen in Tables 6.11 and 6.12, respectively.

Table 6.11. Probability distribution of customers having X minutes service

BL		BM
72	Frequency Distribution of Customers having X minutes service	
73	Minutes	Frequency
74	5	=COUNTIF(D:D,"<="&BL74)
75	=IF(BL74>MAX(D:D);"";BL74+5)	=IF(BL75="";"";COUNTIF(D:D,"<="&BL75)-SUM(BM\$74:\$BM\$74))
76	=IF(BL75>MAX(D:D);"";BL75+5)	=IF(BL76="";"";COUNTIF(D:D,"<="&BL76)-SUM(BM\$74:\$BM\$75))
77	=IF(BL76>MAX(D:D);"";BL76+5)	=IF(BL77="";"";COUNTIF(D:D,"<="&BL77)-SUM(BM\$74:\$BM\$76))
78	=IF(BL77>MAX(D:D);"";BL77+5)	=IF(BL78="";"";COUNTIF(D:D,"<="&BL78)-SUM(BM\$74:\$BM\$77))
79	=IF(BL78>MAX(D:D);"";BL78+5)	=IF(BL79="";"";COUNTIF(D:D,"<="&BL79)-SUM(BM\$74:\$BM\$78))
80	=IF(BL79>MAX(D:D);"";BL79+5)	=IF(BL80="";"";COUNTIF(D:D,"<="&BL80)-SUM(BM\$74:\$BM\$79))

dataSpreadsheetSimulationSimulation_SummaryArrivingControlValidation

In Table 6.11., at column BL, the minutes are written. They start with 5 and increase by 5 minutes until the maximum service time is reached, (MAX(D:D)). At column BM, by using COUNTIF Excel function, the number of customers having less than or equal to the service time in column BL are calculated. The COUNTIF function searches column D (service times) and counts and sums the number which are less than or equal to the value in column BL.

Table 6.12. Probability distribution of customers waiting Y minutes

BO		BP
72	Frequency Distribution of Customers waiting Y minutes in queue	
73	Minutes	Frequency
74	5	=COUNTIF(AX:AX,"<="&BO74)
75	=IF(BO74>MAX(BM\$64);"";BO74+5)	=IF(BO75="";"";COUNTIF(AX:AX,"<="&BO75)-SUM(BP\$74:\$BP\$74))
76	=IF(BO75>MAX(BM\$64);"";BO75+5)	=IF(BO76="";"";COUNTIF(AX:AX,"<="&BO76)-SUM(BP\$74:\$BP\$75))
77	=IF(BO76>MAX(BM\$64);"";BO76+5)	=IF(BO77="";"";COUNTIF(AX:AX,"<="&BO77)-SUM(BP\$74:\$BP\$76))
78	=IF(BO77>MAX(BM\$64);"";BO77+5)	=IF(BO78="";"";COUNTIF(AX:AX,"<="&BO78)-SUM(BP\$74:\$BP\$77))
79		
80		

dataSpreadsheetSimulationSimulation_SummaryArrivingControlValidation

In Table 6.12, in column BO, the minutes are written until the maximum waiting time is reached. This is controlled by taking the max of maximum waiting time in the whole system calculated in cell BM\$64 (Table 6.10). At column BP, again by using the COUNTIF function, the number of customers waiting less than or equal to the waiting time in column BM are calculated. The function searches column AX, then counts and sums the number which are less than or equal to the value in column BO.

These results are stored in different arrays called `Service_Time` and `Waiting_Time` in simulation macro `Run_Sim`, and listed in the “ServiceTime” and “WaitingTime” Excel worksheets, respectively (Tables A.33 and A.34).

In section 6.1.1 (the distribution of arrival times) we mentioned that after simulation runs, the number of arriving customers between 09:00 and 12:00 is listed in the “ArrivalControl” Excel worksheet (Table A.32). The formula for calculating this measure can be seen in Table 6.13.

Table 6.13. Finding number of arriving customers between 09:00 and 12:00

	BN	BO
3		
4		
5	Number of Arriving Customers between 09:00 and 12:00	
6	=COUNTIF(B:B;"<"&180)	
7		
8		

dataSpreadsheetSimulationSimulation_Sur

6.3. Simulation by Using Visual Basic Application

After making the necessary revision in the “data” worksheet, i.e. entering start time-end time and break times of the system and mean values of interarrival and service times, the simulation can be started by just pressing the button called `Run_Sim` in the “data” Excel worksheet. When `n` is equal to 1000, the execution time is 40 minutes. The execution time of the model in this chapter is the longest when compared to the models in Chapter 5, since this model is the most complicated one, and is the one having the most number of columns in process. The simulation is done by the macro, named `Run_Sim`, written in VBA. Below we list the `Run_Sim` macro which executes the simulation and `Clear_Sim` macro which is used for cleaning the previous results;

```
1 Public data, Simulation, ServiceTime, WaitingTime, Validation, ArrivingControl
  As Worksheet
2 Public r, time As Double
3 Public I As Double
```

```

4      Sub Run_Sim()
5          Set data = Sheet1
6          Set Simulation = Sheet3
7          Set ServiceTime = Sheet4
8          Set WaitingTime = Sheet6
9          Set Validation = Sheet7
10         Dim Service_Time(100), Waiting_Time(100) As Double
11         For k = 1 To 100
12             Service_Time(k) = 0
13             Waiting_Time(k) = 0
14         Next
15         Clear_Sim
16         n = data.Cells(2, 2)
17     For m = 1 To n
18         r = 0
19         time = 0
20         I = 0
21         Sheets("Spreadsheet").Select
22         Range("B6").Select
23         ActiveCell.FormulaR1C1 = "=NonHexpo()"
24         Simulation.Cells(m + 1, 1) = Range("BM10")
25         Simulation.Cells(m + 1, 2) = Range("BM12")
26         Simulation.Cells(m + 1, 3) = Range("BM14")
27         Simulation.Cells(m + 1, 4) = Range("BM16")
28         Simulation.Cells(m + 1, 5) = Range("BM18")
29         Simulation.Cells(m + 1, 6) = Range("BM20")
30         Simulation.Cells(m + 1, 7) = Range("BM22")
31         Simulation.Cells(m + 1, 8) = Range("BK24")
32         Simulation.Cells(m + 1, 9) = Range("BM28")
33         Simulation.Cells(m + 1, 10) = Range("BM30")
34         Simulation.Cells(m + 1, 11) = Range("BM32")
35         Simulation.Cells(m + 1, 12) = Range("BM34")
36         Simulation.Cells(m + 1, 13) = Range("BM36")

```

```

37      Simulation.Cells(m + 1, 14) = Range("BM38")
38      Simulation.Cells(m + 1, 15) = Range("BM40")
39      Simulation.Cells(m + 1, 16) = Range("BM42")
40      Simulation.Cells(m + 1, 17) = Range("BM44")
41      Simulation.Cells(m + 1, 18) = Range("BM46")
42      Simulation.Cells(m + 1, 19) = Range("BM50")
43      Simulation.Cells(m + 1, 20) = Range("BM52")
44      Simulation.Cells(m + 1, 21) = Range("BM54")
45      Simulation.Cells(m + 1, 22) = Range("BM56")
46      Simulation.Cells(m + 1, 23) = Range("BM58")
47      Simulation.Cells(m + 1, 24) = Range("BM60")
48      Simulation.Cells(m + 1, 25) = Range("BM66")
49      Simulation.Cells(m + 1, 25) = Range("BM68")
50      Validation.Cells(m + 1, 1) = Range("BM16")
51      Validation.Cells(m + 1, 2) = Range("BM34")
52      ArrivingControl.Cells(m + 1, 1) = Range("BO6")
53      For k = 0 To 23
54      If Cells(k + 74, 64) <> "" Then
55      Service_Time(k) = Cells(k + 67, 65) + Service_Time(k)
56      End If
57      Next
58      For k = 0 To 43
59      If Cells(k + 74, 67) <> "" Then
60      Waiting_Time(k) = Cells(k + 74, 68) + Waiting_Time(k)
61      End If
62      Next
63      data.Cells(1, 1) = data.Cells(r + 1, 1)
64      r=0
65      Next
66      For k = 0 To 23
67      ServiceTime.Cells(k + 2, 2) = Service_Time(k)
68      Next
69      For k = 0 To 43

```

```

70             WaitingTime.Cells(k + 2, 2) = Waiting_Time(k)
71             Next
72     End Sub

15     Sub Clear_Sim()
15.1         Sheets("Simulation").Select
15.2         Range("A2:A2000,B2:B2000,C2:C2000,D2:D2000,E2:E2000,F2:F2000,_
15.3         G2:G2000,H2:H2000,I2:I2000,J2:J2000,K2:K2000,L2:L2000,M2:M2000,_
15.4         N2:N2000,O2:O2000,P2:P2000,Q2:Q2000,R2:R2000,S2:S2000,T2:T2000,
15.5         U2:U2000,V2:V2000,W2:W2000,X2:X2000,Y2:Y2000, Z2:Z2000").Select
15.6         Selection.ClearContents
15.7         Sheets("ServiceTime").Select
15.8         Range("B2:B50").Select
15.9         Selection.ClearContents
15.10        Sheets("WaitingTime").Select
15.11        Range("B2:B50").Select
15.12        Selection.ClearContents
15.13        Sheets("Validation").Select
15.14        Range("A2:A2000,B2:B2000").Select
15.15        Selection.ClearContents
15.16        Sheets("ArrivingControl").Select
15.17        Range("A2:A2000").Select
15.18        Selection.ClearContents
15.19        End Sub

```

First, data, Simulation, ServiceTime, WaitingTime, Validation and ArrivingControl are defined as worksheet and some other variables are defined through command lines 1, 2 and 3. Then, the arrays of Service_Time and Waiting_Time, which store the service times and waiting times in order to find the distributions respectively are defined (command line 10) and set to 0 through command lines 11,12, 13 and 14 through a “Do Loop” . A macro called Clear_Sim is run at line 15. This macro clears the previous simulation results before the new simulation starts. Then the run time n is taken form the “data” Excel worksheet at cell B\$2. The simulation starts, the variable, r, which identifies the row number that the

random number will be taken is set to 0. And, also the variables, time and I which are required for generating nonhomogeneous arrivals are set to 0 at the beginning of the simulation. The results are collected through a "Do Loop" with n replication. In the "Do Loop", the spreadsheet implementation is recalculated n times. This is achieved by selecting the formula in cell B6 in the "Spreadsheet" Excel worksheet (lines 21, 22, 23). The command in line 23 gives the stimulus to Excel for recalculating the spreadsheet implementation. Then the performance measures are written in the "Simulation" Excel worksheet row by row through the command lines 24 to 49 (Table 6.14). And the performance calculated for validation of the model are collected in the "Validation" Excel worksheet through command line 50 and 51 (Table 6.15).

Table 6.14. Simulation output at "Simulation" Excel worksheet

	A	B	C	D	E	F	G	H	I	J
	Max time spent at floor 1	Average time spent at floor 1	Max waiting time in queue at floor 1	Average waiting time in queue at floor 1	Server 1 Utilization (1 st floor)	Server 2 Utilization (1 st floor)	Server 3 Utilization (1 st floor)	Server 4 Utilization (1 st floor)	Max time spent at floor 2	Average time spent at floor 2
1										
2	100,0639468	36,67985787	76,90138547	19,0259271	72,47%	69,74%	73,21%	65,92%	120,9431862	58,47296783
3	38,96130023	23,19042865	22,33896068	5,987406079	68,14%	60,90%	66,90%	71,21%	66,74580786	36,09358118
4	53,10899403	26,24601015	34,28206448	8,983433937	73,30%	77,04%	76,68%	77,35%	112,1240005	39,92294988
5	61,31449487	30,82873077	44,7526137	13,48943909	83,79%	81,07%	81,98%	80,09%	107,5174781	32,57979302
6	55,39435366	29,72512343	34,32790152	12,57754061	74,86%	73,77%	80,29%	70,06%	107,3689296	41,62135197
7	37,1399383	22,96180639	19,24238593	4,855309057	75,83%	72,14%	73,90%	76,99%	97,99257925	48,57771242
8	53,86936076	25,56942754	35,58707279	7,587241138	72,62%	63,99%	75,28%	69,62%	103,1658523	31,22358051
9	61,58009014	29,99723759	36,82842956	11,91394441	85,59%	80,13%	87,60%	88,84%	112,1463096	45,08900135
10	59,52209752	26,92861492	37,34849496	9,515033997	73,51%	71,84%	79,19%	74,93%	57,00837677	30,96293356
11	94,01357161	36,67830167	71,50635325	18,73335353	79,79%	79,95%	76,88%	76,37%	82,67298239	39,36150824
12	102,3874393	45,23416283	86,79785075	28,02406196	86,14%	83,79%	83,30%	83,24%	111,6732274	43,82837129
13	44,88652784	22,7287511	21,97869338	5,121129985	73,42%	69,80%	70,71%	70,31%	87,97925023	32,81369698
14	92,6377267	33,18873218	76,14313779	15,29252911	79,91%	78,52%	77,02%	73,56%	146,9062496	39,12907135
15	51,013655	21,98250165	27,17521738	4,267816983	81,38%	74,70%	77,62%	73,35%	116,5788105	37,93088571
16	97,07819126	54,18778066	73,70967311	36,20138345	91,81%	95,27%	90,54%	92,16%	90,650702	35,78936934
17	46,92478169	22,73923706	22,56606791	5,013490611	75,07%	67,24%	71,88%	70,16%	115,1693809	57,55640834
18	99,47670098	41,00480311	76,09307014	23,27667444	79,11%	78,34%	80,95%	78,49%	86,14387641	29,20537359
19	86,35506898	37,27940276	70,17245186	18,95467838	71,34%	77,26%	73,95%	71,58%	50,30285785	29,21542643
20	55,36984131	34,07873275	37,68646059	16,23790733	81,08%	77,55%	83,38%	78,29%	92,96619577	37,91377733
21	43,85548277	23,77140002	22,70217497	6,178503073	74,05%	69,43%	72,19%	74,99%	81,24852045	38,68166857
22	76,26774837	35,83089048	54,43826771	18,56611363	82,87%	82,75%	84,17%	80,20%	122,7823417	51,67307408
23	49,25820565	24,28612859	33,60541197	6,613446874	85,53%	82,69%	81,52%	80,68%	47,75060228	27,54424011
24	57,03055642	23,65429355	39,35910562	6,423241973	82,60%	81,57%	77,32%	77,99%	141,9880781	51,00126348
25	54,15435918	24,89727805	37,14694194	7,787452356	73,07%	73,03%	73,03%	70,31%	113,8871525	43,77998444

Table 6.15. Simulation output at “Validation” Excel worksheet

	A	B	C	D	E
1	Average Waiting Time in queue at floor 1	Average Waiting Time in queue at floor 2		Average Waiting Time in queue at floor 1	Average Waiting Time in queue at floor 2
2	5,931214007	19,32661116	Mean	14,8113	16,5129
3	8,468925116	17,11003536	Standart Deviation	11,5424	11,8397
4	5,796902485	17,25736998			
5	12,1437414	15,96346			
6	8,366032457	21,48635265			
7	10,80423998	5,102607749			
8	23,92109091	12,94905392			
9	27,38538068	4,161053596			
10	7,900685164	47,18062406			

ArrivingControlValidationService_ChartServiceTimeWaitingTimeWaiting_Chart

In line 52, the command line lists the performance measure found in Table 6.13 in “ArrivalControl” worksheet. Moreover, for every run, the performance measures calculated for the distribution of service times and waiting time in columns BM and BP are stored in arrays Service_Time and Waiting_Time in “spreadsheet” Excel worksheet through lines 53 to 62. Before passing to the next iteration, at line 63, the Random Number stream is revised, and the seed to be used next is set to the last Random Number used in the previous iteration.

After the end of n Runs, for finding the distribution of customers having X service time and waiting Y minutes, the values collected in ServiceTime and WaitingTime arrays are listed in Excel worksheets “ServiceTime” and “WaitingTime” through command lines 66 to 71 (Table 6.16 and Table 6.17). Line 72 is the end of the program.

Table 6.16. Simulation output at “Service_Time” Excel worksheet

	A	B	C	D	E	F
1	X minutes	Frequency	Probability	Cumulative Pr.	Sum	=SUM(B:B)
2	5	0	=IF(ISNUMBER(B2);B2/F\$1;"")	=C2	Probability Sum	=SUM(C:C)
3	=A2+5	0	=IF(ISNUMBER(B3);B3/F\$1;"")	=IF(ISNUMBER(C3);C3+D2;"")		
4	=A3+5	29248	=IF(ISNUMBER(B4);B4/F\$1;"")	=IF(ISNUMBER(C4);C4+D3;"")		
5	=A4+5	62720	=IF(ISNUMBER(B5);B5/F\$1;"")	=IF(ISNUMBER(C5);C5+D4;"")		
6	=A5+5	58064	=IF(ISNUMBER(B6);B6/F\$1;"")	=IF(ISNUMBER(C6);C6+D5;"")		
7	=A6+5	28702	=IF(ISNUMBER(B7);B7/F\$1;"")	=IF(ISNUMBER(C7);C7+D6;"")		
8	=A7+5	8149	=IF(ISNUMBER(B8);B8/F\$1;"")	=IF(ISNUMBER(C8);C8+D7;"")		
9	=A8+5	0	=IF(ISNUMBER(B9);B9/F\$1;"")	=IF(ISNUMBER(C9);C9+D8;"")		
10	=A9+5	0	=IF(ISNUMBER(B10);B10/F\$1;"")	=IF(ISNUMBER(C10);C10+D9;"")		
11	=A10+5	0	=IF(ISNUMBER(B11);B11/F\$1;"")	=IF(ISNUMBER(C11);C11+D10;"")		
12	=A11+5	0	=IF(ISNUMBER(B12);B12/F\$1;"")	=IF(ISNUMBER(C12);C12+D11;"")		
13	=A12+5	0	=IF(ISNUMBER(B13);B13/F\$1;"")	=IF(ISNUMBER(C13);C13+D12;"")		

ValidationService_ChartServiceTimeWaitingTimeWaiting_Charttotal_time(floor1)total_tir

Table 6.17. Simulation output at “Waiting_Time” Excel worksheet

A		B	C	D	E	F
1	Y minutes	Frequency	Probability	Cumulative Pr.	Sum	=SUM(B:B)
2	5	32218	=IF(ISNUMBER(B2);B2/F\$1;"")	=C2	Probability Sum	=SUM(C:C)
3	=A2+5	14636	=IF(ISNUMBER(B3);B3/F\$1;"")	=IF(ISNUMBER(C3);C3+D2;"")		
4	=A3+5	13923	=IF(ISNUMBER(B4);B4/F\$1;"")	=IF(ISNUMBER(C4);C4+D3;"")		
5	=A4+5	13416	=IF(ISNUMBER(B5);B5/F\$1;"")	=IF(ISNUMBER(C5);C5+D4;"")		
6	=A5+5	12876	=IF(ISNUMBER(B6);B6/F\$1;"")	=IF(ISNUMBER(C6);C6+D5;"")		
7	=A6+5	12246	=IF(ISNUMBER(B7);B7/F\$1;"")	=IF(ISNUMBER(C7);C7+D6;"")		
8	=A7+5	11540	=IF(ISNUMBER(B8);B8/F\$1;"")	=IF(ISNUMBER(C8);C8+D7;"")		
9	=A8+5	10405	=IF(ISNUMBER(B9);B9/F\$1;"")	=IF(ISNUMBER(C9);C9+D8;"")		
10	=A9+5	9181	=IF(ISNUMBER(B10);B10/F\$1;"")	=IF(ISNUMBER(C10);C10+D9;"")		
11	=A10+5	8014	=IF(ISNUMBER(B11);B11/F\$1;"")	=IF(ISNUMBER(C11);C11+D10;"")		
12	=A11+5	7208	=IF(ISNUMBER(B12);B12/F\$1;"")	=IF(ISNUMBER(C12);C12+D11;"")		
13	=A12+5	6266	=IF(ISNUMBER(B13);B13/F\$1;"")	=IF(ISNUMBER(C13);C13+D12;"")		

Validation / Service_Chart / ServiceTime \ WaitingTime / Waiting_Chart / total_time(floor1) / total_time

The work of VBA finish at this point. Then, these collected “n” performance measures are summarized and analyzed.

It is important to be sure that, the repetitions are statistically independent in order to make statistical analysis. In our example, the generated random numbers are independent. The seed at the beginning of each run is the following last seed used in the previous run. Therefore, statistically independent runs are achieved.

6.4. Verification and Validation of Simulation Model

After collection of the data and the model building and the, the next stage in the modeling cycle is to check that the model is an adequate representation of the system [11]. One of the most important tasks is the verification and validation of the simulation model [3]. There are two basic goals of the validation process.

- To produce a model that represents true system behavior closely enough for the model to be used as a substitute for the actual system for the experimentation purpose.
- To increase the credibility of the model to an acceptable level, so that the model will be used by managers and other decision makers.

Verification asks the questions: Is the model implementation correctly? Are the input parameters and the logical structure of the model correctly represented?

Validation is concerned with building a valid model. It is utilized to determine whether a model is an accurate representation of the real system. Validation is usually achieved through the calibration of the model, an iterative process of comparing the model to actual system behavior and using the discrepancies between the two, and the insights gained to improve the model. This process is repeated until the model accuracy is judged to be acceptable.

Validation is the process by which model users gain confidence that output analysis is making valid inferences about the real system under study [3].

6.4.1. Verification of the Model

While constructing the model in the Bank, we questioned people familiar with the system (officials, employees, directors). We also obtained the real data of summer 2002 as computer output. Then, we collected the assumptions on the components and we formulated a model. We translated the conceptual model to a computer recognizable form using spreadsheet implementation in Excel.

For verification, we manually trace first 30 arriving customers. We see that they are divided into two groups after the transaction types are generated as A, B, C or D. Then, the service times are generated. The queue discipline is FIFO. Columns E, F, G and H represents the potential start times of the servers, and column I finds the first available server number, who will serve customers having A or B transaction type (floor 1). We see that column I gives the number of the server that first becomes available. We see that customers coming earlier start service earlier. Column I works only for the customers who choose transaction type A or B. Otherwise, it returns an empty cell. Since column J calculates the start service times of the customers at floor 1, we check whether the end times of the service at columns K, L, M and N are found by summing the start time of the service in column J and the generated service time in column D, or not. We also do the same check for floor 2. We see that only the customers having transaction types C or D have their service in this part. And in column AC, the end service times from both floor 1 and floor 2 are listed, where in column AD, only the customers having transaction type A

or C remain. At this point, we examine whether the customers having transaction type B or D leave the system or not. And we see that the rest of the row belong to the customers having transaction type B or D is empty, which shows that these customers leave the system. In column AF, the end service times of column AD are sorted in order to make the queue FIFO. We check this sorted list by hand. After that, the service times are generated, and customers are started to have service at the pay-office (second stage) as in the first stage. Again the customers, who come first, have the service earlier. And in column AN, the end service times of the second stage are listed. In column, AO, the number of customer who have taken service at second stage is found. We compare these customer numbers with the ones who enter the second stage (who have transactions A and C), whether the VLOOKUP() Excel function works correctly when choosing the customer number. In columns AP, AQ, AR and AS, customers waiting time in the queue and in the system are found. We check these numbers by manual calculation. In columns AT, AU, AV and AW, these waiting times are corrected according to the customer line at the beginning (column A). We again check that only the customers having transaction type A or C exist. From columns AX to BI, all servers busy time are calculated. We sum up all servers busy times. These number must be equal to the sum of generated service times at both stage 1 and stage 2, which is really the case (i.e. $SUM(D:D)+SUM(AG:AG)=SUM(AX:BI)$). And we check manually whether the performance measures are calculated correctly.

6.4.2. Validation of the Model

Validation is the overall process of comparing the model and its behavior to the real system and its behavior [3].

Since we have the real data results for customer average waiting times in queue, we compare the customer's average waiting time in queue to the real system values for validation of the model. At the beginning of the construction of the model, when we talked to the people working there, they gave us the service times of each transaction types. They said that the average service times of transactions A, B, C and D are 20, 17, 18 and 60 minutes, respectively. After we run 1000 times repetition with these values, we found

average waiting time at floor 1 is about 26.821299, and at floor 2 is about 5.246751. The actual values are 14,152361 and 16,7025641, respectively (Table A.35). The simulated results are far away from the actual values. Then, we tried many runs with different service times. We discovered that the simulation results are very sensitive to the change at the service times. We went to the department in the Bank again; we talked to more people, told them what is triangular distribution, so they gave us information about the transactions in detail, and about the service times. We found out that the numbers they gave us before were not the mean service times, but the rough values of mode, minimum or maximum service times. We got information about the minimum, maximum and the mode values of the each transaction times detailly. Moreover, we learnt that, transactions A, C and D have two different type of transaction inside which were discussed in Section 6.1.2.

6.4.2.1. Taking Averages of Transaction Times. Instead of taking into account the related probabilities (Table 6.2), we take the averages of the transactions times of each type. And at transaction C, we take the parameters which has the probability 0.96, we ignore the other probability, 0.04) (Table 6.18).

Table 6.18. Triangular distribution parameters

Transactions	a	m	B
A	12,5	17,5	25
B	10	15	20
C	15	20	35
D	35	52	67,5
Pay-Office Desk	3	5	7

Let X and Y be the average waiting time of customers in queue at floor 1 and at floor 2 respectively which are found by simulation. X is equal to 17,4745 and Y is equal 16,7168. We construct below hypothesis tests to answer the questions:

- Does the simulated value of the average waiting times at floor 1 reflect the real data result?

$H_0: E(X)= 14,1526 \text{ minutes}$

$H_1: E(X) \neq 14,1526 \text{ minutes}$

We choose the level of significance $\alpha=0,05$, $n=1000$. Since n is large, we can use z test and $z(0,025)=1,96$.

$$S=13,0486$$

$$s.e.(\bar{X}) = \frac{s}{\sqrt{n}}$$

$$s.e.(\bar{X}) = \frac{13,0486}{\sqrt{1000}} = 0,4126$$

$$z = \frac{\bar{X} - 14,1526}{s.e.(\bar{X})} = \frac{17,4745 - 14,1526}{0,4126} = 8,05$$

Since z value, 8,05 is very large, we reject the hypothesis that the simulated average waiting time in queue at floor 1 reflect the real data result.

- Does the simulated value of the average waiting time at floor 2 reflect the real data result?

$$H_0: E(Y) = 16,7026 \text{ minutes}$$

$$H_1: E(Y) \neq 16,7026 \text{ minutes}$$

$$S=11,74998$$

$$s.e.(\bar{Y}) = 0,37157$$

$$z = \frac{16,7168 - 16,7025641}{0,37157} = 0,038$$

Since $-1,96 < 0,038 < 1,96$, we can not reject the hypothesis that the simulated average waiting time in queue at floor 2 reflect the real data result.

6.4.2.2. Taking Transaction Times with True Probabilities. When we specify the distribution of the transaction times as given in Section 6.1.2 (Table 6.2), we arrive the following results.

The average waiting time at floor 1 and 2 are collected in the “Validation” Excel worksheet through the Run_Sim macro. Let X and Y be the average waiting time of customers in queue at floor 1 and at floor 2 respectively which are found by simulation. X is equal to 14,8113 and Y is equal 16,5129 (Table 6.15). We construct below hypothesis tests to answer the questions:

- Does the simulated value of the average waiting times at floor 1 reflect the real data result?

$$H_0: E(X) = 14,1526 \text{ minutes}$$

$$H_1: E(X) \neq 14,1526 \text{ minutes}$$

We choose the level of significance $\alpha = 0,05$, $n = 1000$. Since n is large, we can use z test and $z(0,025) = 1,96$.

$$S = 11,5424 \quad (\text{From table 6.15})$$

$$s.e.(\bar{X}) = \frac{11,5424}{\sqrt{1000}} = 0,365$$

$$z = \frac{\bar{X} - 14,1526}{s.e.(\bar{X})} = \frac{14,8113 - 14,1526}{0,365} = 1,805$$

Since $1,805 < 1,96$, we can not reject the hypothesis that the simulated average waiting time in queue reflect the real data result.

- Does the simulated value of the average waiting time at floor 2 reflect the real data result?

$H_0: E(Y) = 16,7026 \text{ minutes}$

$H_1: E(Y) \neq 16,7026 \text{ minutes}$

$S = 11,8397$ (From table 6.15)

$s.e.(\bar{Y}) = 0,3744$

$$z = \frac{16,5129 - 16,7026}{0,3744} = -0,507$$

Since $-1,96 < -0,507 < 1,96$, we can not reject the hypothesis that the simulated average waiting time in queue at floor 2 reflect the real data result.

6.4.2.3. Taking Transaction Times with Different Probabilities. When we specify the probabilities of the transaction times as in Table 6.19, we arrive the following results.

Table 6.19. Triangular distribution parameters, 50% probabilites

Transactions		a	m	b
A	0,4	10	15	20
	0,6	15	20	30
B		10	15	20
C		15	20	35
D	0,45	60	90	120
	0,55	10	14	15
Pay-Office Desk		3	5	7

Let X and Y be the average waiting time of customers in queue at floor 1 and at floor 2 respectively which are found by simulation. X is equal to 20,697 and Y is equal 16,1636. We construct below hypothesis tests to answer the questions:

- Does the simulated value of the average waiting times at floor 1 reflect the real data result?

$$H_0: E(X) = 14,1526 \text{ minutes}$$

$$H_1: E(X) \neq 14,1526 \text{ minutes}$$

We choose the level of significance $\alpha=0,05$, $n=1000$. Since n is large, we can use z test and $z(0,025)=1,96$.

$$s.e.(\bar{X}) = 0,4784$$

$$z = \frac{\bar{X} - 14,1526}{s.e.(\bar{X})} = \frac{20,697 - 14,1526}{0,4784} = 13,69$$

Since z value, 13,69 is very large, we reject the hypothesis that the simulated average waiting time in queue at floor 1 reflect the real data result.

- Does the simulated value of the average waiting time at floor 2 reflect the real data result?

$$H_0: E(Y) = 16,7026 \text{ minutes}$$

$$H_1: E(Y) \neq 16,7026 \text{ minutes}$$

$$s.e.(\bar{Y}) = 0,3658$$

$$z = \frac{16,1636 - 16,7026}{0,3658} = -1,47$$

Since $-1,96 < -1,47 < 1,96$, we can not reject the hypothesis that the simulated average waiting time in queue at floor 2 reflect the real data result.

As a result, the second parameters in transaction C ($a=17$, $m=21$ and $c=25$ with probability 0,04) can be ignored and only the ones with probability 0,96 can be taken into consideration. The second parameters do not affect the simulation results significantly. And also at transaction D, we can either take the equal probabilities for two kinds of parameters, or take the reverse of the probabilities. The probabilities of the related parameters don't affect the simulation results significantly. But, the probability distribution of parameters for transaction A must be taken into account as in section 6.4.2.2. the simulation results are sensitive to the parameters of transaction A.

6.5. Output Analysis

Output analysis refers to analyzing the data produced by a simulation and drawing inferences from the data about the behavior of the real system [3].

We analyzed the system for only August since it is the month with the highest workload. There are four servers at floor 1 and six servers at floor 2, also 2 servers at the pay-office desk. All servers are available to serve.

The summary statistics are collected in the "Simulation_Summary" Excel worksheet. The mean, standard deviation, min, max values are found. Moreover, the %95 confidence intervals for the mean are calculated. The formulas can be seen in Table A.37. Moreover, the probability distributions of average time spend in system for floor 1, floor 2 and stage 2 are found. Also comparison of the server utilizations is made.(Table A.38). The graphs of the distributions are shown in Figure B.13, B.14 and B.15, respectively, where the graph of servers' utilization comparison is shown in Figure 6.4.

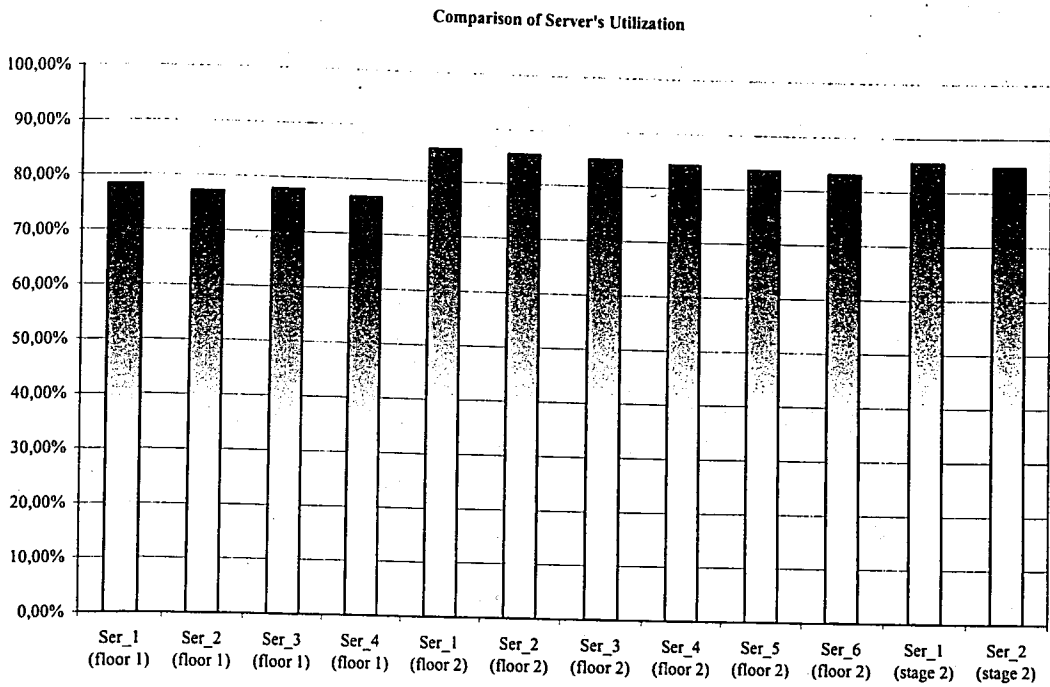


Figure 6.4. The graph of comparison of server’s utilization

Moreover, the probability distribution of customers having X service time and waiting Y minutes are analyzed in the worksheets “ServiceTime” and “WaitingTime”. The results are shown at Table 6.20 and Table 6.21. (The formulas are at Table 6.16 and Table 6.17). Their graphs are shown at Figures B.16 and B.17, respectively.

Table 6.20. Probability distribution of customers having X service time

	A	B	C	D
1	X minutes	Frequency	Probability	Cumulative Pr.
2	5	0	0,00000	0,000000000
3	10	0	0,00000	0,000000000
4	15	29380	0,15562	0,155621001
5	20	63151	0,33450	0,490121403
6	25	57879	0,30658	0,796696894
7	30	29015	0,15369	0,950384550
8	35	8011	0,04243	0,992817492
9	40	0	0,00000	0,992817492
10	45	0	0,00000	0,992817492
11	50	0	0,00000	0,992817492
12	55	0	0,00000	0,992817492
13	60	0	0,00000	0,992817492
14	65	0	0,00000	0,992817492
15	70	0	0,00000	0,992817492
16	75	0	0,00000	0,992817492
17	80	153	0,00081	0,993627908
18	85	285	0,00151	0,995137506
19	90	256	0,00136	0,996493495
20	95	186	0,00099	0,997478707
21	100	170	0,00090	0,998379169
22	105	113	0,00060	0,998977711
23	110	121	0,00064	0,999618628
24	115	49	0,00026	0,999878173
25	120	23	0,00012	1,000000000

ServiceTime / WaitingTime / Waiting_Chart / tot

Table 6.21. Probability distribution of customers waiting Y minutes

	A	B	C	D
1	Y minutes	Frequency	Probability	Cumulative Pr.
2	5	31833	0,16861	0,168614136
3	10	14817	0,07848	0,247097335
4	15	13922	0,07374	0,320839866
5	20	13746	0,07281	0,393650155
6	25	12842	0,06802	0,461672105
7	30	12153	0,06437	0,526044536
8	35	11297	0,05984	0,585882876
9	40	9976	0,05284	0,638724098
10	45	9032	0,04784	0,686565109
11	50	8029	0,04253	0,729093394
12	55	7111	0,03767	0,766759185
13	60	6342	0,03359	0,800351710
14	65	5723	0,03031	0,830665494
15	70	4995	0,02646	0,857123183
16	75	4487	0,02377	0,880890080
17	80	4139	0,02192	0,902813679
18	85	3518	0,01863	0,921447943
19	90	2968	0,01572	0,937168948
20	95	2514	0,01332	0,950485190
21	100	2083	0,01103	0,961518497
22	105	1791	0,00949	0,971005127
23	110	1400	0,00742	0,978420696
24	115	1165	0,00617	0,984591508
25	120	893	0,00473	0,989321581

ServiceTime / WaitingTime / Waiting_Chart / tc

When we examine Table 6.20, we get such inference;

- About %80 of the customers have 25 minutes service time or less.

When we examine Table 6.21, we get such inferences;

- About %70 of the customers wait less than or equal to 45 minutes in the queue.
- About %10 of the customers wait at least 80 minutes in the queue.

When we examine the server's utilization at Figure 6.4, it is seen that the utilization of servers at floor 2 and at the second stage are higher than the server utilization at floor 1. All servers use at least %75 of their time for serving.

It is generally desired to have a point estimate and a confidence interval estimate of the desired performance measure. The point and %95 confidence interval estimates of the performance measures are found in "Simulation_Summary" worksheet (Table A.36.).

Below are the some examples;

- Customers average waiting time in queue at floor 1, w_Q ;

The point estimation according to 6.2 is;

$$\hat{w}_Q = 14,8113 \quad (\text{Table A.36, cell C7})$$

The %95 Confidence Interval Estimation is:

$$s.e.(\hat{w}_Q) = 0,350 \quad (\text{According to Equation 6.4})$$

$$\hat{w}_Q \pm z(0,975)s.e(\hat{w}_w) = (14,0859;15,5267) \quad (\text{Table A.36, cells F7 \& G7})$$

- Server 1 Utilization at the second floor, ρ_1 ;

The point estimation according to 6.1 is:

$$\hat{\rho}_1 = 86,0963\% \quad (\text{Table A.36, cell B17})$$

The %95 Confidence Interval Estimation is:

$\hat{\rho}_1 \pm z(0,975)s.e.(\hat{\rho}_1) = (85,585\%;86,6077\%)$ (Table A.36, cells F17 & G17)

6.6. What-if Analysis

6.6.1. Decreasing Server Number by one at the 1st floor: Case 1

We suppose that there are three servers at floor 1 instead of 4. We run the simulation for n independent repetitions (Table A.39). We set the end time of server 4 at cell to 1000 at floor 1 to discard this server from the system. The comparisons of summary statistics are as follows;

6.6.1.1. Independent Sampling with Equal Variances. We use different and independent random number streams to simulate two systems (App. D.1). This implies that all the observations of simulated basic system are statistically independent of all the observations of the simulated system in Case 1.

Table 6.22. 95% c.i. for the difference of the means of performance measures, independent sampling for Case 1

	BASIC	CASE 1	<u>95% c.i. for the difference of the means</u>		
	<u>p.e. of the means</u>				
FLOOR 1 -TRANSACTIONS "A" AND "B"			Lower	Upper	<u>c.i.widths</u>
Average time spent in the system	32,33	77,49	42,52	47,80	5,28
Average waiting time in queue	14,81	59,95	42,50	47,77	5,27
Server 1 Utilization	78,33%	93,6%	14,45%	16,1%	1,65%
Server 2 Utilization	77,2%	92,88%	14,82%	16,53%	1,71%
Server 3 Utilization	77,85%	92,26%	13,61%	15,19%	1,58%
Server 4 Utilization	76,79%	0			
WHOLE SYSTEM					
Average time spent in the whole system	59,44	70,63	9,27	13,12	3,85
Average waiting time in queue in the whole system	34	45,25	9,32	13,18	3,86

According to Table 6.22, since the confidence intervals are totally greater than zero, then there is a strong evidence that decreasing the server number by one at floor 1 increases the average waiting time of customers in the system and in queue, while the server utilizations increases.

If we examine waiting time distributions;

- In the Case 1 model; about 60% of the customers wait less than or equal to 45 minutes (Table A.40), whereas in the basic model, about 70% of the customers wait less than or equal to 45 minutes in the queue (Table 6.21).
- Moreover, in the Case 1 model; about 30% customers wait at least 80 minutes (Table A.40), whereas in the basic model, about 10% of the customers wait at least 80 minutes in the queue (Table 6.21).

6.6.1.2. Correlated Sampling. We use the same random numbers for each replication to simulate both systems (App D.2).

Table 6.23. 95% c.i. for the mean difference, correlated sampling for Case 1

	BASIC	CASE 1	<u>95% c.i for the</u> <u>mean difference,</u>		
	<u>p.e. of the mean</u> <u>difference, $\bar{D}$</u>		<u>$\bar{D}$</u>		
FLOOR 1 -TRANSACTIONS "A" AND "B"			Lower	Upper	<u>c.i.widths</u>
Average time spent in the system	46		44,86	47,14	2,28
Average waiting time in queue	46		44,86	47,14	2,28
Server 1 Utilization	15,22		14,82%	15,62%	0,80%
Server 2 Utilization	15,8		15,39%	16,21%	0,82%
Server 3 Utilization	14,46		14,1%	14,83%	0,73%
Server 4 Utilization	-				
WHOLE SYSTEM					
Average time spent in the whole system	11,65		11,23	12,08	0,85
Average waiting time in queue in the whole system	11,72		11,29	12,14	0,85

At floor 1, since the generated arrival times and service times are exactly same at the basic system and at Case 1, the difference between average waiting times in the system are equal to the average waiting times in the queue. Thus the confidence interval for the difference, $\bar{D}$ is same (44,86-47,14).

When we compare Table 6.23 with Table 6.22, we notice that using correlated sampling is approximately 50% of the length of the confidence interval based on independent sampling. Thus, correlated sampling produces estimates which are twice as precise in our example.

6.6.2. Changing Periods of Break Hours: Case 2

In the basic model, the break hour is between 12:00 and 13:00. At floor 1, two servers have break time between 12:00 and 12:30, and the other two have 12:30 and 13:00. In case 2, we assume they have break hours between 12:00 and 14:00. And they have taken break times at half hours period. There are three servers in the system between 12:00 and 14:00. In the basic model, the six servers of the second floor take half hour break times three by three between 12:00 and 13:00. In case 2, they have taken again half hours break times two by two in 40 minutes periods. The break hours for the second stage are not changed. Server 1 at the second stage has break between 12:00 and 12:30, while Server 2 has break between 12:30 and 13:00. We run the simulation for n times (Table A.41). We redesigned the “data” worksheet in order to enter the new break hour periods. The comparisons of summary statistics are as follows;

6.6.2.1. Independent Sampling with Equal Variances: We use different and independent random number streams to simulate two systems (App D.1).

Table 6.24. 95% c.i. for the difference of the means of performance measures, independent sampling for Case 2

	BASIC	CASE 2	<u>95% c.i. for the difference of the means</u>		
	<u>p.e. of the means</u>				
			Lower	Upper	<u>c.i.widths</u>
FLOOR 1 -TRANSACTIONS "A" AND "B"					
Average time spent in the system	32,33	31,90	-1,8343	0,9754	2,8097
Average waiting time in queue	14,81	14,36	-1,8489	0,9396	2,7885
Server 1 Utilization	78,33%	77,92%	-1,47%	0,66%	2,13%
Server 2 Utilization	77,2%	78,29%	0,04%	2,14%	2,10%
Server 3 Utilization	77,85%	77,92%	-0,94%	1,08%	2,02%
Server 4 Utilization	76,79%	76,13%	-1,71%	0,39%	2,09%
FLOOR 2 -TRANSACTIONS "C" AND "D"					
Average time spent in the system	40,67	41,35	-0,8804	2,2269	3,1073
Average waiting time in queue	16,51	17,21	-0,8226	2,2217	3,0443
Server 1 Utilization	86,10%	86,43%	-0,69%	1,35%	2,04%
Server 2 Utilization	85,57%	85,77%	-0,83%	1,23%	2,06%
Server 3 Utilization	85,05%	85,35%	-0,73%	1,31%	2,04%
Server 4 Utilization	84,37%	84,63%	-0,77%	1,29%	2,06%
Server 5 Utilization	83,66%	84,03%	-0,68%	1,40%	2,07%
Server 6 Utilization	83,20%	83,50%	-0,74%	1,34%	2,07%
WHOLE SYSTEM					
Average time spent in the whole system	59,44	64,40	3,0258	6,8998	3,87
Average waiting time in queue in the whole system	34,01	38,98	3,0398	6,9151	3,88

According to Table 6.24, the performance measures at floor 1 and floor 2 do not affected so much by the change of break hours. But, since the confidence intervals for the average waiting time in the whole system and in the queue are greater than zero, there is evidence that changing the break hours causes average waiting times in the whole system and in the queue to increase.

If we examine waiting time distributions;

- In the Case 2 model; about %60 of the customers wait less than or equal to 45 minutes (Table A.42), where in the basic model, about %70 of the customers wait less than or equal to 45 minutes in the queue (Table 6.21).
- And in the Case 2 model; %14 customers wait at least 80 minutes (Table A.42), and in the basic model, it was found that again %10 of the customers wait at least 80 minutes in the queue (Table 6.21).

The results at Case 2 may be interpreted as follows; since the arrival rate does not change, people are still not likely to come between 12:00 and 13:00, and likely to come after 13:00. In case 2, we let three servers work between 12:00 and 14:00. And after 13:00, the server number is 3 instead of 4. And since people are coming more after 13:00, the average waiting times increase.

6.6.2.2. Correlated Sampling: We use the same random numbers for each replication to simulate both systems (App D.2).

Table 6.25. 95% c.i. for the mean difference, correlated sampling for Case 2

	BASIC	CASE 1	95% c.i for the		
	<u>p.e. of the mean</u>		<u>mean difference,</u>		
	<u>difference, $\bar{D}$</u>		<u>$\bar{D}$</u>		
FLOOR 1 -TRANSACTIONS "A" AND "B"			Lower	Upper	<u>c.i.widths</u>
Average time spent in the system	-0,22		-0,31	-0,12	0,19
Average waiting time in queue	-0,22		-0,31	-0,12	0,19
Server 1 Utilization	-0,32		-0,46	-0,19	0,27
Server 2 Utilization	1,11		0,92	1,30	0,39
Server 3 Utilization	-0,07		-0,23	0,09	0,31
Server 4 Utilization	-0,75		-0,92	-0,58	0,34
FLOOR 2 -TRANSACTIONS "C" AND "D"					
Average time spent in the system	0				
Average waiting time in queue	0				
Server 1 Utilization	0				
Server 2 Utilization	0				
Server 3 Utilization	0				
Server 4 Utilization	0				
Server 5 Utilization	0				
Server 6 Utilization	0				
WHOLE SYSTEM					
Average time spent in the whole system	4,52		4,43	4,60	0,17
Average waiting time in queue in the whole system	4,54		4,45	4,63	0,18

In Case 2 model, since the generated arrival times and service times are exactly same as in the basic model, the performance measures which belong to floor 2 are exactly the same as in the basic model. That is, changing the break hours of the servers at floor 2 as in Case 2, yields exactly the same results in the basic model at floor 2. Thus, point estimates of the mean difference are 0.

When we compare Table 6.25 with Table 6.24, we notice that using correlated sampling is approximately 20% of the length of the confidence interval based on independent sampling.

6.6.3. Increasing Server Number by one at the 1st floor: Case 3

We suppose that there are 5 servers at floor 1 instead of 4. We run the simulation for n independent repetitions (Table A.43). The comparisons of summary statistics are as follows;

6.6.3.1. Independent Sampling with Equal Variances: We use different and independent random number streams to simulate two systems (App D.1).

Table 6.26. 95% c.i. for the difference of the means of performance measures, independent sampling for Case 3

	BASIC	CASE 3	95% c.i. for the difference of the means		
	p.e. of the means				
FLOOR 1 -TRANSACTIONS "A" AND "B"			Lower	Upper	c.i.widths
Average time spent in the system	32,33	21,08	-12,3080	-10,1915	2,12
Average waiting time in queue	14,81	3,56	-12,3042	-10,2065	2,1
Server 1 Utilization	78,33%	62,61%	-16,71%	-14,73%	1,98%
Server 2 Utilization	77,2%	61,48%	-16,74%	-14,71%	2,02%
Server 3 Utilization	77,85%	63,38%	-15,44%	-13,51%	1,93%
Server 4 Utilization	76,79%	62,21%	-15,54%	-13,61%	1,94%
Server 5 Utilization	-	61,45%			
WHOLE SYSTEM					
Average time spent in the whole system	59,44	57,76	-3,53	0,18	3,71
Average waiting time in queue in the whole system	34	32,33	-3,53	0,18	3,71

According to Table 6.26, since the confidence intervals are totally less than zero, then there is a strong evidence that increasing the server number by one at floor 1 decreases the average waiting time of customers in the system and in queue at floor 1 significantly, while the server utilizations decrease. The average waiting time in the whole system and in queue in the whole system are not affected significantly when the server number is increased by one, since the second floor constitutes a bottleneck Thus, a further improvement in the system might be to increase the number of servers in the second stage.

If we examine waiting time distributions;

- Both in the basic system and the proposed system in the Case 3; about 70% of the customers wait less than or equal to 45 minutes (Table A.44).
- Moreover, in the Case 3 model; about 9% customers wait at least 80 minutes (Table A.44), whereas in the basic model, about 10% of the customers wait at least 80 minutes in the queue (Table 6.21).

6.6.3.2. Correlated Sampling: We use the same random numbers for each replication to simulate both systems (App D.2).

Table 6.27. 95% c.i. for the mean difference, correlated sampling for Case 3

	BASIC	CASE 1	95% c.i for the mean		
	<u>p.e. of the mean</u> <u>difference, $\bar{D}$</u>		<u>difference, $\bar{D}$</u>		
FLOOR 1 -TRANSACTIONS "A" AND "B"			Lower	Upper	<u>c.i.widths</u>
Average time spent in the system	-11,13		-11,84	-10,42	1,42
Average waiting time in queue	-11,13		-11,83	-10,44	1,4
Server 1 Utilization	-15,59%		-16,19%	-14,99%	1,2%
Server 2 Utilization	-15,73%		-16,35%	-15,12%	1,23%
Server 3 Utilization	-14,59%		-15,16%	-14,01%	1,15%
Server 4 Utilization	-21,98		-22,66%	-21,3%	1,37%
WHOLE SYSTEM					
Average time spent in the whole system	-1,42		-2,63	-0,21	2,42
Average waiting time in queue in the whole system	-1,42		-2,62	-0,21	2,4

When we compare Table 6.27 with Table 6.26, we notice that using correlated sampling is approximately 50% of the length of the confidence interval based on independent sampling. Thus, correlated sampling produces estimates which are twice as precise in our example.

As a result, changing the break hours as in Case 2 may affect the system performance slightly, while decreasing or increasing the server number even only by one at floor 1 as in Case 1 and Case 3 respectively, affect the average waiting times of customers significantly. In Case 1, where the server number is decreased by one, the server's utilization gets nearly close to 100%.

6.7. Discussion

Managers could want to achieve a level of service, e.g. they want that 90% of the customers wait less than 10 minutes. Simulation can be used to determine about the necessary number of servers. It is possible that for a fixed number of servers, simulation determines the probability distribution of the waiting time in queue. By comparing the alternative designs through the cost of hiring an extra server and the waiting times, they can control the waiting times of customers by determining the required server number. They can use simulation to forecast the change in waiting times when changing the server number.

7. CONCLUSION

Spreadsheets (such as Microsoft Excel and Lotus 1-2-3) are popular software products, and the skills to program and use them are widely distributed [20]. In this study, we find out if it is possible to perform the simulation of a simple real world queuing model in an Excel spreadsheet. So we perform implementations of spreadsheet simulation for several queuing models in Excel.

We conclude that process-interaction approach is most useful. Although papers argue that spreadsheet implementations of queuing networks can not count the number of customers in the system and can only implement very basic queuing system, we are able to implement spreadsheet simulation of a simple network queuing model with balk and renege properties of customers and break hours of the servers. We also find all the performance measures that can not be found by analytic queuing theory, i.e. percentage of customers that waits at least X minutes. Moreover, we implement a queuing model where the service times are changing according to the queue length.

We also apply spreadsheet simulation to a real world queuing problem in the Central Bank of the Republic of Turkey. After we have performed the verification and validation of the simulation model, we conclude that the spreadsheet-simulation model is a good representation of the real system. Thus, we can use spreadsheet simulation for real world queuing problems. This is certainly the main conclusion of this thesis.

Of course, we cannot say that “all” queuing models can be implemented in spreadsheets. In our experience, spreadsheet simulation has the following advantages;

- We have the full control of performance measures. We are able to find the distribution of queue length in process approach which hasn't been found in the literature before. Moreover, we reach the statistics, e.g. percentage of the customers who wait at least X minutes, which we can not find through analytic queuing theory.

- We can implement queuing models without knowing a programming language and learning a simulation package. This could motivate companies to use spreadsheet simulation, i.e. they would not have to hire trained people to analyze their systems with expensive simulation packages.
- In spreadsheet simulation, it is easy to make “what-if” or sensitivity analysis. Modelers want to determine how sensitive the performance measure is to variations in the unknown parameters. In the real world problem, we apply three different cases and perform what-if and sensitivity analysis.
- When we need something extra about the program, we can rearrange the program by ourselves. In the real world problem, it doesn’t take too much time to redesign the spreadsheet as desired.

If we list the disadvantages of spreadsheet simulation;

- The execution time is slow when compared to dedicated simulation packages. Actually, it depends on the kind of the implementation. As the model gets more complicated, the execution time gets slower. But due to the speed of modern computers, it is still possible to get results.
- It is not possible to implement highly complicated queuing models. For instance, spreadsheets do not have convenient facilities to implement a while-loop or a for-loop.

Finally, spreadsheet simulation can be a useful tool for teaching queuing simulation. When compared to simulation software and analytic queuing theory, spreadsheet queuing simulation is an ideal platform to give students insight into queuing behavior. We can demonstrate why queues form, why they persist and can show the role of variability, and the non-linear effect of increasing utilization on queue performance.

APPENDIX A: EXCEL WORKSHEETS

In this part, spreadsheet implementations of queuing models are shown. Generally numerical outputs are sampled. And in A.5, Crystal Ball simulation results are shown. From A.1 to A.15, the spreadsheet implementations through process interaction approach in the literature are examined; i.e. M/M/1, M/M/2 and M/M/3 queuing models. From A.16 to A.28, the spreadsheet implementations of the models in chapter 5 are given.

A.29, A.30 and A.31 are the data tables of the number of arriving customers for the model in Chapter 6. A.32, A.33 and A.34 are the excel sheets after the simulation runs n times. And A.35 is the data table for the average waiting time of the customers. In A.36, the “Simulation_Summary” excel worksheet is seen, where the formulas are illustrated in A.37. In A.38, the probability distributions of some performance measures are seen. And from A.39 to A.44, the simulation results of Case 1, Case 2 and Case 3 are shown.

Table A.1. Spreadsheet implementation of M/M/1 queuing model, Evans and Olson

	A	B	C	D	E	F	G	H
1	Dan's Carwash							
2				<i>Mean arrival rate/min</i>		0.250 <i>Average</i>		<i>Percent</i>
3				<i>Mean service rate/min</i>		0.333 <i>Wait time</i>		<i>Idle Time</i>
4						13:01		17.92%
5								
6	Customer	TBA	Arrival Time	Start Time	Service Time	Completion Time	Wait Time	Idle Time
7	1	8,367	8,367	8,367	2,029	10,396	0,000	8,367
8	2	1,355	9,722	10,396	7,722	18,118	0,674	0,000
9	3	1,985	11,707	18,118	0,366	18,484	6,410	0,000
10	4	3,885	15,592	18,484	2,541	21,025	2,892	0,000
11	5	0,540	16,132	21,025	2,222	23,246	4,893	0,000
12	6	5,691	21,823	23,246	2,862	26,109	1,424	0,000
13	7	1,874	23,697	26,109	3,863	29,972	2,412	0,000
14	8	4,691	28,388	29,972	14,830	44,802	1,585	0,000

Table A.2. Spreadsheet implementation of M/M/1 queuing model formulas, Evans and Olson

	A	B	C	D	E	F	G	H
1	Dan's Carwash							
2				Mean arrival rate/min		0.250	Average	Percent
3				Mean service rate/min		0.333	Wait time	Idle Time
4						=SUM(G7:G106)/100		
						=SUM(H7:H106)/106		
5	Customer	TBA	Arrival Time	Start Time	Service Time	Completion Time	Wait Time	Idle Time
7	1	=-1*(1/\$F\$1)*LN(RAND())	=B7	=C7	=-1*(1/\$F\$2)*LN(RAND())	=D7+E7	=D7-C7	=D7
8	=A7+1	=-1*(1/\$F\$1)*LN(RAND())	=C7+B8	=MAX(C8,F7)	=-1*(1/\$F\$2)*LN(RAND())	=D8+E8	=D8-C8	=D8-F7
9	=A8+1	=-1*(1/\$F\$1)*LN(RAND())	=C8+B9	=MAX(C9,F8)	=-1*(1/\$F\$2)*LN(RAND())	=D9+E9	=D9-C9	=D9-F8
10	=A9+1	=-1*(1/\$F\$1)*LN(RAND())	=C9+B10	=MAX(C10,F9)	=-1*(1/\$F\$2)*LN(RAND())	=D10+E10	=D10-C10	=D10-F9
11	=A10+1	=-1*(1/\$F\$1)*LN(RAND())	=C10+B11	=MAX(C11,F10)	=-1*(1/\$F\$2)*LN(RAND())	=D11+E11	=D11-C11	=D11-F10
12	=A11+1	=-1*(1/\$F\$1)*LN(RAND())	=C11+B12	=MAX(C12,F11)	=-1*(1/\$F\$2)*LN(RAND())	=D12+E12	=D12-C12	=D12-F11
13	=A12+1	=-1*(1/\$F\$1)*LN(RAND())	=C12+B13	=MAX(C13,F12)	=-1*(1/\$F\$2)*LN(RAND())	=D13+E13	=D13-C13	=D13-F12
14	=A13+1	=-1*(1/\$F\$1)*LN(RAND())	=C13+B14	=MAX(C14,F13)	=-1*(1/\$F\$2)*LN(RAND())	=D14+E14	=D14-C14	=D14-F13

Table A.3. Spreadsheet implementation of M/M/1 queuing model, Evans

	A	B	C	D	E	F	G	H	I
1	Single Server Queueing								
2	Simulation Spreadsheet								
3									
4			1	1,1887	1,1887	0,0000	1,1887	0,4474	1,6361
5	Mean arrival rate	2	2	0,3298	1,5185	1,0000	1,6361	1,1595	2,7956
6	Mean service rate	3	3	0,1209	1,6394	1,0000	2,7956	0,4633	3,2589
7			4	0,8017	2,4411	2,0000	3,2589	0,1008	3,3597
8	Expected No. in Queue	1,333	5	0,1452	2,5863	3,0000	3,3597	0,2486	3,6083
9	Expected Waiting Time	0,667	6	0,3398	2,9262	3,0000	3,6083	0,2296	3,8379
10	Expected % idle time	33%	7	0,7169	3,6431	1,0000	3,8379	0,1182	3,9561
11			8	0,0869	3,7300	2,0000	3,9561	0,7667	4,7228

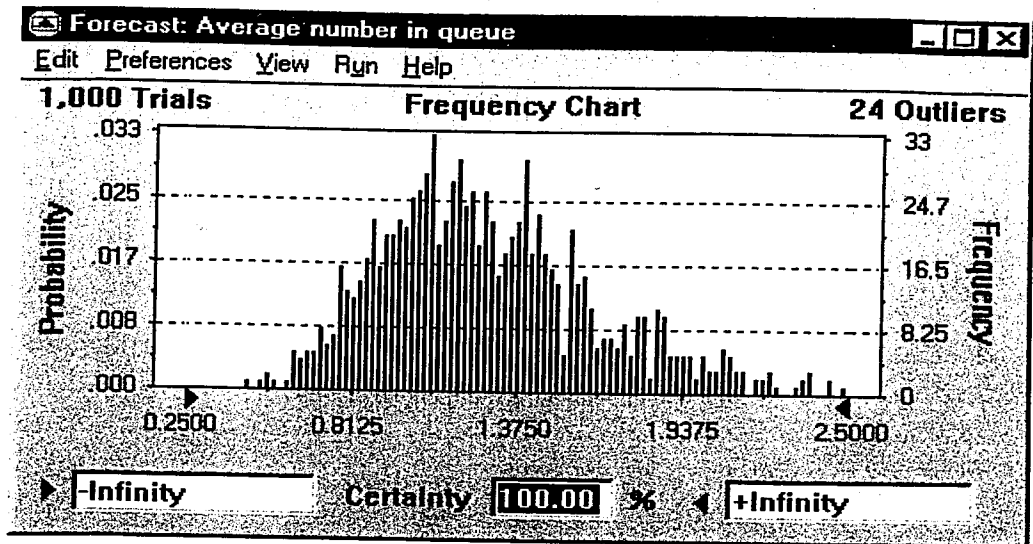
	J	K	L	M	N	O	P	Q
1	WAIT	IDLE	CUMULATIVE AVERAGE		CUMULATIVE AVERAGE		CUMULATIVE AVERAGE	
2	TIME	NO. IN QUEUE	NO. IN QUEUE	NO. IN QUEUE	WAIT TIME	WAIT TIME	IDLE TIME	% IDLE TIME
3								
4	0,0000	1,1887	0,0000	0,0000	0,0000	1,1887	73%	
5	0,1176	0,0000	0,3298	0,2172	0,1176	1,1887	43%	
6	1,1562	0,0000	0,4507	0,2749	1,2737	1,1887	36%	
7	0,8177	0,0000	2,0541	0,8415	2,0915	1,1887	35%	
8	0,7733	0,0000	2,4898	0,9627	2,8648	1,1887	33%	
9	0,6821	0,0000	3,5093	1,1993	3,5469	1,1887	31%	
10	0,1948	0,0000	4,2262	1,1601	3,7418	1,1887	30%	
11	0,2261	0,0000	4,4000	1,1796	3,9678	1,1887	25%	

Table A.4. Spreadsheet implementation of M/M/1 queuing model formulas, Evans

	A	B	C	D	E	F	G	H	I
1	Single Server Queuing		CUSTOMER TIME BET.		ARRIVAL	NUMBER	START	SERVICE	COMPLETION
2	Simulation Spreadsheet		ARRIVALS	TIME	IN QUEUE	TIME	TIME	TIME	TIME
3									
4			1	=-(1/\$B\$5)*LN(RAND())	=D4	0	=MAXA(E4:I3)	=-(1/\$B\$6)*LN(RAND())	=G4+H4
5	Mean arrival rate	2	2	=-(1/\$B\$5)*LN(RAND())	=E4+D5	=C5-MATCH(E5,\$\$3:I4,1)	=MAXA(E5:I4)	=-(1/\$B\$6)*LN(RAND())	=G5+H5
6	Mean service rate	3	3	=-(1/\$B\$5)*LN(RAND())	=E5+D6	=C6-MATCH(E6,\$\$3:I5,1)	=MAXA(E6:I5)	=-(1/\$B\$6)*LN(RAND())	=G6+H6
7			4	=-(1/\$B\$5)*LN(RAND())	=E6+D7	=C7-MATCH(E7,\$\$3:I6,1)	=MAXA(E7:I6)	=-(1/\$B\$6)*LN(RAND())	=G7+H7
8	Expected No. in Queue	=B\$5*2/(\$B\$6*(\$B\$6-\$B\$5))	5	=-(1/\$B\$5)*LN(RAND())	=E7+D8	=C8-MATCH(E8,\$\$3:I7,1)	=MAXA(E8:I7)	=-(1/\$B\$6)*LN(RAND())	=G8+H8
9	Expected Waiting Time	=B8/B5	6	=-(1/\$B\$5)*LN(RAND())	=E8+D9	=C9-MATCH(E9,\$\$3:I8,1)	=MAXA(E9:I8)	=-(1/\$B\$6)*LN(RAND())	=G9+H9
10	Expected % idle time	=1-B5/B6	7	=-(1/\$B\$5)*LN(RAND())	=E9+D10	=C10-MATCH(E10,\$\$3:I9,1)	=MAXA(E10:I9)	=-(1/\$B\$6)*LN(RAND())	=G10+H10

	K	L	M	N	O	P	Q
1	IDLE	CUMULATIVE	AVERAGE	CUMULATIVE	AVERAGE	CUMULATIVE	AVERAGE
2	TIME	NO. IN QUEUE	NO. IN QUEUE	WAIT TIME	WAIT TIME	IDLE TIME	% IDLE TIME
3							
4	=G4-I3	=F4*E4	=L4/E4	=J4	=N4/C4	=K4	=P4/I4
5	=G5-I4	=F5*(E5-E4)+L4	=L5/E5	=N4+J5	=N5/C5	=P4+K5	=P5/I5
6	=G6-I5	=F6*(E6-E5)+L5	=L6/E6	=N5+J6	=N6/C6	=P5+K6	=P6/I6
7	=G7-I6	=F7*(E7-E6)+L6	=L7/E7	=N6+J7	=N7/C7	=P6+K7	=P7/I7
8	=G8-I7	=F8*(E8-E7)+L7	=L8/E8	=N7+J8	=N8/C8	=P7+K8	=P8/I8
9	=G9-I8	=F9*(E9-E8)+L8	=L9/E9	=N8+J9	=N9/C9	=P8+K9	=P9/I9
10	=G10-I9	=F10*(E10-E9)+L9	=L10/E10	=N9+J10	=N10/C10	=P9+K10	=P10/I10

Table A.5. Crystal Ball simulation results for single server queuing model



Forecast: Average number in queue

Edit Preferences View Run Help

Cell 1:503 **Statistics**

Statistic	Value
Trials	1,000
Mean	1.3233
Median	1.2440
Mode	---
Standard Deviation	0.4502
Variance	0.2026
Skewness	1.59
Kurtosis	7.69
Coeff. of Variability	0.34
Range Minimum	0.4547
Range Maximum	4.2016
Range Width	3.7469
Mean Std. Error	0.0142

Table A.6. Basic queuing model, "Data" sheet, Grossman

	A	B	C	D	E	F	G	H	I
1	Spreadsheet Simulation Queueing Engine:1 Server, Basic Model								
2									
3	Start Time	Close Time							
4	09:00	10:00							
5									
6									
7	Interarrival Time Probability Distribution								
8		Lower	Upper	Interarrival					
9		Bound	Bound	Time					
10		Probability		(min)					
11	0,45	0	0,45	1					
12	0,25	0,45	0,7	3					
13	0,1	0,7	0,8	5					
14	0,2	0,8	1	10					
15									
	Service Time Probability Distribution								
		Lower	Upper	Service					
		Bound	Bound	Time					
		Probability		(min)					
	0,3	0	0,3	3					
	0,35	0,3	0,65	6					
	0,35	0,65	1	9					

Table A.8. Basic queuing model, "Simulation" sheet formulas, Grossman

A		B	C	D
	Cust. #	Interarrival Time (min)	Arrival Time (hr:min)	Service Time (min)
7				
8				
9				
10				
11	start		=ComputationsIF11	
12	=ComputationsIA12	=IF(ISTEXT(C12),"",ComputationsIB12)	=IF(ComputationsIF12="closed",ComputationsIF11+ComputationsIC12,ComputationsIF12)	=IF(ISTEXT(A12),"",ComputationsID12)
13	=ComputationsIA13	=IF(ISTEXT(C13),"",ComputationsIB13)	=IF(ComputationsIF13="closed",ComputationsIF12+ComputationsIC13,ComputationsIF13)	=IF(ISTEXT(A13),"",ComputationsID13)
14	=ComputationsIA14	=IF(ISTEXT(C14),"",ComputationsIB14)	=IF(ComputationsIF14="closed",ComputationsIF13+ComputationsIC14,ComputationsIF14)	=IF(ISTEXT(A14),"",ComputationsID14)
15	=ComputationsIA15	=IF(ISTEXT(C15),"",ComputationsIB15)	=IF(ComputationsIF15="closed",ComputationsIF14+ComputationsIC15,ComputationsIF15)	=IF(ISTEXT(A15),"",ComputationsID15)

E		F	G	H
	Server #1	End (hr:min)	Wait Time (hr:min)	Total Time (hr:min)
7				
8				
9				
10				
11				
12	=IF(ComputationsI12=1,ComputationsIH12,"")	=IF(ISTEXT(\$E12),"",E12+ComputationsIE12)	=IF(ISTEXT(\$A12),"",ComputationsIF12+ComputationsIE12)	=IF(ISTEXT(A12),"",G12+ComputationsIE12)
13	=IF(ComputationsI13=1,ComputationsIH13,"")	=IF(ISTEXT(\$E13),"",E13+ComputationsIE13)	=IF(ISTEXT(\$A13),"",ComputationsIF13+ComputationsIE13)	=IF(ISTEXT(A13),"",G13+ComputationsIE13)
14	=IF(ComputationsI14=1,ComputationsIH14,"")	=IF(ISTEXT(\$E14),"",E14+ComputationsIE14)	=IF(ISTEXT(\$A14),"",ComputationsIF14+ComputationsIE14)	=IF(ISTEXT(A14),"",G14+ComputationsIE14)
15	=IF(ComputationsI15=1,ComputationsIH15,"")	=IF(ISTEXT(\$E15),"",E15+ComputationsIE15)	=IF(ISTEXT(\$A15),"",ComputationsIF15+ComputationsIE15)	=IF(ISTEXT(A15),"",G15+ComputationsIE15)

Table A.9. Basic queuing model, "Computation" sheet formulas, Grossman

A	B	C	D	E	F
Cust. #	Interarrival Time User Input (min)	Interarrival Time (serial)	Service Time User Input (min)	Service Time (serial)	Arrival Time (hr:min)
10					
11	start				=start_time
12	=IF(ISTEXT(F12),"closed",1)	=VLOOKUP(RAND0;arrival;3)=B12/1440	=VLOOKUP(RAND0;service;3)=D12/1440		=IF(ISTEXT(F11),"",IF(F11+C12>=close_time,"closed",F11+C12))
13	=IF(ISTEXT(F13),"closed",A12+1)	=VLOOKUP(RAND0;arrival;3)=B13/1440	=VLOOKUP(RAND0;service;3)=D13/1440		=IF(ISTEXT(F12),"",IF(F12+C13>=close_time,"closed",F12+C13))
14	=IF(ISTEXT(F14),"closed",A13+1)	=VLOOKUP(RAND0;arrival;3)=B14/1440	=VLOOKUP(RAND0;service;3)=D14/1440		=IF(ISTEXT(F13),"",IF(F13+C14>=close_time,"closed",F13+C14))
15	=IF(ISTEXT(F15),"closed",A14+1)	=VLOOKUP(RAND0;arrival;3)=B15/1440	=VLOOKUP(RAND0;service;3)=D15/1440		=IF(ISTEXT(F14),"",IF(F14+C15>=close_time,"closed",F14+C15))

G	H	I
Potential #1 Start Time (hr:min)	Actual Start Time (hr:min)	Next Server
6		
7		
8		
9		
10		
11		
12	=IF(ISTEXT(F12),"",MAX(SimulationIF\$11:F11;start_time,\$F12))=IF(ISTEXT(F12),"",MIN(G12;G12))=IF(ISTEXT(F12),"",MATCH(H12;G12;G12;0))	
13	=IF(ISTEXT(F13),"",MAX(SimulationIF\$11:F12;start_time,\$F13))=IF(ISTEXT(F13),"",MIN(G13;G13))=IF(ISTEXT(F13),"",MATCH(H13;G13;G13;0))	
14	=IF(ISTEXT(F14),"",MAX(SimulationIF\$11:F13;start_time,\$F14))=IF(ISTEXT(F14),"",MIN(G14;G14))=IF(ISTEXT(F14),"",MATCH(H14;G14;G14;0))	
15	=IF(ISTEXT(F15),"",MAX(SimulationIF\$11:F14;start_time,\$F15))=IF(ISTEXT(F15),"",MIN(G15;G15))=IF(ISTEXT(F15),"",MATCH(H15;G15;G15;0))	

Table A.10. Two server queuing model, Evans and Olson

	A	B	C	D	E	F	G	H	I	J
1	Dan's Carwash									
2	Two server model									
3					Mean arrival rate/min				0,250	Average
4					Mean service rate/min				0,333	Wait time
5										01:43
6	Customer	TBA	Arrival Time	Server	Start Time	Service Time	Customer	Completion Time	Server 2	Wait Time
7	1	7,450	7,450	1	7,450	2,695	10,144	10,144	0,000	0,000
8	2	1,446	8,896	2	8,896	6,827	15,723	10,144	15,723	0,000
9	3	4,836	13,732	1	13,732	2,293	16,025	16,025	15,723	0,000
10	4	4,356	18,088	2	18,088	0,940	19,028	16,025	19,028	0,000

	A	B	C	D	E	F	G	H	I	J
1	Dan's Carwash									
2	Two server model									
3					Mean arrival rate/min				0,25	Average
4					Mean service rate/min				0,333	Wait time
5										=SUM(J7:J106)/100
6	Customer	TBA	Arrival Time	Server	Start Time	Service Time	Customer	Completion Time	Server 2	Wait Time
7	1	=1*(1/\$\$1)*LN(RAND())	=B7	=1	=C7	=-1*(1/\$\$2)*LN(RAND())	=E7+F7	=F(D7=1;G7;0)	=F(D7=2;G7;0)	=E7-C7
8	=A7+1	=1*(1/\$\$1)*LN(RAND())	=C7+B8	=2	=C8	=-1*(1/\$\$2)*LN(RAND())	=E8+F8	=H7	=G8	=E8-C8
9	=A8+1	=1*(1/\$\$1)*LN(RAND())	=C8+B9	=IF(H8<B8;1;2)	=MAX(C9;MIN(H8;B8))	=-1*(1/\$\$2)*LN(RAND())	=E9+F9	=IF(D9=1;G9;H8)	=IF(D9=2;G9;H8)	=E9-C9
10	=A9+1	=1*(1/\$\$1)*LN(RAND())	=C9+B10	=IF(H9<B9;1;2)	=MAX(C10;MIN(H9;B9))	=-1*(1/\$\$2)*LN(RAND())	=E10+F10	=IF(D10=1;G10;H9)	=IF(D10=2;G10;H9)	=E10-C10

Table A.12. Three server queuing model, "Simulation" sheet formulas, Grossman

	E	F	G	H
	Server #1	Server #2	Server #3	Server #4
	Start (hr:min)	End (hr:min)	Start (hr:min)	End (hr:min)
7				
8				
9				
10				
11				
12	=IF(Computations\$K12=1;ComputationsJ12;"")	=IF(ISTEXT(\$E12),"",E12+Computations\$E12)	=IF(Computations\$K12=2;ComputationsJ12;"")	=IF(ISTEXT(\$G12),"",G12+Computations\$E12)
13	=IF(Computations\$K13=1;ComputationsJ13;"")	=IF(ISTEXT(\$E13),"",E13+Computations\$E13)	=IF(Computations\$K13=2;ComputationsJ13;"")	=IF(ISTEXT(\$G13),"",G13+Computations\$E13)
14	=IF(Computations\$K14=1;ComputationsJ14;"")	=IF(ISTEXT(\$E14),"",E14+Computations\$E14)	=IF(Computations\$K14=2;ComputationsJ14;"")	=IF(ISTEXT(\$G14),"",G14+Computations\$E14)
15	=IF(Computations\$K15=1;ComputationsJ15;"")	=IF(ISTEXT(\$E15),"",E15+Computations\$E15)	=IF(Computations\$K15=2;ComputationsJ15;"")	=IF(ISTEXT(\$G15),"",G15+Computations\$E15)
16	=IF(Computations\$K16=1;ComputationsJ16;"")	=IF(ISTEXT(\$E16),"",E16+Computations\$E16)	=IF(Computations\$K16=2;ComputationsJ16;"")	=IF(ISTEXT(\$G16),"",G16+Computations\$E16)

	I	J	K	L
	Server #3	Wait Time (hr:min)	Total Time (hr:min)	
	Start (hr:min)	End (hr:min)		
7				
8				
9				
10				
11				
12	=IF(Computations\$K12=3;ComputationsJ12;"")	=IF(ISTEXT(\$I12),"",I12+Computations\$E12)	=IF(ISTEXT(\$A12),"",ComputationsJ12-Computations\$E12)	=IF(ISTEXT(A12),"",K12+Computations\$E12)
13	=IF(Computations\$K13=3;ComputationsJ13;"")	=IF(ISTEXT(\$I13),"",I13+Computations\$E13)	=IF(ISTEXT(\$A13),"",ComputationsJ13-Computations\$E13)	=IF(ISTEXT(A13),"",K13+Computations\$E13)
14	=IF(Computations\$K14=3;ComputationsJ14;"")	=IF(ISTEXT(\$I14),"",I14+Computations\$E14)	=IF(ISTEXT(\$A14),"",ComputationsJ14-Computations\$E14)	=IF(ISTEXT(A14),"",K14+Computations\$E14)
15	=IF(Computations\$K15=3;ComputationsJ15;"")	=IF(ISTEXT(\$I15),"",I15+Computations\$E15)	=IF(ISTEXT(\$A15),"",ComputationsJ15-Computations\$E15)	=IF(ISTEXT(A15),"",K15+Computations\$E15)
16	=IF(Computations\$K16=3;ComputationsJ16;"")	=IF(ISTEXT(\$I16),"",I16+Computations\$E16)	=IF(ISTEXT(\$A16),"",ComputationsJ16-Computations\$E16)	=IF(ISTEXT(A16),"",K16+Computations\$E16)

Table A.13. Three server queuing model, "Calculation" sheet formulas, Grossman

G		H		I
	Potential #1 Start Time (hr:min)	Potential #2 Start Time (hr:min)	Potential #3 Start Time (hr:min)	
6				
7				
8				
9				
10				
11				
12	=IF(ISTEXT(F12),"",MAX(SimulationF\$11:F11,start_time,\$F1	=IF(ISTEXT(F12),"",MAX(SimulationH\$11:H11,start_time,\$F12))	=IF(ISTEXT(F12),"",MAX(SimulationJ\$11:J11,start_time,\$F12))	
13	=IF(ISTEXT(F13),"",MAX(SimulationF\$11:F12,start_time,\$F1	=IF(ISTEXT(F13),"",MAX(SimulationH\$11:H12,start_time,\$F13))	=IF(ISTEXT(F13),"",MAX(SimulationJ\$11:J12,start_time,\$F13))	
14	=IF(ISTEXT(F14),"",MAX(SimulationF\$11:F13,start_time,\$F1	=IF(ISTEXT(F14),"",MAX(SimulationH\$11:H13,start_time,\$F14))	=IF(ISTEXT(F14),"",MAX(SimulationJ\$11:J13,start_time,\$F14))	
15	=IF(ISTEXT(F15),"",MAX(SimulationF\$11:F14,start_time,\$F1	=IF(ISTEXT(F15),"",MAX(SimulationH\$11:H14,start_time,\$F15))	=IF(ISTEXT(F15),"",MAX(SimulationJ\$11:J14,start_time,\$F15))	
16	=IF(ISTEXT(F16),"",MAX(SimulationF\$11:F15,start_time,\$F1	=IF(ISTEXT(F16),"",MAX(SimulationH\$11:H15,start_time,\$F16))	=IF(ISTEXT(F16),"",MAX(SimulationJ\$11:J15,start_time,\$F16))	
17	=IF(ISTEXT(F17),"",MAX(SimulationF\$11:F16,start_time,\$F1	=IF(ISTEXT(F17),"",MAX(SimulationH\$11:H16,start_time,\$F17))	=IF(ISTEXT(F17),"",MAX(SimulationJ\$11:J16,start_time,\$F17))	
18	=IF(ISTEXT(F18),"",MAX(SimulationF\$11:F17,start_time,\$F1	=IF(ISTEXT(F18),"",MAX(SimulationH\$11:H17,start_time,\$F18))	=IF(ISTEXT(F18),"",MAX(SimulationJ\$11:J17,start_time,\$F18))	

	J	K
6		
7		
8		
9		
10		
11		
12	=IF(ISTEXT(F12),"",MIN(G12:H12))	=IF(ISTEXT(F12),"",MATCH(J12:G12:H12:0))
13	=IF(ISTEXT(F13),"",MIN(G13:H13))	=IF(ISTEXT(F13),"",MATCH(J13:G13:H13:0))
14	=IF(ISTEXT(F14),"",MIN(G14:H14))	=IF(ISTEXT(F14),"",MATCH(J14:G14:H14:0))
15	=IF(ISTEXT(F15),"",MIN(G15:H15))	=IF(ISTEXT(F15),"",MATCH(J15:G15:H15:0))
16	=IF(ISTEXT(F16),"",MIN(G16:H16))	=IF(ISTEXT(F16),"",MATCH(J16:G16:H16:0))
17	=IF(ISTEXT(F17),"",MIN(G17:H17))	=IF(ISTEXT(F17),"",MATCH(J17:G17:H17:0))
18	=IF(ISTEXT(F18),"",MIN(G18:H18))	=IF(ISTEXT(F18),"",MATCH(J18:G18:H18:0))

Table A.15. G/G/c queuing model formulas, Hora

A	B	C	D	E	F	G	H	I
			Spreadsheet Simulation of the G/G/c System Using Array Functions					
1				Customer Number	Rand 1	Interarrival Time	Arrival Time	Number in the System
2	Input Parameters			1	=RANDO	=(1/Arrival_rate)*LN(F2)	=G2	0
3				2	=RANDO	=(1/Arrival_rate)*LN(F3)	=G3+H2	=COUNTIF(\$M\$2:M2,"> "&H3)
4				3	=RANDO	=(1/Arrival_rate)*LN(F4)	=G4+H3	=COUNTIF(\$M\$2:M3,"> "&H4)
5	Arrival rate			4	=RANDO	=(1/Arrival_rate)*LN(F5)	=G5+H4	=COUNTIF(\$M\$2:M4,"> "&H5)
6	1			5	=RANDO	=(1/Arrival_rate)*LN(F6)	=G6+H5	=COUNTIF(\$M\$2:M5,"> "&H6)
7	Servers			6	=RANDO	=(1/Arrival_rate)*LN(F7)	=G7+H6	=COUNTIF(\$M\$2:M6,"> "&H7)
8	4			7	=RANDO	=(1/Arrival_rate)*LN(F8)	=G8+H7	=COUNTIF(\$M\$2:M7,"> "&H8)
9				8	=RANDO	=(1/Arrival_rate)*LN(F9)	=G9+H8	=COUNTIF(\$M\$2:M8,"> "&H9)
10	Prob			9	=RANDO	=(1/Arrival_rate)*LN(F10)	=G10+H9	=COUNTIF(\$M\$2:M9,"> "&H10)
11	0	0.5	2	10	=RANDO	=(1/Arrival_rate)*LN(F11)	=G11+H10	=COUNTIF(\$M\$2:M10,"> "&H11)
12	0.5	0.55	3	11	=RANDO	=(1/Arrival_rate)*LN(F12)	=G12+H11	=COUNTIF(\$M\$2:M11,"> "&H12)
13	0.55	0.65	4	12	=RANDO	=(1/Arrival_rate)*LN(F13)	=G13+H12	=COUNTIF(\$M\$2:M12,"> "&H13)
14	0.65	1	5	13	=RANDO	=(1/Arrival_rate)*LN(F14)	=G14+H13	=COUNTIF(\$M\$2:M13,"> "&H14)
15				14	=RANDO	=(1/Arrival_rate)*LN(F15)	=G15+H14	=COUNTIF(\$M\$2:M14,"> "&H15)
16				15	=RANDO	=(1/Arrival_rate)*LN(F16)	=G16+H15	=COUNTIF(\$M\$2:M15,"> "&H16)
17				16	=RANDO	=(1/Arrival_rate)*LN(F17)	=G17+H16	=COUNTIF(\$M\$2:M16,"> "&H17)
18				17	=RANDO	=(1/Arrival_rate)*LN(F18)	=G18+H17	=COUNTIF(\$M\$2:M17,"> "&H18)
19				18	=RANDO	=(1/Arrival_rate)*LN(F19)	=G19+H18	=COUNTIF(\$M\$2:M18,"> "&H19)
20	Wait Time			19	=RANDO	=(1/Arrival_rate)*LN(F20)	=G20+H19	=COUNTIF(\$M\$2:M19,"> "&H20)
21	Average	#VAL UEI	#VAL UEI	20	=RANDO	=(1/Arrival_rate)*LN(F21)	=G21+H20	=COUNTIF(\$M\$2:M20,"> "&H21)
22	Std Dev	#VAL UEI	#VAL UEI	21	=RANDO	=(1/Arrival_rate)*LN(F22)	=G22+H21	=COUNTIF(\$M\$2:M21,"> "&H22)
23	Estimates from Little's flow equations			22	=RANDO	=(1/Arrival_rate)*LN(F23)	=G23+H22	=COUNTIF(\$M\$2:M22,"> "&H23)
24				23	=RANDO	=(1/Arrival_rate)*LN(F24)	=G24+H23	=COUNTIF(\$M\$2:M23,"> "&H24)
25				24	=RANDO	=(1/Arrival_rate)*LN(F25)	=G25+H24	=COUNTIF(\$M\$2:M24,"> "&H25)
26				25	=RANDO	=(1/Arrival_rate)*LN(F26)	=G26+H25	=COUNTIF(\$M\$2:M25,"> "&H26)

Table A.15. (Continued)

	J	K	L	M	N	O	P	Q
1	Start Service							
2	=H2 =IF(I3>=Servers;PERCENTILEQ(IP(\$M\$2:M2>H3,\$M\$2:M2); (I3-Servers)/(I3-1));H3)	Rand 2 =RAND()	Service Time =VLOOKUP(K2,\$A\$11:\$C\$14,3)	End Service =L2+I2	Wait time =J2-H2	Total time =M2-H2	Queue Length at Arrival =MAX(Q0:I2,\$A\$8)	Servers Utilized at Arrival =MIN(I2,\$A\$8)
3	=IF(I4>=Servers;PERCENTILEQ(IP(\$M\$2:M3>H4,\$M\$2:M3); (I4-Servers)/(I4-1));H4)	=RAND()	=VLOOKUP(K3,\$A\$11:\$C\$14,3)	=L3+I3	=J3-H3	=M3-H3	=MAX(Q0:I3,\$A\$8)	=MIN(I3,\$A\$8)
4	=IF(I5>=Servers;PERCENTILEQ(IP(\$M\$2:M4>H5,\$M\$2:M4); (I5-Servers)/(I5-1));H5)	=RAND()	=VLOOKUP(K4,\$A\$11:\$C\$14,3)	=L4+I4	=J4-H4	=M4-H4	=MAX(Q0:I4,\$A\$8)	=MIN(I4,\$A\$8)
5	=IF(I6>=Servers;PERCENTILEQ(IP(\$M\$2:M5>H6,\$M\$2:M5); (I6-Servers)/(I6-1));H6)	=RAND()	=VLOOKUP(K5,\$A\$11:\$C\$14,3)	=L5+I5	=J5-H5	=M5-H5	=MAX(Q0:I5,\$A\$8)	=MIN(I5,\$A\$8)
6	=IF(I7>=Servers;PERCENTILEQ(IP(\$M\$2:M6>H7,\$M\$2:M6); (I7-Servers)/(I7-1));H7)	=RAND()	=VLOOKUP(K6,\$A\$11:\$C\$14,3)	=L6+I6	=J6-H6	=M6-H6	=MAX(Q0:I6,\$A\$8)	=MIN(I6,\$A\$8)
7	=IF(I8>=Servers;PERCENTILEQ(IP(\$M\$2:M7>H8,\$M\$2:M7); (I8-Servers)/(I8-1));H8)	=RAND()	=VLOOKUP(K7,\$A\$11:\$C\$14,3)	=L7+I7	=J7-H7	=M7-H7	=MAX(Q0:I7,\$A\$8)	=MIN(I7,\$A\$8)
8	=IF(I9>=Servers;PERCENTILEQ(IP(\$M\$2:M8>H9,\$M\$2:M8); (I9-Servers)/(I9-1));H9)	=RAND()	=VLOOKUP(K8,\$A\$11:\$C\$14,3)	=L8+I8	=J8-H8	=M8-H8	=MAX(Q0:I8,\$A\$8)	=MIN(I8,\$A\$8)
9		=RAND()	=VLOOKUP(K9,\$A\$11:\$C\$14,3)	=L9+I9	=J9-H9	=M9-H9	=MAX(Q0:I9,\$A\$8)	=MIN(I9,\$A\$8)

Table A.16. M/G/c queuing model, "data" sheet

	A	B	C	D	E	F
1	5850182	How many runs?	Start Time:	09:00:00	time period	
2	339294735	1	Close Time:	17:00:00	480	
3	1406484740		Mean Interarrival Time:	25	alfa	10
4	924206133		Mean Service Time:	20	beta	0.5
5	2189447026		Number of Servers:	2		
6	1413113931					
7	3529265936					
8	2000049105					
9	2358491998					
10	3459174471					
11	1280795612					
12	4125696813					
	data	Simulation	Simulation Summary	waiting system	waiting queue	

Table A.17. M/G/c queuing model, "Spreadsheet" sheet

PROCESS INTERACTION QUEUEING SYSTEM SPREADSHEET, M/G/c										
A	B	C	D	E	F	G	H	I		
1	PROCESS INTERACTION QUEUEING SYSTEM SPREADSHEET, M/G/c									
2										
3	Customer	Interarrival	Arrival	Number of	Start	Service	End	Wait	Total	
4		Time	Time	customers	Service	Time	Service	Time	Time	
5	Number	(min)	(min)	in the system	(min)	(min)	(min)	(min)	(min)	
6	1	0,0004	0,0004	0	0,0004	1,5738	1,5742	0,0000	1,5738	
7	2	2,9342	2,9346	0	2,9346	0,0000	2,9346	0,0000	0,0000	
8	3	2,9342	5,8687	0	5,8687	158,0105	163,8792	0,0000	158,0105	
9	4	36,0008	41,8696	1	41,8696	26,8568	68,7264	0,0000	26,8568	
10	5	4,9606	46,8302	2	68,7264	25,3800	94,1064	21,8962	47,2762	
11	6	58,2947	105,1249	1	105,1249	2,3240	107,4488	0,0000	2,3240	
12	7	55,1456	160,2705	1	160,2705	1,6201	161,8906	0,0000	1,6201	
13	8	12,2352	172,5057	0	172,5057	6,2223	178,7281	0,0000	6,2223	
14	9	16,1539	188,6596	0	188,6596	0,0576	188,7172	0,0000	0,0576	
15	10	53,4677	242,1273	0	242,1273	24,6371	266,7643	0,0000	24,6371	
16	11	5,1381	247,2654	1	247,2654	0,2447	247,5100	0,0000	0,2447	
17	12	75,4581	272,7235	0	272,7235	0,4967	273,2202	0,0000	0,4967	
18	data	Simulation	Simulation Summary	Simulation	waiting system	waiting queue				

Table A.18. M/G/c queuing model, "Spreadsheet" sheet, data for performance measures

Q		R		S	T	U	V	W
Row Number		21						
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
data		Simulation		Simulation Summary		waiting system		waiting queue

Table A.20. M/G/c queuing model – “Simulation” worksheet

A		B		C		D		E	
Max no.in the system		Average no.in the system ,L		Max queue length		Average queue length, Lq		Max time spent in system	
1	4	0,804339847		2		0,049678696		80,534091	
2	2	0,547171161		0		0		376,1893611	
3	2	0,360613983		0		0		92,27434041	
4	2	0,663627153		0		0		149,922491	
5	3	0,750582065		1		0,042034367		195,0679198	
6	5	1,030545955		3		0,179107525		125,1314818	
7	2	0,399479096		0		0		64,92466492	
Simulation		Simulation_Summary		waiting_system		waiting_queue		Servers utilization	
Spreadsheet		Simulation		Simulation		Simulation		Simulation	

F		G		H		I	
Average time spent in the system, W		Max waiting time in queue		Average waiting time in queue, Wq		Servers Utilization	
1	18,97684875	17,76754542		0,935133969		30,4924%	
2	49,05591834	0,714150288		0,039675016		51,3127%	
3	75,71750752	170,8031997		33,14829246		52,1266%	
4	8,866317748	1,666610394		0,087716337		17,3743%	
5	8,844071267	0		0		14,7401%	
6	19,07661202	4,963179222		0,291951719		23,9049%	
7	38,94109797	17,07276213		2,237143769		43,1022%	
Simulation		Simulation_Summary		waiting_system		Servers utilization	
Spreadsheet		Simulation		Simulation		Simulation	

Table A.22. M/G/c queuing model, "Simulation_cust" worksheet

	A	B	C	D
1	No. in the system	probability	sum:	1,000
2	0	0,503		
3	1	0,326		
4	2	0,081		
5	3	0,039		
6	4	0,021		
7	5	0,011		
8	6	0,007		
9	7	0,004		
10	8	0,003		
11	9	0,001		
12	10	0,001		
13	11	0,000		
14	12	0,000		
15	13	0,000		
16	14	0,000		

Simulation_cust / Simulation_que / Custoi

Table A.23. M/G/c queuing model, “Simulation_que” worksheet

	A	B	C	D
1	QUEUE LENGTH	probability	sum:	1,000
2	0	0,910		
3	1	0,039		
4	2	0,021		
5	3	0,011		
6	4	0,007		
7	5	0,004		
8	6	0,003		
9	7	0,001		
10	8	0,001		
11	9	0,000		
12	10	0,000		
13	11	0,000		
14	12	0,000		
15	13	0,000		
16	14	0,000		

Simulation_custSimulation_queCustoi

Table A.24. M/G/c queuing model with balking property, "Spreadsheet" sheet

A	B	C	D	E	F	G	H	I	J	K
1	PROCESS INTERACTION QUEUING SYSTEM SPREADSHEET, M/G/c, Balk									
2										
3	Customer	Interarrival	Arrival	Number of						
4	Number	Time (min)	Time (min)	customers	Balk?	Start	Service	End	Wait	Total
5				in the system		Service	Time (min)	Service	Time (min)	Time (min)
6	1	13,2451	13,2451	0	0	13,2451	0,0791	13,3242	0,0000	0,0791
7	2	2,6564	15,9015	0	0	15,9015	6,5623	22,4638	0,0000	6,5623
8	3	28,2803	44,1818	0	0	44,1818	9,7570	53,9388	0,0000	9,7570
9	4	14,2972	58,4790	0	0	58,4790	3,9413	62,4203	0,0000	3,9413
10	5	14,9801	73,4591	0	0	73,4591	2,6583	76,1174	0,0000	2,6583
11	6	69,1869	142,6460	0	0	142,6460	8,4728	151,1188	0,0000	8,4728
12	7	1,4224	144,0684	1	0	144,0684	33,8274	177,8958	0,0000	33,8274
13	8	2,0938	146,1622	2	0	151,1188	0,1562	151,2749	4,9566	5,1127
14	9	3,5494	149,7116	3	1	Balk				
15	10	65,1487	214,8604	0	0	214,8604	2,9156	217,7760	0,0000	2,9156
16	11	14,0943	228,9547	0	0	228,9547	27,4465	256,4012	0,0000	27,4465
17	12	27,7068	256,6615	0	0	256,6615	41,2990	297,9605	0,0000	41,2990
18	13	9,5970	266,2585	1	0	266,2585	39,7465	306,0050	0,0000	39,7465
19										
20										
		Simulation			Simulation Summary	waiting system	waiting queue	Servers busy	Balk	Simulation
		Spreadsheet								

Table A.25. M/G/c queuing model with balking and renegeing properties, "Spreadsheet" sheet

	A	B	C	D	E	F	G	H	I	J	K	L
1	PROCESS INTERACTION QUEUEING SYSTEM SPREADSHEET, M/G/c, Balk and Renege											
2												
3												
4	Customer	Interarrival	Arrival	Number of	Number of							
5	Number	Time (min)	Time (min)	customers	customers	Balk?	Renege?	Start	Service	End	Wait	Total
6	1	50,93238	50,93238	0	0			50,93238	0,01109	50,94347	0,00000	0,01109
7	2	48,20315	99,13553	0	0			99,13553	15,46609	114,60162	0,00000	15,46609
8	3	43,90576	143,04129	0	0			143,04129	24,29633	167,33761	0,00000	24,29633
9	4	20,28829	163,32958	1	1			163,32958	7,09915	170,42873	0,00000	7,09915
10	5	16,53267	179,86225	0	0			179,86225	83,65613	263,51838	0,00000	83,65613
11	6	9,13747	188,99972	1	1			188,99972	1,73509	190,73481	0,00000	1,73509
12	7	7,72332	196,72304	1	1			196,72304	16,06207	212,78510	0,00000	16,06207
13	8	0,13375	196,85679	2	2		Renege	212,78510				
14	9	9,13644	205,99323	2	2		Renege	212,78510				
15	10	1,77082	207,76405	2	2		Renege	212,78510				
16	11	14,30511	222,06915	1	1			222,06915	9,21189	231,28105	0,00000	9,21189
17	12	7,56125	229,63040	2	2			231,28105	21,86594	253,14699	1,65064	23,51659
18	13	1,15934	230,78975	3	3	Balk						
19	14	46,40208	277,19182	0	0			277,19182	1,30281	278,49463	0,00000	1,30281
20	15	0,95371	278,14554	1	1			278,14554	12,45079	290,59632	0,00000	12,45079
21	16	7,15335	285,29889	1	1			285,29889	0,01044	285,30932	0,00000	0,01044
22	17	?? 28486	307,58375	0	0			307,58375	7,96777	315,55152	0,00000	7,96777
14	18	19	20	21	22	23	24	25	26	27	28	29
		Spreadsheet	Simulation	Simulation Summary	waiting system	waiting queue	Servers busy	Balk	Renege			

Table A.26. (Continued)

	U	V	W
1			
2			
3	Total time	Wait time	Total time
4	at second stage	at both stages	at both stages
5	(true line)	(min)	(min)
6	0,123197751	0	0,123209325
7	0,208563117	0	5,475282904
8	80,76440639	0	82,24775294
9	56,63225994	55,46339473	67,43252727
10			
11			
12	11,22784921	8,600002988	31,54471136
13			
14	0,340782877	0,202412373	5,270114845
15	384,5293687	0	384,5373892
16	499,3684656	362,0783506	509,2234579
17	490,5278348	488,7630775	492,5513109
18	425,3681636	419,8524847	425,8780333
19	420,2253732	419,04548	426,4939176
20	317,0956631	315,923833	390,4222691
21			
22	317,3132122	306,5097502	329,4735198
23	294,2872646	286,9904436	297,6551564
24	data \ Spreadsheet / Simulation \ SI		

Table A.28. Simple_network queuing models with M/G/c queue at serial servers with balking and reneging properties and servers break time, “Spreadsheet” sheet

	H	I	J	K
1				
2				
3	Start	Actual	Service	End
4	Service	Start Service	Time	Service
5	(min)	(min)	(min)	(min)
6	6,4652182	6,4652182	0,00019	6,4654034
7	8,9120386	8,9120386	0,77287	9,684913352
8	39,169959	39,169959	8,06920	47,2391589
9	49,412831	49,412831	4,88627	54,29910512
10	57,293729	57,293729	13,12556	70,41929308
11	60,006502	60,006502	1,08976	61,09626011
12	73,897634	73,897634	1,99851	75,89614702
13	173,46508	173,46508	3,16019	176,6252706
14	226,16716	240	0,19489	226,3620548
15	230,24379	240	5,01374	235,2575338
16	243,77322	243,77322	1,12290	244,8961211
17	261,34671	261,34671	0,18255	261,5292553
18	262,05396	262,05396	61,87988	323,9338303
19	283,12821	283,12821	6,01547	289,1436824
20	307,35336	307,35336	7,04796	314,4013201
21	314,40132	314,40132		
	data	Spreadsheet	Simulation	Simulati

Table A.29. Number of arriving customers day by day at June, July and August

		JUNE																			
Between		3rd	4th	5th	6th	7th	10th	11th	12th	13th	14th	17th	18th	19th	20th	21th	24th	25th	26th	27th	28th
08:30	09:00	0	0	1	0	0	0	0	3	0	0	0	0	1	0	1	0	0	0	1	0
09:00	09:30	15	8	2	7	13	9	9	7	7	3	15	9	19	7	11	16	9	9	10	8
09:30	10:00	13	4	15	7	3	9	12	6	9	11	10	8	11	13	8	20	8	13	12	6
10:00	10:30	14	1	8	9	3	6	12	7	8	6	9	4	2	9	8	10	3	12	5	14
10:30	11:00	18	7	6	5	8	14	9	12	6	9	7	12	11	12	9	15	10	10	12	6
11:00	11:30	12	5	7	8	7	9	6	7	2	9	15	4	4	4	8	14	7	6	11	8
11:30	12:00	7	8	5	6	2	2	8	1	6	7	9	7	9	4	2	12	10	3	10	5
12:00	12:30	0	0	0	2	0	0	1	0	0	0	1	0	1	1	1	6	1	0	2	1
12:30	13:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13:00	13:30	9	9	6	6	8	8	5	13	12	5	8	9	9	7	11	13	6	8	7	6
13:30	14:00	3	9	4	4	2	8	7	7	7	3	8	2	6	6	9	9	10	0	6	7
14:00	14:30	4	6	4	0	1	5	8	4	4	3	5	5	9	4	2	11	8	2	9	10
14:30	15:00	3	6	2	1	0	1	0	2	5	3	4	2	9	3	2	6	6	1	9	2
15:00	15:30	4	1	0	2	0	3	0	2	2	2	9	3	5	0	4	2	3	1	10	3
15:30	16:00	3	3	1	4	0	2	2	4	4	3	0	0	4	1	0	1	2	1	0	2
16:00	16:30	0	1	0	0	0	2	0	0	0	0	0	0	0	2	0	1	1	0	1	4
Total Sum		105	68	61	61	47	78	79	75	72	64	100	65	100	73	76	136	84	66	105	82

Table A.29. (Continued)

		JULY																								
Between		1st	2nd	3rd	4th	5th	8th	9th	10th	11th	12th	15th	16th	17th	18th	19th	22th	23th	24th	25th	26th	29th	30th	31th		
	08:30	09:00	2	5	4	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	09:00	09:30	13	14	9	8	11	10	10	14	9	15	19	19	12	16	22	14	11	24	13	16	22	19	12	
	09:30	10:00	12	10	9	11	5	15	10	5	15	11	21	12	5	6	13	20	27	24	20	12	23	24	8	
	10:00	10:30	9	7	3	14	6	12	10	15	7	11	22	12	13	7	7	20	17	12	10	11	26	10	15	
	10:30	11:00	16	7	10	5	9	10	12	8	3	12	18	13	10	8	11	42	8	10	16	15	21	15	11	
	11:00	11:30	17	10	5	8	7	9	12	4	7	8	16	8	4	13	20	27	21	10	11	10	22	17	11	
	11:30	12:00	7	8	9	4	6	18	11	8	7	6	15	11	6	8	8	26	12	15	10	9	20	14	12	
	12:00	12:30	5	1	0	1	1	5	0	1	0	0	7	5	1	1	0	22	7	8	0	0	14	5	6	
	12:30	13:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	2	1	0		
	13:00	13:30	14	10	10	11	11	15	17	11	10	16	17	16	10	17	13	15	13	15	16	16	20	15	11	
	13:30	14:00	14	10	12	10	4	15	8	11	6	6	18	8	4	5	7	13	18	9	18	13	20	20	12	
	14:00	14:30	7	6	7	4	0	18	12	6	13	6	12	11	11	9	11	18	18	7	8	12	17	18	11	
	14:30	15:00	9	5	10	7	4	6	8	14	13	5	8	10	10	6	4	10	11	7	10	10	16	14	11	
	15:00	15:30	5	8	4	5	4	6	7	5	3	4	7	8	6	4	3	14	9	9	10	9	11	17	8	
	15:30	16:00	2	6	7	2	4	11	4	2	4	3	11	4	6	3	4	11	10	6	10	6	8	9	3	
	16:00	16:30	1	2	4	0	1	2	1	3	1	0	2	0	6	0	4	0	8	2	3	2	2	3	2	
Total Sum		133	109	103	95	74	152	122	107	98	103	193	137	104	103	127	254	190	158	155	141	244	201	133		

Table A.29. (Continued)

		AUGUST																					
Between	1st	2nd	5th	6th	7th	8th	9th	12th	13th	14th	15 th	16th	19th	20th	21th	22th	23th	26th	27th	28th	29th		
08:30	09:00	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
09:00	09:30	11	20	7	21	18	21	20	21	17	18	17	19	22	12	14	15	24	23	21	13	12	
09:30	10:00	22	15	28	18	20	22	15	20	13	25	16	13	11	10	10	17	19	20	14	16	16	
10:00	10:30	12	16	23	15	11	20	13	19	15	13	16	15	17	14	20	9	15	21	20	13	19	
10:30	11:00	15	14	10	11	12	14	16	20	12	15	18	16	26	14	20	16	18	16	20	18	17	
11:00	11:30	14	13	19	26	15	8	15	15	13	9	7	14	19	12	17	16	15	28	25	14	15	
11:30	12:00	16	11	15	8	15	11	20	24	13	16	21	12	15	16	15	12	10	23	13	15	12	
12:00	12:30	0	6	19	10	2	2	15	16	4	9	2	10	25	6	0	3	3	5	0	3	8	
12:30	13:00	0	0	8	0	0	0	0	8	0	1	0	0	3	0	0	0	0	0	0	0	0	
13:00	13:30	9	17	8	15	10	15	9	10	17	11	16	15	11	19	11	19	16	21	15	12	20	
13:30	14:00	14	14	12	21	18	15	16	14	20	18	15	17	19	15	21	10	18	18	17	20	17	
14:00	14:30	19	8	12	18	18	10	12	12	14	10	8	7	12	23	21	9	13	19	13	14	14	
14:30	15:00	12	8	14	12	9	11	10	17	10	17	10	9	22	16	10	12	9	12	9	14	11	
15:00	15:30	11	11	17	11	9	5	5	11	6	12	11	6	17	11	11	6	11	11	11	14	6	
15:30	16:00	8	9	11	10	11	9	6	9	8	4	5	5	12	10	4	7	10	4	14	6	7	
16:00	16:30	5	3	8	2	0	0	1	1	0	2	2	0	2	4	4	2	1	5	1	3	1	
Total Sum		169	165	211	198	168	163	173	217	162	181	164	158	233	182	178	153	182	226	193	175	175	

Table A.30. Mean number of arriving customers in time periods at August

Between		Average	
08:30	09:00	0,0952	(=AVERAGE(C2:W2))
09:00	09:30	17,4286	(=AVERAGE(C3:W3))
09:30	10:00	17,1429	(=AVERAGE(C4:W4))
10:00	10:30	16,0000	(=AVERAGE(C5:W5))
10:30	11:00	16,0952	(=AVERAGE(C6:W6))
11:00	11:30	15,6667	(=AVERAGE(C7:W7))
11:30	12:00	7,0476	(=AVERAGE(C8:W8))
12:00	12:30	0,9524	(=AVERAGE(C9:W9))
12:30	13:00	0,9048	(=AVERAGE(C10:W10))
13:00	13:30	14,09524	(=AVERAGE(C11:W11))
13:30	14:00	16,61905	(=AVERAGE(C12:W12))
14:00	14:30	13,61905	(=AVERAGE(C13:W13))
14:30	15:00	12,09524	(=AVERAGE(C14:W14))
15:00	15:30	10,14286	(=AVERAGE(C15:W15))
15:30	16:00	8,04762	(=AVERAGE(C16:W16))
16:00	16:30	2,23810	(=AVERAGE(C17:W17))

Table A.31. Average number of arriving customers in time periods at August

Between		Average
08:30	09:00	0,09524
09:00	09:30	17,42857
09:30	10:00	17,14286
10:00	10:30	16,00000
10:30	11:00	16,09524
11:00	11:30	15,66667
11:30	12:00	14,90476
SUM		97,33333 =AVERAGE(C9:W9)
12:00	12:30	7,04762
12:30	13:00	0,95238
SUM		8,00000 =AVERAGE(C12:W12)
13:00	13:30	14,09524
13:30	14:00	16,61905
14:00	14:30	13,61905
14:30	15:00	12,09524
SUM		56,42857 =AVERAGE(C17:W17)
15:00	15:30	10,14286
15:30	16:00	8,04762
16:00	16:30	2,23810
SUM		20,42857 =AVERAGE(C21:W21)

Table A.32. Average number of arriving customers in time period 09:00-12:00

	A	B	C	D
	Number of arriving customers up to 12:00	Mean		
1				
2	96	97,3	=AVERAGE(A:A)	
3	99			
4	103			
5	100			
6	96			
7	86			
8	100			
9	97			
10	109			
11	94			
12	93			
13	95			
14	111			
« ◀ ▶ » ArrivingControl / Validation / Service_Chart / Servi				

Table A.35. Data for average waiting time of customers in the queue

	OPEN AN ACCOUNT (A)	EXPRESS TRANSACTION DESK (B)	PAYMENT AND OTHER OPERATIONS (C)	INHERITANCE OPERATIONS (D)
AUGUST 2002	Average Waiting Time	Average Waiting Time	Average Waiting Time	Average Waiting Time
1	0:10:46		0:01:48	0:33:46
2	0:03:00		0:01:02	0:54:10
5	0:08:30	0:10:44	0:17:44	0:17:10
6	0:26:24	0:20:46	0:11:20	0:06:06
7	0:05:42	0:13:32	0:04:44	0:15:34
8	0:13:08	0:12:06	0:05:44	
9	0:17:54	0:09:42	0:19:36	0:00:00
12	0:56:54	0:30:54	1:01:28	
13	0:08:18	0:08:56	0:22:12	
14	0:18:34	0:11:02	0:27:58	0:16:30
15	0:10:28	0:01:14	0:16:34	0:34:28
16	0:11:54	0:25:38	0:25:44	0:05:22
19	0:23:46	0:13:08	0:15:50	0:05:28
20	0:17:28	0:00:26	0:05:34	0:15:28
21	0:09:28		0:04:10	0:00:16
22	0:10:54		0:06:52	0:15:46
23	0:19:44	0:09:58	0:20:04	0:33:00
26	0:22:16	0:07:46	0:34:14	0:17:28
27	0:18:50	0:05:58	0:13:22	0:25:22
28	0:04:46	0:00:00	0:11:10	0:12:24
29	0:24:58	0:12:16	0:08:44	0:07:12
Customers average waiting time at floor 1 (A and B)				14,1526
Customers average waiting time at floor 2 (C and D)				16,7026

Table A.36. Simulation results after 1000 runs at "Simulation_Summary" Excel worksheet

A		B	C	D	E	F	G
		Mean	Standard Deviation	Min	Max	%95 Confidence Interval Lower	%95 Confidence Interval Upper
1							
2							
3	FLOOR 1 - TRANSACTIONS "A" AND "B"						
4	Max time spent in system	68,5933	24,3376	27,2255	171,5079	67,0849	70,1018
5	Average time spent in the system	32,3258	11,6318	18,0747	96,5267	31,6049	33,0468
6	Max waiting time in queue	49,0416	24,2317	6,8558	151,1361	47,5397	50,5435
7	Average waiting time in queue	14,8113	11,5424	0,6695	78,4304	14,0959	15,5267
8	Server 1 Utilization	78,3269%	8,5963%	52,5972%	98,0586%	77,7941%	78,8597%
9	Server 2 Utilization	77,2026%	8,9338%	48,6697%	96,8253%	76,6488%	77,7563%
10	Server 3 Utilization	77,8541%	8,1356%	52,8420%	96,8208%	77,3499%	78,3584%
11	Server 4 Utilization	76,7913%	8,2564%	52,9711%	96,7949%	76,2796%	77,3030%
12	FLOOR 2 - TRANSACTIONS "C" AND "D"						
13	Max time spent in system	102,6216	30,7856	33,5046	213,1384	100,7135	104,5297
14	Average time spent in the system	40,6746	12,1038	23,4060	94,3332	39,9244	41,4248
15	Max waiting time in queue	48,0138	21,6620	8,5858	139,9412	46,6711	49,3564
16	Average waiting time in queue	16,5129	11,8397	0,7421	66,9411	15,7791	17,2467
17	Server 1 Utilization	86,0963%	8,2495%	59,5285%	99,9624%	85,5850%	86,6077%
18	Server 2 Utilization	85,5737%	8,2192%	58,4107%	99,9037%	85,0643%	86,0831%
19	Server 3 Utilization	85,0544%	8,1630%	58,6515%	99,5590%	84,5484%	85,5603%
20	Server 4 Utilization	84,3712%	8,2410%	56,2533%	99,3350%	83,8604%	84,8819%
21	Server 5 Utilization	83,6636%	8,3661%	52,7946%	99,1167%	83,1451%	84,1821%
22	Server 6 Utilization	83,1978%	8,3445%	54,5683%	98,8162%	82,6806%	83,7150%
23	SECOND STAGE - TRANSACTIONS "A" AND "C"						
24	Max time spent in system	51,1130	11,8117	18,9212	98,7090	50,3809	51,8451
25	Average time spent in the system	24,3138	9,6449	7,8399	58,8243	23,7160	24,9116
26	Max waiting time in queue	45,9772	11,7426	13,0728	92,7496	45,2494	46,7051
27	Average waiting time in queue	19,3130	9,6284	2,8584	53,7900	18,7162	19,9097
28	Server 1 Utilization	85,4505%	5,0464%	65,3660%	92,3219%	85,1377%	85,7633%
29	Server 2 Utilization	84,8454%	5,0450%	65,4116%	91,7050%	84,5327%	85,1581%
30	WHOLE SYSTEM						
31	Average time spent in the whole system	59,4365	15,2066	29,7842	110,4689	58,4940	60,3790
32	Average waiting time in queue in the whole system	34,0052	15,2256	4,7402	85,3978	33,0615	34,9489
33							
Simulation_Summary		Simulation	ArrivingControl	Validation	Service Chart	ServiceTime	WaitingTime

Table A.37. Simulation results after 1000 runs at "Simulation_Summary" Excel worksheet formulas

A	B	C	D	E
	Mean	Standard Deviation	Min	Max
1				
2				
3	FLOOR 1 - TRANSACTIONS "A" AND "B"			
4	Max time spent in system	=AVERAGE(Simulation!A:A)	=MIN(Simulation!A:A)	=MAX(Simulation!A:A)
5	Average time spent in the system	=AVERAGE(Simulation!B:B)	=MIN(Simulation!B:B)	=MAX(Simulation!B:B)
6	Max waiting time in queue	=AVERAGE(Simulation!C:C)	=MIN(Simulation!C:C)	=MAX(Simulation!C:C)
7	Average waiting time in queue	=AVERAGE(Simulation!D:D)	=MIN(Simulation!D:D)	=MAX(Simulation!D:D)
8	Server 1 Utilization	=AVERAGE(Simulation!E:E)	=MIN(Simulation!E:E)	=MAX(Simulation!E:E)
9	Server 2 Utilization	=AVERAGE(Simulation!F:F)	=MIN(Simulation!F:F)	=MAX(Simulation!F:F)
10	Server 3 Utilization	=AVERAGE(Simulation!G:G)	=MIN(Simulation!G:G)	=MAX(Simulation!G:G)
11	Server 4 Utilization	=AVERAGE(Simulation!H:H)	=MIN(Simulation!H:H)	=MAX(Simulation!H:H)
12	FLOOR 2 - TRANSACTIONS "C" AND "D"			
13	Max time spent in system	=AVERAGE(Simulation!I:I)	=MIN(Simulation!I:I)	=MAX(Simulation!I:I)
14	Average time spent in the system	=AVERAGE(Simulation!J:J)	=MIN(Simulation!J:J)	=MAX(Simulation!J:J)
15	Max waiting time in queue	=AVERAGE(Simulation!K:K)	=MIN(Simulation!K:K)	=MAX(Simulation!K:K)
16	Average waiting time in queue	=AVERAGE(Simulation!L:L)	=MIN(Simulation!L:L)	=MAX(Simulation!L:L)
17	Server 1 Utilization	=AVERAGE(Simulation!M:M)	=MIN(Simulation!M:M)	=MAX(Simulation!M:M)
18	Server 2 Utilization	=AVERAGE(Simulation!N:N)	=MIN(Simulation!N:N)	=MAX(Simulation!N:N)
19	Server 3 Utilization	=AVERAGE(Simulation!O:O)	=MIN(Simulation!O:O)	=MAX(Simulation!O:O)
20	Server 4 Utilization	=AVERAGE(Simulation!P:P)	=MIN(Simulation!P:P)	=MAX(Simulation!P:P)
21	Server 5 Utilization	=AVERAGE(Simulation!Q:Q)	=MIN(Simulation!Q:Q)	=MAX(Simulation!Q:Q)
22	Server 6 Utilization	=AVERAGE(Simulation!R:R)	=MIN(Simulation!R:R)	=MAX(Simulation!R:R)
23	SECOND STAGE - TRANSACTIONS "A" AND "C"			
24	Max time spent in system	=AVERAGE(Simulation!S:S)	=MIN(Simulation!S:S)	=MAX(Simulation!S:S)
25	Average time spent in the system	=AVERAGE(Simulation!T:T)	=MIN(Simulation!T:T)	=MAX(Simulation!T:T)
26	Max waiting time in queue	=AVERAGE(Simulation!U:U)	=MIN(Simulation!U:U)	=MAX(Simulation!U:U)
27	Average waiting time in queue	=AVERAGE(Simulation!V:V)	=MIN(Simulation!V:V)	=MAX(Simulation!V:V)
28	Server 1 Utilization	=AVERAGE(Simulation!W:W)	=MIN(Simulation!W:W)	=MAX(Simulation!W:W)
29	Server 2 Utilization	=AVERAGE(Simulation!X:X)	=MIN(Simulation!X:X)	=MAX(Simulation!X:X)
30	WHOLE SYSTEM			
31	Average time spent in the whole system	=AVERAGE(Simulation!Y:Y)	=MIN(Simulation!Y:Y)	=MAX(Simulation!Y:Y)
32	Average waiting time in queue in the whole system	=AVERAGE(Simulation!Z:Z)	=MIN(Simulation!Z:Z)	=MAX(Simulation!Z:Z)
33				
data / Spreadsheet / Simulation / Simulation_Summary / ArrivingControl / Validation / Service Chart / Service				

Table A.37. (Continued)

F		G	
Lower		Upper	
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
data		Simulation	Simulation Summary
Spreadsheet		ArrivingControl	Validation

Table A.39. Simulation results after 1000 runs at "Simulation_Summary" Excel worksheet, Case 1

A		B	C	D	E	F	G
1		Mean	Standard Deviation	Min	Max	%95 Confidence Interval	
2						Lower	Upper
3	FLOOR 1 - TRANSACTIONS "A" AND "B"						
4	Max time spent in system	133,99691	39,94692775	40,04605022	286,4038787	131,5209731	136,4728456
5	Average time spent in the system	77,48791	27,76664794	21,90687724	170,2626853	75,76691074	79,20890008
6	Max waiting time in queue	114,56220	39,71917061	18,80101281	273,4506552	112,1003776	117,024017
7	Average waiting time in queue	59,94642	27,69711558	4,336307625	152,5135382	58,22973113	61,66310115
8	Server 1 Utilization	93,6019%	3,8849%	67,6042%	99,1642%	93,3611%	93,8427%
9	Server 2 Utilization	92,8772%	3,9821%	71,7444%	98,5087%	92,6304%	93,1240%
10	Server 3 Utilization	92,2570%	3,8248%	68,2122%	98,0021%	92,0199%	92,4941%
11	Server 4 Utilization	0,0000%	0,0000%	0,0000%	0,0000%	0,0000%	0,0000%
12	FLOOR 2 - TRANSACTIONS "C" AND "D"						
13	Max time spent in system	101,33313	30,39760147	32,92628525	241,3634529	99,44906728	103,217193
14	Average time spent in the system	40,20736	12,12053257	23,1500033	129,2455325	39,45612522	40,958602
15	Max waiting time in queue	47,69162	21,79297527	7,973838922	171,4166743	46,34087658	49,04236178
16	Average waiting time in queue	16,07993	11,83099447	0,527101004	102,8146493	15,34668404	16,81322567
17	Server 1 Utilization	85,9748%	8,0138%	58,1929%	99,9963%	85,4781%	86,4715%
18	Server 2 Utilization	85,4010%	8,0756%	55,3655%	99,8072%	84,9005%	85,9015%
19	Server 3 Utilization	84,7044%	8,1079%	59,7102%	99,6109%	84,2018%	85,2069%
20	Server 4 Utilization	84,0834%	8,0990%	58,4365%	99,4979%	83,5814%	84,5854%
21	Server 5 Utilization	83,5827%	8,0457%	53,1197%	99,2514%	83,0841%	84,0814%
22	Server 6 Utilization	82,9105%	8,2298%	57,4750%	99,2213%	82,4004%	83,4206%
23	SECOND STAGE - TRANSACTIONS "A" AND "C"						
24	Max time spent in system	35,78991	7,77209	14,22549	60,25001	35,30819	36,27163
25	Average time spent in the system	14,04869	4,91384	6,42655	31,88385	13,74433	14,35346
26	Max waiting time in queue	30,64001	7,73886	9,37828	55,65051	30,16035	31,11967
27	Average waiting time in queue	9,04684	4,90093	1,49633	26,87785	8,74308	9,35060
28	Server 1 Utilization	79,7296%	5,5080%	61,9211%	91,5582%	79,3882%	80,0710%
29	Server 2 Utilization	79,1895%	5,4380%	62,1697%	91,1152%	78,8524%	79,5265%
30	WHOLE SYSTEM						
31	Average time spent in the whole system	70,62960	15,81572	35,68475	125,39784	69,64933	71,60987
32	Average waiting time in queue in the whole syste	45,25134	15,90197	10,45977	100,97055	44,26572	46,23695
33							
34							

Table A.40. Probability distribution of customers waiting Y minutes in Case 1

A		B	C	D
1	Y minutes	Frequency	Probability	Cumulative Pr.
2	5	28974	0,15382	0,153823284
3	10	12572	0,06674	0,220568170
4	15	12569	0,06673	0,287297129
5	20	12206	0,06480	0,352098917
6	25	11162	0,05926	0,411358098
7	30	10170	0,05399	0,465350740
8	35	9067	0,04814	0,513487542
9	40	8030	0,04263	0,556118901
10	45	7248	0,03848	0,594598612
11	50	6598	0,03503	0,629627467
12	55	6178	0,03280	0,662426537
13	60	5653	0,03001	0,692438376
14	65	5336	0,02833	0,720767258
15	70	5034	0,02673	0,747492820
16	75	4718	0,02505	0,772540733
17	80	4217	0,02239	0,794928833
18	85	3797	0,02016	0,815087147
19	90	3616	0,01920	0,834284531
20	95	3426	0,01819	0,852473203
21	100	3202	0,01700	0,869472656
22	105	3021	0,01604	0,885511178
23	110	2706	0,01437	0,899877362
24	115	2451	0,01301	0,912889748
25	120	2341	0,01243	0,925318142
Σ		187411	1,00000	0,925318142
		ServiceTime	WaitingTime	Waiting_Cha

Table A.41. Simulation results after 1000 runs at "Simulation_Summary" Excel worksheet, Case 2

A		B	C	D	E	F	G
1			Standard			%95 Confidence Interval	
2		Mean	Deviation	Min	Max	Lower	Upper
3	FLOOR 1 - TRANSACTIONS "A" AND "B"						
4	Max time spent in system	64,97246	22,16958742	26,21504526	164,2807679	63,59837785	66,34654838
5	Average time spent in the system	32,34874	11,23547197	18,24806693	93,06813293	31,85235644	33,04511998
6	Max waiting time in queue	45,08548	22,38006174	7,675529771	142,3330029	43,69834854	46,47260974
7	Average waiting time in queue	14,81478	11,16597068	0,779714384	75,09072744	14,12270262	15,50685069
8	Server 1 Utilization	78,0230%	8,5772%	53,6887%	97,8306%	77,4914%	78,5546%
9	Server 2 Utilization	78,5290%	8,0851%	56,3051%	97,1614%	78,0279%	79,0301%
10	Server 3 Utilization	78,0410%	8,1409%	52,7180%	97,8752%	77,5365%	78,5456%
11	Server 4 Utilization	76,2300%	8,9634%	43,9232%	97,1550%	75,6744%	76,7855%
12	FLOOR 2 - TRANSACTIONS "C" AND "D"						
13	Max time spent in system	102,41241	30,86513239	37,41745705	206,2624773	100,4993676	104,3254491
14	Average time spent in the system	40,94223	12,86270305	23,93767828	103,367384	40,14498702	41,73946421
15	Max waiting time in queue	48,21959	22,53728672	7,760445082	137,171645	46,82271361	49,61646462
16	Average waiting time in queue	16,76999	12,61231844	0,882877093	77,35694306	15,98826656	17,55170575
17	Server 1 Utilization	86,1167%	8,3932%	56,8428%	99,9949%	85,5964%	86,6369%
18	Server 2 Utilization	85,7012%	8,2851%	58,9246%	99,8799%	85,1877%	86,2147%
19	Server 3 Utilization	85,0877%	8,2937%	56,1028%	99,6939%	84,5736%	85,6017%
20	Server 4 Utilization	84,5466%	8,3591%	53,7594%	99,2889%	84,0285%	85,0647%
21	Server 5 Utilization	83,8057%	8,4018%	59,2551%	99,1633%	83,2849%	84,3264%
22	Server 6 Utilization	83,3100%	8,4161%	56,4421%	98,5546%	82,7884%	83,8316%
23	SECOND STAGE - TRANSACTIONS "A" AND "C"						
24	Max time spent in system	61,41802	12,80039	18,10274	107,01914	60,62464	62,21139
25	Average time spent in the system	29,83474	10,50581	7,85375	62,21613	29,18358	30,48589
26	Max waiting time in queue	56,25638	12,75941	13,56417	102,13072	55,46555	57,04722
27	Average waiting time in queue	24,83618	10,49338	2,92631	57,06903	24,18579	25,48656
28	Server 1 Utilization	84,8155%	4,1613%	66,5772%	90,8211%	84,5576%	85,0734%
29	Server 2 Utilization	84,2261%	4,1666%	67,7486%	90,5529%	83,9678%	84,4843%
30	WHOLE SYSTEM						
31	Average time spent in the whole system	64,69977	16,18377	30,69475	126,10409	63,69669	65,70285
32	Average waiting time in queue in the whole syste	39,27743	16,19675	6,53898	100,82633	38,27355	40,28132
33							
		data / Spreadsheet / Simulation / Simulation_Summary / Service Chart / ServiceTime / WaitingTime / Waiting					

Table A.42. Probability distribution of customers waiting Y minutes in Case 2

	A	B	C	D
1	Y minutes	Frequency	Probability	Cumulative Pr.
2	5	27728	0,14659	0,146591101
3	10	12528	0,06623	0,212823549
4	15	12235	0,06468	0,277506979
5	20	12088	0,06391	0,341413255
6	25	11689	0,06180	0,403210117
7	30	10993	0,05812	0,461327398
8	35	10525	0,05564	0,516970479
9	40	10054	0,05315	0,570123499
10	45	9402	0,04971	0,619829555
11	50	8671	0,04584	0,665670995
12	55	7924	0,04189	0,707563230
13	60	7218	0,03816	0,745723016
14	65	6430	0,03399	0,779716841
15	70	5693	0,03010	0,809814329
16	75	5140	0,02717	0,836988242
17	80	4836	0,02557	0,862554982
18	85	4122	0,02179	0,884346980
19	90	3676	0,01943	0,903781086
20	95	3450	0,01824	0,922020386
21	100	2953	0,01561	0,937632169
22	105	2391	0,01264	0,950272796
23	110	2049	0,01083	0,961105354
24	115	1724	0,00911	0,970219717
25	120	1327	0,00702	0,977235239
26	125	1104	0,00584	0,983071815
ServiceTime		WaitingTime	Waiting Chart	total

Table A.43. Simulation results after 1000 runs at "Simulation_Summary" Excel worksheet, Case 3

A		B	C	D	E	F	G
1		Mean	Standard Deviation	Min	Max	%95 Confidence Interval	
2						Lower	Upper
FLOOR 1 - TRANSACTIONS "A" AND "B"							
3	Max time spent in system	40,83298	11,19502556	23,6701197	101,30335	40,13910979	41,52689593
4	Average time spent in the system	21,07610	3,211344578	16,4848291	40,889162	20,87706233	21,27514473
5	Max waiting time in queue	20,07837	11,36851268	0	79,42286	19,37374102	20,78299645
6	Average waiting time in queue	3,55595	3,131813561	0	23,310176	3,361842531	3,750066162
7	Server 1 Utilization	62,6053%	7,2753%	42,0275%	85,1604%	62,1544%	63,0562%
8	Server 2 Utilization	61,4781%	7,2815%	42,4535%	86,3950%	61,0267%	61,9294%
9	Server 3 Utilization	63,3794%	7,4373%	39,8397%	86,3297%	62,9184%	63,8404%
10	Server 4 Utilization	62,2147%	7,3241%	39,0879%	83,4833%	61,7608%	62,6687%
11	Server 5 Utilization	61,4453%	7,2750%	37,6226%	83,4140%	60,9944%	61,8962%
FLOOR 2 - TRANSACTIONS "C" AND "D"							
12	Max time spent in system	101,75039	30,4068128	34,9089968	210,87667	99,86575564	103,6350232
13	Average time spent in the system	40,78794	12,56074968	23,9219811	112,50205	40,00941689	41,56646355
14	Max waiting time in queue	48,50346	21,84001139	9,22487162	153,20742	47,14980392	49,85711978
15	Average waiting time in queue	16,66665	12,28857251	1,04889769	86,608031	15,90499997	17,4283072
16	Server 1 Utilization	85,8078%	8,4999%	60,1526%	99,9687%	85,2810%	86,3346%
17	Server 2 Utilization	85,3640%	8,3119%	60,7723%	99,8488%	84,8488%	85,8792%
18	Server 3 Utilization	84,7762%	8,3747%	57,9607%	99,7786%	84,2571%	85,2953%
19	Server 4 Utilization	84,2508%	8,2281%	56,4486%	99,6567%	83,7408%	84,7608%
20	Server 5 Utilization	83,5506%	8,4028%	55,7831%	99,0925%	83,0298%	84,0714%
21	Server 6 Utilization	82,9615%	8,5134%	55,8217%	98,9654%	82,4338%	83,4891%
SECOND STAGE - TRANSACTIONS "A" AND "C"							
22	Max time spent in system	59,24087	14,28804	18,53661	101,33262	58,35529	60,12645
23	Average time spent in the system	28,10447	11,28524	6,77629	61,81144	27,40500	28,80393
24	Max waiting time in queue	54,09691	14,22769	13,29569	96,29873	53,21507	54,97875
25	Average waiting time in queue	23,10315	11,27525	1,85511	56,77337	22,40430	23,80200
26	Server 1 Utilization	85,6185%	4,8569%	64,1072%	92,1774%	85,3173%	85,9197%
27	Server 2 Utilization	85,0153%	4,9029%	61,9775%	92,0659%	84,7115%	85,3192%
WHOLE SYSTEM							
28	Average time spent in the whole system	57,76173	14,73662	28,32445	109,91451	56,84833	58,67513
29	Average waiting time in queue in the whole system	32,33416	14,70913	3,59095	84,54942	31,42248	33,24584
30							
31							
32							
33							
		data		Simulation_Summary	ArrivingControl	Validation	Service_Chart / Ser

Table A.44. Probability distribution of customers waiting Y minutes in Case 3

A		B	C	D
1	Y minutes	Frequency	Probability	Cumulative Pr.
2	5	36876	0,19545	0,195450305
3	10	15811	0,08380	0,279251823
4	15	14042	0,07443	0,353677281
5	20	13179	0,06985	0,423528664
6	25	12226	0,06480	0,488328952
7	30	11219	0,05946	0,547791935
8	35	10594	0,05615	0,603942291
9	40	9825	0,05207	0,656016791
10	45	8846	0,04689	0,702902391
11	50	7798	0,04133	0,744233379
12	55	7043	0,03733	0,781562712
13	60	6319	0,03349	0,815054698
14	65	5484	0,02907	0,844121014
15	70	4870	0,02581	0,869933005
16	75	4267	0,02262	0,892548974
17	80	3690	0,01903	0,911576705
18	85	3095	0,01640	0,927980834
19	90	2609	0,01383	0,941809065
20	95	2303	0,01221	0,954015434
21	100	1918	0,01017	0,964181225
22	105	1617	0,00857	0,972751654
23	110	1413	0,00749	0,980240841
24	115	1051	0,00557	0,985811355
25	120	797	0,00422	0,990035617
26	125	542	0,00287	0,992908328
		Service Time	Waiting Time	Waiting Cha

APPENDIX B: FIGURES

In this part, after the simulation runs for n times, the graphical representations of some performance measures are shown. From B.1 to B.7, some examples of graphical illustrations of spreadsheet simulation in the literature are given. From, B.8 to B.12, the graphs for the probability distributions of the performance measures for M/G/c the model in chapter 5 are shown.

From B.13 to B.17, the graphs for the probability distributions of performance measures for the model in chapter 6 are given.

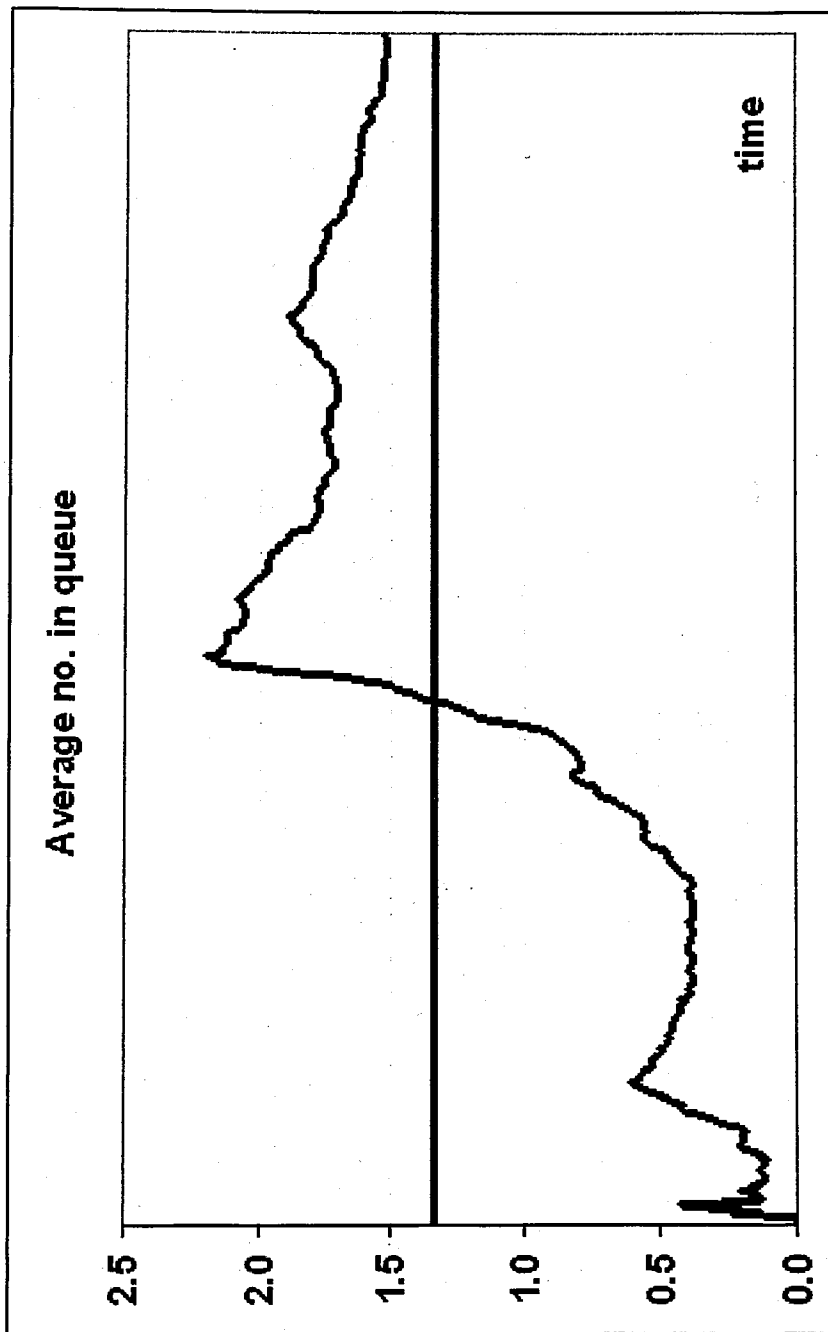


Figure B.1. Event-driven $M/M/1$, approximate time plot of average number in queue, Evans

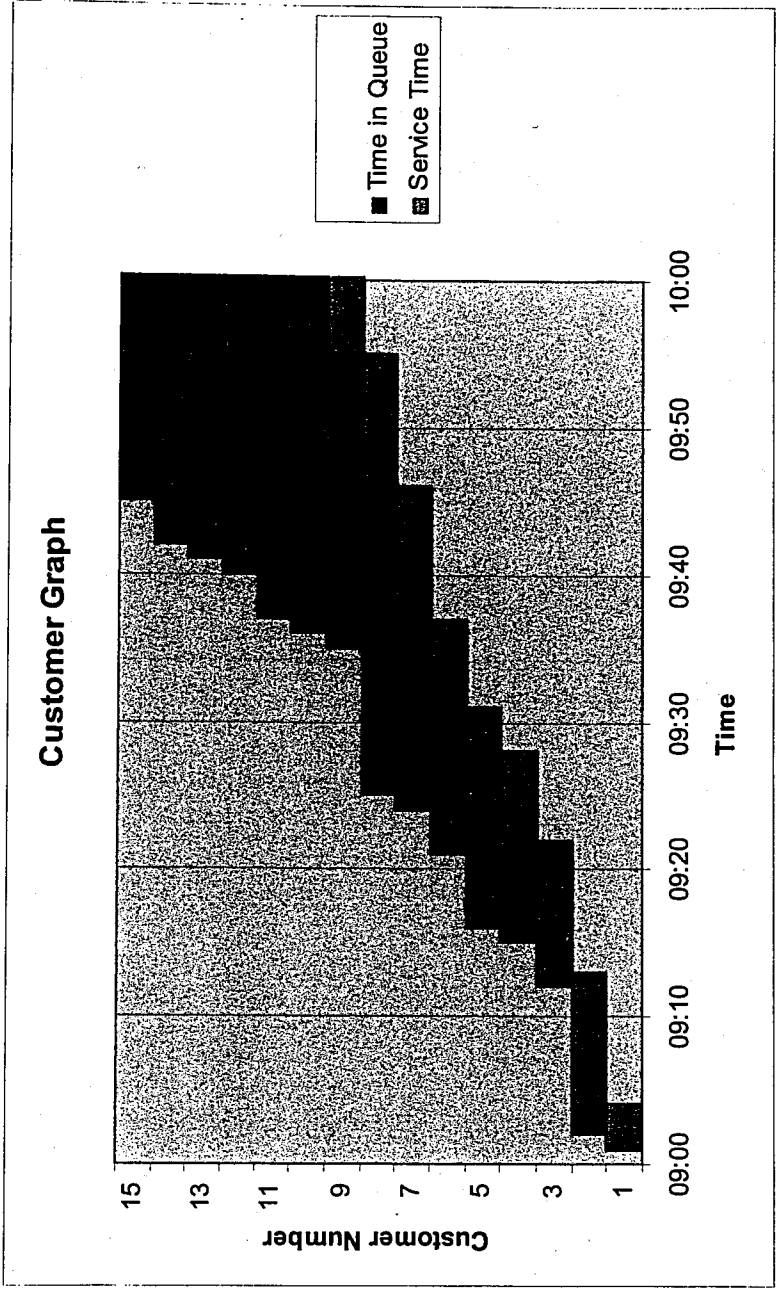


Figure B.2. Basic queuing model, customer graph, Grossman

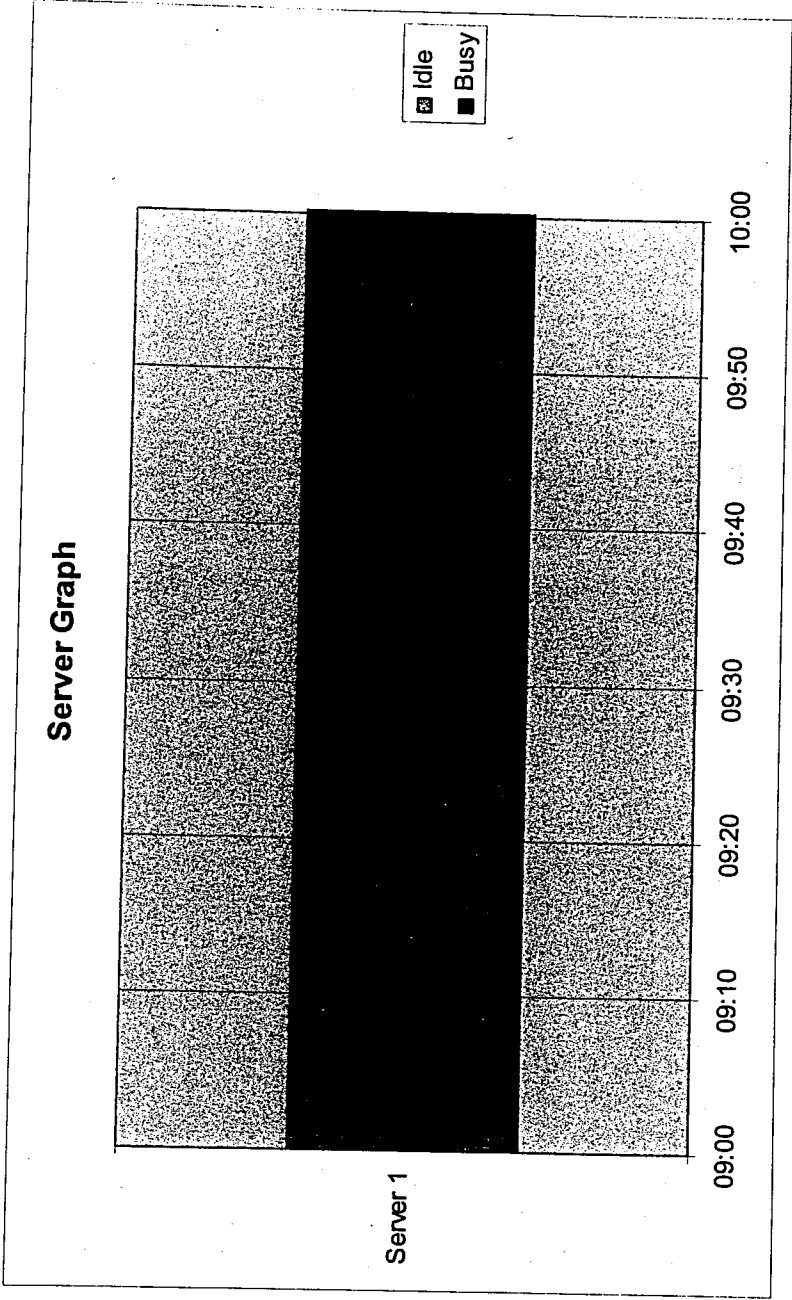


Figure B.3. Basic queuing model, server graph, Grossman

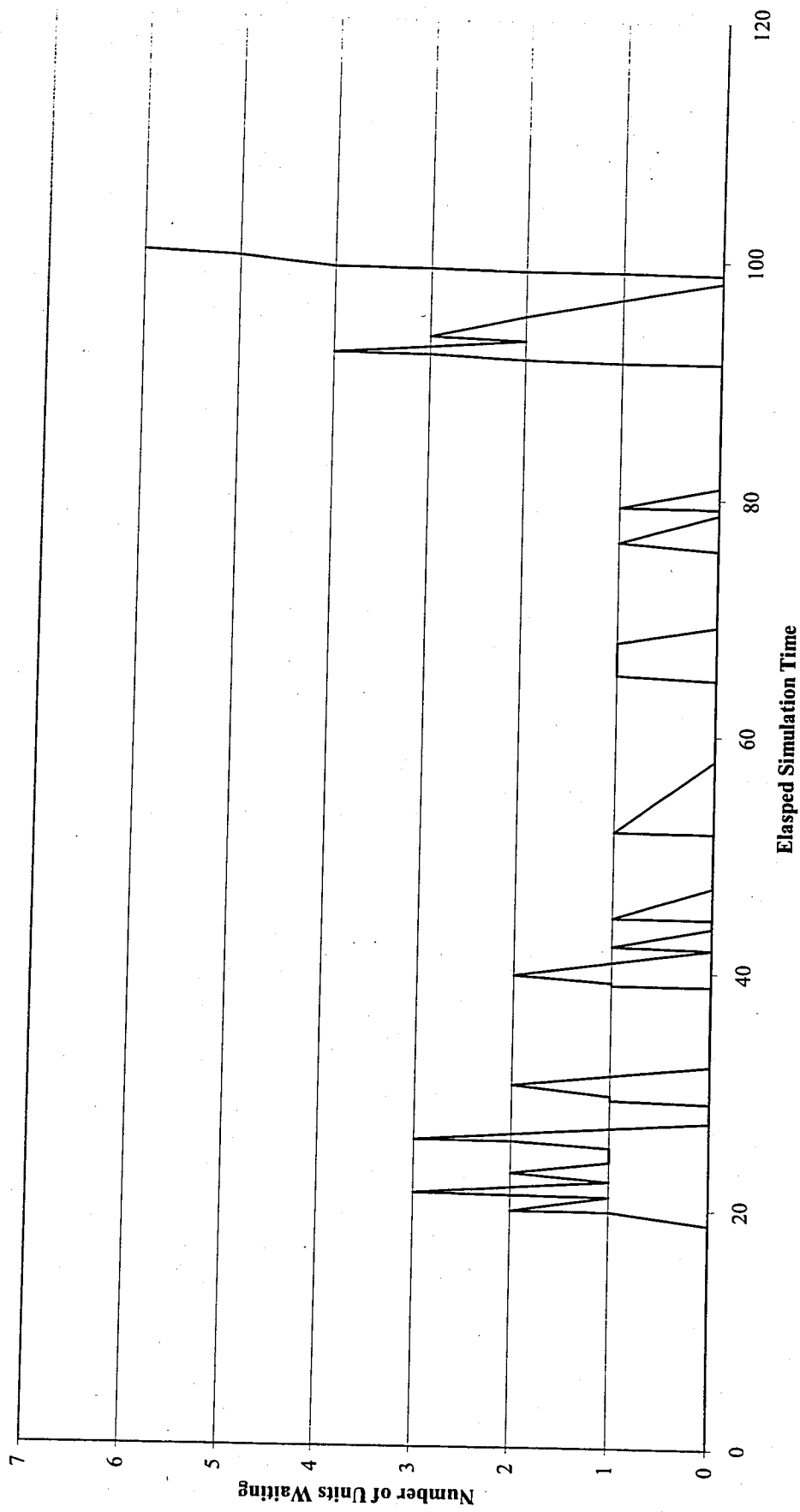
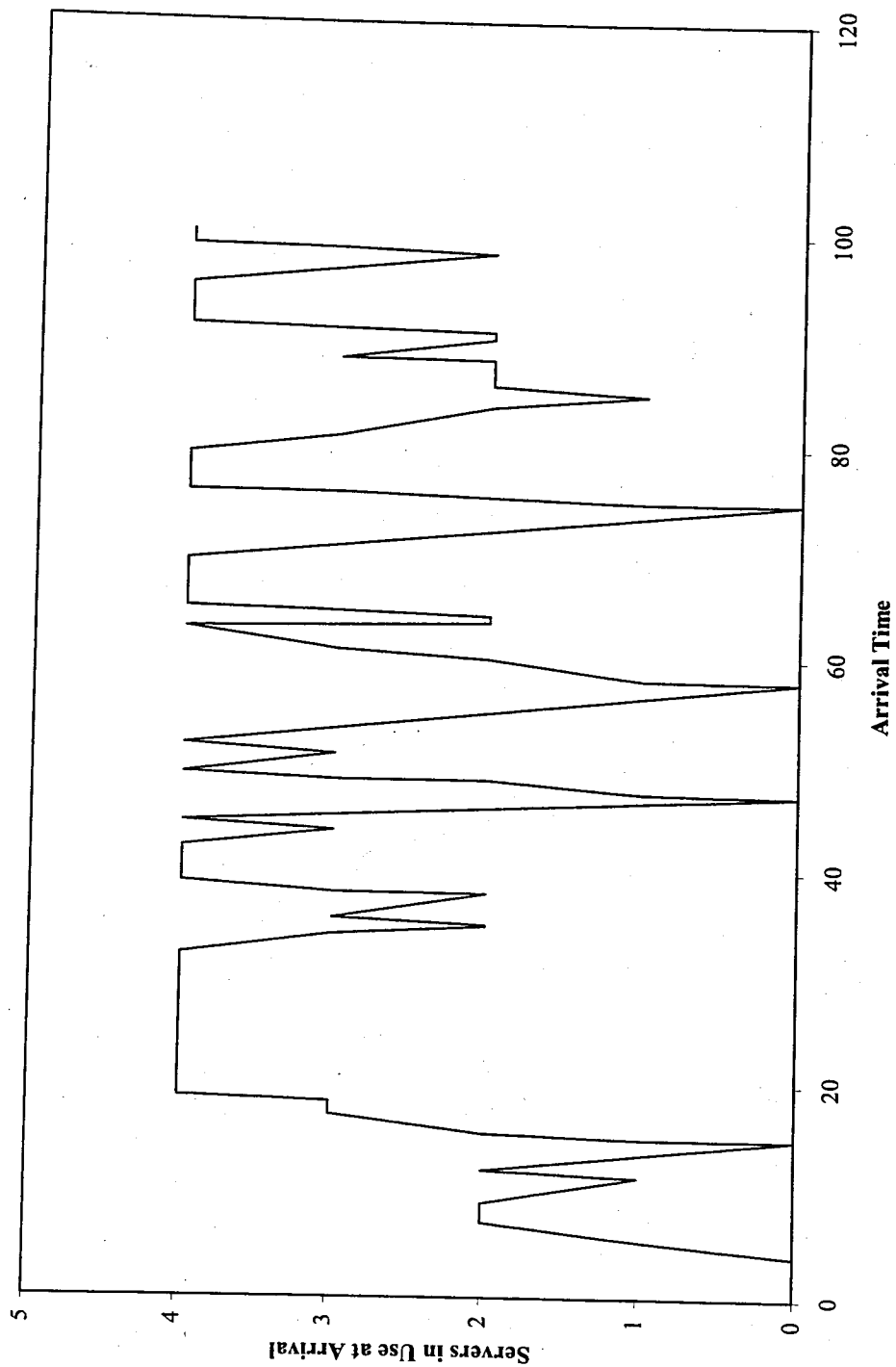


Figure B.4. The graph of queue length at customer arrivals, Hora



— Servers Utilized at Arrival

Figure B.5. The graph of number of server utilized with respect to customer arrivals, Hora

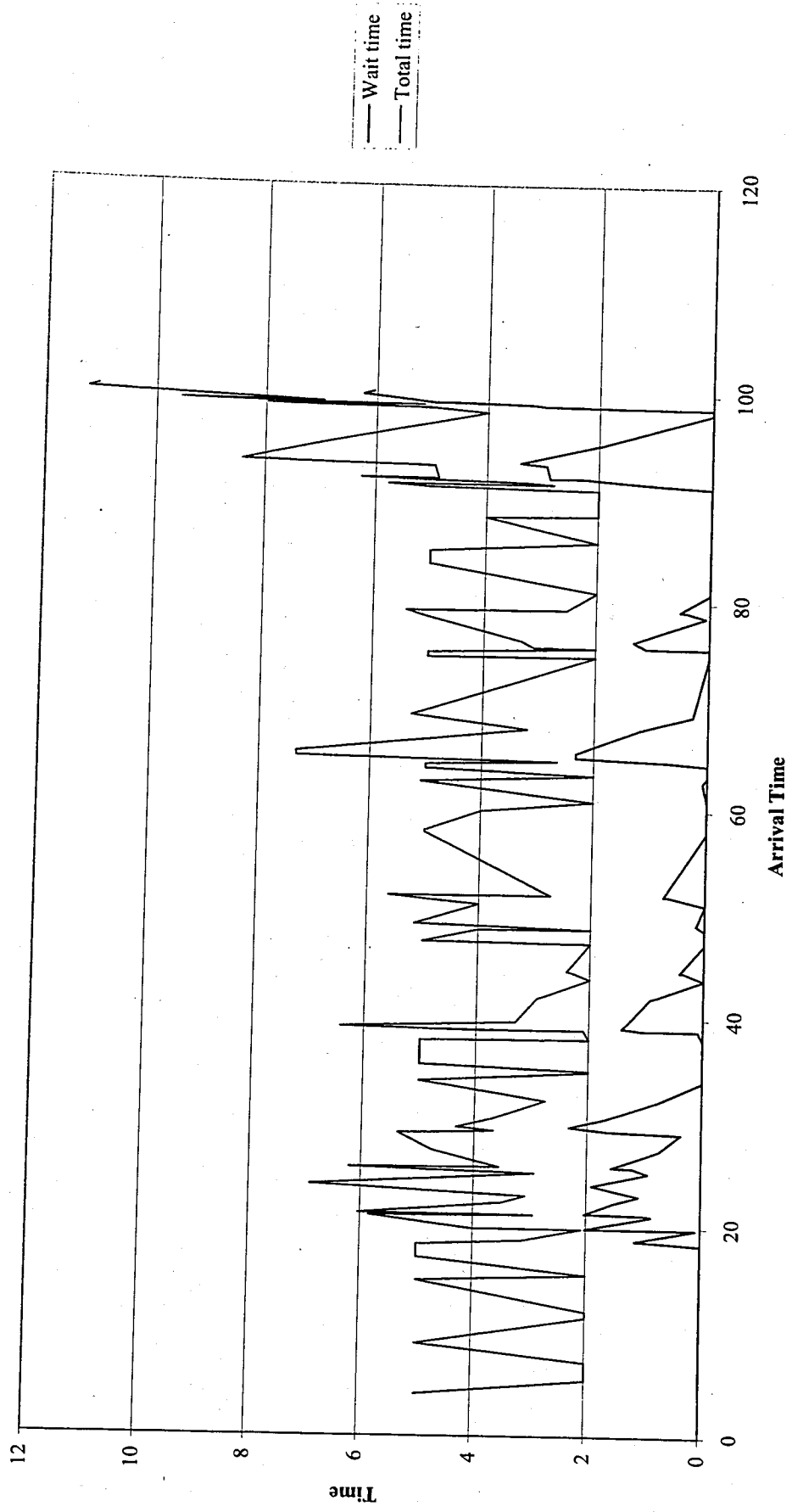


Figure B.6. The graph of waiting time and total time that customers spend in the system with respect to customer arrivals, Hora

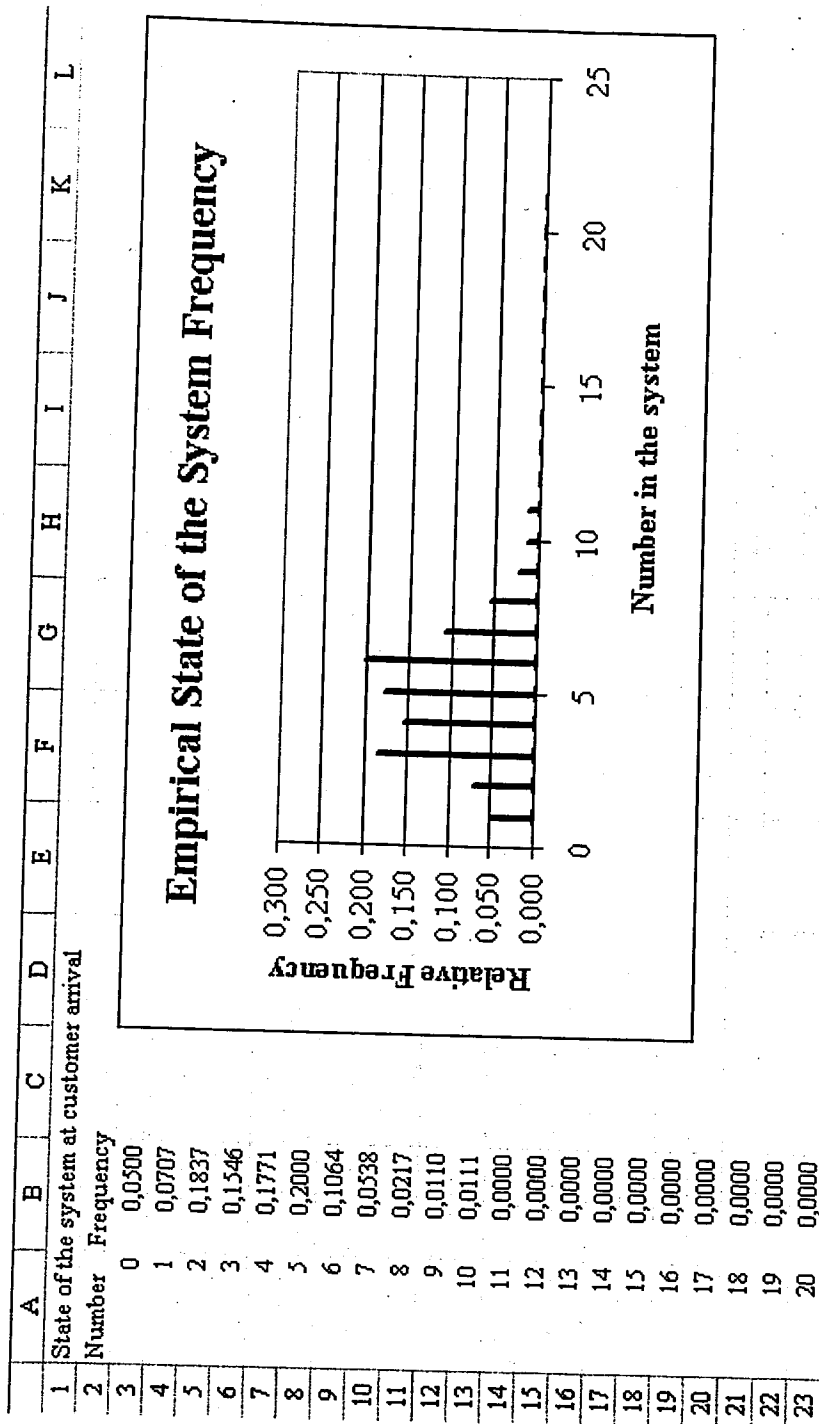


Figure B.7. Empirical state of the system frequency, Hora

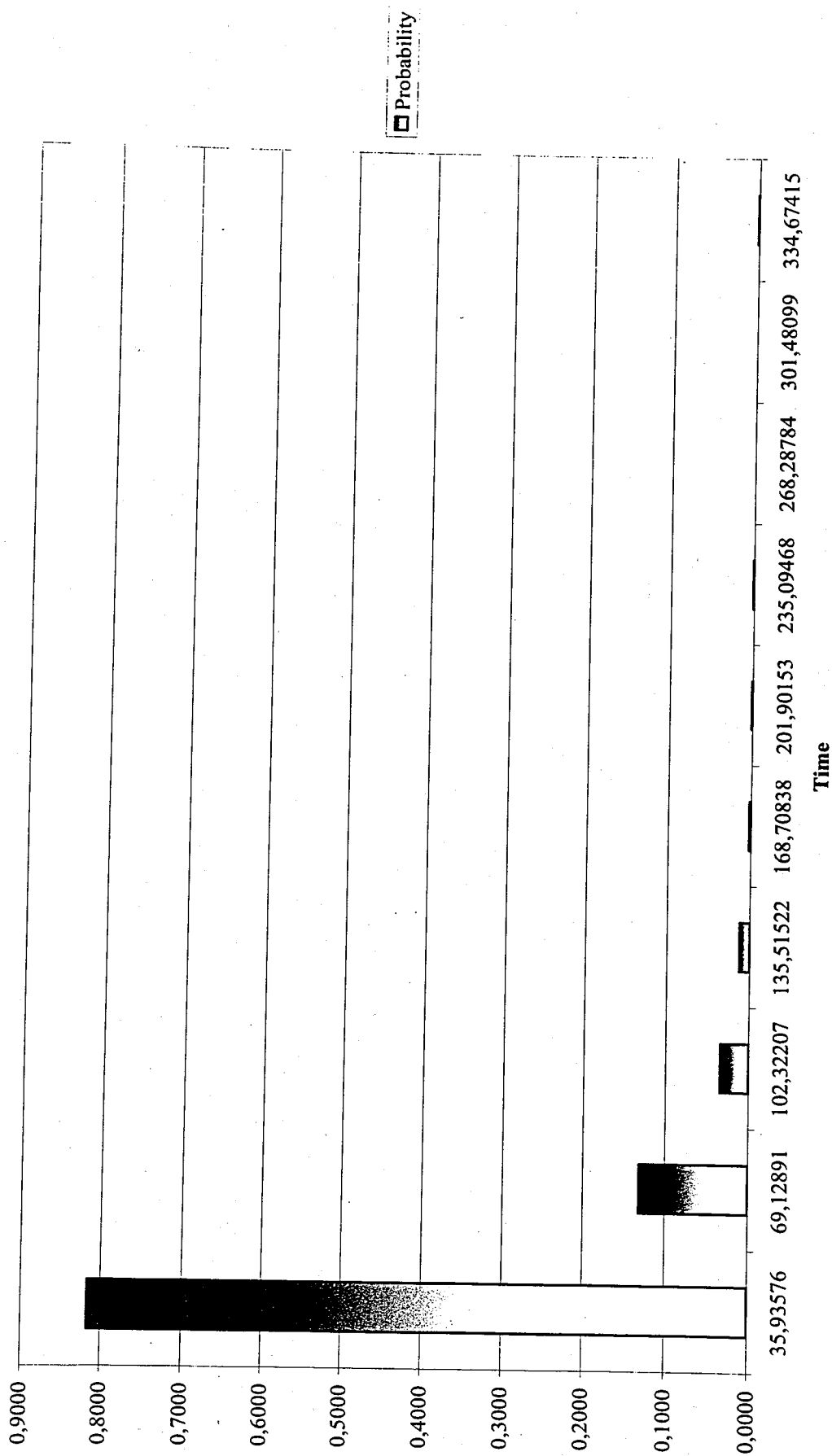


Figure B.8. M/G/c queueing model, the graph of distribution of average time spent in the system

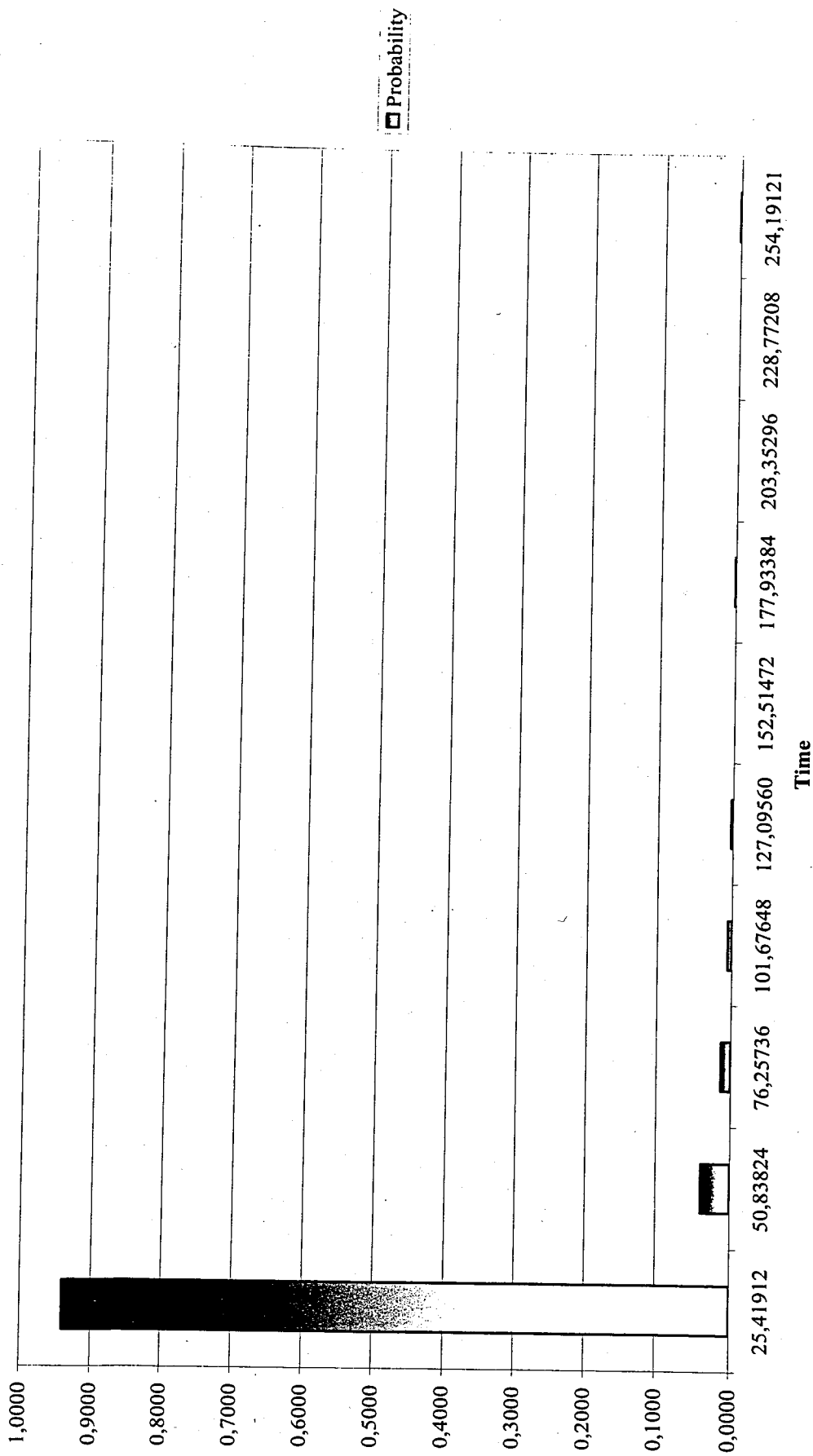


Figure B.9. M/G/c queueing model, the graph of distribution of average waiting time in queue

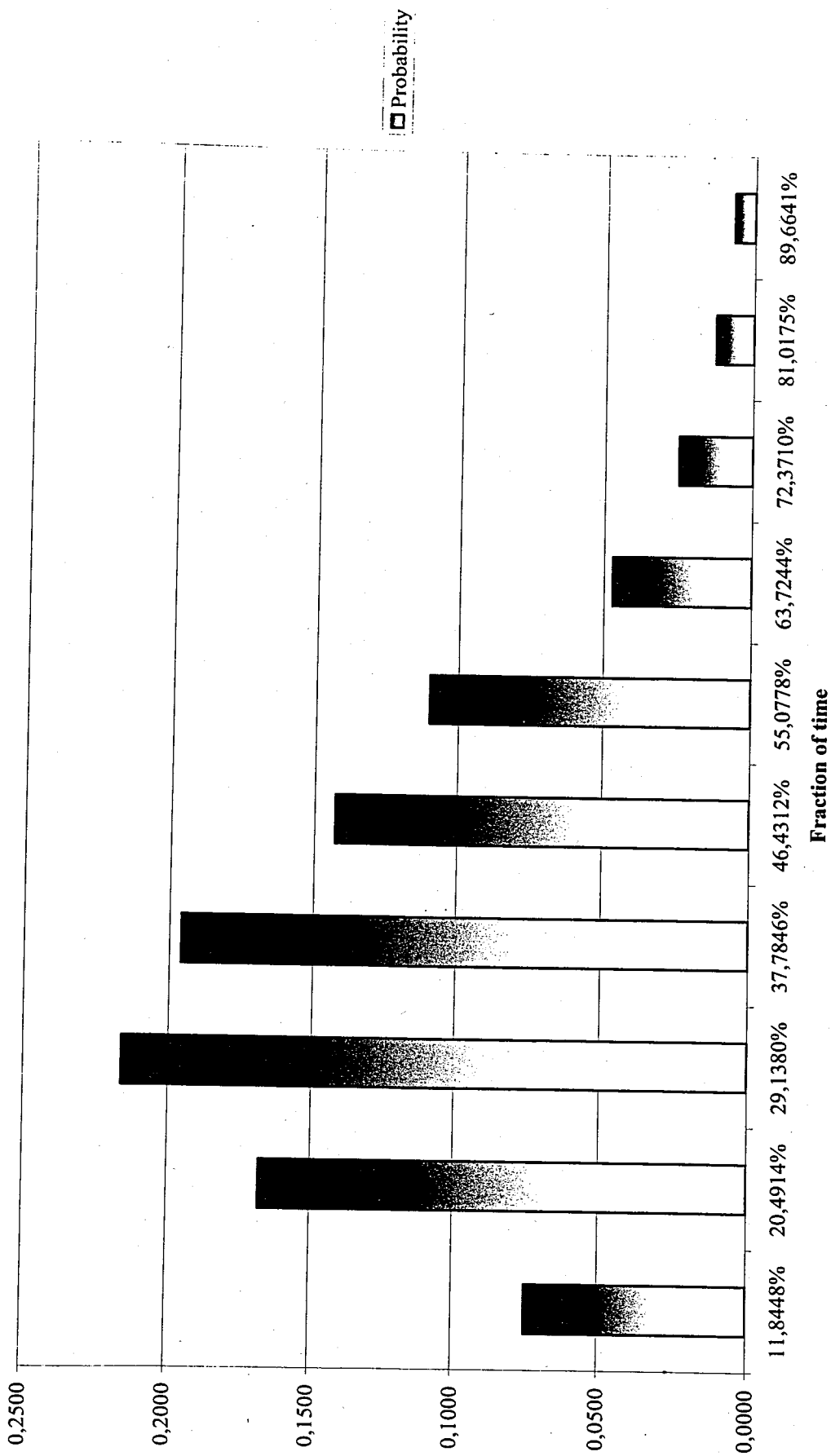


Figure B.10. M/G/c queueing model, the graph of distribution of servers utilization

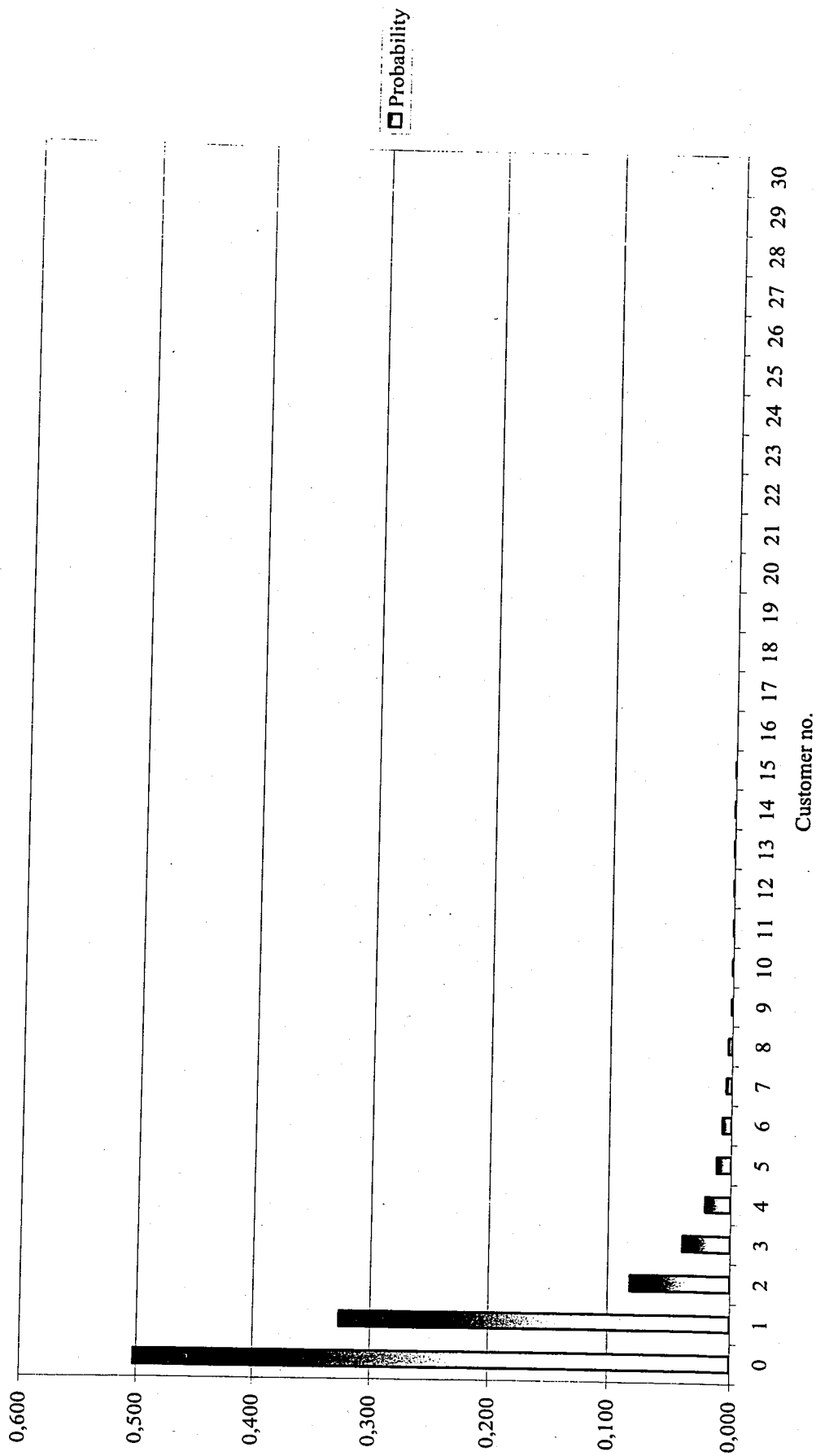


Figure B.11. M/G/c queueing model, the graph of distribution of customer number in the system

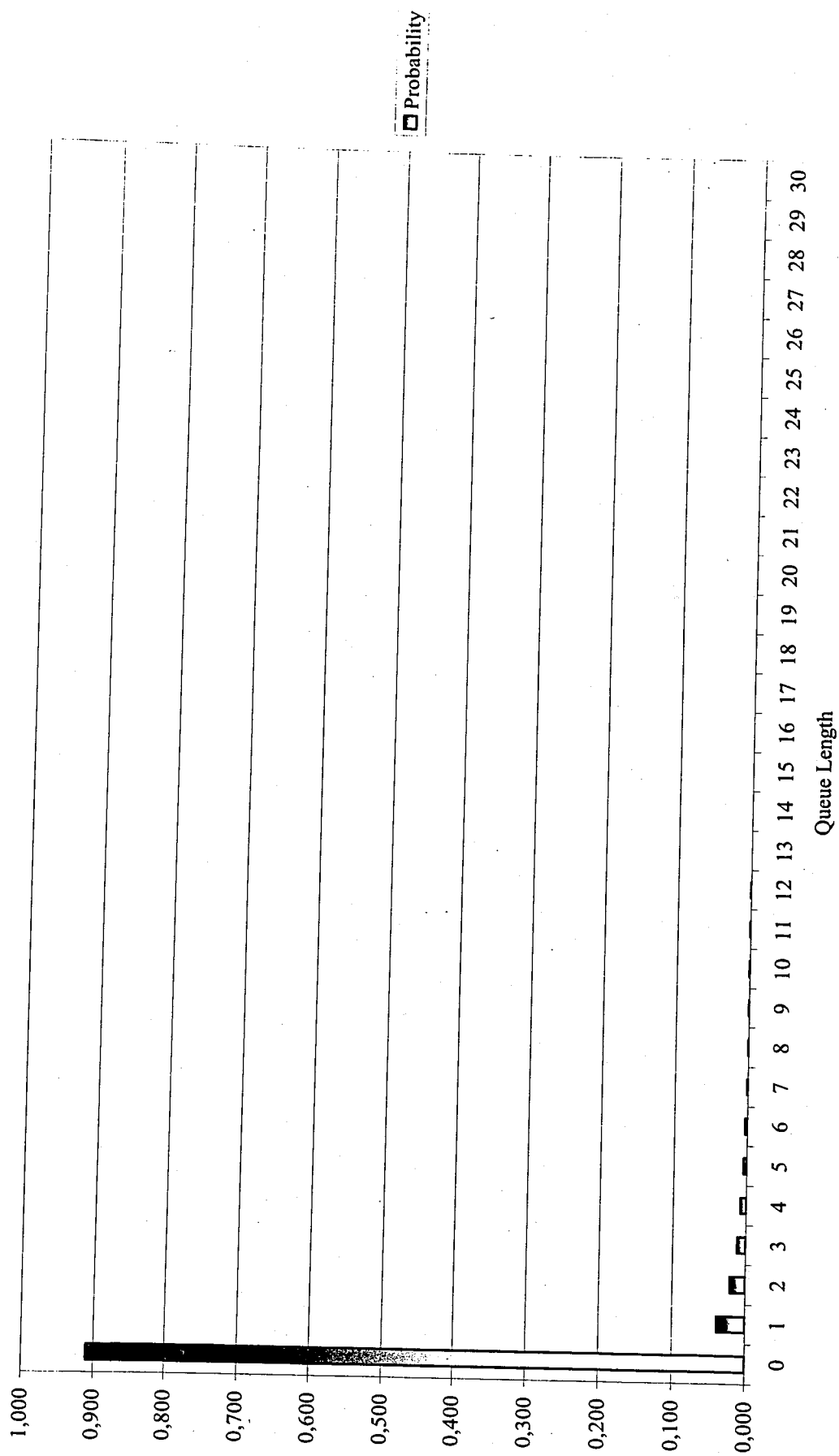


Figure B.12. M/G/c queueing model, the graph of distribution of queue length

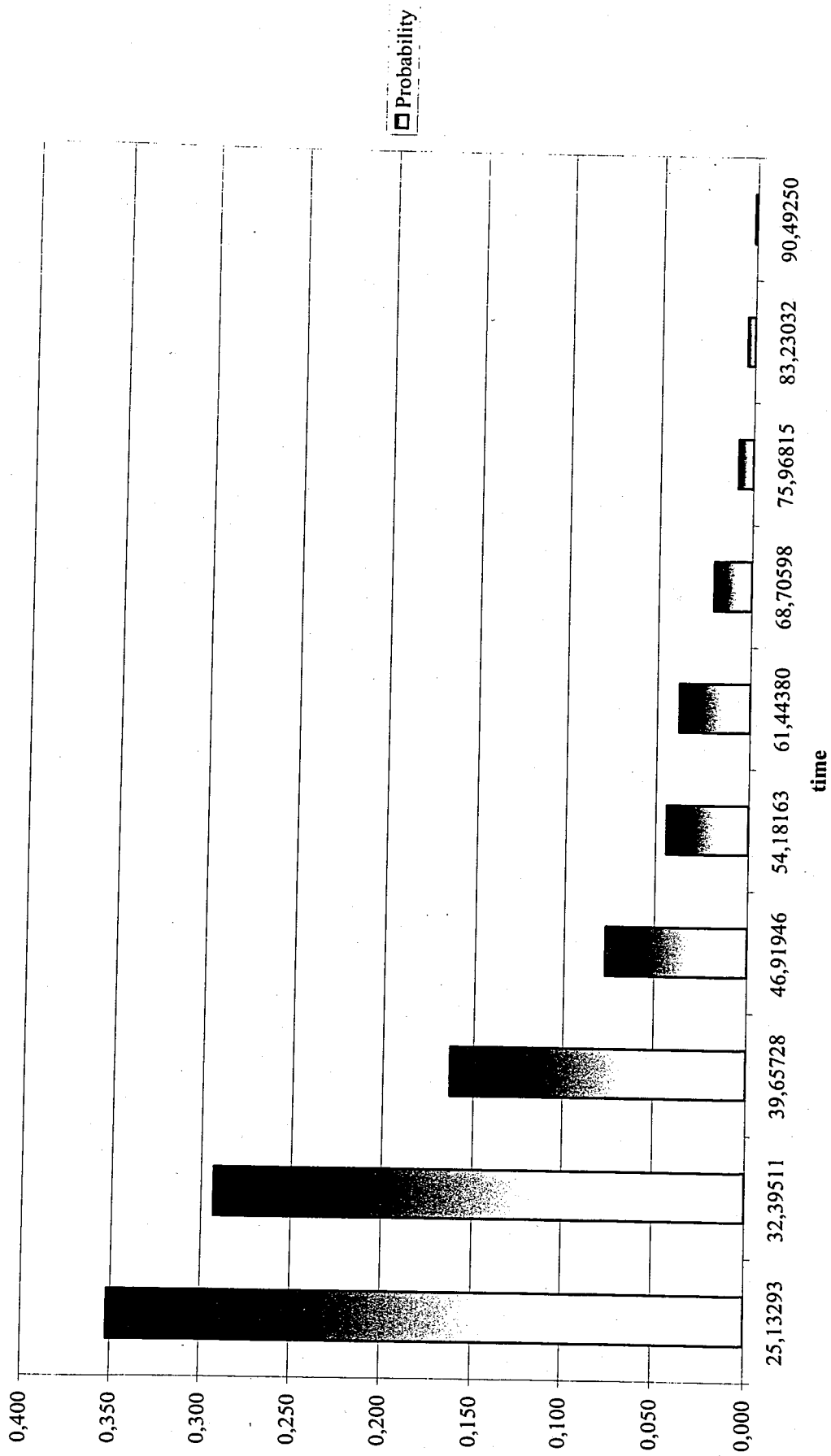


Figure B.13. The graph of distribution of average time spent at floor 1

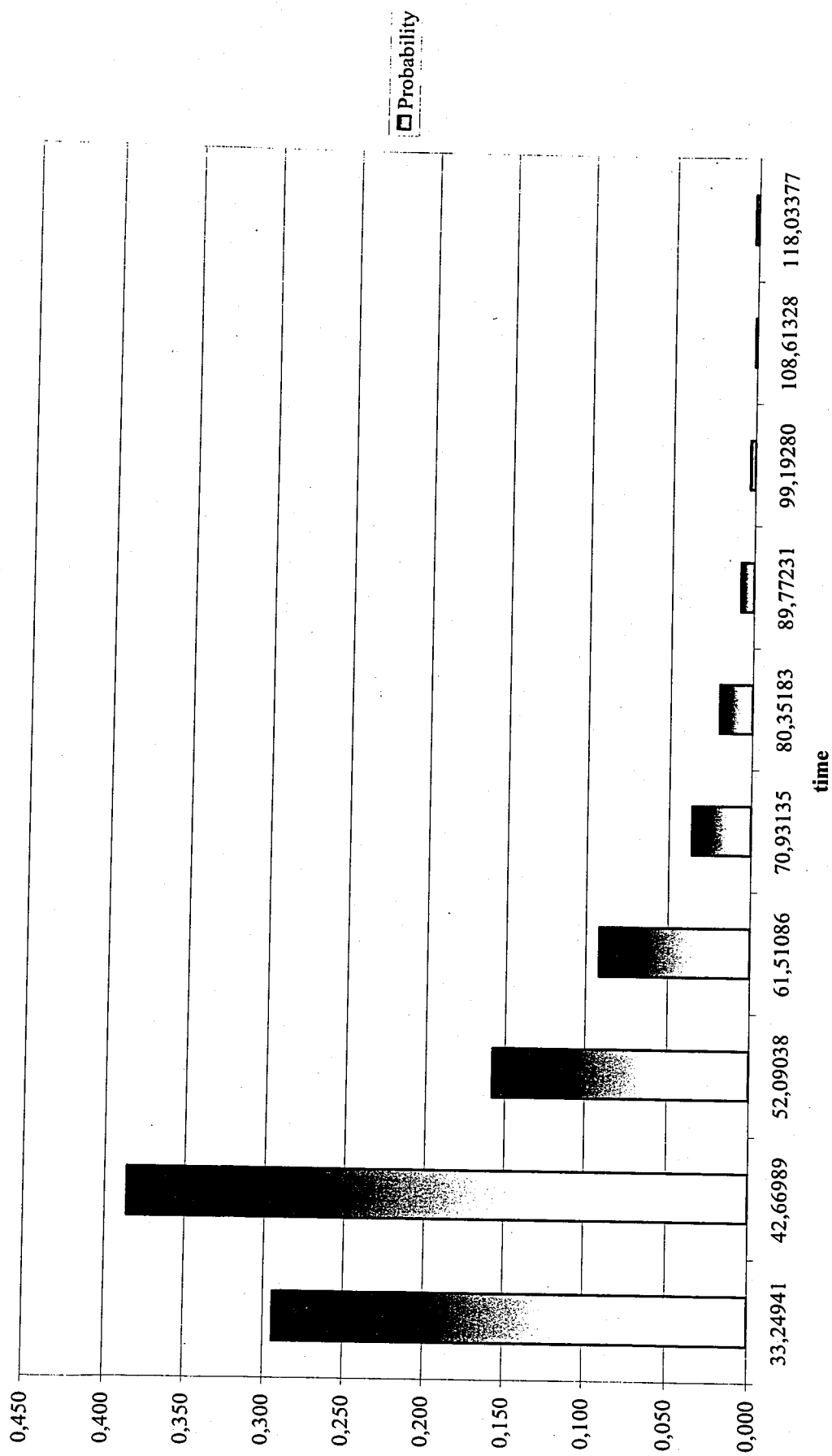


Figure B.14. The graph of distribution of average time spent at floor 2

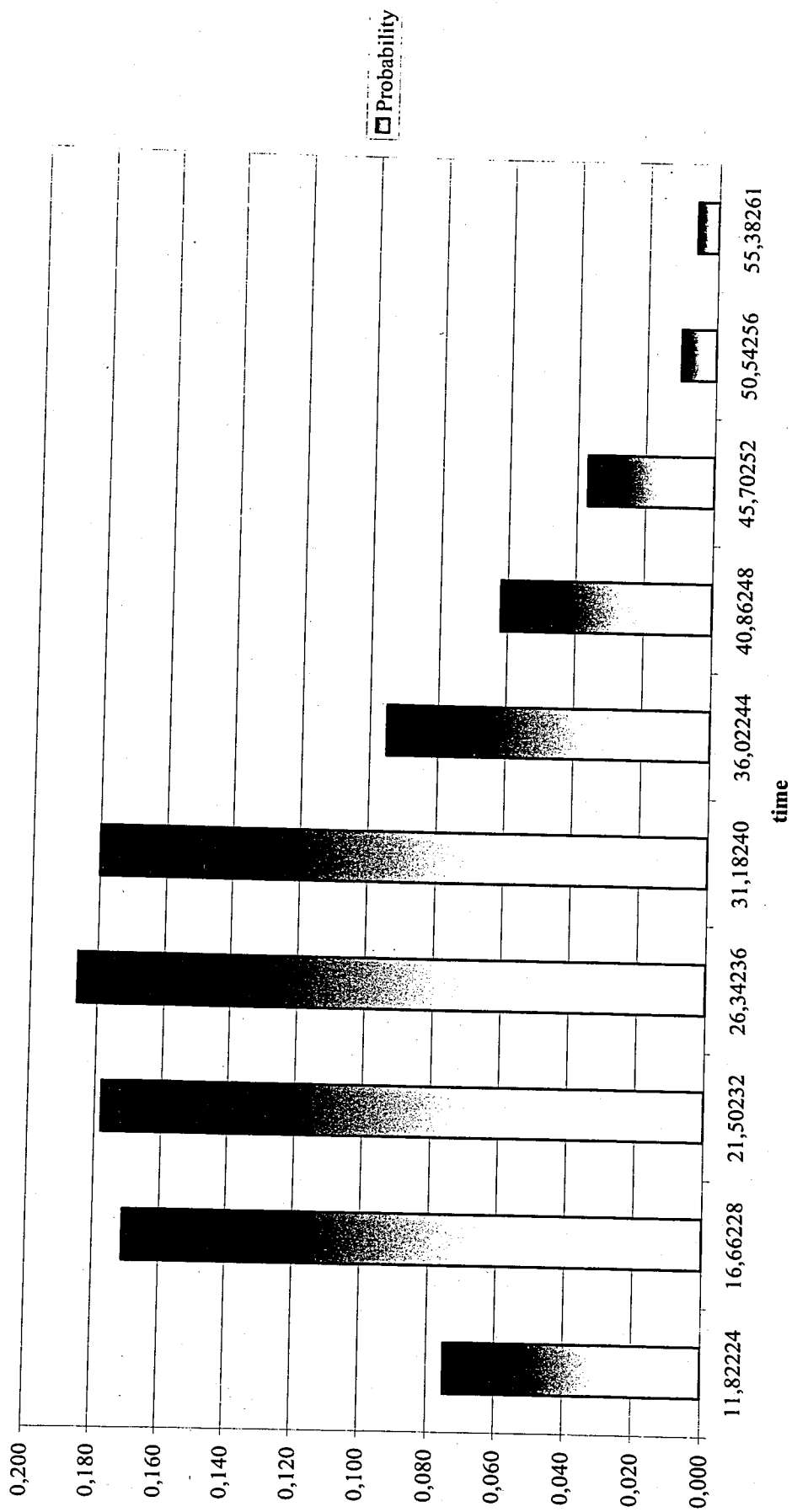


Figure B.15. The graph of distribution of average time spent at stage 2

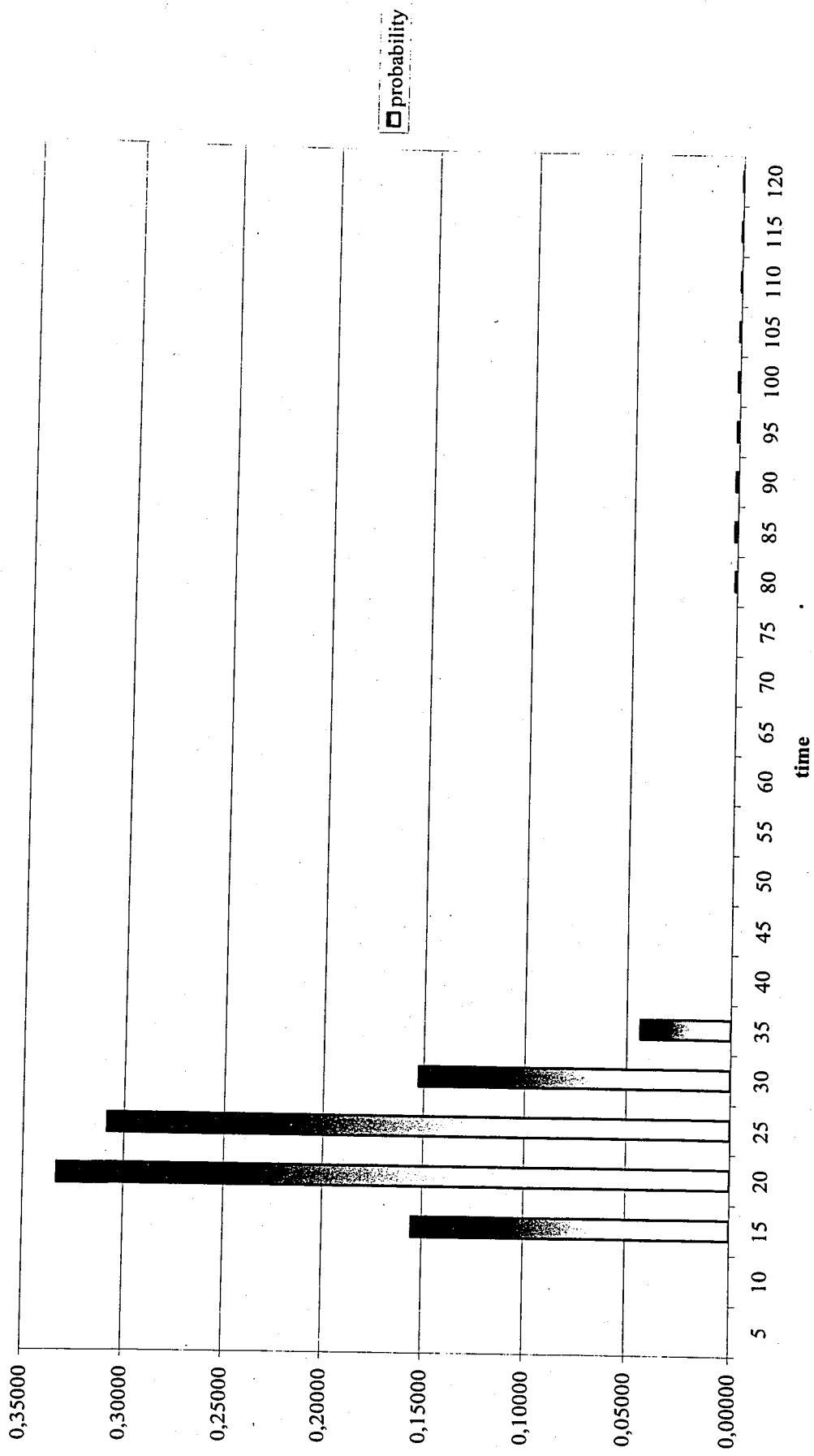


Figure B.16. The graph of probability distribution of customers having X minutes service times

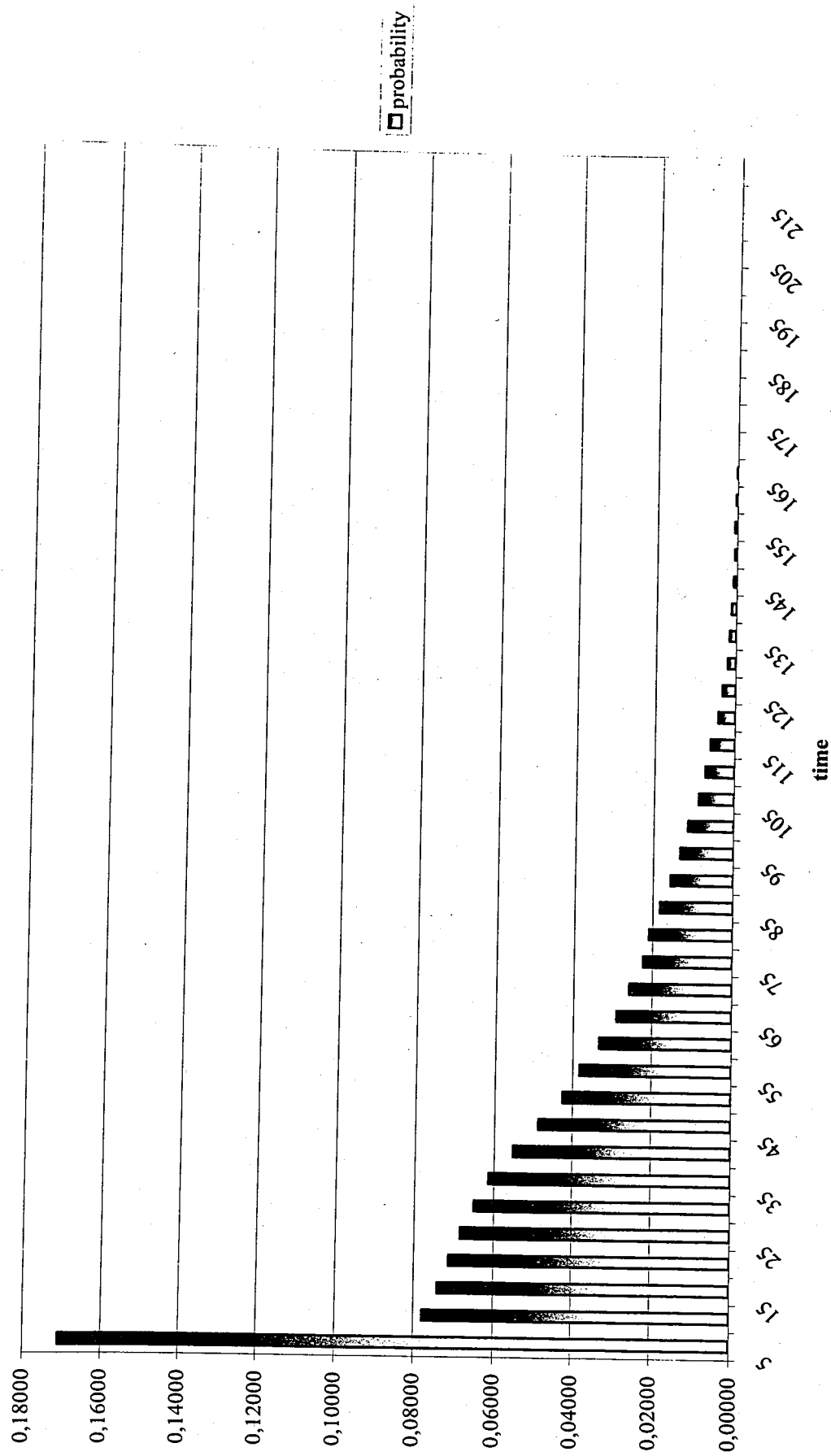


Figure B.17. The graph of probability distribution of customers waiting Y minutes in queue

APPENDIX C: PROGRAMS CODED IN VBA AND EXCEL FILES

In this part, some of the simulation programs coded in VBA are given. Belows are the programs that generate random number variates from exponential, nonhomogeneous Poisson Process, Weibull, Triangular distribution and Triangular distribution with parameter, respectively. Moreover, last two are the simulation programs of the first two examples, 5.2.1.1 and 5.2.2.1 in Chapter 5, which runs the simulation for n times.

And the Excel files are listed below;

- 5.1.1.xls = Spreadsheet implementation of M/G/1 model in event-scheduling approach.
- 5.1.2.xls = Spreadsheet implementation of M/G/c model in event-scheduling approach.
- 5.2.1.1.xls = Spreadsheet implementation of M/G/1 model in process-interaction approach.
- 5.2.2.1.xls = Spreadsheet implementation of M/G/c model in process-interaction approach, with the distributions of no. customers in the system and queue length.
- 5.2.2.2.xls = Spreadsheet implementation of M/G/c model with balking property.
- 5.2.2.3.xls = Spreadsheet implementation of M/G/c model with balking and reneging properties.
- 5.2.2.4.xls = Spreadsheet implementation of simple network queuing models with M/G/c queue at serial servers with balking and reneging properties.
- 5.2.2.5.xls = Spreadsheet implementation of simple network queuing models with M/G/c queue at serial servers with balking and reneging properties and servers break time.
- 5.2.2.6.xls = Spreadsheet implementation of M/G/c models with changing service times in case of long queue length.
- 6.2.1.xls = Spreadsheet implementation of the real queuing problem in the Central Bank.
- 6.2.1_Case1.xls, 6.2.1_Case2.xls and 6.2.1_Case3.xls = Spreadsheet implementations of some proposed models to the real queuing problem in the Central Bank.

VBA Function, expo(mean), for Generating Interarrival Times from Exponential Distribution:

```
Function expo(mean As Double) As Double
    Dim x As Double
    expo = Log(1 - random()) * (-1) * mean
End Function
```

The expo(mean) function has one parameter, which is the mean of the desired exponential distribution. The expo function calls uniform random number, U, from random() function, and returns exponential random variate with mean = $\frac{1}{\lambda}$.

VBA Function, Nonhexpo(), for Generating Interarrival Times from Nonhomogeneous Poisson Process:

```
Public time As Double
Public I As Double
Function NonHexpo() As Double
    Dim x As Double
    Set data = Sheet1
    Dim mean(4) As Double
    Dim Time_Int(4) As Double
    mean(0) = data.Cells(24, 5)
    mean(1) = data.Cells(25, 5)
    mean(2) = data.Cells(26, 5)
    mean(3) = data.Cells(27, 5)
    Time_Int(0) = data.Cells(24, 4)
    Time_Int(1) = data.Cells(25, 4)
    Time_Int(2) = data.Cells(26, 4)
    Time_Int(3) = data.Cells(27, 4)
    x = Log(1 - random()) * (-1) * mean(I)
```

```

Do Until x + time < Time_Int(I)
    x = (x - Time_Int(I) + time) * mean(I + 1) / mean(I)
    time = Time_Int(I)
    I = I + 1
Loop
NonHexpo = x + time
time = NonHexpo
End Function

```

The function first takes the required data from the excel sheet. It stores the mean arrival times in mean(4) matrix and four specified t values in Time_Int matrix. Then, it calculates X random variable that is exponentially distributed with mean of the first time interval (e.g. 1,849315068). "time" variable in the function represents the time passed. Its first value is 0, and it shows the current time passed as the function is called. "time" is set to 0, at the beginning of a new simulation in the main simulation program, Run_Sim. Then a loop is constructed. This loop supplies the calculated X value within the true time interval. The loop checks this X value, whether it is in the correct time interval by summing the generated X and time. If it is in the correct interval, then this X is output as the random variable, and the function returns the arrival time of the next customer. If new time with generated X passes the next interval, the true X is calculated by dividing the old mean and multiplying the new mean. Only the time period that falls into the next time interval is revised (i.e. $(x - \text{Time_Int}(I) + \text{time})$). The time is set to the beginning of the next time interval, while the variable I, which indicates the time interval, is set to the next time interval. By defining time and I as variable type Public at the beginning of the program provides that these variables keep their value until a series of random variables are generated for 1 run.

VBA Function, zweibull(alfa,beta), for Generating Service Times from Weibull Distribution:

```

Function zweibull(alfa, beta As Double) As Double
    zweibull = alfa * (-1 * Log(1 - random())) ^ (1 / beta)

```

End Function

VBA Function, Tri_Pay(a,m,b), for Generating Service Times from Triangular Distribution:

Function Tri_Pay(a, m, b As Double) As Double

Dim U As Double

U = random()

If U < (m - a) / (b - a) Then

Tri_Pay = a + Sqr(U * (m - a) * (b - a))

Else

Tri_Pay = b - Sqr((1 - U) * (b - m) * (b - a))

End If

End Function

VBA Function, Tri(kind), for Generating Service Times from Triangular Distribution:

Function Tri(kind As String) As Double

Dim U As Double

Set data = Sheet1

If kind = "A" Then

If random() < 0.6 Then

a = data.Cells(31, 4)

m = data.Cells(31, 5)

b = data.Cells(31, 6)

Else

a = data.Cells(32, 4)

m = data.Cells(32, 5)

b = data.Cells(32, 6)

End If

Else

```

If kind ="B" Then
    a = data.Cells(33, 4)
    m = data.Cells(33, 5)
    b = data.Cells(33, 6)
Else
    If kind = "C" Then
        If random() < 0.96 Then
            a = data.Cells(34, 4)
            m = data.Cells(34, 5)
            b = data.Cells(33, 6)
        Else
            a = data.Cells(35, 4)
            m = data.Cells(35, 5)
            b = data.Cells(35, 6)
        End If
    Else
        If kind = "D" Then
            If random() < 0.55 Then
                a = data.Cells(36, 4)
                m = data.Cells(36, 5)
                b = data.Cells(36, 6)
            Else
                a = data.Cells(37, 4)
                m = data.Cells(37, 5)
                b = data.Cells(37, 6)
            End If
        End If
    End If
End If
U = random()
If r < (m - a) / (b - a) Then
    Tri = a + Sqr(U * (m - a) * (b - a))

```

Else

Tri = b - Sqr((1 - U) * (b - m) * (b - a))

End If

End Function

The Tri(kind) function has a parameter “kind” as string. It determines the values of the parameter a, m and b according to the value of “kind”. It represents the transaction type in the model. There are four kinds of transaction type as A, B, C and D in the model at Chapter 6. The parameters of the triangular distribution a, m and b are determined afterwards.

Run_Sim VBA Program for example 5.2.1.1:

Public data, Simulation As Worksheet

Public r As Double

Sub Clear_Sim()

Sheets("Simulation").Select

Range("A2:A2000,B2:B2000,C2:C2000,D2:D2000,E2:E2000").Select

Selection.ClearContents

End Sub

Sub Run_Sim()

Set Simulation = Sheet3

Set data = Sheet1

Clear_Sim

r = 0

n = data.Cells(2, 2)

For m = 1 To n

Sheets("Spreadsheet").Select

Range("B6").Select

ActiveCell.FormulaR1C1 = "=expo(Interarrival)"

Simulation.Cells(m + 1, 1) = Range("J8")

```

Simulation.Cells(m + 1, 2) = Range("J10")
Simulation.Cells(m + 1, 3) = Range("J12")
Simulation.Cells(m + 1, 4) = Range("J14")
Simulation.Cells(m + 1, 5) = Range("J16")
data.Cells(1, 1) = data.Cells(r + 1, 1)
r = 0
Next
End Sub

```

Run_Sim VBA Program for example 5.2.2.1:

```

Public data, Simulation, Simulation_que, Simulation_cust As Worksheet
Public r As Double
Sub Clear_Sim()
    Sheets("Simulation").Select
    Range("A2:A2000,B2:B2000,C2:C2000,D2:D2000,E2:E2000,F2:F2000,
        G2:G2000,H2:H2000,I2:I2000").Select
    Selection.ClearContents
    Sheets("Simulation_que").Select
    Range("B2:B100").Select
    Selection.ClearContents
    Sheets("Simulation_cust").Select
    Range("B2:B100").Select
End Sub

Sub Run_Sim()
    Set Simulation = Sheet3
    Set Simulation_que = Sheet4
    Set Simulation_cust = Sheet6
    Dim data As Worksheet
    Set data = Sheet1
    Dim Arr_que(100) As Double

```

```
Dim Arr_cust(100) As Double
```

```
Dim Max_que As Double
```

```
Dim Max_cust As Double
```

```
Clear_Sim
```

```
    r = 0
```

```
    n = data.Cells(2, 2)
```

```
    For i = 1 To 100
```

```
        Arr_que(i) = 0
```

```
        Arr_cust(i) = 0
```

```
    Next
```

```
    For m = 1 To n
```

```
        Sheets("Spreadsheet").Select
```

```
        Range("B6").Select
```

```
        ActiveCell.FormulaR1C1 = "=expo(Interarrival)"
```

```
        Simulation.Cells(m + 1, 1) = Range("K8")
```

```
        Simulation.Cells(m + 1, 2) = Range("K10")
```

```
        Simulation.Cells(m + 1, 3) = Range("K12")
```

```
        Simulation.Cells(m + 1, 4) = Range("K14")
```

```
        Simulation.Cells(m + 1, 5) = Range("K16")
```

```
        Simulation.Cells(m + 1, 6) = Range("K18")
```

```
        Simulation.Cells(m + 1, 7) = Range("K20")
```

```
        Simulation.Cells(m + 1, 8) = Range("K22")
```

```
        Simulation.Cells(m + 1, 9) = Range("K24")
```

```
        Max_cust = Range("K8")
```

```
        For i = 0 To Max_cust
```

```
            Arr_cust(i) = Cells(i + 29, 12) + Arr_cust(i)
```

```
        Next
```

```
        Max_que = Range("K12")
```

```
        For i = 0 To Max_que
```

```
            Arr_que(i) = Cells(i + 29, 15) + Arr_que(i)
```

```
        Next
```

```

        data.Cells(1, 1) = data.Cells(r + 1, 1)
        r = 0
    Next
    For i = 0 To 50
        Simulation_cust.Cells(i + 2, 2) = Arr_cust(i) / n
        Simulation_que.Cells(i + 2, 2) = Arr_que(i) / n
    Next
End Sub

```

First, variables and worksheet names are defined. The run time n is taken from the "data" excel worksheet at cell B\$2. The macro called Clear_Sim is run in order to delete the previous simulation results. Then, the simulation starts by declaring the formula in cell C1 in excel sheet. This gives the stimulus to excel for recalculating the spreadsheet. The performance measures in column K are collected at the end of each run row by row in the worksheet "Simulation". The values of no. of customers in the system and in the queue are collected also in Arr_cost and Arr_que arrays. The values in these arrays are then written in "Simulation_cust" and "Simulation_que" worksheets.

APPENDIX D: FORMULAS

In this part, formulas for Independent sampling with equal variances and Correlated sampling are given.

Independent Sampling with Equal Variances: The point estimate of the mean difference is;

$$\hat{\theta}_1 - \hat{\theta}_2 = \bar{Y}_1 - \bar{Y}_2$$

When using independent sampling, $\bar{Y}_1$ and $\bar{Y}_2$ are statistically independent; hence,

$$\begin{aligned} \text{var}(\bar{Y}_1 - \bar{Y}_2) &= \text{var}(\bar{Y}_1) + \text{var}(\bar{Y}_2) \\ &= \frac{\sigma_1^2}{R_1} + \frac{\sigma_2^2}{R_2} \end{aligned}$$

Assuming $\sigma_1^2 = \sigma_2^2 = \sigma^2$; S_p^2 is a pooled estimate of σ^2 ;

$$S_p^2 = \frac{(R_1 - 1)S_1^2 + (R_2 - 1)S_2^2}{R_1 + R_2 - 2} \text{ which has } \nu = R_1 + R_2 - 2$$

and the standard error is calculated by;

$$s.e.(\bar{Y}_1 - \bar{Y}_2) = S_p \sqrt{\frac{1}{R_1} + \frac{1}{R_2}}$$

And the c.i. for $\theta_1 - \theta_2$ is given by;

$$(\bar{Y}_1 - \bar{Y}_2) \pm t_{\alpha/2, \nu} s.e.(\bar{Y}_1 - \bar{Y}_2)$$

When n is large, t distribution approaches z distribution.

Correlated Sampling: The point estimate of the mean difference is;

$$D_r = Y_{r1} - Y_{r2}$$

$$\bar{D} = \frac{1}{R} \sum_{r=1}^R D_r$$

The sample variance and the standard error of the difference $\{D_r\}$ are computed by;

$$S_D^2 = \frac{1}{R-1} \sum_{r=1}^R (D_r - \bar{D})^2 \text{ which has } \nu = R-1$$

$$s.e.(\bar{D}) = s.e.(\bar{Y}_1 - \bar{Y}_2) = \frac{S_D}{\sqrt{D}}$$

And the c.i. for $\theta_1 - \theta_2$ is given by;

$$\bar{D} \pm t_{\alpha/2, \nu} s.e.(\bar{D})$$

When n is large, t distribution approaches z distribution.

REFERENCES

1. Grossman, T.A., "Spreadsheet Modeling Simulation Improves Understanding of Queues", *Interfaces* Vol.29, No.3, pp.88-103, <http://www.ucalgary.ca/~grossman/simulation/>, 1999.
2. Hora, S. C., "Spreadsheet Modeling of the G/G/c Queuing System without Macros or Add-Ins" *INFORMS Transactions on Education*, Vol. 3, No. 3, <http://ite.pubs.informs.org/Vol3No3/Hora/>, 2003.
3. Banks, J., J. S. Carson, II. B. L. Nelson, and D. M. Nicol, *Discrete-Event System Simulation*, Prentice Hall, Inc., 3rd edition, New Jersey, 2001.
4. Lapin, L. L., W.D. Whisler, *Quantitative Decision Making with Spreadsheet Applications*, Belmont, CA:Duxbury/Thomson Learning, 2002.
5. Pritsker, A.A. B., C.E. Sigal, and R.D.J. Hammesfahr, *SLAM II Network Models for Decision Support*, The Scientific Press, 1994.
6. Evans, J.R. "Spreadsheets as a Tool for Teaching Simulation," *INFORMS Transactions on Education*, Vol. 1, No. 1, [http://ite.informs.org/Vol1No1/ Evans/](http://ite.informs.org/Vol1No1/Evans/), 2000.
7. Seila, A.F., "Spreadsheet Simulation", *Proceedings of the 2001 Winter Simulation Conference*, 2001.
8. Mattesich, R., "Budgeting Models and System Simulation", *The Accounting Reviews*, 36:384:397, 1961.
9. Seal, K.C., Przasnyski, Z. H., and Leon, L., "A Literature Review of MS/OR Models in Spreadsheets", *OR Insight*, Vol. 13, Issue 4, pp. 21-31, 2000.

10. Evans, J. R. and D.L. Olson, *Introduction to Simulation and Risk Analysis*, Prentice Hall, New Jersey, 1998.
11. Oakshott, L., *Business Modeling and Simulation*, 1st edition, Pitman Publishing, London, 1997.
12. Swain, J. J., "1997 Simulation Software Survey", *OR/MS Today*, Vol. 24, No. 5 (October), pp.38-46, 1997.
13. Grossman , T.A. and Ingolfsson A., "Graphical Spreadsheet Simulation of Queues", *INFORMS Transactions on Education*, Vol. 2, No. 2, <http://ite.informs.org/Vol2No2/IngolfssonGrossman/>, 2001.
14. Little, J. D. J, "A Proof for the Queuing Formulas $L = \lambda w$ ", *Operations Research*, Vol. 16, pp.651-65, 1961.
15. Lehmer, D. H., *Proceedings of the Second Symposium on Large-Scale Digital Computing Machinery*, Harvard University Press, Cambridge, MA, 1951.
16. Ross, S. M., *Introduction to Probability Models*, pp.284, 7th edition, Academic Press, London, 2000.
17. Winston, W. L., *Simulation Modeling Using @RISK*, Duxbury Press, Cheshire, 1996.
18. Chase, R. B. and Aquilano, N. J., *Production and Operations Management*, 6th edition, Richard D. Irwin, Chicago, Illinois, 1992.
19. Plane, D. R., *Management Science, a Spreadsheet Approach*, Boyd and Fraser, Danvers, Massachusetts, 1994.

20. Grossman, T. A., Tran, K., Ursu, G.D., "Prospects and Techniques for Dynamic Discrete Event Simulation in Spreadsheets", *Proceedings of the IASTED International Conference on Applied Modeling and Simulation*, Banff, Canada, 1997.

