2011

Models To Estimate Arrival Counts And Staffing Requirements In Nonstationary Queueing Systems Applied To Long Distance Road Races

Lindon P. Fairweather
University of Central Florida

MODELS TO ESTIMATE ARRIVAL COUNTS AND STAFFING REQUIREMENTS IN NONSTATIONARY QUEUEING SYSTEMS APPLIED TO LONG DISTANCE ROAD RACES

by

LINDON P. FAIRWEATHER
B.A., Eastern Connecticut State University, 1999
M.A., University of Connecticut, 2002
M.S., University of Central Florida, 2005

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the department of Industrial Engineering and Management Systems
in the College of Engineering and Computer Science
University of Central Florida
Orlando, Florida

Summer Term
2011

Major professor: Charles H. Reilly

ABSTRACT

We examine the problem of staffing refreshment stations at a long distance road race. A race is modeled as a mixed queueing network in which the required number of servers at each service station has to be estimated. Two models to represent the progress of runners along a long distance road race course are developed. One model is a single-class model that allows a road race manager to staff service stations assuming the runners are identical to those in some historical dataset. Another model is a multi-class simulation model that allows a road race manager to simulate a race of any number of runners, classified based on their running pace into different runner classes.

Both the single-class model and the multi-class model include estimates for the rates at which the runners arrive at specified locations along the course. The arrival rates, combined with assumed service rates, allow us to base staffing decisions on the Erlang loss formula or a lesser known staffing rule that gives a lower bound for the required number of servers. We develop a staffing strategy that we call the Peak Arrival Staffing Bound (PASB), which is based on this staffing bound.

The PASB and the Erlang loss formula are implemented in the single-class model and the multi-class simulation model. By way of numerical experiments, we find that the PASB is numerically stable and can be used to get staffing results regardless of the traffic intensity. This finding is in contrast to the Erlang loss formula, which is known to become numerically unstable and overflows when the traffic intensity exceeds 171. We compare numerical results of the PASB and the Erlang loss formula with a blocking probability level of 5% and find that when the traffic intensity is high, staffing results based on the PASB are more conservative than staffing results based on the Erlang loss formula. As the traffic intensity gets lower, we find that staffing results based on the PASB are similar to staffing results based on the Erlang loss formula. These findings suggest that the PASB can be a valuable tool to aid race directors in making staffing decisions for races of all traffic intensities.

To Bubbles, Peppy, and Vicky

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION TO ROAD RACE MODELING

1.1 Motivating Problem

The motivating problem for this research is matching the service capacity at refreshment stations along a long distance road race course over time to the arrivals of runners at those refreshment stations. Examples of systems that are similar in principle include a convoy of vehicles that are traveling on a defined route and arriving at predetermined locations for servicing, and street parades in which participants move along a planned course with aid stations at predetermined locations.

1.2 Problem Description

The road-race course that is described in this manuscript is equipped with $K+1$ timing detectors placed at selected locations along the course, including the start and finish lines. The first detector is at the start line and the last detector is at the finish line. We let d_k , $k=1,2,\cdots,K$, denote the distance of the k^{th} detector location from the start line, d_0, and assume that $0=d_0 < d_1 < d_2 <\ldots< d_K$.

There are a total of N runners traversing the course, and data on their arrival times are only available at each of the $K+1$ detector locations. For example, there were 9336 runners in the 2004 WDW Marathon. Each of the N runners has a unique identifying number, n, for $n=1,2,\cdots,N$. Runner n enters the course at time $t_{0,n}$, and reaches d_k at time $t_{k,n}$. All runners are assumed to traverse the entire course and eventually reach the finish line (d_K) at $t_{K,n}$. Because each runner wears a chip that transmits a signal so that the current time may be recorded when that runner makes contact with a detector, we are able to collect arrival data on each runner at each detector location.

Road race managers need some method of approximating the numbers of runners that are expected to arrive at the refreshment stations over time in order to make volunteer staffing decisions. To approximate the service requirement along the race course, we model a long distance road race as a network of queues in which we consider refreshment stations to be service stations that are to be staffed with the appropriate numbers of volunteer servers for the runners that approach each refreshment station. We assume that all the runners in the race enter the queueing network at the start line, enter each station along the race course where they seek service for refreshments, and exit at the finish line. This description of the queueing system gives rise to a special type of queueing network that will be described in Chapter 2.

In this research, we address the problem of estimating the staffing requirements in time-dependent (nonstationary) traffic systems, such as long distance road races, that have high server requirements. We develop models that allow road race managers to track the flow of runners at each service station, as well as a staffing model that allows road race managers to estimate the required numbers of servers needed in a long-distance road race with L service stations.

One example of the type of traffic system that is described above is a marathon, the principal application motivating this research. The data set that will be used for the experiments herein comes from the 2004 WDW Marathon. (This data set will be described in greater detail in Chapter 4.) It is assumed here that there are $L = 25$ service stations, one located at each of the first 25 mile markers.

1.3 The Research Plan

The major contributions of this research are: (a) the development of a model for determining staffing requirements for service stations along a marathon course that is modeled as a queueing network, and is characterized by nonstationary arrival processes at each service station, and (b) the development of models to determine how runners transition along the course of a long distance road race, as well as the development of models to estimate the arrival rates of the runners at each service station. The models developed include a simulation model that allows road race managers to simulate the transition of multiple classes of runners along the course of a long distance road race that is modeled as a queueing network. Each of these models is embedded with a staffing model at each service station in the queueing network in order to provide service level requirements at each service station.

We use the Erlang loss formula as the basis for estimating staffing requirements. We also develop another staffing model that uses a modification of a staffing lower bound in Hall [17] for estimating the required numbers of servers. We call this model the Peak Arrival Staffing Bound (PASB).

The rest of this manuscript is structured in the following way. In Chapter 2, we briefly review the literature relating to the basic principles of simulation and Monte Carlo methods, queueing networks and their applications, and some of the methods that are used to model time-dependent arrival rates for no-waiting queueing systems. The Erlang loss formula is intended for this kind of queueing system; it is reviewed in Section 2.4. In Chapter 3, we develop a framework for estimating the peak arrival rates at each of the service stations in the queueing network. We also develop the PASB staffing rule that is based on a server lower bound given by Hall [17]. Chapter 4 contains experimental results based on the WDW 2004 data. In Chapter 5, we develop a simulation model of the queueing network representation of a long distance road race with multiple classes of runners. Experimental results from the simulation model are also presented in Chapter 5. Finally, in Chapter 6, we discuss the benefits and shortcomings of the models, as well as areas for future research.

CHAPTER 2
LITERATURE REVIEW

2.1 Introduction

In this chapter, we present an introduction to simulation and Monte Carlo methods, an introduction to queueing networks and some applications of queueing networks, and a review of some of the methods that are used to model time-dependent arrival rates for no-waiting queueing systems, such as the Erlang loss formula.

An exhaustive review of the literature did not turn up a single model for staffing a long distance road race such as the one described in Chapter 1. In Sections 2.2 and 2.3, we give a brief review of the literature related to estimating long distance running times and some of the relevant queueing theory literature, with special focus on some application areas of queueing networks. In Section 2.4, we review the Erlang loss formula, and give a brief review of the literature related to modeling the time-dependent arrival rate for use in the Erlang loss formula. In Section 2.5, we give an introduction to simulation and Monte Carlo methods. In Section 2.6, we present a brief summary of some application areas of simulation and Monte Carlo methods. In Section 2.7, we give a brief review of variance reduction techniques for Monte Carlo methods.

2.2 Road Race Modeling

As the popularity of recreational running increases, so does the body of literature dedicated to running. However, Rae et al. [32] notes that the current body of work around recreational running is focused primarily on the health, medical, and physical aspects of runners.

An extensive review of the endurance running literature turned up no articles related to modeling running times of long distance athletes for the purpose of race management and staffing. During the literature review, one article was found with relevance to modeling the times of runners. In this article, Strand and Boes [37] used extreme value theory (EVT) to model the times of competitive runners. EVT was used to estimate the expected rate of change in performance for a runner of any age. The authors did not discuss the impact of the availability of service at refreshment stations on the runners' predicted performance.

2.3 Queueing Networks

The queueing theory literature is replete with examples of physical systems that are modeled as queueing networks. See, for example, Kelly [23], Saaty [34], or Stidham [36]. However, a thorough review of the literature did not turn up any articles in which a long distance road race is modeled as a network of queues. For this research, we model a long distance road race as a network of queues in which refreshment stations are considered as service stations that are to be staffed with the appropriate numbers of volunteer servers. We assume that the service time duration at a service station can be approximated by any choice of continuous probability distribution and has a service rate per busy server μ, and that the time-varying arrival rates at the ℓ^{th} service station of interest, $\ell = 1, 2, \ldots, L$, are based on a nonstationary arrival process with arrival rate $\lambda_\ell(t)$. The queueing network framework is described in Section 3.1.

According to Bolch et al. [6], a queueing network is a connected queueing system that consists of at least two service stations. In such networks, the entities of interest, which may be single-class entities or multi-class entities, are transferred between service stations. The queueing network is called an open queueing network if the entities of interest are allowed to enter or exit the network at any of the service stations in the network. The queueing network is called closed if the entities of interest can neither enter nor leave the network. A mixed queueing network combines some of the features of both types of queueing networks. For example, a mixed network may only allow entities to enter or exit the network at predetermined service stations. In the next section, we cover the basics of queueing system notation.

2.3.1 Queueing Theory Nomenclature

This section covers the basics of queueing system notation. For additional details on conventions for queueing notations, see Chapter 1 of Gross and Harris [16] or Chapter 5 of Hall [17]. Most queueing systems are classified according to the Kendall-Lee notation, characterized by the following six factors: arrival process, service process, number of servers, system capacity, calling population (of potential customers), and queue discipline. Such a classification is represented by a series of six parameters separated by slashes, as shown below:

$$A / B / s / m / n / x$$

where

A represents the interarrival-time distribution

B represents the probability distribution of service times

s represents the number of parallel service channels

m represents the system capacity

n represents the size of the calling population

x represents the queue discipline

Borrowing from Hall [17], some of the standard probability distributions are denoted as follows:

$M =$ exponential (Markovian) distribution

$G =$ general distribution

$D =$ constant (deterministic) service times

For example, an $M / G / s / s / n / FIFO$ queue has exponential interarrival times, general service times, s servers, a system capacity of s, a calling population of n, and FIFO queue discipline. The $M(t) / G(t) / s / s / n / LIFO$ queue has time-dependent exponential interarrival times, time-dependent general service times, s servers, a system capacity of s, a calling population of n, and LIFO queue discipline.

2.3.2 Examples of Queueing Network Models

Applications of queueing models abound in the literature; however, we have not seen any applications of a queueing network representation of a long distance road race. We now examine some of the application areas of queueing theory. A compendium of research related to queueing networks can be found in Kelly [23].

2.3.2.1 *Communication Network*

An extensive coverage of communication networks is given by Reiser [33] and Kelly [23]. Stidham [36] gives a condensed coverage of the subject. Whitt [40] derives and describes some functional limit theorems that are applicable to communication networks.

Communication networks are telegraph systems in which a graph represents the system with vertices and directed edges corresponding to cities and directed channels, respectively. In a communication network, messages originate in a city and are to be transmitted, possibly via some intermediate relay city or cities, to their destination city. Each channel in the communication network has a maximum capacity, and the various messages interact.

In the queueing network representation of a communication network, each message is regarded as a customer and each channel is regarded as a queue. Messages are assumed to arrive from outside the system in independent Poisson streams. Additionally, each message is assumed to have its own route through the channels of the system.

Kelly [23] discusses two models for the time that it takes a message to pass along a channel. The system has L channels in which n_ℓ represents the number of messages waiting at the ℓ^{th} channel (including the message being transmitted), and $n_1, n_2, \ldots, n_L$ are assumed to be independent. The first model assumes that the time that it takes a message to pass along the ℓ^{th} channel is distributed as an exponential random variable with mean μ_ℓ, and is independent of the time that it takes to pass along other channels along its route in the communication network. The author derived an analytical expression for the steady state behavior of this type of communication network.

The second model assumes that the service requirement for a message along its route in the communication network is the same at every queue, follows a general distribution with mean μ_ℓ, and is independent from customer to customer. The author derived an expression for the steady state behavior of this type of communication network that shows that the mean number of messages waiting at the ℓ^{th} queue is $\lambda_\ell / (\mu_\ell - \lambda_\ell)$, and the mean waiting time of a message at the ℓ^{th} queue is $1 / (\mu_\ell - \lambda_\ell)$.

2.3.2.2 *Machine Interference*

The basic form of the machine interference problem is described by Kelly [23], and an extensive treatment is discussed by Saaty [34]. For this type of network, there are N machines under the care of a single operator. From time to time, a machine stops working and requires the attention of the operator before it can resume working. The operator can only attend to one machine at a time; hence, the machines interfere with each other if two or more machines are stopped and must wait on the attention of the operator.

The system is represented as a closed queueing network in which there are N identical machines, and the operating time of a machine before it is stopped is represented by a general distribution with mean μ_n, and the time that it takes the operator to service a machine is distributed as an exponential random variable with mean μ. The queue discipline is FIFO. The machines that are stopped queue to receive service, and all service times and operating times are assumed to be independent.

Kelly [23] derived an analytical expression that describes the steady state behavior of the system, which makes it possible to calculate the proportion of time that the operator is busy, or the amount of time a machine is busy.

2.3.2.3 *Timesharing Computers*

Modeling timesharing computers as a network of queues is described in Chapter 4 of Kelly [23], as well as by Huessmann and Goldberg [19] who describes programs for simulating computer systems, and gives a very good description of time sharing computer systems.

In timesharing computer systems, there is a central processing unit (CPU) of the computer to which L terminals are attached. These terminals are assumed to be queues. There are N jobs, which are considered to be customers. Job $n, n=1, 2, \ldots, N,$ is either being serviced by the CPU or is with the computer at terminal $\ell, \ell=1, 2, \ldots, L$. The service times for all queues in this system, including at the CPU, follows a general distribution.

2.3.2.4 *Teletraffic Network*

According to Whitt [38] and Whitt et al. [41], the field of teletraffic networks has benefited significantly from the developments in queueing theory. Kelly [23] shows a teletraffic network can be modeled as an open queueing network. The typical teletraffic network has J exchanges, $e_1, e_2, \ldots, e_J$, that are connected to exchange f, which is connected to exchange g. There are R_j lines between e_j and f, and S lines between f and g. $R_j \leq S$ and $S < R_1 + R_2 + \ldots + R_J$. Calls that require a line between e_j and g are initiated as a Poisson process with intensity λ_j; however, these calls are lost if all of the lines from e_j to f or all the lines from f to g are busy. Kelly [23] derived an expression that describes the steady state behavior of the teletraffic network.

2.3.2.5 *Compartmental Models*

A comprehensive coverage of compartmental models can be found in Allen [2], Kelly [23], and Stidham [36]. The queueing network representation of compartmental models give rise to systems that have multiple compartments that are visited by customers. These systems have the property that after customers enter the system, they move independently through it. We discuss two types of compartmental models below.

2.3.2.5.1 *Biology Compartmental Model*

Reviews of the use of compartmental models in the field of biology can be found in Kelly [23] and Stidham [36]. A very comprehensive review of biology compartmental models is covered by Allen [2], who uses a biology compartmental model in his doctoral research to model tumor growth.

Compartmental models are used in biology to model the movement of particles through the various parts of an animal's body. In such systems, there are J compartments, and particles enter the system, move around the various compartments, and then exit the system. The arrival rate of particles to the system is governed by a Poisson process with intensity λ, and the time that a particle spends in the j^{th} compartment of the system is well approximated by any general continuous random variable with mean α_j. Kelly [23] shows that, in steady state, the number of particles in the j^{th} compartment is independent of the number of particles in the other compartments, and is distributed as a Poisson random variable with mean $\lambda\alpha_j$.

2.3.2.5.2 *Birth-Illness-Death Compartmental Model*

The birth-illness-death compartmental model is summarized by Stidham [36], and discussed in greater detail by Allen [2] and by Kelly [23].

In birth-illness-death compartmental models, an individual is born, passes through various states of health, and finally dies. There are J states of the system, which are chosen at birth, and the individual moves independently through each of them. The length of time that the individual spends in the j^{th} state follows a general distribution.

2.3.2.6 ***Some Other Applications of Queueing Network Models***

An open queueing network model of a computer system with virus infection is described by Chao [8]. In this application, a job that is infected with a virus enters the system, and with some probability, transmits the virus when it is transferred to other computer processors in the network (for example, CPU, diskettes, I/O, etc.). The service time distribution at each service station in the network is approximated by independent exponential distributions, and the arrival rate at each service station in the network is modeled as independent Poisson processes.

When a job finishes its service at a service station in the network, it either goes to another service station for service, or it leaves the system. However, when a job goes to another service station, there is a non-zero probability that it brings a computer virus to that service station. Computer viruses can also arrive from outside the system, for example, jobs arriving on an infected floppy diskette. The arrival of a computer virus arrives at a service station leads to the immediate destruction of all the customers at that service station, and the server is sent for repair. The time to repair a server is assumed to be negligible, and the system resumes operation once the server repair is completed.

George and Xia [10] describe a closed queueing network model of a vehicle rental operation that has multiple rental locations in its vehicle rental network. In this application, each rental location is modeled as a service location. The arrival rate at each service station in the queueing system is approximated by an independent Poisson process. Customers enter the system and rent a vehicle that they keep for a period of time that follows a general distribution. The vehicle is returned to one of the service stations in the network. If a vehicle is unavailable when a customer arrives at a service station, the customer immediately leaves the system; thus, there is no queueing at the service stations. Using this framework, with respect to fleet size at an arbitrary service station, the authors derived the asymptotic properties of vehicle availability.

Bar-Lev et al. [4] describe an open queueing network model of a HIV blood testing operation. In this model, blood samples undergo two different tests – an enzyme linked immuno-sorbent assay (or ELISA) test and a polymerase chain reaction (or PCR) test – before the blood samples are assessed to be negative for the HIV virus. There is a processor-sharing service station and a delay service station. Service times at both service stations are assumed to be exponentially distributed, and the arrival rate at the service stations are approximated by independent Poisson processes.

Blood samples arrive at the processor-sharing service station for ELISA testing. If the blood samples are found to be positive for the HIV virus, they leave the system. Otherwise, the blood samples are forwarded to the delay service station where they are subjected to PCR testing. The blood samples then they leave the system either as HIV positive or HIV negative.

Haskose et al. [18] describe a queueing network model of a manufacturing production process. In this application, a job shop is formulated as an open queueing network with multiple service stations. The service discipline at each service station is FIFO. Arrivals at the service stations are modeled as independent Poisson processes, and the service times at each service station in the network are modeled by independent exponential distributions.

Jobs enter the system at the first service station in the network, and if there are other jobs waiting for service, the arriving jobs queue and wait for service. Otherwise, the arriving jobs enter service. When service is completed, the jobs are transported to the next service station in the routing sequence. This process is repeated until the jobs visit all their predetermined service stations in the system.

A second queueing model of a manufacturing process is described by Jain et al. [21]. In this application, a flexible manufacturing system in which multiple material-handling devices and multiple pallets circulate among a fixed number of loading and unloading processing stations is modeled as a closed queueing network. Service times at the various service stations are modeled by independent exponential distributions, and parts are transported to processing stations according to predetermined routing schedules or process plans.

During the parts transportation process, a pallet makes several visits to various service stations in the system. When the processing of the pallet is completed at a service station, it (the pallet) awaits the arrival of a material-handling device. When the material-handling device gets to the service station, it is seized by the pallet and immediately exits the station. If a service station downstream is available, it is seized by the pallet. Otherwise, the pallet moves to the central buffer and awaits the availability of a downstream service station. Once a downstream service station becomes available, the pallet reserves it and the material-handling device moves towards the downstream service station. At the unloading station, the pallet is unloaded, and once empty, it moves to a loading station.

A third queueing network model of a manufacturing process is described by Juang and Huang [22], in which a semiconductor manufacturing process is modeled as an open, multi-class, priority $G/G/s/\infty/\infty/\infty$ queueing network. Each tool in the semiconductor manufacturing process is modeled as a single-queue station with infinite buffer space. Lots are modeled as individual entities that originate from outside the network and receive service at predetermined stations. Then they leave the system. The queueing network model includes a system analyzer module and a system predictor module. The arrival and service patterns of each tool group are analyzed by the system analyzer module. The system predictor module provides forecasts of the performance measures of interest (for example products cycle time, lots remaining cycle time, tool utilization, queue length, tool moves, and stage moves).

An open queueing network model of a railway network in which trains traverse predetermined routes is described by Huisman et al. [20]. The system consists of multiple multi-server service stations, each having a FIFO service discipline. The arrival rates of trains at a service station along its predetermined route are modeled as independent Poisson processes, and the service times at these service stations are modeled as independent exponential distributions.

Muduli and Yegulalp [29] describe a closed, multi-class queueing network model of a truck-shovel system of an open-pit operation. In this system, shovels, haulage roads, ore dumps, and repair centers are modeled as service stations that are visited by the trucks. Trucks go to a shovel, get loaded, then go to an ore or waste dump and return to the same or a different shovel. The service discipline is FIFO at each service station, and the arrival rates and service rates at the various stations in the network follow independent Poisson processes and independent general distributions, respectively.

Park and Lee [30] describe a closed queueing network model of a maintenance network. The network is made up of three nodes - a parts inventory system, a repair depot, and a base. Items in operation are subject to failure, and the operating times of any item before failure at a base in the network are assumed to be exponentially distributed. Items that fail require a part, which must be obtained from the parts inventory node. The immediate repair of a failed part depends on the availability of that part at the parts inventory node. If a replacement part is available at the parts inventory for the failed part, then the failed item is immediately sent to the repair depot. Otherwise, the failed item awaits the arrival of a replacement part at the parts inventory. It enters the repair depot upon the arrival of the replacement part at the parts inventory. The repair depot is equipped with multiple, identical repair channels, and times between repairs are assumed to be Poisson distributed. Lead times between orders are also assumed to follow a Poisson distribution.

2.4 Erlang Loss Formula

The life and works of A.K. Erlang are detailed in Brockmeyer et al. [7]. In 1917, Erlang, an employee of the Copenhagen Telephone Exchange, published the following loss formula for an $M/M/s/s/\infty/FIFO$ queue:

$$P_b = \frac{\dfrac{\left(\lambda/\mu\right)^s}{s!}}{\displaystyle\sum_{i=0}^{s}\frac{\left(\lambda/\mu\right)^i}{i!}} = \frac{\dfrac{r^s}{s!}}{\displaystyle\sum_{i=0}^{s}\frac{r^i}{i!}}$$

(2.1)

where λ is the arrival rate, μ is the service rate per busy server, s is the number of servers and the system capacity, and $r = \lambda/\mu$ is the traffic intensity.

The Erlang loss formula gives the blocking probability or the proportion of customers that are lost (or cannot be served) because all servers are busy when a customer arrives. Alternatively, equation (2.1) can be used to (heuristically) determine the minimum number of servers for a given blocking probability. According to Brockmeyer et al. [7], the

Erlang loss formula has been used with much success to determine staffing levels of telephone call centers. Whitt [40] used a modified version of the Erlang loss formula to staff a call center. The Erlang loss formula will be used as the benchmark for the staffing model that is proposed in Section 3.4.

Massey and Whitt [26 - 28] observe that the most restrictive assumption in the $M/M/s/s/\infty/FIFO$ model is the constant arrival rate assumption. Massey and Whitt [26] developed a stationary process approximation for use with the Erlang loss formula when the arrival process is known to be nonstationary. However, some of the articles that are reviewed later in this chapter present significant progress toward relaxing this assumption.

A second problem that affects the Erlang loss formula is that it becomes unstable as r^s grows. Qiao and Qiao [31] showed that equation (2.1) becomes unstable as the parameter s increases. Qiao and Qiao [31] showed that 171! overflows to $+\infty$. Additionally, Qiao and Qiao [31] showed that equation (2.1) becomes unstable for $s > 154$ because r^{155} overflows to $+\infty$. Therefore, for systems with high server requirements, and by "high server requirement" we mean $s > 154$, equation (2.1) is not practical.

In the next few sections, we summarize some approximation methods for nonstationary arrival processes that have been used to model the arrival rate in loss models such as the Erlang loss formula.

2.4.1 The Simple Stationary Approximation (SSA) Method

Alnowibet and Perros [3] observed that the SSA method is applicable to a wide variety of queueing systems. Green et al. [12] show that the SSA reasonably approximates nonstationary systems with arrival rates that do not deviate more than 10% from the average arrival rate at any time.

While the SSA method reasonably approximates nonstationary systems that do not deviate more than 10% from the average arrival rate at any time, Alnowibet and Perros [3] discussed its shortcomings with respect to the underestimation of the average performance measures of nonstationary systems with arrival rates that deviate more than 10% from the mean arrival rate. This finding was corroborated by Green et al. [12] who numerically investigated the level of nonstationarity at which the SSA method provides reasonably accurate average performance measures.

The applicability of the SSA method to relatively small problems, as pointed out by Green and Kolesar [13, 14] seems to imply that it would not be appropriate for the type of problem that is addressed herein.

2.4.2 The Pointwise Stationary Approximation (PSA) Method

Grassman [11] used the PSA method to estimate upper bounds on the expected number of entities in a queue, and Green and Kolesar [13] used the PSA method to approximate steady state average performance measures for queues with multiple servers and periodic arrival rates. Green and Kolesar [14] used the PSA to get approximate steady-state average performance measures for the $M(t)/M/s/s/\infty/FIFO$ queue with periodic arrival input function. Whitt [38] showed that the PSA gives an upper bound on the expected number of entities in the system and delay probabilities when the system is stable (with traffic intensity less than one).

Alnowibet and Perros [3] noted that the PSA method can be applied to many queueing systems; however, it requires that $r < 1$ for all times t. Alnowibet and Perros [3] also observed that implementation of the PSA method leads to an overestimation of the peak of the average number of entities, regardless of the type of queueing system involved.

2.4.3 The Simple Peak Hour Approximation (SPHA) Method

Green and Kolesar [15] proposed the Simple Peak Hour Approximation (SPHA) method. The SPHA first uses the PSA method to obtain a performance measure of interest. Then, the time at which the performance measure achieves its maximum value is estimated. The average of the performance measure of interest over some time interval of interest in which the peak time is the center of the interval is the SPHA value for the performance measure of interest.

It will be seen in Chapter 3 that the proposed models make use of a variation of the SPHA in determining the maximum arrival rate at a service station. The model that is proposed herein then uses the arrival estimate to estimate different levels of the stationary arrival rates for the system.

2.4.4 The Fixed Point Approximation (FPA) Method

The fixed point approximation (FPA) method was proposed by Alnowibet and Perros [3] to analyze nonstationary loss queues in which the arrivals are governed by nonhomogeneous Poisson processes and the service times are Markovian. This method can be used to calculate time-dependent performance measures, such as the mean number of entities in the nonstationary loss system as well as the blocking probability functions in the nonstationary loss system.

Alnowibet and Perros [3] noted that the mean number of entities in the $M(t)/M/s/s/\infty/FIFO$ system at an arbitrary time t can be approximated by assessing the difference between the arrival and departure rates at time t.

2.5 Simulation and Monte Carlo Methods

One of the contributions of this dissertation is the development of a multi-class simulation model of a long distance road race in which runners are classified based on their overall anticipated pace for the entire race. We develop a simulation model of a long distance road race, and use a Monte Carlo method to estimate the number of runners in each class that arrive at each service station of interest for service. A review of the literature did not turn up any similar models for estimating the arrival rate of runners of different classes that arrive at service stations in a queueing network.

2.5.1 Simulation

According to Dagpunar [9], simulation is a broad collection of methods and applications that are usually done on a computer with specialized software that use pseudo-random numbers. The computer is used to evaluate the model numerically by repeatedly replicating the model and collecting data after each replication for analysis. The pseudo-random number generator allows for the sampling of a random number sequence $x_1, x_2, \dots$ from the continuous uniform distribution such that $x_i \sim U(0,1)$ for all i, and x_i is independent of x_j for all $i \neq j$.

2.5.2 Monte Carlo Methods

In this section, we present a brief introduction of the Monte Carlo method, which is covered in great detail by Lafortune [24], who noted that Monte Carlo methods can be regarded as methods or techniques for numerical integration. Monte Carlo integration relies on the existence of the square-integrable function, $f \in L^2(0,1)$, which implies that $\int_0^1 f^2(x)dx$ exists.

If one is interested in the following estimand:

$$I = \int_0^1 f(x)dx, \tag{2.2}$$

then, given a uniform random number generator, equation (2.2) can be estimated by drawing a sample of Q uniform random numbers, $x_1, x_2, \ldots, x_Q$, and then using them to evaluate the function $f(\xi)$, which is known as the primary estimator. Lafortune [24] shows that the expected value of the primary estimator, $E[f(\xi)]$, is equal to the actual integral, and is consequently unbiased. Therefore, the expected value of the primary estimator, μ_P, is given as follows:

$$\mu_P = E[f(\xi)] = \int_0^1 f(x)dx$$

(2.3)

The variance of the primary estimator, σ_P^2, is given as follows:

$$\sigma_P^2 = V[f(\xi)] = \int_0^1 [f(x) - I]^2 dx$$

(2.4)

The primary objective of Monte Carlo methods is to minimize the variance of the primary estimator, which, as shown by Dagpunar [9] and by Lafortune [24], can be accomplished by taking W independent samples, and creating another estimator by averaging the primary estimators. This new estimator is called a secondary estimator, I_S, which is the sum of the primary estimators, and can be written as follows:

$$I_S = \sum_{w=1}^{W} \int_0^1 \frac{f(x)}{W} dx = \sum_{w=1}^{W} I_w \ , \tag{2.5}$$

Lafortune [24] shows that relative to the primary estimator, the variance of the secondary estimator, σ_S^2, is reduced by a factor of W. That is,

$$\begin{aligned} \sigma_S^2 = V[I_S] &= \int_0^1 \dots \int_0^1 [\frac{1}{W} \sum_{w=1}^{W} f(x_w)]^2 dx_1 \dots dx_W - I^2 \\ &= \frac{1}{W} \int_0^1 f^2(x) dx - \frac{1}{W} I^2 = \frac{\sigma_P^2}{W} \end{aligned} ,$$

(2.6)

and has a standard error that is proportional to $1/\sqrt{W}$. Therefore, variance reduction techniques, two of which we discuss in Section 2.7, are concerned with reducing the variance of the primary estimator.

2.6 Some Applications of Monte Carlo Methods and Simulation

A review of the literature turned up no applications of a simulation model of a long distance road race that uses Monte Carlo methods to estimate the required number of servers at each service station of interest. In the next few sections, we briefly review some areas in which Monte Carlo methods and simulation have been used to estimate parameters and random variables of interest in a variety of fields.

2.6.1 Monte Carlo Methods and Simulation: Biochemistry

According to Berney and Danuser [5], Monte Carlo simulation is widely used in biochemistry and biological research to model molecular activity. In one application, Berney and Danuser [5] developed a simulation model of the fluorescence resonance energy transfer (FRET) technique, which is used to measure how two molecules interact with each other. In another application, LeBlanc et al. [25] described how important principles of a complex molecular system were better understood through the development of a Monte Carlo simulation model of the molecular system.

2.6.2 Monte Carlo Methods and Simulation: Environmental and Water Engineering

Monte Carlo methods and simulation techniques have been used in the field of environmental and water engineering to understand important principles of such systems. In one application, Abhulimen [1] developed a simulation model of a reservoir system, and compared its output to that of a standard deterministic reservoir engineering model, the Darcy Equation, which describes the flow of a fluid through a porous medium. The results show that the simulation model provided reliable estimates of a reservoir system's performance. Additionally, by varying input parameters of the simulation model, insights into areas of strength and weaknesses of the reservoir system were acquired.

2.6.3 Monte Carlo Methods and Simulation: Maritime Science

Monte Carlo methods and simulation techniques have been applied to the field of maritime science. Santos et al. [35] proposed a simulation model for assessing damaged ship survivability. The probability that a ship survives, conditional on the severity of the damages that the ship sustained, is evaluated using a model that is proposed by the International Maritime Organization, which is the conventional approach. In their simulation model, Santos et al. [35] proposed probability models to generate estimates of the severity of the damage to a ship, and the location of the damage. Data to parameterize the probability models was acquired. Additionally, Santos et al. [35] proposed probability models to estimate the vertical position of an affected vessel, the center of gravity of the vessel, and the state of the sea at the moment of accident. The proposed simulation methodology was applied to a passenger roll-on, roll-off ship, and estimates of the variables of interest were estimated. Additionally, by conducting sensitivity analyses on the various model inputs, the authors were able to gain important insights into damaged ship survivability.

2.7 Variance Reduction Techniques

According to Dagpunar [9], a "simple" Monte Carlo method is one that reduces the variance of the estimate by evaluating a large number of independent samples and averaging the results; however, as was pointed out earlier, the variance in (2.5) decreases as $1/\sqrt{W}$. The goal of variance reduction techniques is to improve the rate of variance reduction.

An excellent coverage of variance reduction techniques is found in Chapter 5 of Dagpunar [9]. In this section, we summarize two well known variance reduction techniques, importance sampling and stratified sampling, both of which, as shown by Dagpunar [9], typically result in higher accuracy of the Monte Carlo integral while using fewer function evaluations, at least in some instances.

2.7.1 Importance Sampling

According to Dagpunar [9], the basic idea behind importance sampling is to use a simple, well-known function to approximate a more complex function from which it may be difficult to obtain samples, perhaps because of the complexity of the function. Lafortune [24] shows that if an integrand f can be constructed by multiplying two different functions h and g, where h is almost constant and g has positive support, then the integral over the multidimensional surface V is as follows:

$$\int f\,dV = \int (f\,/\,g)\,g\,dV = \int h\,g dV \quad , \tag{2.7}$$

which implies that one can integrate f by taking samples from h instead, with nonuniform density gdV. Dagpunar [9] shows that a good choice for h, the sampling density, is one that has a shape that is similar to f, the integrand of interest.

2.7.2 Stratified Sampling

According to Dagpunar [9], the basic idea behind stratified sampling is to split the integration domain of a complicated function that one is trying to estimate with Monte Carlo integration into several strata or subdomains, and estimate the partial integral in each subdomain. Lafortune [24] shows that the variance resulting from stratified sampling is the sum of the variances of each stratum, and is always less than or equal to the simple Monte Carlo integral with the same number of samples.

2.8 Chapter Summary

The objective of this research is to develop models to estimate staffing requirements in time-dependent traffic systems such as long distance road races that have high server requirements. We propose a queueing network model of a long distance road race in which we consider refreshment stations as service stations that are to be staffed with the appropriate numbers of volunteer servers at each refreshment station.

We reviewed the literature to assess the gap, if any, that the proposed queueing network modeling framework of a long distance road race would fill. The literature review did not turn up any similar modeling approach for the problem that this research addresses. This finding suggests that, as far as we know, no other researcher has developed a queueing network model of a long distance road race to use as a management tool, or a management framework to handle the staffing needs for a long distance road race.

In Chapter 5, we develop a multi-class simulation tool that incorporates the principles that we reviewed in this chapter. We model a long distance road race as a mixed $M_\ell(t)/G_\ell/s_\ell/s_\ell/N/FIFO$ queueing network in which we assume that the service time duration at the ℓ^{th} service station can be approximated by any choice of continuous probability distribution, and where s_ℓ, the required number of servers at the ℓ^{th} service station, can be estimated (heuristically) by using (2.1), the Erlang loss formula. As will be seen in Chapter 5, a simple Monte Carlo method is used to estimate the number of runners in each class that arrive at each service station of interest for service. The multi-class simulation tool allows the race to be repeated as many times as the user specifies (that is, for multiple runs of the same race), and for the results to be averaged, thereby potentially reducing the variance of the variable of interest.

Therefore, this dissertation will help to address the staffing needs of the long distance running community, as well as add to the knowledge of approximating and staffing certain types of systems that can be modeled as nonstationary queueing networks. Examples of such systems include street parades or processions, military convoys, and so on.

CHAPTER 3
MODEL DEVELOPMENT AND METHODOLOGY

3.1 <u>Introduction</u>

The long distance road race that we model in this dissertation is equipped with L service stations, located at each mile marker beginning at Mile 1 and ending at Mile 25. We assume that runners enter the course, traverse the entire length of the course, and seek service (for refreshments) at each of the L service stations. Then the runners exit the course at the finish line. Because runners enter the course at the beginning of the race, seek service at each of the L service stations, are not able to enter or exit at any of the service stations, and exit the course at the end of the race, the system can be defined as a mixed $M_\ell(t)/G_\ell/s_\ell/s_\ell/N/FIFO$ queueing network in which we assume that the service time duration at the ℓ^{th} service station can be approximated by any choice of continuous probability distribution, and where s_ℓ, the number of servers at the ℓ^{th} service station, can be estimated (heuristically) by using (2.1). In Section 3.3, we will develop a staffing rule that can be used as an alternate to (2.1) to bound s_ℓ.

The key assumptions that we make in this chapter are listed below.

- The pace at which a runner travels for all points between any two consecutive detector locations, d_{k-1} and d_k is constant.
- There that there are $L=25$ service stations, one located at each of the first 25 mile markers.

- The long distance road-race course is equipped with $K+1$ timing detectors placed at selected locations along the course, including the start and finish lines. The first detector is at the start line and the last detector is at the finish line.
- There are a total of N runners traversing the course, and data on their arrival times are only available at each of the $K+1$ detector locations.
- All runners traverse the entire course, seek service at each of $L=25$ the service stations, and eventually reach the finish line.

The remainder of this chapter is organized as follows: In Section 3.2, we develop methodologies to estimate the paces and running times of the runners in the race, as well as a methodology to estimate the peak arrival rates at each service station in the $M_\ell(t)/G_\ell/s_\ell/s_\ell/N/FIFO$ queueing network. In Section 3.3, we develop a methodology to estimate the staffing requirements at each service station in the $M_\ell(t)/G_\ell/s_\ell/s_\ell/N/FIFO$ queueing network.

3.2 A Model of a Road Race Traffic System Based on Available Historical Data

Given historical road race traffic data, the procedure that is described in this section allows us to estimate the peak arrival rates at each service station in a long distance road race. In Section 3.2.1, we estimate the pace of a runner. In Section 3.2.2, we estimate the distance that is traveled by a runner at a specified cumulative time. In Section 3.2.3, we develop a procedure to estimate the arrival rates of runners at service stations. In Section 3.2.4, we develop a procedure to estimate the peak arrival rate at each service station.

3.2.1 Estimating the Pace of a Runner

Each runner's arrival time at d_k is denoted by $t_{k,n}$, $k=0,1,2,\ldots,K$; $n=1,2,\ldots,N$. Let $p_{k,n}$, $k=1,2,\ldots,K$; $n=1,2,\ldots,N$, denote the pace of the nPthP runner between detector locations d_{k-1} and d_k for all k and all n. Then the pace at which the runner travels between detector locations d_{k-1} and d_k is given by the following equation:

$$p_{k,n} = \frac{d_k - d_{k-1}}{t_{k,n} - t_{k-1,n}}$$

(3.1)

We assume that $p_{k,n}$ is constant for all points between any two consecutive detector locations, d_{k-1} and d_k. Because there are $K+1$ detector locations, for each runner that exits from the queueing network, there are K pace calculations of the form shown in equation (3.1) for each runner n.

3.2.2 Estimating Running Times

For a long distance road race with $K+1$ detectors placed at arbitrary locations along its course, the cumulative time that it takes the n^{th} runner to get to the ℓ^{th} service station from the start line can be approximated by the following equation:

$$T_{\ell,n} = \begin{cases} \dfrac{\ell}{p_{1,n}} & \ell \le d_1 \\ \dfrac{d_1}{p_{1,n}} + \dfrac{\ell - d_1}{p_{2,n}} & d_1 < \ell \le d_2 \\ \vdots & \vdots \\ \sum_{k=1}^{K-2} \dfrac{d_k}{p_{k,n}} + \dfrac{\ell - d_{K-2}}{p_{K-1,n}} & \ell > d_{K-1} \end{cases} \qquad (3.2)$$

Applying (3.2) to the data for the 2004 WDW Marathon, which, excluding the detector at the start line, had detectors at miles 10, 13.109375, 20, and 26.21875 (finish), the approximate the cumulative time that it took the n^{th} runner to get to the ℓ^{th} service station from the start line is given by the following equation:

$$T_{\ell,n}=\begin{cases}\dfrac{\ell}{p_{1,n}} & \ell\le 10\\ \dfrac{10}{p_{1,n}}+\dfrac{\ell-10}{p_{2,n}} & 10<\ell\le 13.109375\\ \dfrac{10}{p_{1,n}}+\dfrac{3.109375}{p_{2,n}}+\dfrac{\ell-13.109375}{p_{3,n}} & 13.109375<\ell\le 20\\ \dfrac{10}{p_{1,n}}+\dfrac{3.109375}{p_{2,n}}+\dfrac{6.890625}{p_{3,n}}+\dfrac{\ell-20}{p_{4,n}} & \ell>20\end{cases}.$$

(3.3)

Equation (3.3) is helpful in characterizing the rates of arrivals of the runners at service stations of interest in the long distance road race applications that are developed and presented in this manuscript.

3.2.3 Estimating the Arrival Rates

Because the paces of the runners are different, the *N* runners are spread out along the race course. To estimate the arrival rate at any refreshment station ℓ, one can employ the procedure in Chapter 5 of Hall [17], and count the number of arriving runners in the interval $[a,b]$ during every τ time frame for some specified period of time G. These counts are converted into a time-dependent arrival rate, $\lambda(t)$, by dividing each of the interval counts by τ / G. Therefore, if the number of arrivals in a selected time interval is α, then the arrival rate at service station ℓ is estimated as $\lambda_\ell(t)$. That is, $\lambda_\ell(t)=\alpha/_{\tau/G}=\alpha G/_{\tau}$.

The arrival rate for the process that we have developed can be derived in a similar fashion. For example, suppose a race manager collects a total of J counts, each one in a time interval that is τ time units long, at some location ℓ. Suppose further that the number of arrivals in the first, second, and up to the final time interval are $\alpha, \beta, \ldots, \upsilon$, respectively. Then the time-varying arrival rates at location ℓ are computed as,

$$\lambda_\ell(t) = \begin{cases} \dfrac{\alpha}{\tau/G} & 0 < t \le \tau \\ \dfrac{\beta}{\tau/G} & \tau < t \le 2\tau \\ \vdots & \\ \dfrac{\upsilon}{\tau/G} & (J-1)\tau < t \le J\tau \end{cases}$$

(3.4)

For the case where $\tau = G = 1$ minute, the arrival rates are the observed counts per minute. For example, the arrival rate for time $0 < t \le 1$ is α runners per minute. As pointed out by Hall [17], the choice of the length of a time frame is usually a matter of practical judgment. The physical context of the modeling problem should dictate the frame length so that whatever it is that one is trying to capture is well represented. For example, if a traffic engineer was interested in the number of automobiles that pass a point on a street, a time frame of, say, one second may be appropriate if the street of interest is a busy thoroughfare in a major city, but a longer time frame may be appropriate if the street is in a rural area with light traffic.

In the next section, we develop a procedure that uses observed per minute arrival rates to estimate the time of peak arrivals and the associated rate of arrivals. The estimates from this procedure can be used as inputs for (2.1) and for the staffing model that is described in Section 3.3. The procedure that is developed in the next section makes use of (3.3) and (3.4).

3.2.4 Procedure 1: A Procedure to Estimate Peak Arrival Rates from Historical Data

In this section, we develop a procedure for estimating the maximum arrival rate, λ_ℓ^*, at the ℓ^{th} service station from sample data. We denote this estimate as $\overset{\wedge}{\lambda}_\ell^*$. Given sample historical data, do the following:

Step 1: Begin by setting up a time-counts table where time (in minutes) is recorded in the first column, and the number of arriving runners at each of the L service stations is recorded in the L other columns. (See Table 1)

Step 2: Increment distance in (3.3) and record the time of arrival of each runner at each service station until all the runners have exited the course.

Step 3: Count the number of arriving runners at each service station ℓ in each one minute interval. This is the per minute arrival rate, and it is recorded in the appropriate column in a time-counts table such as the one shown in Table 1.

Table 3.1: Time-Counts Table

Minute	Location 1	Location 2	…	Location L
1	$\lambda_1(1)$	$\lambda_2(1)$	…	$\lambda_L(1)$
2	$\lambda_1(2)$	$\lambda_2(2)$	…	$\lambda_L(2)$
3	$\lambda_1(3)$	$\lambda_2(3)$	…	$\lambda_L(3)$
4	$\lambda_1(4)$	$\lambda_2(4)$	…	$\lambda_L(4)$

Step 4: When all the runners' times have been recorded for each $\ell,\ \ell=1,2,\ldots,L$, record the maximum value in each column of the time-counts table and the time associated with this value. The estimated maximum arrival rate is denoted $\hat{\lambda}_\ell^*$, and the time associated with $\hat{\lambda}_\ell^*$ is denoted $\hat{t}_\ell^*$.

Suppose that $\hat{\lambda}_\ell^*$ is an estimate of an arrival rate λ_ℓ^* for a Poisson process. Then, as suggested by Green and Kolesar [15], and by Hall [17], $\hat{\lambda}_\ell^*$ can be used as an estimate for the arrival rate parameter in the Erlang loss formula. We use $\hat{\lambda}_\ell^*$ in the long distance road race applications that are developed and presented in this manuscript.

If we assume that the service time duration at the ℓ^{th} service station can be approximated by any choice of continuous probability distribution, then we can model the long distance road race traffic system as a mixed $M_\ell(t)/G_\ell/s_\ell/s_\ell/N/FIFO$ queueing network in which s_ℓ is the number of servers at the ℓ^{th} service station, and can be estimated (heuristically) by using (2.1). In the next section, we will develop a staffing rule that can be used to bound s_ℓ.

Table 2: Peak Arrival Rates and Time of Peak Arrivals

Service Station	Peak Arrival Rate	Time of Peak Arrivals (minutes)
Mile 1	1381	12
Mile 2	741	22
Mile 3	497	33
Mile 4	379	43
Mile 5	310	58
Mile 6	252	66
Mile 7	227	81
Mile 8	201	85
Mile 9	182	107
Mile 10	173	115
Mile 11	152	117
Mile 12	153	137
Mile 13	144	138
Mile 14	142	160
Mile 15	124	171
Mile 16	113	173
Mile 17	109	183
Mile 18	104	194
Mile 19	97	219
Mile 20	96	222
Mile 21	91	236
Mile 22	86	246
Mile 23	82	257
Mile	90	274

Servi ce Station	Peak Arrival Rate	Time of Peak Arrivals (minutes)
24		
Mile 25	78	283

3.3 A Framework to Estimate Staffing Requirements in a Long Distance Road Race

A problem that affects the Erlang loss formula is its propensity to become unstable, as discussed by Qiao and Qiao [31]. In this section, we develop a staffing rule that can be used for staffing purposes in instances where the Erlang loss formula is subject to the overflow problem.

3.3.1 Background

For a stationary queueing system with Poisson arrivals and general service time distributions, Hall [17] observes that the queue length will be small when the number of servers, s, is bounded below as follows:

$$s \geq \left\lceil \frac{\lambda}{\mu} + \sqrt{\frac{\lambda}{\mu}} \right\rceil .$$

(3.5)

In other words, Hall [17] suggests that the queue length will be short when the number of servers is at least as many as the sum of the mean and standard deviation of the number of busy servers (or traffic intensity) expressed as an integer.

3.3.2 A Staffing Equation Based on Peak Arrival Rates

Staffing based on (3.5) relies on the queueing system being stationary (or time-independent). However, the system for which we seek staffing is nonstationary (or time-dependent), and staffing levels based on the time-varying average may not be conservative enough for the mixed $M_\ell(t)/G_\ell/s_\ell/s_\ell/N/FIFO$ queueing network that is considered in this research. To remedy this situation, we modify (3.5) to allow us to compute staffing estimates at each service station at the peak arrival rate, λ_ℓ^*. To estimate the staffing requirement, s_ℓ, at each service station $\ell=1,2,\ldots,L$ in our $M_\ell(t)/G_\ell/s_\ell/s_\ell/N/FIFO$ queueing network, we estimate the peak arrival rate, λ_ℓ^* with $\overset{\wedge}{\lambda_\ell^*}$, as described in Section 3.2.4, and modify (3.5) as follows:

$$s_\ell \geq \left\lceil \frac{\lambda_\ell^*}{\mu} + \sqrt{\frac{\lambda_\ell^*}{\mu}} \right\rceil, \qquad \ell=1,2,\ldots,L\ . \tag{3.6}$$

We call this staffing equation the Peak Arrival Staffing Bound (PASB). The PASB is a function of the peak traffic intensity and yields more conservative numbers of servers than (3.5) for the same service rate.

3.4 Chapter Summary

In this chapter we developed a framework that allows us to model a long distance road race (or similar types of traffic systems) as a mixed $M_{\ell}(t)/G_{\ell}/s_{\ell}/s_{\ell}/N/FIFO$ queueing network. We developed a staffing rule to estimate a conservative requirement for the number of servers at each service station in the queueing network.

In the next chapter, we implement the framework that we developed in this chapter.

CHAPTER 4
EXPERIMENTS AND RESULTS

4.1 Introduction

We report on an experiment that we conducted for a hypothetical long distance road race. We model the long distance road race as a mixed $M_\ell(t)/G_\ell/s_\ell/s_\ell/N/FIFO$ queueing network. Our objective is to determine the staffing requirement at each service station in the queueing network. In the next section, we describe the data from the 2004 WDW Marathon. In Section 4.3, we estimate the peak arrival rate for each service station of interest, λ_ℓ^*. In Section 4.4, we compare the staffing estimates based on (2.1) to the staffing estimates based on (3.6). In Section 4.5, we conduct 8 additional experiments with various values of the number of runners in the race (N), the service rate per busy server at all service stations (μ), and P_b (which is given by (2.1)). In Section 4.6, we analyze the results of the experiments from Section 4.5.

4.2 Data Description and Summary Statistics

The data that is used for the experiments was provided by the company that managed the 2004 Walt Disney World Marathon, EMMI (Events Marketing and Management, Incorporated), which is based in Orlando, Florida. The data provided by EMMI for the 2004 WDW Marathon contained 9336 runners that completed the 26.21875 mile race. For this data set, there were detector locations at Miles 0 (start), 10, 13.109375 (half-marathon), 20, and 26.21875 (finish).

4.3 Estimating the Peak Arrival Rates

In this section, (3.3) is applied to all 25 service stations in the queueing network (for each mile beginning with Mile 1 and ending with Mile 25) of the 2004 WDW Marathon. A truncated time-counts table (Step 1 through Step 3 of Procedure 1) for three arbitrarily selected stations at Mile 3, Mile 7, and Mile 11 is shown in Table 3.

Table 3: Arrival Counts for Miles 3, 7, and 11

Time (minute)	ile 3	ile 7	ile 11
1			
…			
20	6		
…			
33	**97**		
…			
45		1	
…			
81		**27**	6
…			

117		4	**52**
…			
180			7
…			

Next, we execute Step 4 of Procedure 1 from Section 3.2.4, and record the largest observed arrival rates, $\hat{\lambda}_{\ell}^{*}$, for each service station. The results of this step are highlighted in Table 3. Table 4 gives the largest observed arrival rate values for each service station. For each service station, we also give the estimated values for the expected number of busy servers (or traffic intensity) $\hat{\lambda}_{\ell}^{*}/\mu$ when $\mu=12$ customers per minute.

Table 4: Peak Arrival Rates and Traffic Intensity Estimates

ervice tation	Ar rival Rate	Traffic Intensity, $\mu=12$
ile 1	13 81	115.08
ile 2	74 1	61.75
ile 3	49 7	41.42

ervice tation	Arrival Rate	Traffic Intensity, $\mu=12$
ile 4	379	31.58
ile 5	310	25.83
ile 6	252	21.00
ile 7	227	18.92
ile 8	201	16.75
ile 9	182	15.17
ile 10	173	14.42
ile 11	152	12.67
ile 12	153	12.75
ile 13	144	12.00
ile 14	142	11.83

ervice tation	Arrival Rate	Traffic Intensity, $\mu=12$
ile 15	124	10.33
ile 16	113	9.42
ile 17	109	9.08
ile 18	104	8.67
ile 19	97	8.08
ile 20	96	8.00
ile 21	91	7.58
ile 22	86	7.17
ile 23	82	6.83
ile 24	90	7.50
ile 25	78	6.50

A noteworthy observation from the results in Table 4 is that it appears that as the runners become more spread out, the arrival rates begin to gradually decrease at a decreasing rate, thereby becoming more homogeneous. This pattern is most evident for the last third of the race.

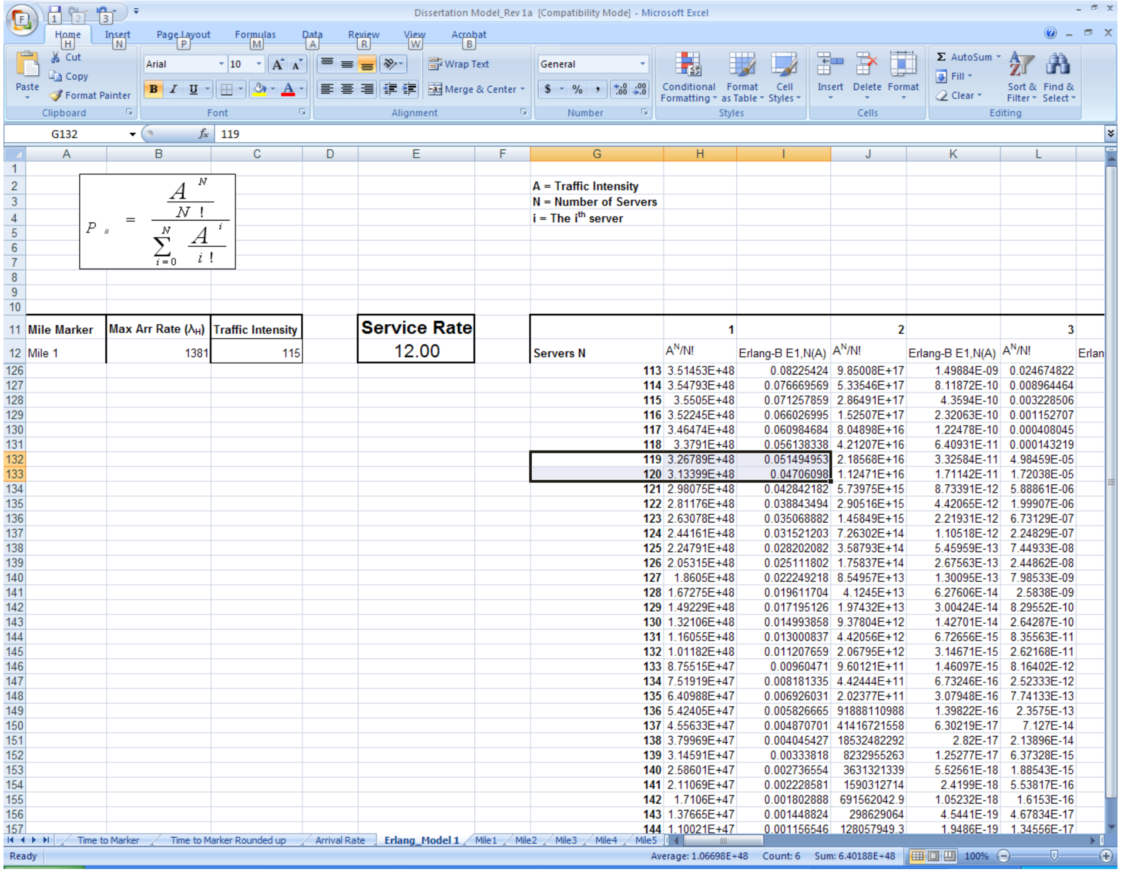
4.4 Experiment A: Estimating Staffing Requirements

In this section, the traffic intensity estimates in Table 4 are used in the Erlang loss formula (2.1) and in the PASB (3.6) to get staffing estimates.

4.4.1 Experiment A-1: Staffing with Erlang Loss Formula

For staffing with the Erlang loss formula (2.1), a maximum blocking probability of 5% was specified. A screenshot of the implementation of (2.1) is shown in Figure 4.1 below. From Figure 4.1, cell B12 contains the value of the estimated peak arrival rate, $\hat{\lambda}_{\ell}^{*}$, for Mile 1. Cell C12 contains the traffic intensity, which is a function of the peak arrival rate and the service rate (given in cell E12). The blocking probabilities for Mile 1 are given in column I, and cell I132 shows a blocking probability of approximately 0.0471, which corresponds to an estimated server requirement of 120 servers. For all staffing levels below 120 servers, the blocking probability is greater than 0.05.

The staffing estimates based on the Erlang loss formula that are given in Table 5 show that the numbers of servers change very little for the service stations (Miles 17 - 25) where the arrival rates appear to begin to settle down (or become more homogeneous).



G132 | 119

$$P_B = \frac{\dfrac{A^N}{N!}}{\sum_{i=0}^{N}\dfrac{A^i}{i!}}$$

A = Traffic Intensity
N = Number of Servers
i = The i^th server

Mile Marker	Max Arr Rate (λ_H)	Traffic Intensity
Mile 1	1381	115

Service Rate
12.00

	1		2		3	
Servers N	$A^N/N!$	Erlang-B E1,N(A)	$A^N/N!$	Erlang-B E1,N(A)	$A^N/N!$	Erlan
113	3.51453E+48	0.08225424	9.85008E+17	1.49884E-09	0.024674822	
114	3.54793E+48	0.076669569	5.33546E+17	8.11872E-10	0.008964464	
115	3.5505E+48	0.071257859	2.86491E+17	4.3594E-10	0.003228506	
116	3.52245E+48	0.066026995	1.52507E+17	2.32063E-10	0.001152707	
117	3.46474E+48	0.060984684	8.04898E+16	1.22478E-10	0.000408045	
118	3.3791E+48	0.056138338	4.21207E+16	6.40931E-11	0.000143219	
119	3.26789E+48	0.051494953	2.18568E+16	3.32584E-11	4.98459E-05	
120	3.13399E+48	0.04706098	1.12471E+16	1.71142E-11	1.72038E-05	
121	2.98075E+48	0.042842182	5.73975E+15	8.73391E-12	5.88861E-06	
122	2.81176E+48	0.038843494	2.90516E+15	4.42065E-12	1.99907E-06	
123	2.63078E+48	0.035068882	1.45849E+15	2.21931E-12	6.73129E-07	
124	2.44161E+48	0.031521203	7.26302E+14	1.10518E-12	2.24829E-07	
125	2.24791E+48	0.028202082	3.58793E+14	5.45959E-13	7.44933E-08	
126	2.05315E+48	0.025111802	1.75837E+14	2.67563E-13	2.44862E-08	
127	1.8605E+48	0.022249218	8.54957E+13	1.30095E-13	7.98533E-09	
128	1.67275E+48	0.019611704	4.1245E+13	6.27606E-14	2.5838E-09	
129	1.49229E+48	0.017195126	1.97432E+13	3.00424E-14	8.29552E-10	
130	1.32106E+48	0.014993858	9.37804E+12	1.42701E-14	2.64287E-10	
131	1.16055E+48	0.013000837	4.42056E+12	6.72656E-15	8.35563E-11	
132	1.01182E+48	0.011207659	2.06795E+12	3.14671E-15	2.62168E-11	
133	8.75515E+47	0.00960471	9.60121E+11	1.46097E-15	8.16402E-12	
134	7.51919E+47	0.008181335	4.42444E+11	6.73246E-16	2.52333E-12	
135	6.40988E+47	0.006926031	2.02377E+11	3.07948E-16	7.74133E-13	
136	5.42405E+47	0.005826665	91888110988	1.39822E-16	2.3575E-13	
137	4.55633E+47	0.004870701	41416721558	6.30219E-17	7.127E-14	
138	3.79969E+47	0.004045427	18532482292	2.82E-17	2.13896E-14	
139	3.14591E+47	0.00333818	8232955263	1.25277E-17	6.37328E-15	
140	2.58601E+47	0.002736554	3631321339	5.52561E-18	1.88543E-15	
141	2.11069E+47	0.002228581	1590312714	2.4199E-18	5.53817E-16	
142	1.7106E+47	0.001802888	691562042.9	1.05232E-18	1.6153E-16	
143	1.37665E+47	0.001448824	298629064	4.5441E-19	4.67834E-17	
144	1.10021E+47	0.001156546	128057949.3	1.9486E-19	1.34556E-17	

Figure 1: Staffing Based on Erlang Loss formula (Screenshot for Mile 1)

Table 5: Staffing Estimates for Experiment A-1 Using Erlang Loss Formula with Blocking Probability <=0.05 and Service Rate = 12

ervice tation	Number of servers based on Erlang loss formula
ile 1	120
ile 2	68
ile 3	47
ile 4	37
ile 5	32
ile 6	27
ile 7	24
ile 8	22
	20

ervice tation	Number of servers based on Erlang loss formula
ile 9	
ile 10	20
ile 11	18
ile 12	18
ile 13	17
ile 14	17
ile 15	15
ile 16	14
ile 17	14
ile 18	13
	13

ervice tation	Number of servers based on Erlang loss formula
ile 19	
ile 20	13
ile 21	12
ile 22	12
ile 23	11
ile 24	12
ile 25	11

4.4.2 Experiment A-2: Staffing with the Peak Arrival Staffing Bound

In this section, we report on the results of another experiment on the original 2004 WDW Marathon data set. For this experiment, we again assume a constant service rate of 12 customers per minute. Table 6 contains staffing estimates that are based on the PASB (3.6). Once the staffing estimates based on the PASB are determined, they are substituted into (2.1) to get a blocking probability estimate. For comparison purposes, the staffing estimates based on (2.1) are included as well.

Table 6 shows that the PASB, in terms of blocking probabilities and numbers of servers, gives values that are approximately similar to those calculated with the Erlang loss formula at the 5% blocking probability level. When the numbers of servers based on the PASB are greater than or equal to the numbers of servers based on the Erlang loss formula with $P_b = 0.05$, the estimated blocking probabilities for the PASB are less than or equal to 5%. On the other hand, when the numbers of servers based on the PASB are less than the numbers of servers based on the Erlang loss formula with $P_b = 0.05$, the estimated blocking probabilities for the PASB staffing level are greater than or equal to 5%.

A clear pattern that emerges from Tables 4.3 and 4.4 is that the arrival rates become more homogeneous as the runners become more spread out, especially for the last third of the race. With a fixed service rate, we observe that more servers are needed earlier in the race than are needed later in the race. For example, for both staffing methodologies, the staffing requirement at Mile 1 is almost twice as large as the staffing requirement at Mile 2 and the staffing requirement at Mile 2 is substantially greater than the staffing requirement at Mile 3. However, as the runners become more spread out, both staffing methodologies show that the staffing requirements change slower as the distance from the start location increases. For example, the numbers of servers that are required at both Mile 11 and Mile 12 are only marginally different from the numbers of servers that are required at both Mile 13 and Mile 14. For the last third of the race, because the arrival rates begin to settle down, there is very little change in the number of servers over the final eight miles.

Table 6: Arrival Rates and Staffing Estimates for Experiment A Using Erlang Loss Formula with Blocking Probability <=0.05 and μ=12 and the Peak Arrival Staffing Bound (PASB) with μ=12

ervice tation	Arrival Rate	# of Servers - based on Erlang loss formula	# of Servers – based on PASB	Blocking Probability for PASB Results
ile 1	1381	120	127	0.03
ile 2	741	68	70	0.03
ile 3	497	47	49	0.04
ile 4	379	37	38	0.04
ile 5	310	32	32	0.05
ile 6	252	27	26	0.06

ervice tation	Arrival Rate	# of Servers - based on Erlang loss formula	# of Servers – based on PASB	Blocking Probability for PASB Results
ile 7	227	24	24	0.05
ile 8	201	22	22	0.05
ile 9	182	20	20	0.05
ile 10	173	19	19	0.05
ile 11	152	18	17	0.06
ile 12	153	18	17	0.06
ile 13	144	17	16	0.06
ile 14	142	17	16	0.06
ile 15	124	15	15	0.05

ervice tation	Arrival Rate	# of Servers - based on Erlang loss formula	# of Servers – based on PASB	Blocking Probability for PASB Results
ile 16	113	14	14	0.05
ile 17	109	14	14	0.05
ile 18	104	13	12	0.07
ile 19	97	13	12	0.06
ile 20	96	13	11	0.08
ile 21	91	12	11	0.06
ile 22	86	12	11	0.06
ile 23	82	11	10	0.06
ile 24	90	12	11	0.06

ervice tation	Arrival Rate	# of Servers - based on Erlang loss formula	# of Servers – based on PASB	Blocking Probability for PASB Results
ile 25	78	11	10	0.06

In the next experiment, we will investigate the effects on the staffing recommendations when the expected number of busy servers (or traffic intensity) is quadrupled. Then, it is possible to effectively investigate the situations where the number of runners is quadrupled or where the service rate is cut by a factor of 4. For demonstration purposes, the service rate is reduced from 12 (customers per server per minute) to 3.

4.4.3 Experiment B

The objective of this experiment is to investigate the effects of high traffic demands on (2.1) and the PASB. We quadrupled the traffic intensity by dividing the original service rate $(\mu = 12)$ by 4, thereby getting a new service rate of $\mu = 3$. Table 7 gives the results of the quadrupled traffic intensity values.

Table 7: Peak Arrival Rates and Traffic Intensity Estimates

ervice tation	Arrival Rate	Traffic Intensity, $\mu=3$
ile 1	1381	460
ile 2	741	247
ile 3	497	166
ile 4	379	126
ile 5	310	103
ile 6	252	84
ile 7	227	76
ile 8	201	67
ile 9	182	61
ile 10	173	58

ervice tation	Ar rival Rate	Traffic Intensity, $\mu=3$
ile 11	15 2	51
ile 12	15 3	51
ile 13	14 4	48
ile 14	14 2	47
ile 15	12 4	41
ile 16	11 3	38
ile 17	10 9	36
ile 18	10 4	35
ile 19	97	32
ile 20	96	32
ile 21	91	30

ervice tation	Arrival Rate	Traffic Intensity, $\mu = 3$
ile 22	86	29
ile 23	82	27
ile 24	90	30
ile 25	78	26

These traffic intensities will be used in (2.1) and (3.5) to get staffing estimates for each of the service stations.

4.4.4 Experiment B-1: Staffing with Erlang Loss Formula

For staffing with the Erlang loss formula, a maximum blocking probability of 5% was specified, and (2.1) was executed for each of the 25 service stations. Table 8 contains the staffing requirements based on the Erlang loss formula.

The results in Table 8 confirm the findings of Qiao and Qiao [31]. We can see that the Erlang loss formula overflows for the first three service stations because the number of servers needed exceeds 154.

Table 8: Arrival Rates, Traffic Intensity Estimates, and Staffing Estimates for Experiment B-1 Using Erlang Loss Formula with Blocking Probability <=0.05 and $\mu = 3$

ervice tation	rrival ate	Traffic Intensity	# of Servers – based on Erlang loss formula
ile 1	381	460	overflow
ile 2	41	247	overflow
ile 3	97	166	overflow
ile 4	79	126	131
ile 5	10	103	108
ile 6	52	84	104
ile 7	27	76	81
ile 8	01	67	73

ile 9	82	6 1	67
ile 10	73	5 8	64
ile 11	52	5 1	57
ile 12	53	5 1	57
ile 13	44	4 8	54
ile 14	42	4 7	53
ile 15	24	4 1	47
ile 16	13	3 8	44
ile 17	09	3 6	42
ile 18	04	3 5	41
ile 19	7	3 2	38
ile 20	6	3 2	38
ile 21	1	3 0	36

ile 22	6	2 9	34
ile 23	2	2 7	33
ile 24	0	3 0	36
ile 25	8	2 6	33

4.4.5 Experiment B-2: Staffing with the Peak Arrival Staffing Bound

In this section, we report on the staffing results based on the PASB. Table 9 contains the staffing estimates for Experiment B-2. For comparison purposes, the staffing levels prescribed by the Erlang loss formula are also included in the table.

Table 9: Arrival Rates, Traffic Intensity Estimates, and Staffing Estimates for Experiment B-2 Using Erlang Loss Formula with Blocking Probability <=0.05 and $\mu = 3$ and the Peak Arrival Staffing Bound (PASB) with $\mu = 3$

ervice tation	rrival ate	Traffic I ntensity	# of Servers based on Erlang loss formula	# of Servers ba sed on P ASB	B locking P robability fo r PASB results
ile 1	381	4 60	overfl ow	48 2	N A
ile 2	41	2 47	overfl ow	26 3	N A
ile 3	97	1 66	overfl ow	17 9	N A
ile 4	79	1 26	131	13 8	0. 02
ile 5	10	1 03	108	11 4	0. 03
ile 6	52	8 4	104	94	0. 03
ile 7	27	7 6	81	85	0. 03
ile 8	01	6 7	73	76	0. 03
ile 9	82	6 1	67	69	0. 03
ile 10	73	5 8	64	66	0. 03
ile 11	52	5 1	57	59	0. 04
ile 12	53	5 1	57	59	0. 03

ile 13	44	4 8	54	55	0. 04
ile 14	42	4 7	53	54	0. 03
ile 15	24	4 1	47	48	0. 04
ile 16	13	3 8	44	45	0. 04
ile 17	09	3 6	42	42	0. 05
ile 18	04	3 5	41	41	0. 05
ile 19	7	3 2	38	38	0. 05
ile 20	6	3 2	38	38	0. 05
ile 21	1	3 0	36	36	0. 05
ile 22	6	2 9	34	35	0. 04
ile 23	2	2 7	33	33	0. 05
ile 24	0	3 0	36	36	0. 05
ile 25	8	2 6	33	32	0. 06

The finding by Qiao and Qiao [31] is evident for each of the first three service stations where the Erlang loss formula overflows due to high traffic intensities indicating server requirements that exceed 154 servers for each of the first three service stations. The PASB does not overflow, and shows its usefulness in situations where the relative sizes of the arrival and service rates cause the Erlang loss formula to overflow. Additionally, as the traffic intensity increases, the PASB gives more conservative numbers of servers than the Erlang loss formula with $P_b = 0.05$, and in those instances, the blocking probabilities associated with the PASB is less than 5%. A recurring theme that is seen from Table 9 is that the differences in the staffing numbers between the two staffing methodologies diminish as the rate of change in the traffic intensity decreases. In those instances, the blocking probabilities of both staffing methodologies are approximately similar.

In the next section, we describe additional experiments conducted with the Erlang loss formula (2.1) and the PASB (3.6).

4.5 Additional Experiments

This section presents experiments that were conducted to show the capabilities of the staffing models that we present in this research. In Section 4.5.1, we discuss the experiments, and in Section 4.5.2, we report our findings.

4.5.1 Experiments

We designed the eight experiments that are summarized in Table 10. The experiments were specified in the following way. For Experiments 1 - 4, a blocking probability of 5% was specified, and (2.1) was executed for each of the 25 service stations. Similarly, for Experiments 5 – 8, a blocking probability of 10% was specified, and (2.1) was executed for each of the 25 service stations of interest. The PASB is also implemented here. Because the PASB does not use a blocking probability, it was executed only once for each pair of experiments with the same values of N and μ.

Table 10: List of Experiments

Experiments		N Service Rate (per minute)	Blocking Probabilities for Erlang loss formula
1	336	10	0.05
2	7344	10	0.05
3	336	20	0.05
4	7344	20	0.05
5	336	10	0.10
6	7344	10	0.10
7	336	20	0.10
8	7344	20	0.10

4.5.2 Rationale for Choice of Experiments

The purposes of these experiments are to: (a) assess the impact of the race size on staffing under different blocking probability assumptions, and (b) assess the impact of the service rate on staffing under different blocking probability assumptions.

The impact of the race size on staffing is assessed by varying the size of N. By quadrupling N, we aim to assess the capabilities of the PASB in estimating the prescribed staffing levels when the race size is large. It is worth noting that an increase in N by a factor $\alpha, \alpha \in \mathbb{R}^{+}$, results in an increase in λ_{ℓ}^{*} by the same factor α. Therefore, by increasing N, we are also assessing the impact of λ_{ℓ}^{*} on staffing. For example, doubling N (from WDW) results in the doubling of λ_{ℓ}^{*} at each service station. The impact of the service rate on staffing is assessed by varying the service rate. Our aim is to assess how sensitive our staffing models are to service rate variability. Tables 4.9 – 4.12 (and Figures 4.2 – 4.5, which are based on the data in Tables 4.9 – 4.12, respectively) contain the results of our experiments.

Table 11: Number of Servers for Experiments 1 and 5

Service Station	# of Servers for Experiment 1: based on Erlang loss Formula:blocking probability = 0.05	# of Servers for Experiment 9: based on Erlang loss Formula: blocking probability = 0.10	# of Servers: based on PASB	Blocking Probability for PASB results
Mile 1	142	131	150	NA
Mile 2	80	73	83	0.03
Mile 3	56	51	57	0.04
Mile 4	44	40	45	0.04
Mile 5	37	33	37	0.04
Mile 6	31	28	31	0.04
Mile 7	28	25	28	0.05

Service Station	# of Servers for Experiment 1: based on Erlang loss Formula:blocking probability = 0.05	# of Servers for Experiment 9: based on Erlang loss Formula: blocking probability = 0.10	# of Servers: based on PASB	Blocking Probability for PASB results
Mile 8	26	23	25	0.06
Mile 9	24	21	23	0.06
Mile 10	23	20	22	0.06
Mile 11	20	18	20	0.05
Mile 12	21	18	20	0.06
Mile 13	19	17	19	0.05
Mile 14	19	17	18	0.07
Mile 15	17	15	16	0.07

Service Station	# of Servers for Experiment 1: based on Erlang loss Formula:blocking probability = 0.05	# of Servers for Experiment 9: based on Erlang loss Formula: blocking probability = 0.10	# of Servers: based on PASB	Blocking Probability for PASB results
Mile 16	16	14	15	0.07
Mile 17	16	14	15	0.06
Mile 18	15	13	14	0.07
Mile 19	14	13	13	0.07
Mile 20	14	13	13	0.07
Mile 21	14	12	13	0.06
Mile 22	13	12	12	0.07
Mile 23	13	11	12	0.06

Service Station	# of Servers for Experiment 1: based on Erlang loss Formula:blocking probability = 0.05	# of Servers for Experiment 9: based on Erlang loss Formula: blocking probability = 0.10	# of Servers: based on PASB	Blocking Probability for PASB results
Mobile 24	14	12	12	0.08
Mobile 25	12	11	11	0.07

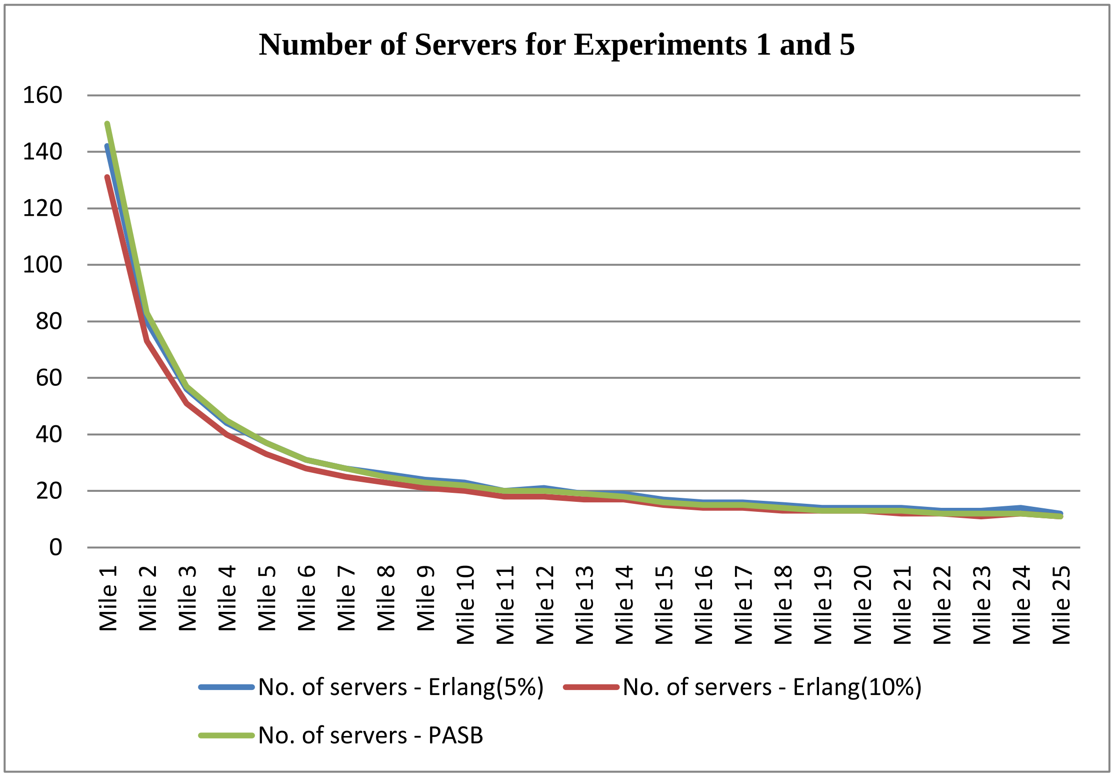


Figure 2: Number of Servers for Experiments 1 and 5

Table 12: Number of Servers for Experiments 2 and 6

Service Station	# of Servers for Experiment 2: based on Erlang loss Formula: blocking probability = 0.05	# of Servers for Experiment 10: based on Erlang loss Formula: blocking probability = 0.10	# of Servers: based on PASB	Blocking Probability for PASB results
Mile 1	overflow	overflow	576	NA
Mile 2	overflow	overflow	314	NA
Mile 3	overflow	overflow	213	NA
Mile 4	overflow	overflow	164	NA
Mile 5	128	119	136	0.02
Mile 6	106	97	111	0.03
Mile 7	96	88	101	0.03
Mile 8	86	79	90	0.03
Mile 9	79	72	82	0.03
Mile 10	75	69	78	0.03
Mile 11	67	61	70	0.03
Mile 12	67	61	69	0.03
Mile 13	63	58	66	0.03
Mile 14	63	57	65	0.03
Mile 15	56	51	57	0.04
Mile 16	51	46	52	0.04
Mile 17	50	45	51	0.03
Mile 18	48	43	49	0.03
Mile 19	45	41	46	0.04
M	44	40	45	0.

Service Station	# of Servers for Experiment 2: based on Erlang loss Formula: blocking probability = 0.05	# of Servers for Experiment 10: based on Erlang loss Formula: blocking probability = 0.10	# of Servers: based on PASB	Blocking Probability for PASB results
ile 20				04
Mile 21	42	38	43	0.04
Mile 22	40	36	41	0.04
Mile 23	39	35	39	0.04
Mile 24	42	38	42	0.04
Mile 25	37	33	37	0.05

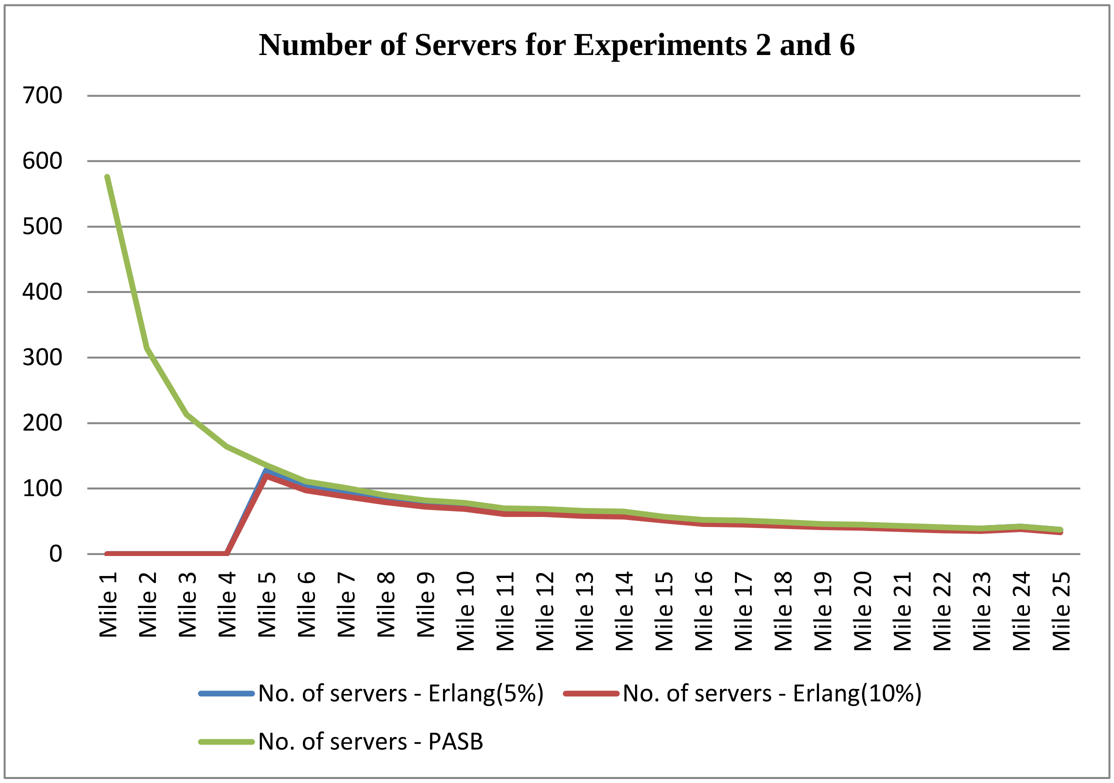


Figure 3: Number of Servers for Experiments 2 and 6

Table 13: Number of Servers for Experiment 3 and 7

Service Station	# of Servers for Experiment 7: based on Erlang loss Formula: blocking probability = 0.05	# of Servers for Experiment 15: based on Erlang loss Formula: blocking probability = 0.10	# of Servers: based on PASB	Blocking Probability for PASB results
Mile 1	75	68	78	0.03
Mile 2	43	39	44	0.04
Mile 3	31	27	29	0.06
Mile 4	24	22	24	0.05
Mile 5	21	18	20	0.06
Mile 6	18	16	18	0.04
Mile 7	16	14	17	0.03
Mile 8	15	13	14	0.06
Mile 9	14	12	13	0.06
Mile 10	13	12	12	0.07
Mile 11	12	11	11	0.07
Mile 12	12	11	11	0.07
Mile 13	12	10	10	0.08
Mile 14	12	10	10	0.08
Mile 15	10	9	9	0.08
Mile 16	10	9	9	0.06
Mile 17	10	8	8	0.09
Mile 18	9	8	8	0.08
Mile 19	9	8	8	0.06
M	9	8	7	0.

Service Station	# of Servers for Experiment 7: based on Erlang loss Formula: blocking probability = 0.05	# of Servers for Experiment 15: based on Erlang loss Formula: blocking probability = 0.10	# of Servers: based on PASB	Blocking Probability for PASB results
ile 20				11
Mile 21	9	7	7	0.09
Mile 22	8	7	7	0.08
Mile 23	8	7	7	0.07
Mile 24	8	7	7	0.09
Mile 25	8	7	6	0.11

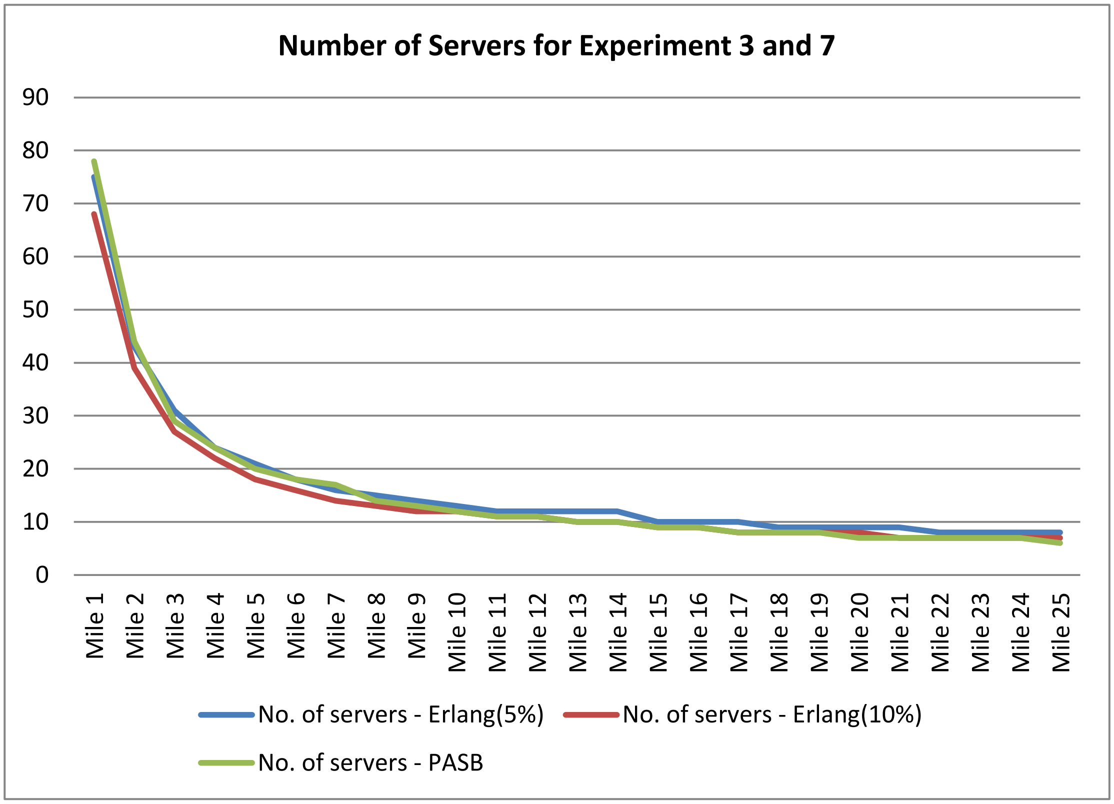


Figure 4: Number of Servers for Experiment 3 and 7

Table 14: Number of Servers for Experiments 4 and 8

Location of Interest	# of Servers for Experiment 8: based on Erlang loss Formula: blocking probability = 0.05	# of Servers for Experiment 16: based on Erlang loss Formula: blocking probability = 0.10	# of Servers: based on PASB	Blocking Probability for PASB results
Mile 1	overflow	overflow	293	NA
Mile 2	overflow	141	161	NA
Mile 3	105	96	110	0.03
Mile 4	81	75	85	0.03
Mile 5	68	62	70	0.03
Mile 6	56	51	58	0.04
Mile 7	51	47	53	0.03
Mile 8	46	42	47	0.04
Mile 9	42	38	43	0.04
Mile 10	41	37	41	0.04
Mile 11	36	33	37	0.04
Mile 12	36	33	36	0.05
Mile 13	34	31	35	0.04
Mile 14	34	31	34	0.05
Mile 15	30	27	30	0.05
Mile 16	28	25	28	0.05
Mile 17	27	25	27	0.05
Mile 18	26	24	26	0.05
Mile 19	25	22	24	0.06
Mi	25	22	24	0.

Location of Interest	# of Servers for Experiment 8: based on Erlang loss Formula: blocking probability = 0.05	# of Servers for Experiment 16: based on Erlang loss Formula: blocking probability = 0.10	# of Servers: based on PASB	Blocking Probability for PASB results
le 20				06
Mile 21	24	21	23	0.06
Mile 22	23	20	22	0.06
Mile 23	22	19	21	0.06
Mile 24	23	21	23	0.05
Mile 25	21	19	20	0.06

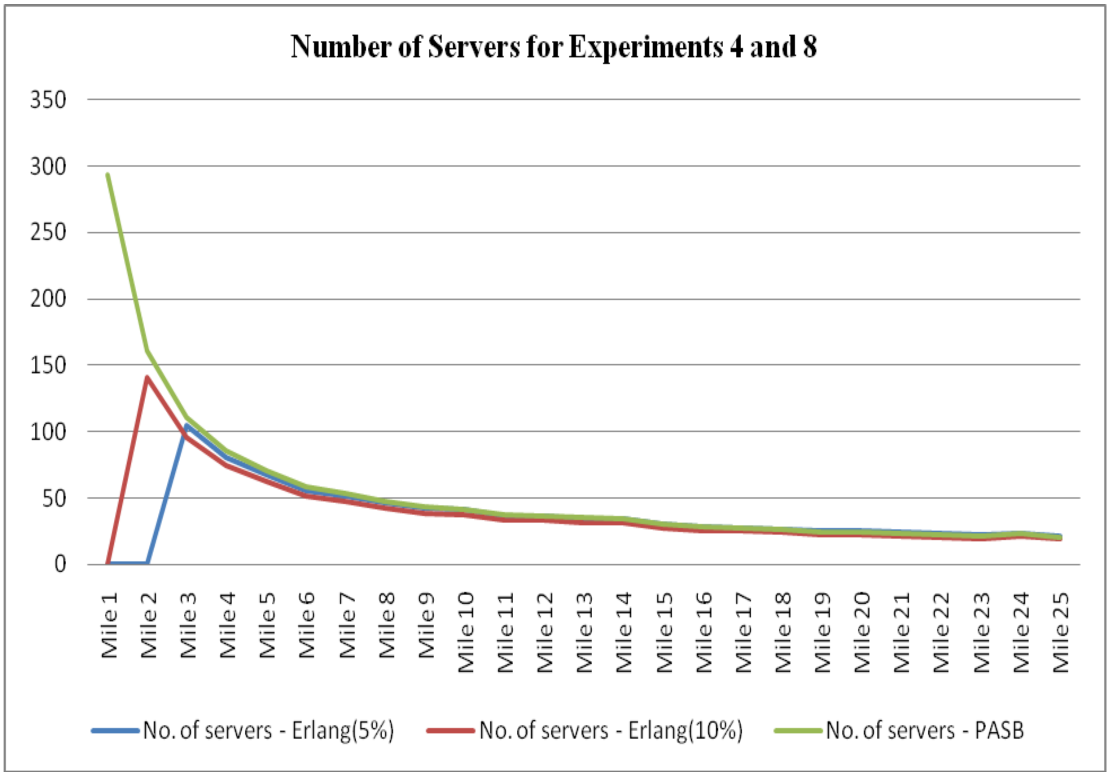


Figure 5: Number of Servers for Experiments 4 and 8

A recurring pattern in Tables 4.9 – 4.12 (and Figures 4.2 – 4.5) is that the PASB gives more conservative numbers of servers than the Erlang loss formula when the traffic intensity is high. However, as the traffic intensity becomes low, the Erlang loss formula and the PASB give similar results, with the Erlang loss formula with $P_b = 0.05$ yielding slightly more conservative numbers of servers than the PASB. Though obvious, it is worth noting that when a blocking probability that is associated with a staffing result that is based on the PASB is less than a specified blocking probability for the Erlang loss formula, the staffing result that is based on the PASB is more conservative than the staffing result of the Erlang loss formula. Conversely, when a blocking probability that is associated with a staffing result that is based on the PASB is equal to or greater than a specified blocking probability for the Erlang loss formula, the staffing result that is based on the PASB is equal to or less conservative than the staffing result of the Erlang loss formula.

For all staffing methodologies, the differences in the numbers of servers decrease over the course of the race, and can be seen clearly in Figures 4.2 to 4.5. The differences in the numbers of servers that is based on the PASB and the two staffing methodologies that are based on the Erlang loss formula get smaller as the runners become more spread out along the course. As the rate of decrease in the traffic intensity slows and the runners become more spread out, the changes in the numbers of servers become small, and the numbers of servers for both staffing methodologies are approximately similar. This finding implies that the numbers of servers based on the PASB and the numbers of servers based on the Erlang loss formula may be approximately similar when the traffic intensity is lowered.

Several intuitive results are evident from our experiments. The inverse relationship between the blocking probability level and the staffing level in Tables 4.9 – 4.12 confirm our intuition regarding how the numbers of servers change as we increase or decrease the blocking probability in the Erlang loss formula. Another intuitive result that the experiments confirm is the inverse relationship between the service rate and the staffing level. Comparing the results of pairs of experiments 1 and 3, (or 2 and 4, or 5 and 7, or 6 and 8) show that an increase (or decrease) in the service rate (while keeping the arrival rate constant) leads to a decrease (or increase) in the traffic intensity, which leads to a decrease (or increase) in the numbers of servers. Another intuitive result is seen from the results of the experiments to assess the impact of the traffic intensity (as a result of an increase in the size of the race) on the numbers of servers. A comparison of the results for the pairs of experiments 1 and 2, (or 3 and 4, or 5 and 6, or 7 and 8) show that a four times increase (decrease) of the traffic intensity leads to a less than four times increase (decrease) of the numbers of servers for all models.

4.6 Analysis of Results

For each of the Tables 4.9 – 4.12, we give the staffing results of the Erlang loss formula at 5% blocking probability and 10% blocking probability, and the maximum staffing levels for the PASB. Thus, for the PASB, the arrival rate at each service station is fixed at the peak level for the entire time that it takes all runners to exit a service station. The results of the experiments that we conducted show that the staffing requirements, based on a constant service rate, decrease over the course of the race and appear to begin to settle down when the traffic intensity start to decrease at a decreasing rate.

The results from the experiments in which $N = 37344$ confirm the findings of Qiao and Qiao [31] and show that the Erlang loss formula overflows at the early service stations during which the traffic intensity is high and would require a server count that exceeds 154. In those instances, the PASB shows an advantage over the Erlang loss formula in providing staffing estimates regardless of the traffic intensity. As the traffic intensity decreases, the PASB gives staffing estimates that are comparable to the staffing estimates of the Erlang loss formula at the 5% blocking probability level. Thus, key findings of our experiments are that the PASB provides more conservative server requirements than the Erlang loss formula when the traffic intensity is high, and that the service requirement based on the PASB is similar to the Erlang loss formula as the traffic intensity gets lower. Consequently, the PASB can be used to estimate the numbers of servers regardless of the traffic intensity, which may vary depending on the size of the race, the arrival rate, or the service rate. However, the same is not true for the Erlang loss formula, which becomes numerically unstable and "overflows" when the traffic intensity is high. These findings suggest that the PASB can be a valuable tool to aid race directors in making staffing decisions in all races regardless of the traffic intensities.

4.7 Chapter Summary

The staffing rule that we have developed is a staffing alternative to the Erlang loss formula, which is known to overflow when the traffic intensity gets large. By way of numerical examples, we find that the results of the PASB are similar to the staffing results of the Erlang loss formula as the traffic intensity gets lower, and that the PASB provides more conservative staffing requirements than the Erlang loss formula when the traffic intensity is high. We also verified that the Erlang loss formula overflows when the traffic intensity is high. Because the PASB does not suffer from the kinds of numerical instability problems that affect the Erlang loss formula, staffing decisions can be made based on the PASB regardless of the traffic intensity. Therefore, the PASB can be a valuable tool for a race director to manage the staffing requirements for the entire race. The generally conservative staffing levels prescribed by the PASB suggests that the PASB is relatively robust with respect to the blocking probability.

To make this research more valuable to practitioners, in the next chapter, we will develop a simulation model that will provide a road race manager with a flexible management tool to manage a long road race with multiple classes of runners.

CHAPTER 5
A MULTI-CLASS MONTE CARLO SIMULATION MODEL OF A LONG DISTANCE ROAD RACE

5.1 Simulation Model Development

In this chapter, we extend the single class $M_\ell(t)/G_\ell/s_\ell/s_\ell/N/FIFO$ queueing network to include multiple classes of runners. The key assumptions that we make in developing this new model are the following:

- Runners can be grouped into distinct groups, based on their overall anticipated paces for the entire race.
- Each group gets a proportion of the total number of runners in the race, and the total of the proportions distributed across all groups of runners sums to one.

For a race with N runners, we partition the N runners into C distinct classes of runners. Runners are placed in Class c, $c=1,2,\ldots,C$, based on their overall anticipated pace for the entire race, and we denote the expected proportion of runners in Class c as ρ_c. Thus, for example, runners in Class 1 occur, on average, $\rho_1 \times 100$ percent of the time; runners in Class 2 occur $\rho_2 \times 100$ percent of the time, on average, etc., as shown in the second column of Table 15. We assume $\sum_{c=1}^{C} \rho_c = 1$.

A pace distribution table gives the pace-ranges for runners in each of the C classes, and is illustrated in the third column of Table 15. Specifically, the range of possible values for the paces of all N runners is given by the pace interval which we define as $[a_1, b_C]$, where a_1 is the fastest pace and b_C is the slowest pace. We assume that $a_c < a_{c+1},\ b_c < b_{c+1} \ \forall\ c$. Also, to account for possibly overlapping intervals, b_c may be smaller than a_{c+1}. We give an example of the multiclass simulation tool with overlapping intervals in the next section.

Table 15: Class and Pace Distribution Table

Runner Class	Percentage Occurrence	Class Pace (seconds/mile)
1	ρ_1	$[a_1, b_1]$
2	ρ_2	$(a_2, b_2]$
$\vdots$	$\vdots$	$\vdots$
$C-1$		$[a_{C-1}, b_{C-1}]$
C	ρ_C	$[a_C, b_C]$

5.1.1 The Multiclass Simulation Model

The following is the procedure that we used to develop the simulation model in Microsoft Excel. This procedure can be repeated as many times as the user specifies (that is, for multiple replications of the same race), and the results of all replications averaged in order to get an estimate of the variable of interest (which in the present case is the numbers of runners arriving at service stations in the $M_\ell(t)/G_\ell/s_\ell/s_\ell/N/FIFO$ queueing network during each one minute interval in the race).

Step 0: Specify the number of runners, N, the number of runner classes, C, the expected proportions in each class, $\rho_1, \rho_2, \ldots, \rho_C$, and the pace ranges for each class of runners.

Step 1: For each runner n, $n = 1, 2, \ldots, N$, generate a random number and determine the runner's class by comparing the random number to the cumulative class proportions in a manner analogous to the inverse CDF transformation for a discrete random variable.

Step 2.0: For each $\ell \in L$ and each $n \in N$, generate the time that it takes a runner to traverse between consecutive service stations $\ell - 1$ and ℓ as follows:

Step 2.1: Set up a time-counts table. (For an illustration, see Table 1.)

Step 2.2: For each runner in Class c, $c = 1, 2, \ldots, C$, that is expected to arrive at service station ℓ, draw a random number ($w_{n,\ell}$, for the n^{th} runner to arrive at location ℓ) that determines the time that it takes the runner to traverse the distance between service station $\ell - 1$ and service station ℓ from the interval $[a_c, b_c]$. Record $w_{n,\ell}$ for each runner at each service station until all the runners have exited the course at d_K.

Step 2.3: When all the runners' times have been recorded for each $\ell \in L$, get the per minute arrival rate by counting the number of arriving runners in each one minute interval at each service station by checking each column of the time-counts table and recording its maximum value, $\overset{\wedge}{\lambda_\ell^*}$, and the time associated with this value, $\overset{\wedge}{t_\ell^*}$.

Step 3: For staffing purposes, implement a staffing equation at each $\ell \in L$ to get staffing estimates.

We have implemented the Erlang loss formula (2.1) and the PASB (3.6) at each service station.

5.1.2 Implementing the Multiclass Simulation Model

The simulation model described above was implemented in Microsoft Excel. To illustrate the workings of the model, we consider a race with $N = 10000$ runners and the classes, probabilities, and paces shown in Table 16, which gives an example of the multiclass simulation model with overlapping pace intervals.

Table 16: Pace Distribution and Probability of Occurrence

lass	Probabilities for Experiments used in Example	Pace (seconds/mile)
1	0.30	300 - 390
2	0.20	330 - 510
3	0.20	450 - 690
4	0.15	630 - 870
5	0.10	810 - 1050
6	0.05	990 - 1200

Figure 5.1 shows the implementation of Step 0 from the procedure from Section 5.1.1. In the illustration that follows, we use seconds as the time-unit of choice. This choice of time-unit is arbitrary; the model that is developed here accepts any time-unit that the user specifies.

One aspect of Step 0 is shown in cell range K2:L7, which shows a race with six distinct classes - Class 1 to Class 6. For example, runners in Class 1 occur in this simulation approximately 30% of the time, runners in Class 2 occur approximately 20% of the time, and so on up to Class 6 runners who occur approximately 5% of the time.

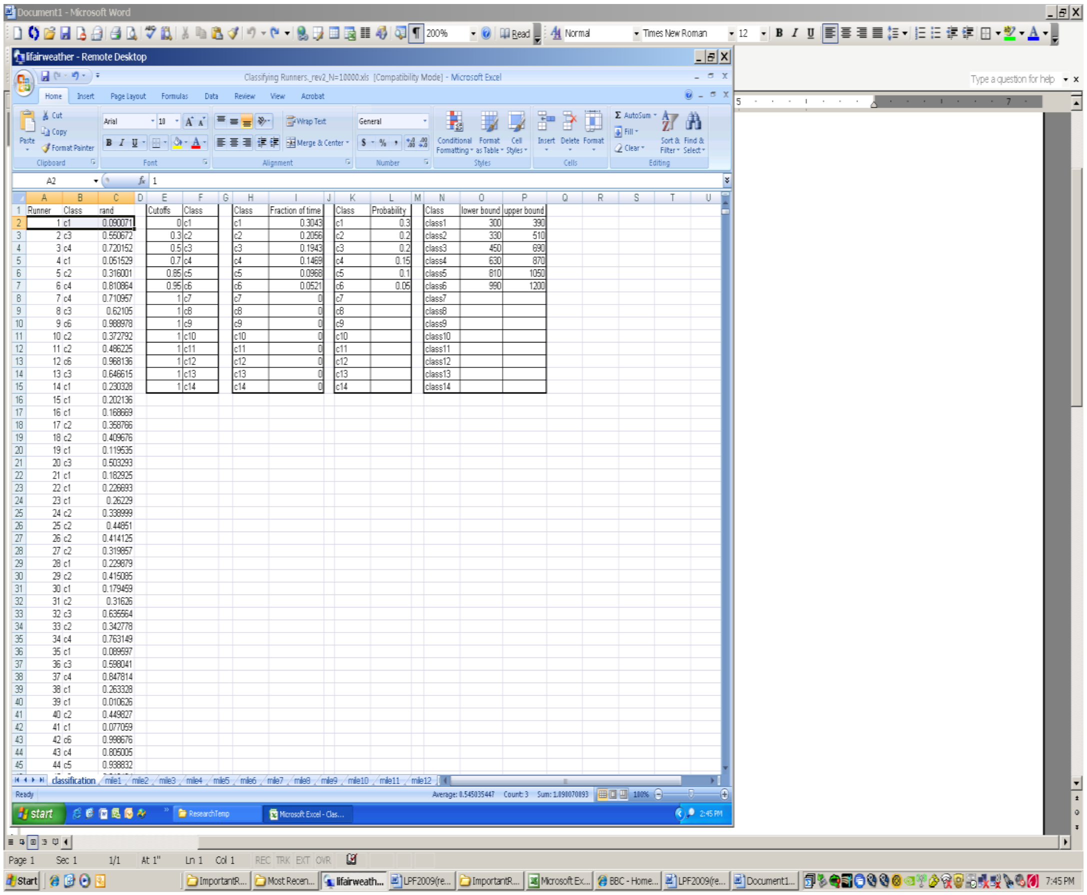
Another aspect of Step 0 is shown in cell range N2:P7; thus, for this example, the athletes in Class 1 run between 300-seconds and 390-seconds pace per mile for the entire race. Like the classes in Step 1, the paces are inputs to the model; thus, the road race manager can change them to whatever the appropriate paces are for the race that is being planned.

The ease with which Steps 1 to 3 can be implemented in Microsoft Excel was the primary motivator for our choice on how to implement the steps from Section 5.1.1. There are drawbacks, however, to choosing to implement the model in Excel. These drawbacks will be discussed later in this chapter.

We use the RAND() function in Excel to execute Step 1. Step 1 is shown in Column C where we use the RAND() function to generate $N = 10000$ random numbers. The first 44 random numbers can be seen in Screenshot 5.1.

The ease with which tables in Excel can be linked was important in our decision on how to implement the model. The LOOKUP() function is one of the functions in Excel that allows values from different tables to be linked in a user-specified manner. The particular LOOKUP()

function that we use is the VLOOKUP() function, which takes a value that the user specifies, and then goes and looks up that user-specified value in a table. The VLOOKUP() function looks for the specified value in the leftmost column of the table, and either finds it or the closest match. The result of the VLOOKUP() function can be seen in column B of Figure 5.1



Classifying Runners_rev2_N=10000.xls [Compatibility Mode] - Microsoft Excel

A2 fx 1

	A	B	C	E	F	H	I	K	L	N	O	P
1	Runner	Class	rand	Cutoffs	Class	Class	Fraction of time	Class	Probability	Class	lower bound	upper bound
2	1	c1	0.090071	0	c1	c1	0.3043	c1	0.3	class1	300	390
3	2	c3	0.550672	0.3	c2	c2	0.2056	c2	0.2	class2	330	510
4	3	c4	0.720152	0.5	c3	c3	0.1943	c3	0.2	class3	450	690
5	4	c1	0.051529	0.7	c4	c4	0.1469	c4	0.15	class4	630	870
6	5	c2	0.316001	0.85	c5	c5	0.0968	c5	0.1	class5	810	1050
7	6	c4	0.810864	0.95	c6	c6	0.0521	c6	0.05	class6	990	1200
8	7	c4	0.710957	1	c7	c7	0	c7		class7		
9	8	c3	0.62105	1	c8	c8	0	c8		class8		
10	9	c6	0.988978	1	c9	c9	0	c9		class9		
11	10	c2	0.372792	1	c10	c10	0	c10		class10		
12	11	c2	0.486225	1	c11	c11	0	c11		class11		
13	12	c6	0.968136	1	c12	c12	0	c12		class12		
14	13	c3	0.646615	1	c13	c13	0	c13		class13		
15	14	c1	0.230328	1	c14	c14	0	c14		class14		
16	15	c1	0.202136									
17	16	c1	0.168669									
18	17	c2	0.358766									
19	18	c2	0.409676									
20	19	c1	0.119535									
21	20	c3	0.503293									
22	21	c1	0.182925									
23	22	c1	0.226693									
24	23	c1	0.26229									
25	24	c2	0.338999									
26	25	c2	0.44851									
27	26	c2	0.414125									
28	27	c2	0.319857									
29	28	c1	0.229879									
30	29	c2	0.415085									
31	30	c1	0.179459									
32	31	c2	0.31626									
33	32	c3	0.635564									
34	33	c2	0.342778									
35	34	c4	0.763149									
36	35	c1	0.089597									
37	36	c3	0.598041									
38	37	c4	0.847814									
39	38	c1	0.263328									
40	39	c1	0.010626									
41	40	c2	0.449827									
42	41	c1	0.077059									
43	42	c6	0.998676									
44	43	c4	0.806005									
45	44	c5	0.938832									

classification / mile1 / mile2 / mile3 / mile4 / mile5 / mile6 / mile7 / mile8 / mile9 / mile10 / mile11 / mile12
Ready
Average: 0.545035447 Count: 3 Sum: 1.090070893

Figure 6: Simulation Implementation: Steps 0 – 1

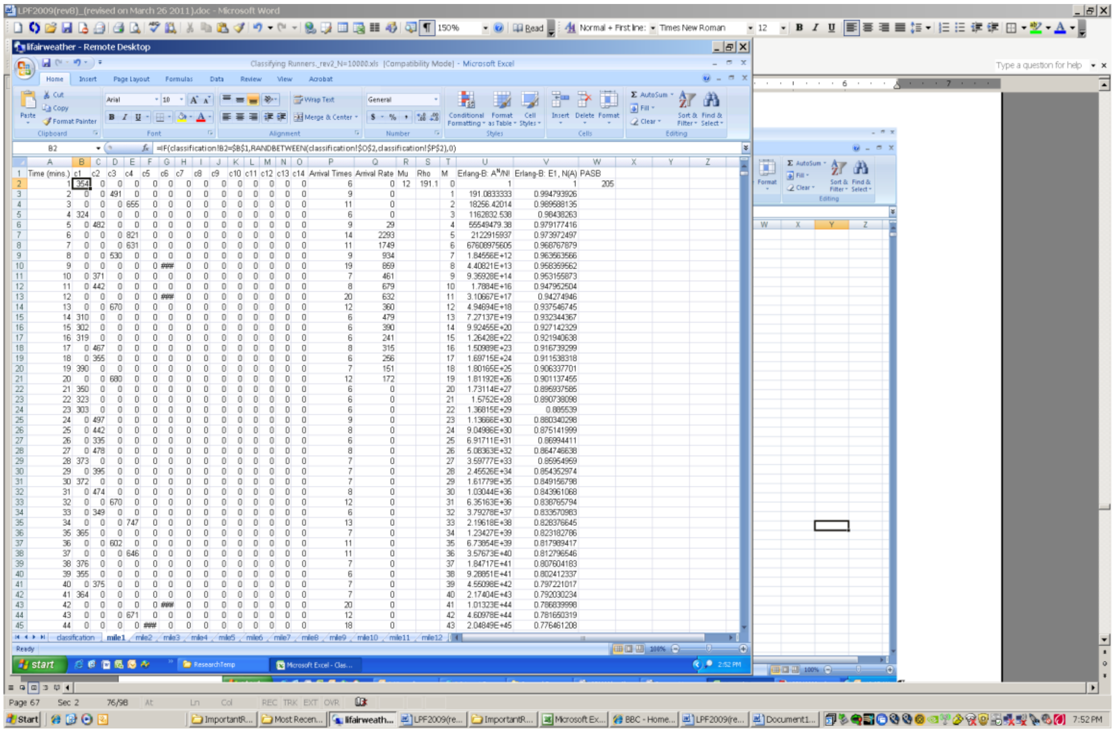
Step 0 of the simulation modeling procedure allows us to create a dataset of runners based on their classification specifications. The remaining steps in the simulation modeling procedure allow for the transitioning of the runners over the entire length of the course, and for determining appropriate staffing levels.

The pace distribution table from Step 0 is shown in cell range O2:P7 of Figure 5.1. For the specific example, we see from the pace distribution table that Runner #1 is a Class 1 runner, and takes between 300 seconds and 390 seconds to run each mile.

For Step 2, we use the RANDBETWEEN() function in Excel to generate a random value between a_c and b_c to represent the time that it takes a Class c runner, $c = 1, 2, \ldots, C$, to traverse the distance between service station $\ell - 1$ and service station, ℓ. Hence, for Step 2, we use the RANDBETWEEN($[a_c, b_c]$) function to simulate the times for each of the N runners at each of the L service stations from the pace distribution table.

Continuing with Runner #1, who we know is a Class 1 runner, for the service station at Mile 1, we execute the RANDBETWEEN($[300, 390]$ function which returns a random value between 300 seconds and 390 seconds, which is the time that it takes Runner #1 to traverse the distance between the start of the race and the service station at the first mile marker. Column B2 of Figure 5.2 shows that the simulated time for Runner #1 to get from the start of the race to the first mile marker is 354 seconds. The paces for the remaining $N - 1$ simulated runners are generated in a similar fashion.

Continuing with Runner #1, we use the RANDBETWEEN($[300, 390]$) function to simulate the elapsed time between the service station at the first mile marker and the service station at the second mile marker for Runner #1. The time that it took Runner #1 to traverse the second mile of the race is 328 seconds, and Column B2 of Figure 5.3 shows that the simulated time for Runner #1 to reach the second mile marker from the start of the race is 682 seconds. The times and paces for the remaining service stations, and for the remaining $N-1$ simulated runners are calculated in a similar fashion.

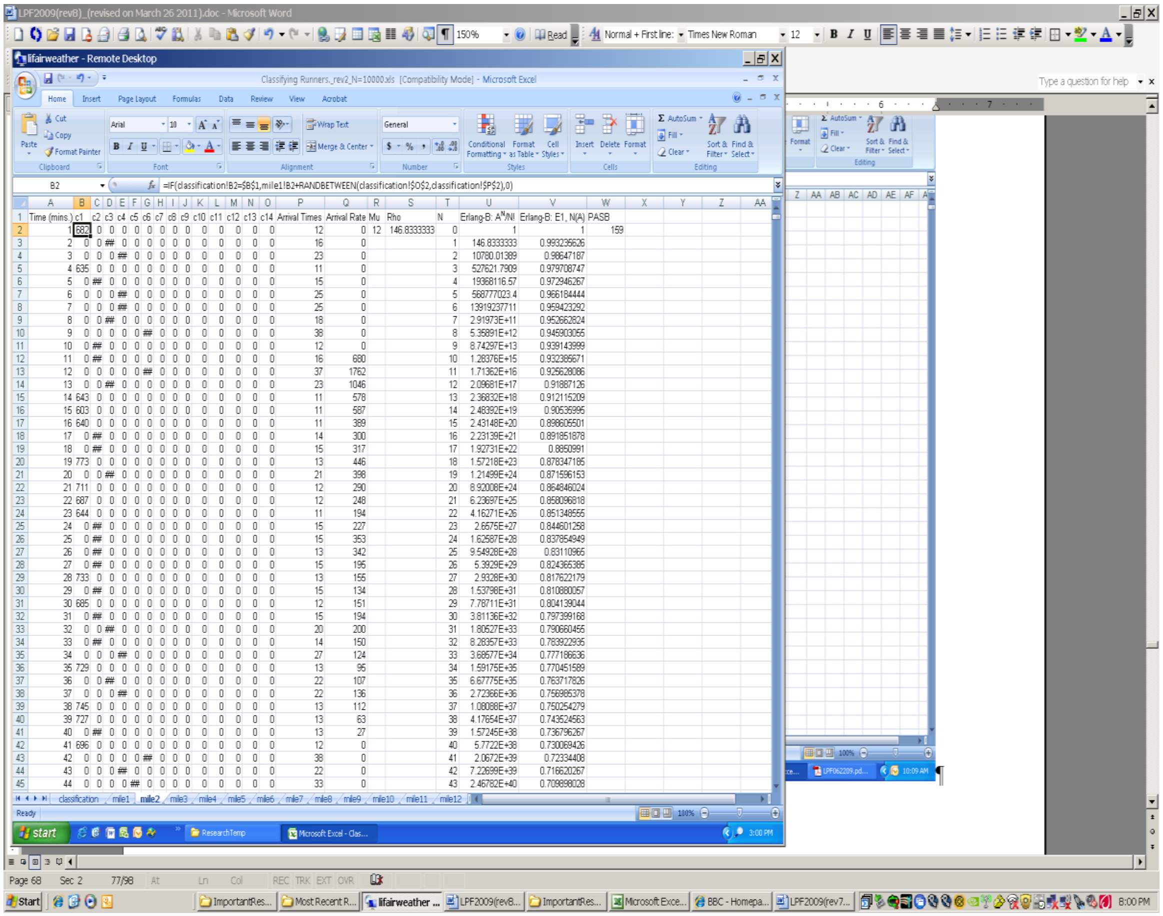

LPF2009(rev8)_(revised on March 26 2011).doc - Microsoft Word
lifairweather - Remote Desktop
Classifying Runners_rev2_N=10000.xls [Compatibility Mode] - Microsoft Excel
Home
Insert
Page Layout
Formulas
Data
Review
View
Acrobat
classification
mile1
mile2
mile3
mile4
mile5
mile6
mile7
mile8
mile9
mile10
mile11
mile12
Ready
Page 67
Sec 2
76/98

B2 =IF(classification!B2=$B$1,RANDBETWEEN(classification!$O$2,classification!$P$2),0)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	Time (mins.)	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	Arrival Times	Arrival Rate	Mu	Rho	M	Erlang-B: A^N/N!	Erlang-B: E1, N(A)	PASB
2	1	354	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	12	191.1	0	1	1	205
3	2	0	0	491	0	0	0	0	0	0	0	0	0	0	0	9	0			1	191.0833333	0.994793926	
4	3	0	0	0	655	0	0	0	0	0	0	0	0	0	0	11	0			2	18256.42014	0.989588135	
5	4	324	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0			3	1162832.538	0.98438263	
6	5	0	482	0	0	0	0	0	0	0	0	0	0	0	0	9	29			4	55549479.38	0.979177416	
7	6	0	0	0	821	0	0	0	0	0	0	0	0	0	0	14	2293			5	2122915937	0.973972497	
8	7	0	0	0	631	0	0	0	0	0	0	0	0	0	0	11	1749			6	67608975605	0.968767879	
9	8	0	0	530	0	0	0	0	0	0	0	0	0	0	0	9	934			7	1.84556E+12	0.963563566	
10	9	0	0	0	0	0	###	0	0	0	0	0	0	0	0	19	859			8	4.40821E+13	0.958359562	
11	10	0	371	0	0	0	0	0	0	0	0	0	0	0	0	7	461			9	9.35928E+14	0.953155873	
12	11	0	442	0	0	0	0	0	0	0	0	0	0	0	0	8	679			10	1.7884E+16	0.947952504	
13	12	0	0	0	0	0	###	0	0	0	0	0	0	0	0	20	632			11	3.10667E+17	0.94274946	
14	13	0	0	670	0	0	0	0	0	0	0	0	0	0	0	12	360			12	4.94694E+18	0.937546745	
15	14	310	0	0	0	0	0	0	0	0	0	0	0	0	0	6	479			13	7.27137E+19	0.932344367	
16	15	302	0	0	0	0	0	0	0	0	0	0	0	0	0	6	390			14	9.92455E+20	0.927142329	
17	16	319	0	0	0	0	0	0	0	0	0	0	0	0	0	6	241			15	1.26428E+22	0.921940638	
18	17	0	467	0	0	0	0	0	0	0	0	0	0	0	0	8	315			16	1.50989E+23	0.916739299	
19	18	0	355	0	0	0	0	0	0	0	0	0	0	0	0	6	256			17	1.69715E+24	0.911538318	
20	19	390	0	0	0	0	0	0	0	0	0	0	0	0	0	7	151			18	1.80165E+25	0.906337701	
21	20	0	0	680	0	0	0	0	0	0	0	0	0	0	0	12	172			19	1.81192E+26	0.901137455	
22	21	350	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0			20	1.73114E+27	0.895937585	
23	22	323	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0			21	1.5752E+28	0.890738098	
24	23	303	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0			22	1.36815E+29	0.885539	
25	24	0	497	0	0	0	0	0	0	0	0	0	0	0	0	9	0			23	1.13666E+30	0.880340298	
26	25	0	442	0	0	0	0	0	0	0	0	0	0	0	0	8	0			24	9.04986E+30	0.875141999	
27	26	0	335	0	0	0	0	0	0	0	0	0	0	0	0	6	0			25	6.91711E+31	0.86994411	
28	27	0	478	0	0	0	0	0	0	0	0	0	0	0	0	8	0			26	5.08363E+32	0.864746638	
29	28	373	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0			27	3.59777E+33	0.85954969	
30	29	0	395	0	0	0	0	0	0	0	0	0	0	0	0	7	0			28	2.45526E+34	0.854352974	
31	30	372	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0			29	1.61779E+35	0.849156798	
32	31	0	474	0	0	0	0	0	0	0	0	0	0	0	0	8	0			30	1.03044E+36	0.843961068	
33	32	0	0	670	0	0	0	0	0	0	0	0	0	0	0	12	0			31	6.35163E+36	0.838765794	
34	33	0	349	0	0	0	0	0	0	0	0	0	0	0	0	6	0			32	3.79278E+37	0.833570983	
35	34	0	0	0	747	0	0	0	0	0	0	0	0	0	0	13	0			33	2.19618E+38	0.828376645	
36	35	365	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0			34	1.23427E+39	0.823182786	
37	36	0	0	602	0	0	0	0	0	0	0	0	0	0	0	11	0			35	6.73854E+39	0.817989417	
38	37	0	0	0	646	0	0	0	0	0	0	0	0	0	0	11	0			36	3.57673E+40	0.812796546	
39	38	376	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0			37	1.84717E+41	0.807604183	
40	39	355	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0			38	9.28851E+41	0.802412337	
41	40	0	375	0	0	0	0	0	0	0	0	0	0	0	0	7	0			39	4.55098E+42	0.797221017	
42	41	364	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0			40	2.17404E+43	0.792030234	
43	42	0	0	0	0	0	###	0	0	0	0	0	0	0	0	20	0			41	1.01323E+44	0.786839998	
44	43	0	0	0	671	0	0	0	0	0	0	0	0	0	0	12	0			42	4.60978E+44	0.781650319	
45	44	0	0	0	0	###	0	0	0	0	0	0	0	0	0	18	0			43	2.04849E+45	0.776461208	

Figure 7: Simulation Implementation: Steps 2 and 3

The final step is to manage the staffing at each service station as specified in Step 3. The Erlang loss formula and the PASB are implemented at each service station. Figure 5.2 shows the implementation of the Monte Carlo simulation at Mile 1. The implementation of the Erlang loss formula is captured in columns R through V, and the implementation of the PASB is captured in cell W2. The service rate per busy server is specified in Cell R2, and the traffic intensity is specified in Cell S2. For the Erlang loss formula, the numbers of servers for a specified blocking probability are shown in Column T; Columns U and V show the implementation of (2.1). Column V is of particular interest because it contains the blocking probabilities for a prescribed staffing level. For example, as can be seen in Figure 5.2, given the mix of runners, the blocking probability is approximately 87 percent when there are only 25 servers. Cell W2 shows that the number of servers based on the PASB is 205 at Mile 1.

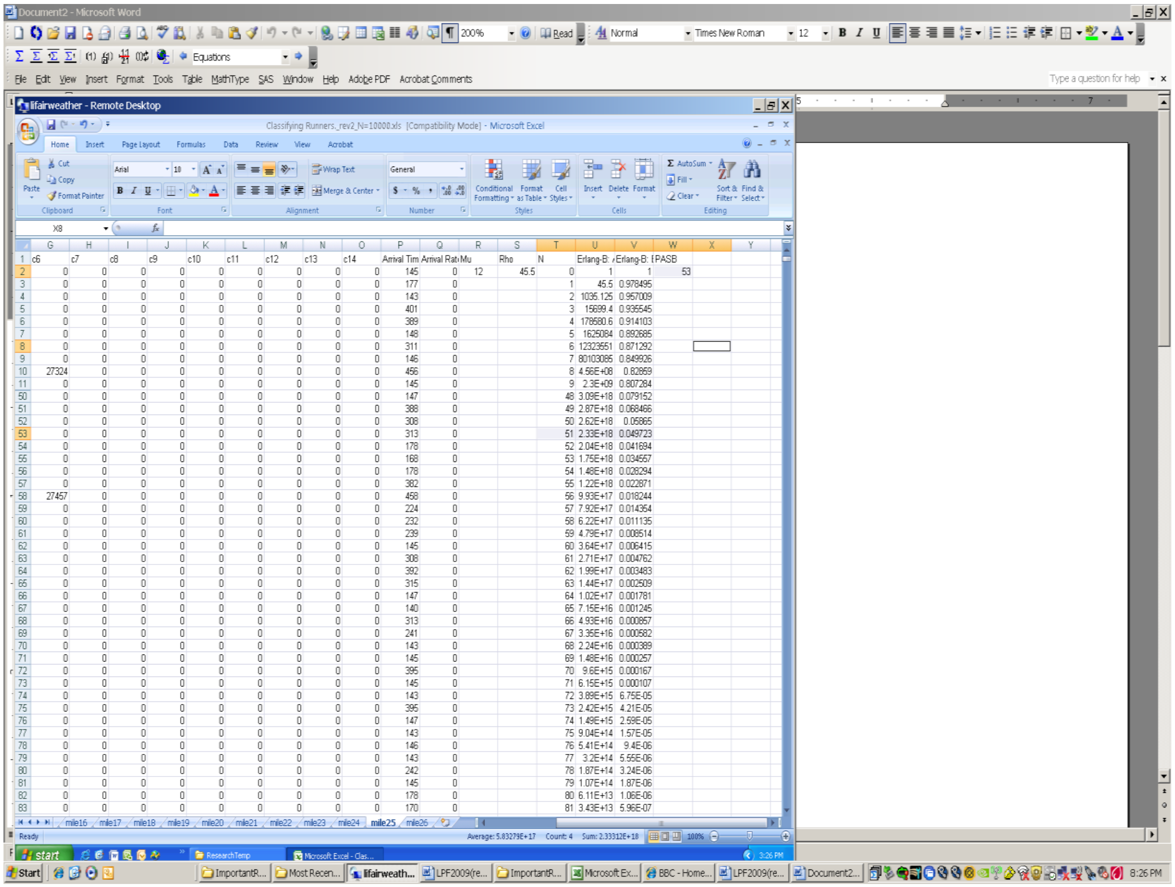

LPF2009(rev8)_(revised on March 26 2011).doc - Microsoft Word
lifairweather - Remote Desktop
Classifying Runners_rev2_N=10000.xls [Compatibility Mode] - Microsoft Excel
Home
Insert
Page Layout
Formulas
Data
Review
View
Acrobat
classification
mile1
mile2
mile3
mile4
mile5
mile6
mile7
mile8
mile9
mile10
mile11
mile12
Ready
start
ResearchTemp
Microsoft Excel - Clas...
Page 68
Sec 2
77/98

B2 =IF(classification!B2=$B$1,mile1!B2+RANDBETWEEN(classification!$O$2,classification!$P$2),0)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	Time (mins.)	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12	c13	c14	Arrival Times	Arrival Rate	Mu	Rho	N	Erlang-B: $A^N/N!$	Erlang-B: E1, N(A)	PASB
2	1	682	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0	12	146.8333333	0	1	1	159
3	2	0	0	##	0	0	0	0	0	0	0	0	0	0	0	16	0			1	146.8333333	0.993236626	
4	3	0	0	0	##	0	0	0	0	0	0	0	0	0	0	23	0			2	10780.01389	0.98647187	
5	4	635	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0			3	527621.7909	0.979708747	
6	5	0	##	0	0	0	0	0	0	0	0	0	0	0	0	15	0			4	19368116.57	0.972946267	
7	6	0	0	0	##	0	0	0	0	0	0	0	0	0	0	25	0			5	568777023.4	0.966184444	
8	7	0	0	0	##	0	0	0	0	0	0	0	0	0	0	25	0			6	13919237711	0.959423292	
9	8	0	0	##	0	0	0	0	0	0	0	0	0	0	0	18	0			7	2.91973E+11	0.952662824	
10	9	0	0	0	0	0	##	0	0	0	0	0	0	0	0	38	0			8	5.35891E+12	0.945903056	
11	10	0	##	0	0	0	0	0	0	0	0	0	0	0	0	12	0			9	8.74297E+13	0.939143999	
12	11	0	##	0	0	0	0	0	0	0	0	0	0	0	0	16	680			10	1.28376E+15	0.932385671	
13	12	0	0	0	0	0	##	0	0	0	0	0	0	0	0	37	1762			11	1.71362E+16	0.925628086	
14	13	0	0	##	0	0	0	0	0	0	0	0	0	0	0	23	1046			12	2.09681E+17	0.91887126	
15	14	643	0	0	0	0	0	0	0	0	0	0	0	0	0	11	578			13	2.36832E+18	0.912115209	
16	15	603	0	0	0	0	0	0	0	0	0	0	0	0	0	11	587			14	2.48392E+19	0.90535995	
17	16	640	0	0	0	0	0	0	0	0	0	0	0	0	0	11	389			15	2.43148E+20	0.898605501	
18	17	0	##	0	0	0	0	0	0	0	0	0	0	0	0	14	300			16	2.23139E+21	0.891851878	
19	18	0	##	0	0	0	0	0	0	0	0	0	0	0	0	15	317			17	1.92731E+22	0.8850991	
20	19	773	0	0	0	0	0	0	0	0	0	0	0	0	0	13	446			18	1.57218E+23	0.878347185	
21	20	0	0	##	0	0	0	0	0	0	0	0	0	0	0	21	398			19	1.21499E+24	0.871596153	
22	21	711	0	0	0	0	0	0	0	0	0	0	0	0	0	12	290			20	8.92008E+24	0.864846024	
23	22	687	0	0	0	0	0	0	0	0	0	0	0	0	0	12	248			21	6.23697E+25	0.858096818	
24	23	644	0	0	0	0	0	0	0	0	0	0	0	0	0	11	194			22	4.16271E+26	0.851348555	
25	24	0	##	0	0	0	0	0	0	0	0	0	0	0	0	15	227			23	2.6575E+27	0.844601258	
26	25	0	##	0	0	0	0	0	0	0	0	0	0	0	0	15	353			24	1.62587E+28	0.837854949	
27	26	0	##	0	0	0	0	0	0	0	0	0	0	0	0	13	342			25	9.54928E+28	0.83110965	
28	27	0	##	0	0	0	0	0	0	0	0	0	0	0	0	15	195			26	5.3929E+29	0.824365385	
29	28	733	0	0	0	0	0	0	0	0	0	0	0	0	0	13	155			27	2.9328E+30	0.817622179	
30	29	0	##	0	0	0	0	0	0	0	0	0	0	0	0	15	134			28	1.53798E+31	0.810880057	
31	30	685	0	0	0	0	0	0	0	0	0	0	0	0	0	12	151			29	7.78711E+31	0.804139044	
32	31	0	##	0	0	0	0	0	0	0	0	0	0	0	0	15	194			30	3.81136E+32	0.797399168	
33	32	0	0	##	0	0	0	0	0	0	0	0	0	0	0	20	200			31	1.80527E+33	0.790660455	
34	33	0	##	0	0	0	0	0	0	0	0	0	0	0	0	14	150			32	8.28357E+33	0.783922935	
35	34	0	0	0	##	0	0	0	0	0	0	0	0	0	0	27	124			33	3.68577E+34	0.777186636	
36	35	729	0	0	0	0	0	0	0	0	0	0	0	0	0	13	95			34	1.59175E+35	0.770451589	
37	36	0	0	##	0	0	0	0	0	0	0	0	0	0	0	22	107			35	6.67775E+35	0.763717826	
38	37	0	0	0	##	0	0	0	0	0	0	0	0	0	0	22	136			36	2.72366E+36	0.756985378	
39	38	745	0	0	0	0	0	0	0	0	0	0	0	0	0	13	112			37	1.08088E+37	0.750254279	
40	39	727	0	0	0	0	0	0	0	0	0	0	0	0	0	13	63			38	4.17654E+37	0.743524563	
41	40	0	##	0	0	0	0	0	0	0	0	0	0	0	0	13	27			39	1.57245E+38	0.736796267	
42	41	696	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0			40	5.7722E+38	0.730069426	
43	42	0	0	0	0	0	##	0	0	0	0	0	0	0	0	38	0			41	2.0672E+39	0.72334408	
44	43	0	0	0	##	0	0	0	0	0	0	0	0	0	0	22	0			42	7.22699E+39	0.716620267	
45	44	0	0	0	0	##	0	0	0	0	0	0	0	0	0	33	0			43	2.46782E+40	0.709898028	

Figure 8: Simulation Implementation: Steps 2 and 3

Continuing with our example, as can be seen from Figure 5.3, given the mix of runners in the example of $N = 10000$ runners, if only 25 servers are placed at Mile 2, then the blocking probability is approximately 83 percent. As seen in cell W2, the number of servers based on the PASB is 159 at Mile 2.


Document2 - Microsoft Word
File Edit View Insert Format Tools Table MathType SAS Window Help Adobe PDF Acrobat Comments
lifairweather - Remote Desktop
Classifying Runners_rev2_N=10000.xls [Compatibility Mode] - Microsoft Excel
Home Insert Page Layout Formulas Data Review View Acrobat
X8
mile16 mile17 mile18 mile19 mile20 mile21 mile22 mile23 mile24 mile25 mile26
start
ResearchTemp
Microsoft Excel - Clas...
Start
ImportantR... Most Recen... lifairweath... LPF2009(re... ImportantR... Microsoft Ex... BBC - Home... LPF2009(re... Document2...

	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	c6	c7	c8	c9	c10	c11	c12	c13	c14	Arrival Tim	Arrival Rat	Mu	Rho	N	Erlang-B:	Erlang-B:	PASB		
2	0	0	0	0	0	0	0	0	0	145	0	12	45.5	0	1	1	53		
3	0	0	0	0	0	0	0	0	0	177	0			1	45.5	0.978495			
4	0	0	0	0	0	0	0	0	0	143	0			2	1035.125	0.957009			
5	0	0	0	0	0	0	0	0	0	401	0			3	15699.4	0.935545			
6	0	0	0	0	0	0	0	0	0	389	0			4	178580.6	0.914103			
7	0	0	0	0	0	0	0	0	0	148	0			5	1625084	0.892685			
8	0	0	0	0	0	0	0	0	0	311	0			6	12323551	0.871292			
9	0	0	0	0	0	0	0	0	0	146	0			7	80103085	0.849926			
10	27324	0	0	0	0	0	0	0	0	456	0			8	4.56E+08	0.82859			
11	0	0	0	0	0	0	0	0	0	145	0			9	2.3E+09	0.807284			
50	0	0	0	0	0	0	0	0	0	147	0			48	3.09E+18	0.079152			
51	0	0	0	0	0	0	0	0	0	388	0			49	2.87E+18	0.068466			
52	0	0	0	0	0	0	0	0	0	308	0			50	2.62E+18	0.05865			
53	0	0	0	0	0	0	0	0	0	313	0			51	2.33E+18	0.049723			
54	0	0	0	0	0	0	0	0	0	178	0			52	2.04E+18	0.041694			
55	0	0	0	0	0	0	0	0	0	168	0			53	1.75E+18	0.034557			
56	0	0	0	0	0	0	0	0	0	178	0			54	1.48E+18	0.028294			
57	0	0	0	0	0	0	0	0	0	382	0			55	1.22E+18	0.022871			
58	27457	0	0	0	0	0	0	0	0	458	0			56	9.93E+17	0.018244			
59	0	0	0	0	0	0	0	0	0	224	0			57	7.92E+17	0.014354			
60	0	0	0	0	0	0	0	0	0	232	0			58	6.22E+17	0.011135			
61	0	0	0	0	0	0	0	0	0	239	0			59	4.79E+17	0.008514			
62	0	0	0	0	0	0	0	0	0	145	0			60	3.64E+17	0.006415			
63	0	0	0	0	0	0	0	0	0	308	0			61	2.71E+17	0.004762			
64	0	0	0	0	0	0	0	0	0	392	0			62	1.99E+17	0.003483			
65	0	0	0	0	0	0	0	0	0	315	0			63	1.44E+17	0.002509			
66	0	0	0	0	0	0	0	0	0	147	0			64	1.02E+17	0.001781			
67	0	0	0	0	0	0	0	0	0	140	0			65	7.15E+16	0.001245			
68	0	0	0	0	0	0	0	0	0	313	0			66	4.93E+16	0.000857			
69	0	0	0	0	0	0	0	0	0	241	0			67	3.35E+16	0.000582			
70	0	0	0	0	0	0	0	0	0	143	0			68	2.24E+16	0.000389			
71	0	0	0	0	0	0	0	0	0	145	0			69	1.48E+16	0.000257			
72	0	0	0	0	0	0	0	0	0	395	0			70	9.6E+15	0.000167			
73	0	0	0	0	0	0	0	0	0	145	0			71	6.15E+15	0.000107			
74	0	0	0	0	0	0	0	0	0	143	0			72	3.89E+15	6.75E-05			
75	0	0	0	0	0	0	0	0	0	395	0			73	2.42E+15	4.21E-05			
76	0	0	0	0	0	0	0	0	0	147	0			74	1.49E+15	2.59E-05			
77	0	0	0	0	0	0	0	0	0	143	0			75	9.04E+14	1.57E-05			
78	0	0	0	0	0	0	0	0	0	146	0			76	5.41E+14	9.4E-06			
79	0	0	0	0	0	0	0	0	0	143	0			77	3.2E+14	5.55E-06			
80	0	0	0	0	0	0	0	0	0	242	0			78	1.87E+14	3.24E-06			
81	0	0	0	0	0	0	0	0	0	145	0			79	1.07E+14	1.87E-06			
82	0	0	0	0	0	0	0	0	0	178	0			80	6.11E+13	1.06E-06			
83	0	0	0	0	0	0	0	0	0	170	0			81	3.43E+13	5.96E-07			

Ready Average: 5.83279E+17 Count: 4 Sum: 2.33312E+18 100%

Figure 9: Simulation Implementation: Steps 2 and 3

Figure 5.4 shows the implementation of Steps 2 and 3 at Mile 25. Given the mix of runners in our sample of $N = 10000$ runners, the number of servers based on the Erlang loss formula with $P_b = 0.05$ is 51, and the number of servers based on the PASB is 53.

The Erlang loss formula and the PASB are implemented at the remaining service stations in a similar fashion to the implementations in the examples in Figures 5.2 – 5.4. With respect to the Erlang formula, for any service station of interest, the number of servers required for smaller blocking probabilities can be found by scrolling down on the spreadsheet, where it will be seen that the required number of servers increases as the blocking probability decreases.

5.1.3 Alternative Approaches and Implementation Limitations

Before we close this section, we discuss alternative ways to develop the multi-class simulation model. We also discuss some pros and cons in our choice of implementation platform - Microsoft Excel. Of course, the simulation model’s accuracy is bound by the accuracy of EXCEL and its random number generator.

The first area in which the simulation model can be improved is in computational efficiency. The way in which we "classify" runners can be improved to make the model less computationally expensive. In the model that we have developed, "classifications" result from the assignment of random numbers to a pre-determined class. In doing so, we draw a “class” for each of N simulated runners when we know what proportion of the runners are expected to belong to each class. Though this approach is theoretically appealing (because it allows us to account for variability in the number of runners in each class), an alternative approach that may be less computationally expensive is to compute the number of runners in Class c by multiplying the total number of runners in the race, N , by the proportion, ρ_c , of runners that is expected to be in that class, Class c . For example, the total number of runners in Class 1 is simply $N \times \rho_1$. The number of runners in Class 2 is $N \times \rho_2$, and so on.

Another area in which the model can be improved is in the choice of implementation platform. The computational efficiency of the LOOKUP algorithms in Excel were not analyzed and compared against LOOKUP algorithms that can be utilized on other popular alternative implementation platforms such as C++, Java, or Matlab. With computational efficiency (and speed of analysis) as the objective, such analyses may lead to a different choice of implementation platform.

We implemented the simulation model in Microsoft Excel, which limits the size of the race to the allowable number of rows in Excel. The number of rows in Excel is 65536 for Excel 2003 users and over 1000000 for Excel 2007 users. Since Excel contains (pre-coded) functions that can be applied to key parts of the simulation model development, this makes Excel an attractive choice for the kinds of problems that this research considers. However, we caution that this choice may not be optimal, at least computationally.

In the next section, we will conduct a set of experiments with our simulation model, and give staffing results based on the PASB.

5.2 <u>Experiments with the Simulation Model</u>

In this section, we conduct a set of experiments to demonstrate some important principles of the simulation model. The aim of these experiments is two-fold. First, we will demonstrate how to simulate races of different mixes of runners, and second, we will demonstrate how to simulate races of different sizes. For each of the experiments, we use λ_ℓ^* as the arrival rates, and $\mu=12$ customers per minute as the service rate at all service stations. Staffing results are based on the PASB and the Erlang loss formula with a blocking probability of 5%, and are based on ten Monte Carlo runs of each of the experiments, thereby illustrating an important principle of Monte Carlo methods that we discussed in Section 2.5. That is, by replicating and averaging the results of multiple runs of the experiments, we can get a good estimate of the variable of interest (which, in our application, is the number of servers at each of the L service stations) in the $M_\ell(t)/G_\ell/s_\ell/s_\ell/N/FIFO$ queueing network.

5.2.1 Runner-Mix Experiments

Using our simulation model to conduct runner-mix experiments, a road race manager can simulate a race with as diverse a group of runners as the manager envisions. For example, if a manager wants to simulate a long distance road race of size N that is comprised of runners of similar capabilities (in terms of their overall paces), then the manager only needs to simulate a single class of runner. On the other hand, if the runners' paces are less homogeneous, then the manager can group them by pace, which forms classes of runners. Then during the simulation, the (approximate) numbers of runners in each of the C classes of runners are generated.

The experiments that are conducted here are examples of the latter scenario. For each experiment, the service rate is fixed at 12 runners per minute. However, it is worth noting that in our implementation, the service rate can be changed at each service station to any desired rate. Similarly, a distinct blocking probability could be specified for each service station as well.

In these experiments, the mixes of runners are different, but the size of the races is the same. We conducted two experiments, Experiment 5.1 and Experiment 5.2, with $N = 10000$ runners in both experiments. Experiment 5.1 has the classifications shown in the second column of Table 17, and the pace distributions that are shown in the fourth column of Table 17, and Experiment 5.2 has the classifications shown in the third column of Table 17, and the pace distributions that are shown in the fourth column of Table 17. For both of these experiments, we use basic principles of Monte Carlo methods and replicated the steps from Section 5.1.1 ten times at each of the 25 service stations.

Table 17: Runners Pace Distribution and Probability of Occurrence

lass	Probabilities for Experiments 5.1 and 5.3	Probabilities for Experiment 5.2	Pace (seconds/mile)
1	0.10	0.30	300 – 390
2	0.20	0.20	390 - 510
3	0.30	0.20	510 - 690
4	0.20	0.15	690 - 870
5	0.15	0.10	870 - 1050
6	0.05	0.05	1050 - 1200

Tables 18 to 21 give the number of servers for each of the ten Monte Carlo experiments or replications (for the PASB and the Erlang loss formula), and the mean and standard deviation (which we round up to the nearest integer value) of the ten replications. The mean and standard deviation (in parentheses) of the prescribed staffing requirements are shown in the tables.

Table 18: Server Requirements Based on 10 Monte Carlo Replications of Experiment 5.1 Using the PASB with μ=12

S ervice S tation	un 1: of Serve rs	un 2: of ervers	un 3: of ervers	un 4: of ervers	un 5: of ervers	un 6: of ervers	un 7: # of ervers	un 8: # of Server s	un 9: # of Server s	un 10: # of Server s	ean of ervers
M ile 1	7	5	6	6	6	5	4	7	7	4	6 (2)
M ile 2	2	9	2	2	8	6	8	9	3	8	0 (3)
M ile 3	0	0	9	2	0	1	9	9	9	0	0 (1)
M ile 4	2	8	0	5	1	2	8	2	6	1	3 (3)
M ile 5	8	8	6	7	6	9	7	9	2	7	8 (2)
M ile 6	2	3	2	6	4	1	2	2	4	3	3 (2)
M ile 7	0	0	0	1	9	0	9	9	0	0	0 (1)
M ile 8	8	8	7	7	0	6	6	6	8	6	8 (2)
M ile 9	6	8	7	6	4	3	4	3	7	6	6 (2)
M ile 10	4	6	3	2	2	1	2	3	3	3	3 (2)
M ile 11	3	3	1	1	2	1	4	2	4	2	3 (2)
M ile 12	1	2	9	2	1	0	9	0	8	2	1 (2)
M ile 13	0	9	8	0	8	9	8	0	6	7	9 (2)
M ile 14	9	7	8	7	6	8	7	9	7	8	8 (1)
M ile 15	6	7	6	6	8	7	6	8	6	6	7 (1)
M ile 16	4	5	8	7	5	4	5	6	6	5	6 (2)
M ile 17	4	5	5	6	3	3	4	5	3	3	5 (2)
M ile 18	3	5	5	3	5	3	4	5	3	4	4 (1)
M ile 19	1	3	2	2	5	4	1	2	3	2	3 (2)
M ile 20	2	2	1	1	0	1	4	2	2	1	2 (2)
M											

ile 21	2	0	1	4	3	1	1	3	1	0	2 (2)
M ile 22	1	3	9	2	2	1	2	1	2	0	2 (2)
M ile 23	1	1	1	0	1	0	2	0	1	9	1 (1)
M ile 24	0	0	9	0	1	9	9	2	1	7	0 (2)
M ile 25	0	0	9	0	0	8	8	9	1	9	0 (1)

Table 19: Server Requirements Based on 10 Monte Carlo Replications of Experiment 5.1 Using Erlang Loss Formula with Blocking Probability <=0.05 and μ=12

S ervice S tation	un 1: of Servers	un 2: of ervers	un 3: of ervers	un 4: of ervers	un 5: of ervers	un 6: of ervers	un 7: # of ervers	un 8: # of Servers	un 9: # of Servers	un 10: # of Servers	ean of ervers
M ile 1	4	5	7	8	7	4	5	5	7	9	7 (2)
M ile 2	4	2	4	3	1	1	2	0	2	5	3 (2)
M ile 3	6	5	7	5	6	7	7	6	7	7	7 (1)
M ile 4	9	6	8	1	8	8	7	9	6	7	8 (2)
M ile 5	2	2	1	3	1	2	0	2	0	1	2 (1)
M ile 6	8	7	7	8	8	7	7	7	7	8	8 (1)
M ile 7	4	4	4	4	3	5	4	3	3	4	4 (1)
M ile 8	2	2	3	3	2	4	1	2	2	1	3 (1)
M ile 9	0	0	9	0	0	0	0	9	0	0	0 (1)
M ile 10	8	8	8	8	8	8	9	9	9	8	9 (1)
M ile 11	7	7	7	8	7	8	8	7	7	8	8 (1)
M ile 12	8	6	7	7	6	6	6	7	7	8	7 (1)
M ile 13	6	6	5	5	5	5	4	6	6	6	6 (1)
M ile 14	5	4	5	6	5	5	5	4	5	5	5 (1)
M ile 15	4	4	4	4	4	4	4	4	5	5	5 (1)
M ile 16	4	4	4	3	4	4	4	4	3	3	4 (1)
M ile 17	3	3	3	3	3	3	3	3	2	3	3 (1)
M ile 18	3	2	2	3	2	2	3	2	13	3	3 (1)
M ile 19	2	2	2	2	3	2	2	2	2	2	3 (1)
M ile 20	2	2	1	2	2	2	1	2	2	2	2 (1)
M ile 21	2	2	1	1	3	1	1	2	1	2	2 (1)
M ile 22	1	2	1	2	1	1	1	2	1	2	2 (1)
M ile 23	0	1	1	0	0	0	0	1	1	1	1 (1)
M											

ile 24	0	1	1	1	0	1	1	1	1	1	1 (1)
M ile 25	0	0	0	0	0	0	0	0	0	1	1 (1)

Table 20: Server Requirements Based on 10 Monte Carlo Replications of Experiment 5.2 Using the PASB with μ=12

S ervice S tation	un 1: of Servers	un 2: of ervers	un 3: of ervers	un 4: of ervers	un 5: of ervers	un 6: of ervers	un 7: # of ervers	un 8: # of Servers	un 9: # of Servers	un 10: # of Servers	ean of ervers
M ile 1	79	75	76	82	75	82	79	72	72	82	78 (4)
M ile 2	50	49	48	52	46	51	48	52	45	53	50 (3)
M ile 3	28	21	20	26	20	26	20	30	23	26	24 (4)
M ile 4	04	02	02	03	04	06	03	06	01	05	04 (2)
M ile 5	03	01	03	04	04	03	03	05	01	06	04 (2)
M ile 6	6	4	8	6	6	8	02	00	1	00	8 (4)
M ile 7	8	1	2	7	0	4	1	1	7	0	1 (3)
M ile 8	3	1	2	6	0	4	4	4	8	3	3 (3)
M ile 9	1	1	4	5	9	0	0	8	6	0	1 (3)
M ile 10	9	7	0	9	8	0	9	9	5	9	9 (2)
M ile 11	4	2	4	5	1	6	2	2	8	6	3 (3)
M ile 12	2	0	9	0	0	1	9	9	8	1	0 (2)
M ile 13	1	6	9	6	0	9	0	0	8	3	0 (3)
M ile 14	0	7	8	5	7	0	7	7	5	2	8 (3)
M ile 15	7	7	4	3	3	5	7	5	3	8	6 (2)
M ile 16	4	1	1	3	0	4	4	2	0	4	3 (2)
M ile 17	3	2	1	2	9	4	3	8	9	6	2 (3)
M ile 18	2	1	9	0	8	0	0	9	1	2	1 (2)
M ile 19	1	9	7	9	9	9	0	6	8	2	9 (2)
M ile 20	1	6	8	7	6	0	7	6	6	9	8 (2)
M ile 21	8	6	8	6	6	7	4	4	4	6	6 (2)
M ile 22	5	3	6	4	7	6	3	1	2	5	5 (2)
M ile 23	7	4	3	6	4	4	1	2	1	6	4 (3)
M											

ile 24	3	1	5	4	2	5	3	3	0	3	3 (2)
M ile 25	3	1	3	3	1	3	1	9	0	6	2 (2)

Table 21: Server Requirements Based on 10 Monte Carlo Replications of Experiment 5.2 Using Erlang Loss Formula with Blocking Probability <=0.05 and μ=12

S ervice S tation	un 1: of Servers	un 2: of ervers	un 3: of ervers	un 4: of ervers	un 5: of ervers	un 6: of ervers	un 7: # of ervers	un 8: # of Servers	un 9: # of Servers	un 10: # of Servers	ean of ervers
M ile 1	/F	/F	/F	/F	/F	/F	/F	/F	/F	/F	/A
M ile 2	43	/F	43	/F	/F	/F	43	/F	40	/F	43 (2)
M ile 3	11	15	14	18	12	16	12	17	14	14	15 (3)
M ile 4	08	07	08	15	13	11	11	08	07	13	11 (3)
M ile 5	9	1	0	4	4	9	0	0	3	9	1 (3)
M ile 6	3	0	9	1	0	5	2	3	1	2	2 (2)
M ile 7	2	3	3	8	1	1	5	2	2	3	3 (3)
M ile 8	7	0	1	3	4	8	8	2	0	8	1 (3)
M ile 9	4	7	8	9	7	6	7	9	4	4	7 (2)
M ile 10	5	4	5	6	4	3	0	3	2	3	4 (2)
M ile 11	2	9	1	4	0	1	0	0	0	2	1 (2)
M ile 12	9	7	7	1	7	9	9	0	0	8	9 (2)
M ile 13	7	6	5	9	8	8	7	8	7	9	8 (2)
M ile 14	6	3	3	7	5	6	5	8	7	8	6 (2)
M ile 15	4	2	2	6	3	4	5	3	6	4	4 (2)
M ile 16	3	3	2	2	2	3	3	3	2	2	3 (1)
M ile 17	0	9	8	0	9	0	1	0	3	3	1 (2)
M ile 18	8	1	7	1	1	0	0	7	2	9	0 (2)
M ile 19	8	8	7	9	9	0	8	8	0	9	9 (1)
M ile 20	7	1	5	8	7	8	9	9	0	7	9 (2)
M ile 21	6	8	5	9	9	8	8	6	6	9	8 (2)
M ile 22	6	7	2	6	6	7	6	6	8	7	7 (2)
M ile 23	4	5	3	6	4	6	6	5	4	6	5 (2)
M											

ile 24	3	3	2	5	4	3	4	4	4	5	4 (1)
M ile 25	3	2	1	4	3	3	3	5	5	5	4 (2)

The relationship between the Erlang loss formula and the PASB that were discussed in Chapter 4 are noticeable from Tables 18 to 21. Two additional points are noticeable from Tables 18 to 21. The first point is that, as indicated by the standard deviations (in parentheses), there is very little variability around the prescribed numbers of servers at any service station in the queueing network. The second point is that, based on the results in Tables 18 to 21, there appears to be some significant impact on the required number of servers when we change the mix of runners as we did in our experiments. We also observe that the number of required servers appears to level off as the distance from the starting line increases.

5.2.2 Race Size Experiment

For this experiment, we want to know the impact on staffing of changing the size of the race. We conducted one additional experiment (Experiment 5.3) that we compared to Experiment 5.1 from Section 5.2.1. In conducting Experiments 5.3, we increased the size of the race in Experiments 5.1 to 20000. For Experiments 5.1 and 5.3, the mix of runners is the same, but the sizes of the two races are different. The probabilities and paces for this experiment are given in Table 17.

Table 22: Server Requirements Based on 10 Monte Carlo Replications of Experiment 5.3 Using the PASB with μ=12

ervice tation	un 1: of Servers	un 2: of ervers	un 3: of ervers	un 4: of ervers	un 5: of ervers	un 6: of ervers	un 7: # of ervers	F un 8: # of Servers	un 9: # of Servers	un 10: # of Servers	ean of ervers
ile 1	83	83	86	82	80	78	82	1 86	82	82	83 (3)
ile 2	53	55	52	53	53	52	59	1 55	52	51	54 (3)
ile 3	18	21	21	19	15	16	16	1 18	17	15	18 (3)
ile 4	13	15	18	18	11	14	13	1 14	14	12	15 (3)
ile 5	6	01	03	03	8	8	8	1 02	6	5	9 (4)
ile 6	5	7	7	7	3	1	3	8 5	5	5	5 (2)
ile 7	9	9	3	9	8	7	0	8 3	1	2	1 (3)
ile 8	3	4	4	3	3	3	5	7 4	6	4	4 (1)
ile 9	2	0	1	3	3	1	0	7 1	2	2	2 (2)
ile 10	8	8	2	7	1	8	1	6 5	7	5	9 (3)
ile 11	4	5	6	4	5	5	7	6 3	5	5	5 (2)
ile 12	3	5	3	4	3	2	4	6 3	2	9	3 (2)
ile 13	1	2	1	3	1	1	2	6 1	3	9	2 (2)
ile 14	7	0	1	8	1	0	9	5 8	1	7	0 (2)
ile 15	9	9	8	6	8	7	8	5 6	6	6	8 (2)
ile 16	5	5	8	7	8	7	4	5 4	6	3	6 (2)
ile 17	3	5	4	5	7	4	3	5 4	4	2	5 (2)
ile 18	3	5	2	2	3	5	4	5 1	4	0	3 (2)
ile 19	0	2	2	0	3	2	2	5 3	2	9	2 (2)
ile 20	1	0	3	0	3	0	3	5 3	9	0	2 (2)
ile 21	0	0	2	9	0	9	0	4 9	9	9	0 (1)
ile 22	1	9	1	7	0	9	0	4 9	8	6	9 (2)
ile 23	9	8	0	7	9	8	8	4 6	7	8	8 (2)
								4			

ile 24	8	8	9	8	8	8	9	6	8	7	8 (1)
ile 25	5	7	8	6	6	8	8	4 7	5	5	7 (2)

Table 23: Server Requirements Based on 10 Monte Carlo Replications of Experiment 5.3 Using Erlang Loss Formula with Blocking Probability <=0.05 and μ=12

ervice tation	un 1: of Servers	un 2: of ervers	un 3: of ervers	un 4: of ervers	un 5: of ervers	un 6: of ervers	un 7: # of ervers	F un 8: # of Servers	un 9: # of Servers	un 10: # of Servers	ean of ervers
ile 1	/F	/F	/F	/F	/F	/F	/F	C /F	/F	/F	/A
ile 2	43	/F	43	/F	/F	/F	43	C /F	40	/F	43 (2)
ile 3	11	15	14	18	12	16	12	1 17	14	14	15 (3)
ile 4	08	07	08	15	13	11	11	1 08	07	13	11 (3)
ile 5	9	1	0	4	4	9	0	9 0	3	9	1 (3)
ile 6	3	0	9	1	0	5	2	8 3	1	2	2 (2)
ile 7	2	3	3	8	1	1	5	7 2	2	3	3 (3)
ile 8	7	0	1	3	4	8	8	7 2	0	8	1 (3)
ile 9	4	7	8	9	7	6	7	6 9	4	4	7 (2)
ile 10	5	4	5	6	4	3	0	6 3	2	3	4 (2)
ile 11	2	9	1	4	0	1	0	6 0	0	2	1 (2)
ile 12	9	7	7	1	7	9	9	6 0	0	8	9 (2)
ile 13	7	6	5	9	8	8	7	5 8	7	9	8 (2)
ile 14	6	3	3	7	5	6	5	5 8	7	8	6 (2)
ile 15	4	2	2	6	3	4	5	5 3	6	4	4 (2)
ile 16	3	3	2	2	2	3	3	5 3	2	2	3 (1)
ile 17	0	9	8	0	9	0	1	5 0	3	3	1 (2)
ile 18	8	1	7	1	1	0	0	4 7	2	9	0 (2)
ile 19	8	8	7	9	9	0	8	4 8	0	9	9 (1)
ile 20	7	1	5	8	7	8	9	4 9	0	7	9 (2)
ile 21	6	8	5	9	9	8	8	4 6	6	9	8 (2)
ile 22	6	7	2	6	6	7	6	4 6	8	7	7 (2)
ile 23	4	5	3	6	4	6	6	4 5	4	6	5 (2)
								4			

ile 24	3	3	2	5	4	3	4	4	4	5	4 (1)
ile 25	3	2	1	4	3	3	3	4 5	5	5	4 (2)

For the results for Experiment 5.3 that are shown in Tables 22 and 23, we use basic principles of Monte Carlo methods and replicated the model (see Section 5.1.1) ten times at each of the 25 service stations. Similar patterns as those in the two earlier experiments regarding the corroboration of the relationship between the PASB and the Erlang loss formula from Chapter 4, the variability of the prescribed staffing levels and the leveling off of the number of servers as the distance from the starting line of the race increases are observed here. Given the same mix of runners, we expect the peak arrival rates in Experiment 5.3 (which has twice as many runners as Experiment 5.1) to be larger than the peak arrival rates in Experiment 5.1, thus, leading to the higher prescribed server levels for Experiment 5.3.

5.3 Chapter Summary

In this chapter, we developed a multi-class simulation model for the mixed $M_\ell(t)/G_\ell/s_\ell/s_\ell/N/FIFO$ queueing network. This simulation model allows road race managers to simulate a race of size N, and to classify those N runners, based on their (historical or assumed) paces, into $C, c=1,2,\ldots,C$ classes of runners. We integrated the Erlang loss formula and the PASB into the simulation model. We conducted numerical experiments using both staffing methodologies. The PASB was highlighted in the experiments that were conducted in this chapter because it is more conservative than the Erlang loss formula when the traffic intensity is high, and because it does not suffer from the numerical instability problems that cause the Erlang loss formula to overflow when the traffic intensity is high.

The multi-class simulation model is very flexible, and allows a road race manager to simulate different kinds of races. For example, a road race manager can use the simulation model to get the staffing requirements for a road race that is made up entirely of elite runners, or for a road race that is comprised of only recreational runners. For a more practical example, such as the New York Marathon or the Boston Marathon which are comprised of runners of various skill levels ranging from elite runners to recreational runners, a road race manager can simulate such types of road races, and get the staffing requirements at each service station.

CHAPTER 6
CONCLUSIONS AND FUTURE RESEARCH

6.1 Contribution Summary

In this chapter, we summarize the outcomes of the work that was undertaken in this dissertation research project.

In this research, we modeled a long distance road race traffic system as a mixed $M_\ell(t)/G_\ell/s_\ell/s_\ell/N/FIFO$ queueing network. We developed models that allow road race managers (or managers of similar types of events) to provide “service” to runners (or other types of similar entities) arriving at service stations at predetermined distances. Deliverables from this research include:

1. A single-class model to address the staffing needs for long distance road race (and related) traffic systems with time-dependent traffic patterns.
2. A multi-class simulation model to address the staffing needs for long distance road race traffic systems (or other related traffic systems) with time-dependent traffic patterns.
3. The development of a mathematical framework to model how runners in long distance road races (or other similar type entities in similar types of traffic systems) transition across distance and time.
4. The development of a mathematical framework to estimate the arrival rates of runners (for example, runners in a long distance road race) at service stations of interest.
5. A staffing equation to provide staffing requirements at service stations in the queueing network.

The analytical framework for the single-class model was developed in Chapter 3. For the single-class model, we developed an analytical framework that allows for the estimation of time-dependent arrivals at a service station of interest. Then, we developed a staffing formula to prescribe the number of servers when there are time-dependent arrivals at service stations. Empirical analyses showed that the PASB yields staffing requirements that are comparable to the Erlang loss formula. However, unlike the Erlang loss formula, the PASB does not overflow.

An advantage of the single-class model is its ability to duplicate the estimated running times of the runners from an available dataset. This feature of the model allows a road race manager to create a larger dataset by making copies (or duplicates) of the (estimated) running times of the runners from the original data set. The primary benefit of this feature is that it allows a manager to easily assess the impact on staffing for a larger (or smaller) race with runners that are identical in mix to those in the original dataset. The disadvantage of this model is that it fails to capture the staffing effects when the new race is comprised of runners whose paces are less homogeneous (or different) from those runners whose records are in the original dataset.

The multi-class simulation model was developed in Chapter 5. This model allows a road race manager to simulate a race of any size, N , and to classify those N runners, based on their (historical or assumed) paces, into different classes of runners. For the purpose of staffing, we integrated the PASB and the Erlang loss formula into the simulation model at each service station. The multi-class simulation model provides road race managers with a very flexible management tool that allows them to simulate different size races with as their choice of runner-mix.

This model improves upon the single-class model by allowing the road race manager to design as diverse a race (in terms of the mix of the runners and the number of runners) as the manager chooses. The main drawback of this model is that it is relatively slow to run in Microsoft Excel as the number of runners in the race gets larger. In our applications, using a Dell Latitude D610 laptop with a 2.00 GHz Pentium M processor (with 2.00 GB of RAM), it took approximately 4 minutes to run a race with $N = 20000$ runners, $C = 10$ classes and $L = 25$ service stations, using this model. For the single-class model, it took about half that time to run a race with the same number of runners and the same number of service stations. However, this is a small price to pay for the added versatility of this model over the single-class model.

By conceiving and constructing a long distance road race as a single or multi class $M_\ell(t)/G_\ell/s_\ell/s_\ell/N/FIFO$ queueing network in which refreshment stations are to be staffed with the appropriate numbers of servers for the runners that approach each refreshment station, we have shown how tools and concepts from operations research can be used to design, develop, and find solutions for a real-world problem. Before we close this section, we mention some other application areas that may benefit from the use of the models that we have developed in this research. The model can also be used by practitioners in other areas in which the traffic flows and runner types are similar to the kinds that are modeled in this research. Two other possible areas where the models can be applied are:

1. Providing service staff to a convoy of vehicles that are traveling on a defined route and arriving at predetermined locations for servicing, and
2. Providing staffing at aid stations along a street parade route, or along a street procession route.

Specifically, these models may be useful for supplying needs to military units and personnel during a long distance deployment to an area where there is a conflict.

6.2 Recommendations and Directions for Future Research

As far as the literature suggests, we are the first researchers to propose queueing network models that address server staffing or volunteer staffing in long distance road race traffic system. We believe that the models that we have developed are just a first step to the development of more sophisticated models.

A possible extension to the models that we have developed in this research is to extend the models to allow for moving volunteers from one service station to another service station during the race at minimum cost.

Another obvious improvement on what we have developed for the multi-class simulation model is to incorporate alternate distributions of running times into the model.

Another area in which the models that we have developed can be improved is to take attrition of runners into account. This improvement can be done by developing attrition models and integrating them in the model at each service station. If runner attrition is a matter that deserves attention, then the incorporation of attrition into the existing model may lead to fewer required servers than would be otherwise needed.

It would be interesting to apply the models suggested here and the historical data from the 2004 WDW Marathon to prescribe service station staffing levels at a different race where the number of runners is different than in the historical data set.

REFERENCES

[1] Abhulimen, K. (2007), "Predicting Reservoir Performance Using Stochastic Monte Carlo Simulation," *e-journal of reservoir engineering,* 1, 1

[2] Allen, P.M. (1973), *The use of generalized birth-illness-death stochastic formalism as a model of tumor growth.* Doctoral Dissertation, University of Virginia, Charlottesville, VA.

[3] Alnowibet, K. and H. G. Perros, "Nonstationary Loss Queues and Loss Queueing Networks", (2TUhttp://www.csc.ncsu.edu/faculty/perros//recentpapers.htmlU2T).

[4] Bar-Lev, S. K., Boxma, O., Stadje, W., and F. A. Van der Duyn Schouten (2009), "Two-Stage Queueing Network Models for Quality Control and Testing," *European Journal of Operational Research*, 198, 859 – 866.

[5] Berney, C. and G. Danuser (2003), "FRET or No FRET: A Quantitative Comparison", *Biophysical Journal*, 84, 3992 – 4010.

[6] 5T Bolch, G., Greiner, S., de Meer, H., and K. Trivedi (2006), *Queueing Networks and Markov Chains,* Wiley & Sons, Hoboken, NJ, USA.

[7] Brockmeyer, E., H. L. Halstrom, and A. Jensen (1948), *The Life and Work of A. K. Erlang.* Copenhagen: The Copenhagen Telephone Company.

[8] Chao, L. (1995), "A Queueing Network Model with Catastrophes and Product Form Solution," *Operations Research Letters*, 18, 75 – 79.

[9] Dagpunar, J.S. (2007), *Simulation and Monte Carlo with Applications in Finance and MCMC*, Wiley, West Sussex, England.

[10] George, K. G. and C. H. Xia (2011), "Fleet-Sizing and Service Availability for a Vehicle Rental System via Closed Queueing Networks," *European Journal of Operational Research,* 211, 198 – 207.

[11] Grassman, W. K. (1988), "Finding the Right Number of Servers in Real-World Queueing Systems," *Interfaces*, 18, 2, 94 – 104.

[12] Green, L.V., Kolesar, P. J. and A. Svoronos (1991), "Some Effects of Nonstationarity on Multiserver Markovian Queues Systems", *Operations Research,* 39, 502 – 511.

[13] Green, L.V. and Kolesar, P. J. (1991), "The Pointwise Stationary Approximation for Queues with Nonstationary Arrivals", *Management Sciences,* 37, 84 – 97.

[14] Green, L.V. and Kolesar, P. J. (1997), "The Lagged PSA for Estimating Peak Congestion in Multiserver Markovian Queues with Periodic Arrival Rates", *Management Sciences,* 43, 80 – 87.

[15] Green, L.V. and Kolesar, P. J. (1995),"On the Accuracy of the Simple Peak Hour Approximation for Markovian Queues", *Management Sciences,* 41, 1353 – 1370.

[16] Gross, D. and C. M. Harris (1998), *Fundamentals of Queueing Theory,* 3rd ed. John Wiley & Sons, New York.

[17] Hall, R.W. (1991), *Queueing Methods for Services and Manufacturing,* Prentice Hall, Englewood Cliffs, New Jersey.

[18] Haskose, A., Kingsman, B. G., and D. Worthington (2002), "Modelling Flow and Jobbing Shops as a Queueing Network for Workload Control," *International Journal of Production Economics*, 78, 271 – 285.

[19] Huessmann, L. R., and R. P. Goldberg (1967), "Evaluating Computer Systems Through Simulation," *The Computer Journal*, 10, 150 - 156.

[20] Huisman, T., Boucherie, R. J., and N, M. van Dijk (2002), “A Solvable Queueing Network Model for Railway Networks and its Validation and Applications for the Netherlands,” *European Journal of Operational Research*, 142, 30 – 51.

[21] Jain, M., Maheshwari, S., and K. P. S. Baghel (2008), “Queueing Network Modelling of Flexible Manufacturing System Using Mean Value Analysis,” *Applied Mathematical Modelling*, 32, 700 – 711.

[22] Juang, J. and H. Huang (April 2000), “Queueing Network Analysis for an IC Foundry,” *Proceedings of the 2000 International Conference on Robotics and Automation*, San Francisco, CA.

[23] Kelly, F. P. (1979), *Reversibility and Stochastic Networks,* Wiley, New York.

[24] Lafortune, E. (1995). *Mathematical Models and Monte Carlo Algorithms for Physically Based Rendering.* PhD thesis, Katholieke Universiteit Leuven.

[25] LeBlanc, B., B. Braunschweig; H. Toulhoath, and E. Lutton (2003), “Improving the Sampling Efficiency of Monte Carlo Molecular Simulations: An Evolutionary Approach,” *Molecular Physics,* 101, 22, 3293 - 3308.

[26] Massey, W. A. and Whitt W. (1996), “Stationary-Process Approximation for the Nonstationary Erlang loss formula”, *Operations Research,* 44, 976 – 983.

[27] Massey, W. A. and Whitt W. (1993), “Networks of Infinite-Server Queues with Nonstationary Poisson Input”, *Queueing Systems*, 13, 183 – 251.

[28] Massey, W. A. and Whitt W. (1994), “An Analysis of the Modified Offered-load Approximation for the Nonstationary Loss Model”, *Annals of Applied Probability,* 4, 1145 – 1160.

[29] Muduli, P. K. and T. M. Yegulalp (1996), “Modeling Truck-Shovel Systems as Closed Queueing Network with Multiple Job Classes,” *International Transactions of Operational Research,* 3, 1, 89 – 98.

[30] Park, C and H. Lee (in press), “A Multi-Class Closed Queueing Maintenance Network with a Parts Inventory System,” *Computers and Operations Research*. Retrieved from http://www.sciencedirect.com

[31] Qiao, S., and L. Qiao (1998), “A Robust and Efficient Algorithm for Evaluating Erlang B Formula. *Technical Report CAS98-03*, Department of Computing and Software, McMaster University, Ontario, Canada L86 4L7, August 1998.

[32] Rae, D., A. N. Bosch, M. Collins, and M. I. Lambert (2005), “The Interaction of Aging and 10 Years of Racing on Ultraendurance Running Performance,” *Journal of Aging and Physical Activity,* 13, 210 – 222.

[33] Reiser M. (1979), “A Queueing Network Analysis of Computer Communication Networks with Window Flow Control,” *IEEE Trans. Comm.,* 27, 8, 1199 – 1209.

[34] Saaty, T. L. (1961), *Elements of Queueing Theory with Applications,* New York: McGraw Hill.

[35] Santos, T. and G. Soares (2005), “Monte Carlo Simulation of Damaged Ship Survivability,” *Proceedings of Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment,* 219, 1, p.25.

[36] Stidham, Jr., S. (2002) "Analysis, Design, and Control of Queueing Systems," *Operations Research,* 50, 1, 197 - 216.

[37] Strand, M., and D. Boes (1998), "Modeling Road Racing Time of Recreational Runners Using Extreme Value Theory," *The American Statistician,* 52, 3, 205 – 210.

[38] Whitt, W., "Decomposition Approximation for Time-Dependent Markovian Queueing Networks", *Operations Research Letters*, 24(1999), 97 – 103.

[39] Whitt, W. (1999), "Dynamic Staffing in a Telephone Call Center Aiming to Immediately Answer All Calls," *Operations Research Letters*, 24, 205 – 212.

[40] Whitt, W. (1980), "some Useful Functional Limit Theorems," *Mathematics of Operations Research,* 5, 1, 67 - 85.

[41] Whitt, W., O. B. Jennings, A. Mandelbaum, and W. A. Massey (1996), "Server Staffing to Meet Time-Varying Demand," *Management Science*, 42, 1383 – 1394.