Dynamic Queueing Systems: Behavior and Approximations for Individual Queues and for Networks

by

Kerry Marie Malone

Submitted to the Sloan School of Management
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Operations Research

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1995

Author . . . .
Sloan School of Management
May 22, 1995

Certified by . . . .
Amedeo R. Odoni
Professor, Departments of Aeronautics and Astronautics and
of Civil and Environmental Engineering
Thesis Supervisor

Accepted by . . . .
Thomas L. Magnanti
Co-Director, Operations Research Center, and
George Eastman Professor of Management Science,
Sloan School of Management

Dynamic Queueing Systems: Behavior and Approximations for Individual Queues and for Networks

by

Kerry Marie Malone

Submitted to the Sloan School of Management
on May 22, 1995, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Operations Research

Abstract

In this thesis, we develop and analyze methods for approximate analysis of networks of single-server dynamic queues. Specifically, we investigate three facets of these problems.

First, we develop and test new approximate methods for analyzing individual queues with time-varying arrival and/or service rates. We approximate queues with time-varying Poisson arrival processes and Erlang, hyperexponential and general non-phase-type service-time distributions. These new methods are simple to implement, efficient and have limited computer memory requirements. In cases in which exact results can be found, the exact methods require 3 – 400 times as much CPU time as the approximations. Our computational tests cover cases of moderate to heavy utilization. We show that under many system conditions, these methods give estimates within 5% of exact system measures.

Second, we propose and test an approximate decomposition method for analyzing open networks of single-server dynamic queues. This method allows analysis of systems for which exact methods of analysis do not exist, or are infeasible to implement due to time or computer memory constraints. It uses an individual queue approximation method to estimate delays at the queues, and a propagation algorithm to model the interactions among the queues. The decomposition approach is computationally and computer-memory efficient. Our results indicate that the exact solution requires two to three orders of magnitude more CPU time than the decomposition method. We determine the sensitivity of the accuracy of the decomposition method to the parameters of the system, and identify under what conditions the methods give estimates within 5% of exact system measures.

Third, we investigate aspects of the behavior of queueing systems with time-varying arrival and service rates. We examine the time lag between a peak in system utilization and corresponding peaks in the mean and variance for the number of customers in an $M(t)/M(t)/1$ system. We establish a necessary condition for the times at which local extremes in the mean will be achieved. In cases in which system utilization exceeds one during some period, we show that the local peak in the mean induced by this period of oversaturation occurs strictly after the end of the oversaturation period. The observations we make for these systems provide some rules of thumb that should help planners and operators of facilities with strongly time-dependent demand and capacity to make better facility management decisions.

Our focus on networks of single-server dynamic queues was motivated by a particular system of great practical importance: the national network of airports. In recent years, congestion in the airport networks has been experienced with increasing frequency in both the United States and Western Europe. The use by the airlines of "hub-and-spoke" network configurations that create a tight "coupling" among flights at geographically dispersed airports causes delays at one major airport to propagate rapidly throughout the Air Traffic Management (ATM) system. The Office of System Capacity and Requirements of the Federal Aviation Administration (FAA) estimated the total cost of delay in US airports to be at least $736 million in 1992. The airlines estimate that cost as being in excess of $1.5 billion. The research presented in this thesis provides, among other results, evidence that the assumptions of a decomposition approach for modeling a network of queues may be appropriate in the context of a network of airports.

Thesis Supervisor: Amedeo R. Odoni
Title: Professor, Departments of Aeronautics and Astronautics and
of Civil and Environmental Engineering

Acknowledgments

Almost six years ago, I became a student at the Operations Research Center at MIT. I have had the opportunity to live in a city I love, and to meet people who have been wonderful colleagues and friends. Now that I have achieved my goal of earning a PhD and must now leave MIT, I would like to thank the people who made this experience possible and so enjoyable.

I have had the pleasure and privilege of working with Amedeo Odoni for five years. His ability to identify important and interesting problems in air transportation led to the research presented in this thesis. During the years we worked together, he has functioned as research supervisor, mentor, and role model. Amedeo has high expectations of his students, and I thank him for challenging me to carry through with the research, and for spending time with me thinking through difficulties encountered while carrying it out. I look forward to working with Amedeo in the future.

I would like to thank Dimitris Bertsimas, Linda Green and Viên Nguyen for agreeing to be members of my thesis committee. Each of them made valuable contributions to the direction and content of this research.

The results contained in this thesis required that much computational work be done. I would like to thank Richard Larson and Thomas Magnanti for providing the financial support for the computer facilities of the Operations Research Center (ORC). Several students maintained the computer facilities in top working order. They helped me with installing and maintaining software, and provided help at all times of the day and night. Without their help, this computational work would have been much more difficult to carry out. To David Markowitz and Robert Shumsky, I say "THANKS!"

I wish to acknowledge partial financial support over the years from the FAA, grant number US DOT/FAA #94-6-030, Volpe National Transportation Systems Center (TSC), the MITRE Corporation, and the American Association of University Women (AAUW) Selected Professions Engineering Fellowship. In particular, the following individuals deserve specific mention: Steve Bradford, Richard John, Larry Kiernan, Robert Rovinsky, and David Winer of the FAA; Zale Anis and David Lev of the TSC; and Andrew Haines, Richard Tibbs, Fred Wieland, and Anthony Zukas of the MITRE Corporation.

I would like to thank the people who make the ORC the wonderful place it is. The Co-

directors of the ORC and faculty associated with care deeply about fostering a cooperative atmosphere at the ORC. The ORC could not function as well as it does, though, without Paulette Mosley, Cheryl Mottley and Laura Rose. I especially thank Laura for her help over the past six years.

There are many current and former students who have also made the ORC an enjoyable place. In particular, Eddie Pinker has been a great friend, confidant, roommate and, of course, a source of editorializing. Pieter Klaassen and I shared the Holland-Amerika Lijn for a few years. During that time, we enjoyed many dinners, movies, and bike rides together. Mitchell Burman and I have known each other for six years, during which Mitchell provided enough sagas for a lifetime. He also provided friendship and delicious dinners. Robert Shumsky has been a wonderful running partner and a sympathetic listener. I also enjoyed many runs, soccer games, movies, parties, and dinners with my former officemate, Raghu. I enjoyed working with Ármann Ingólfsson. The results we obtained appear as Chapter 5 of this thesis.

I thank my family for supporting me in my pursuit of my doctorate. In particular, in the last days of writing the thesis, Mom provided support, dinners, proofreading help, and a clean house to come home to.

Finally, I thank Albert for his support during the years we have been on opposite sides of the Atlantic Ocean. *Nou, dat hebben we ook gehad, hè?*

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Objective

Congestion and its propagation among facilities is a common phenomenon which involves complex interactions within a single queueing facility system as well as in a network of queueing facilities. In many real-world systems, such as airports, air terminals, some manufacturing processes, roads and highways, and telecommunication networks, there are large costs associated with congestion and its propagation. The Office of System Capacity and Requirements of the Federal Aviation Administration (FAA) estimated the total cost of delay in US airports to be at least $736 million in 1992 [5]. The airlines estimate that cost as being in excess of $1.5 billion. There are good economic reasons for studying ways to reduce this delay.

To analyze congestion at real-world facilities, we use the tools of queueing theory. Most analytical results in queueing theory and especially for networks of queues are valid under restricted conditions, one of which is constant arrival and service rates. Many important real-world queueing systems, such as those mentioned above, do not have this critical characteristic. They have demand and capacity which vary strongly as a function of time. We call such time-varying systems "dynamic" or "nonstationary." Some analytical results do exist for individual queues with arrival and/or service rates that vary with time. However, these results are for special queueing systems, or require at least some numerical computation or transform inversion. More general methods of analysis are needed.

Simulation can be used to analyze queues with demand and capacity which are stochastic and vary with time, but many simulation "runs" are needed in order to obtain statistically valid results. Koopman [26] noted that the time required to obtain a sufficient number of data points makes simulation an unattractive and often infeasible approach for the analysis of complex dynamic queues.

The fluid approximation model also provides a first-order analysis of queueing systems with parameters which vary with time [24]. This model predicts a queue only if the arrival rate exceeds the service rate. However, if there is randomness in the arrival or service process, the fluid model severely underestimates queueing congestion. Diffusion approximations try to compensate for this deficiency and provide second-order approximations [24, 33, 34]. They take into account the variance of the arrival and service processes. However, even these approximations are only good if the arrival rate is approximately equal to the service rate over the period of analysis, limiting their applicability.

When a quick back-of-the-envelope calculation is needed, steady-state queueing results are sometimes used to analyze systems with time-varying arrival or service rates. However, Green et al. [16] have shown how poorly stationary results may approximate these types of systems even when the arrival rate exhibits limited variation from average demand as a function of time. Odoni and Roth [35] have shown that the amount of time needed to reach steady-state in peak periods of system utilization is not negligible; in fact, in some systems, it may take several hours to reach steady state, whereas the peak period being analyzed is usually short and depends on the particular system. Hence, applying steady-state results to discrete periods of the day using the average arrival and service rates over the period may yield rather inaccurate results. Furthermore, steady-state results for time-varying systems mask the time lag which occurs between the peak in system utilization and the peak in the congestion level (measured in terms of expected number of customers in the system), as well as the peak in the standard deviation for the number of customers in the system. Applying steady-state results at peak periods may give a a very erroneous indication of when the system is most congested. This time lag can range from minutes to hours depending on the arrival-rate process, service-time distribution, the average system utilization, and how much the instantaneous system utilization peak exceeds the average utilization. Finally, the most significant drawback of using steady-state results to analyze systems with time-varying parameters is that it cannot be applied to systems in which demand exceeds capacity for certain periods of time, as often happens in practice.

Networks of queues with time-varying demands and capacities present even more difficulties. Relatively little is known about networks of queues with time-varying parameters, yet these systems are common and most important. With the exception of networks with infinite-server queues, no exact general solutions exist [30]. Simulation is not a feasible approach for analyzing large and complex networks, due to the number of runs necessary to obtain statistically valid results. If one takes an "exact" computational approach, one can model an open network of queues with an external Poisson arrival process which varies with time, and exponential service. In this case, one can solve the differential equations describing the system using an ordinary differential equation (ODE) solver. We show in Chapter 4, however, that even for simple two-station networks, one encounters serious memory and CPU time constraints when solving such systems. This approach proves infeasible for most real-world networks problems.

In summary, existing methodologies do not satisfactorily address real-world problems with time-varying demands and capacities. In this research, we strive to develop approximate methods for these problems. It is, of course, impossible to address all possible real-world manifestations of queues and networks of queues with time-varying arrival and service rates in a general manner. There is an enormous number of such problems with accompanying attributes specific to each. In addition, these are extremely difficult problems to address. Motivated by practical applications, we study in this thesis a subset of these problems: networks of single-server nonstationary queues.

Our focus on networks of single-server nonstationary queues is motivated by a particular system of great practical importance: the national network of airports. Until the mid-1980's, most congestion problems in the air traffic management (ATM) system were of a local nature, i.e., they were concentrated primarily at a few local airports. More recently, however, both the United States and Western Europe are experiencing system-wide congestion with increasing frequency. This more widespread congestion has resulted from an overall traffic growth that has created many potential "bottlenecks" in the world's two most intensively utilized regional airport systems. Also contributing to the phenomenon is the growing use by the airlines of "hub-and-spoke" network configurations that create a tight "coupling" among flights at geographically dispersed airports. This coupling causes delays at one major airport to propagate rapidly throughout the ATM system. There is then an acute need for models that can assist in analyzing network-wide airport congestion phenomena and understanding the impact of various parameters on the propagation of delays.

An additional motivation for developing network models comes from the growing need to understand the system-wide implications of major local changes, such as the investment of capital funds to achieve a significant expansion of capacity at a particular airport. For example, part of the justification for the contribution of approximately $500 million of federal funds to the construction of the new Denver International Airport was based on the claim that the new airport would have a substantial effect on reducing air traffic delays not only at Denver, but, "through a ripple effect," throughout the United States, as well. Furthermore, in operating the Airport Improvement Program (AIP), the FAA distributes approximately $2 billion annually in grant monies to US airports for facility improvement and expansion. As a federal organization, the FAA is interested not only in the local effect of local improvements, but also the system-wide effect of local improvements.

Understanding the system-wide costs and benefits of facility improvement requires an understanding of the network behavior. Equivalent dollar investments in different facilities may generate significantly different national benefits in terms of delay reduction. For example, following the deregulation of the air transportation industry in 1978, Chicago's O'Hare Airport became a hub for both American and United Airlines and the busiest airport in the world, with more than a thousand flights operating there each day. When Chicago experiences congestion, the delays ripple throughout the national network of airports. This ripple or "network effect" causes delays at airports which may not otherwise experience delay. Therefore, decreasing the chance of excessive delays at Chicago can possibly reduce delay at other airports in the US airport system. Hence, an investment at O'Hare may reap larger benefits than an equivalent investment elsewhere.

The study of delay in networks of airports gives us insights into system performance and into the economic questions of facility improvement and expansion. Moreover, because delays at airports are caused by a large concentration of aircraft in a small area, studying ways to reduce delays may have safety benefits as well.

With the paradigm of the network of airports in mind, we investigate three facets of networks of single-server dynamic queues.

1. We develop and test new approximate methods for analyzing individual queues with time-varying arrival and/or service rates. These computational methods are efficient and have limited computer memory requirements. In addition, they are simple to implement and model a wide variety of queues with different service-time distributions.

2. We investigate the behavior of queueing systems with time-varying arrival and service rates. The observations we make for these systems provide some rules-of-thumb that should help planners and operators of facilities with strongly time-dependent demand and capacity to make better facility management decisions.

3. We propose and test an approximate decomposition method for analyzing open networks of dynamic queues. These methods allow analysis of systems for which exact methods of analysis do not exist, or are infeasible. The decomposition approach is computationally and computer-memory efficient. It uses the individual queue approximation methods to estimate delays at the queues, and a propagation algorithm to model the interactions among the queues.

We will use the paradigm of the national network of airports in much of our discussion and numerical tests.

We have applied the results of our research to networks of airports. During the last four years, we developed a policy-oriented tool which models the national network of airports. This tool is called the Approximate Network Delays (AND) Model [29]. The AND model is an approximate model for a weakly-connected network of queueing systems (nodes) with nonstationary parameters. "Weakly connected" means, informally, that no single node receives a large percentage of its customers from any other single node. The arrival stream at any one node is a combination of streams from sources external to the netwɔrk and from several other nodes in the network. The AND model is macroscopic and is best-suited for use in strategic (or "policy analysis") studies in which the primary objective is to assess the relative performance of a wide range of alternatives. That is, we want to capture the relative changes in delay in response to changes in demand and capacity at airports. For this reason it aspires to be very fast, in terms of both input preparation and execution times, so it can be used to explore a large number of "scenarios." The model can also be used as a screening device to identify the few most promising among many alternative courses of action, which can then be studied in detail through more "microscopic" simulation models. The research presented in this thesis provides, among other results, evidence that the assumptions and algorithm of the AND Model are appropriate. Section 4.4 briefly describes the AND Model.

We next describe the relevant characteristics of the Airport System. We distill these characteristics into important attributes which form the basis of how we model the network of airports in a queueing context.

1.2 Characteristics of Individual Airports and the National Network

In this section we first identify the characteristics of individual airports and of networks of airports which must be captured by our queueing models. We then show how we incorporate these characteristics into our approximate, macroscopic queueing models.

To avoid confusing terminology, we use the word "arrivals" to an airport to indicate the demand for "landings" and "takeoffs." "Arrivals per hour" is equivalent to the hourly demand rate for landings and takeoffs. We define an "operation" to be either a landing or a takeoff.

We identify five attributes of individual airports which we consider significant for a macroscopic airport model. First, the number of operations per hour at airports varies. Figure 1-1 shows the number of scheduled operations per hour at Boston Logan International Airport. This demand profile represents a typical weekday at Logan in 1992. The hourly demand rate varies from 0 during some of the night hours to more than 100 during peak hour periods. This profile is unique to Logan. However, practically all other major US airports have demand profiles which also show large variations in the number of operations per hour over the 24-hour day.

The second attribute is that the demand profiles are not deterministic. Several factors "randomize" the scheduled demand for landings and takeoffs at airports. First, aircraft do not operate exactly according to the published schedule. Air traffic control, weather and wind conditions, routes flown, and changes of aircraft contribute to occasionally large deviations from the scheduled landing and takeoff times. Second, airlines sometimes publish landing times which may not reflect the actual flight time between the airports of origin and destination. Instead, flight times are increased to include an allowance for expected flight delays. The airlines may do this to attract air travelers to their flights by claiming better "on-time" performance, or to protect themselves against unforeseen delays. A third factor which randomizes the number of operations per hour is that not all operations are scheduled. In fact, General Aviation (GA) flights, charter flights, and extra sections of shuttle flights are not scheduled and account for about 10% of the approximately 1200

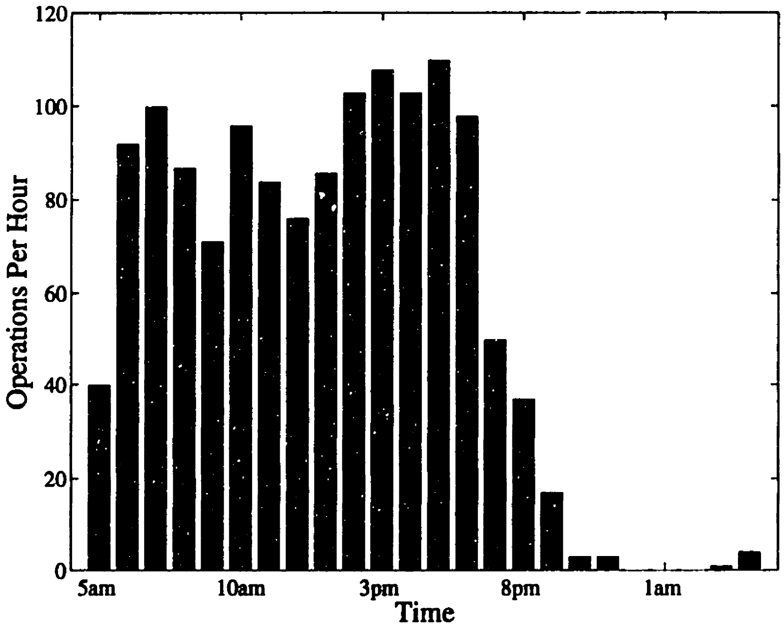


Figure 1-1: Hourly Demand Rate at Boston Logan International Airport

weekday operations at Logan Airport. Logan is actually an example of an airport with very few unscheduled flights. Typically, unscheduled flights represent a much higher percentage of the number of daily operations. Finally, airlines cancel flights. Unscheduled and canceled flights cause the actual number of operations to vary from day to day, and hour to hour, from the number which are scheduled.

Periodicity of demand is the third relevant attribute of airports from the queueing point of view. From a practical point of view, the demand profile at airports is nearly periodic, with a period of 24 hours.

We capture these first three attributes of airport demand by modeling the arrival process to the queues as a nonstationary Poisson process, as in Koopman [26] and Horangic [19]. We define the Poisson arrival rate to be the rate over time of demand for access to the runway by landings and takeoffs. This rate is not necessarily the same as the rate at which aircraft actually land or take off. The two rates will be approximately the same only in the absence of congestion.

Airport capacity also varies with time, in an even less predictable manner than the demand rate. Airport capacity depends on the runway configuration in use. A runway configuration of an airport consists of a set of runways which are simultaneously active. Figure 1-2 shows the runway layout at Boston's Logan Airport. Logan has about 40 runway

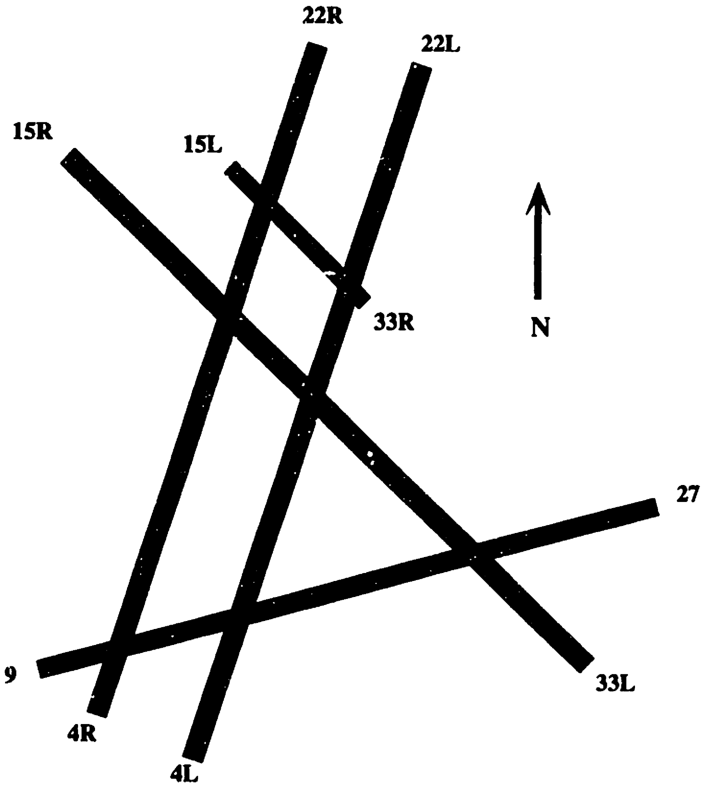


Figure 1-2: Boston Logan International Airport Runway System

configurations in total. For example, one common such configuration is the simultaneous use of runways 4R and 4L for landings and takeoffs and of runway 9 for takeoffs only. Changes in weather and wind direction are the primary causes of runway configuration changes, and therefore changes in capacity. Some weather conditions cause airport capacity to drop so low that the demand rate exceeds the airport capacity for extended periods of time. Thus, the fourth attribute for which airport models must account is that demand can exceed capacity for certain periods of time that may last for several hours.

In our queueing model, we represent the airport as a "black box" which processes landings and takeoffs. Our black box approach ignores the interdependencies among and independence of runway combinations. What impact does this have on our model? First, because we take a macroscopic view of the airport as a server, we do not attempt to model the details of the runway interdependencies. Second, the distinction between single and multiple servers at an airport only makes a difference when there are fewer aircraft in the system than the number of servers. Since we consider busy airports, the congested periods of operation are far more important than the periods of low utilization. During congested periods, there is a relatively large expected number of aircraft in the system. Therefore, this single vs multiple-server distinction is not critical from a macroscopic point of view.

We model the queue discipline at airports as First-Come First-Serve (FCFS). In practice, this is a reasonable assumption. In cases of low to moderate congestion, air traffic controllers allow aircraft to land and take off in FCFS order. In the case of severe congestion, landings have partial priority over takeoffs. However, air traffic controllers fit takeoffs into the sequence of landing aircraft in such a way that although the queue discipline in this circumstance is not strictly FCFS, it is a reasonable approximation. In addition, when departure queues get long, the priorities are sometimes reversed and a string of departures will be allowed to operate ahead of landings.

Finally, the fifth attribute of airports concerns the service times of aircraft. The service times at an airport depends on the types of aircraft that operate there. At busy airports, many different sizes of aircraft operate. This results in a large variability in the service times there. Figure 1-3 shows a sample distribution of 61 interoperation times recorded during a busy period at Logan. In this case, the mean service time was 35.7 seconds, and the standard deviation was 23.5. This means that the coefficient of variation of the service times, or the standard deviation divided by the mean, was 0.66. In general, at busy US airports like Logan, the coefficient of variation is as a rule less than one. In contrast, major European airports, which restrict the classes of aircraft with access to the airport, typically exhibit less variability in service times. It is reasonable to expect much smaller service time coefficients of variation at these airports.

To model the service times observed at airports, we use the Erlang distribution. The Erlang has two parameters which can be adjusted for the variability of the service times. It is also an analytically tractable distribution in a queueing context. Finally, its coefficient of variation can be adjusted to values less than or equal to one. A more detailed description of the Erlang distribution appears in Section 2.2.3.

We now turn to the network of airports: from a modeling perspective, the major attribute of the national network of airports is that it is weakly connected. That is, no single airport receives a dominating percentage of its arrivals from any other single airport. If we examine total arrivals to a particular airport, departures from *any other single* airport in the network typically represent less than 10% of the total arrivals to the airport under consideration. A stronger link exists between landings and takeoffs at a single airport. Aircraft which land at an airport subsequently take off at some time later in the day. Departures

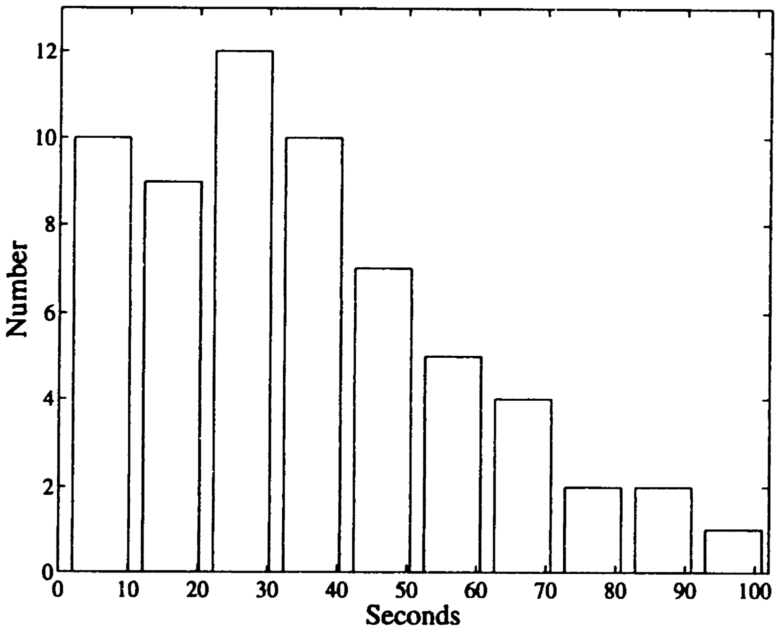


Figure 1-3: Histogram of Interoperation Times at Boston Logan International Airport. October 26, 1994. 5:12 - 5:30 pm.

which are directly linked to a previous arrival during the same day typically represent 30 – 40% of the total operations at an airport. Despite the fact that there is this coupling, even in this case one can argue that variable gate times, airline scheduling practices, and late arrivals and departures lead to a very substantial decoupling of this link.

We model the network of airports using a decomposition method. We intend this decomposition to provide a tractable method for approximate analysis of a network of airports. We intend it for use as a strategic, as opposed to a tactical, model. The goal of this strategic model is to give an indication of the relative changes in delay which occur as a result of changes in demand and/or capacity. We do not account for the tactical decisions of airlines to change itineraries or swap aircraft in response to congestion conditions at airports.

Our approach decomposes the network of airports into individual queues at each single airport. We use the $M(t)/E_k(t)/1$ queueing system to model each individual airport approximately. We use a "propagation method" to link the airports together and to propagate the delays at congested airports to other airports in the network. Since we use the Poisson arrival assumption for each individual airport throughout our analyses, we believe the weak connectivity among the airports in the network is a key assumption in our decomposition method. We experiment with this assumption in Chapter 4. We show that the results of the decomposition method are valid for the levels of connectivity typically encountered in an airport network.

In this section, we have described the practical context we intend to use in this research. In the next section, we review what options exist currently for analysis of the types of queues and networks in which we are interested.

1.3 Review of Previous Research

We now briefly review existing methods for analyzing nonstationary queues which are relevant in light of our stated research goals: fast accurate approximations of the time-dependent probability distribution, mean, and variance for the number of customers in $M(t)/E_k(t)/1$ systems and networks of such systems. Methods can be categorized along two dimensions: exact or approximate methods for transient or periodic systems. We assume that the systems we study are stable. Bambos and Walrand [3], Harrison and Lemoine [17], Heyman and Whitt [18], and Rolski [42, 43] discuss notions of and conditions for stability in queueing systems with nonstationary inputs.

To date, exact numerical analysis of nonstationary models focus on $M(t)/M(t)/s/c$ systems and generalizations in which the number of customers (or phases) in system can be represented by a continuous-time Markov Chain, $M(t)/G/\infty$ systems in which the time-dependent number of busy servers (or customers in the system) is distributed as a time-dependent Poisson random variable, or $M(t)/G(t)/1$ systems in which either a transient analysis using transforms or an asymptotic workload and waiting time analysis is used. Readers interested in infinite-server systems are referred to the papers of Eick, Massey and Whitt [11, 12] and Massey and Whitt [30] and the references therein. See Jennings et al. [20] and Massey and Whitt [31] for infinite-server approximations to finite-server systems. We now discuss exact results for finite-server systems.

Clarke [8] derived an exact expression for the time-dependent probability of i customers in an $M(t)/M(t)/1$ system, for all i. These expressions can be evaluated, given that a solution to several complicated expressions, including an integral equation of the Volterra type, can be found. Due to its complexity, this method is not used.

In the case of $M(t)/M(t)/s/c$ systems and generalizations, the time-dependent distribution of the number of customers in the system can be found by numerically solving a system of time-dependent ordinary differential equations (ODE's), as in Koopman [26]. One can perform transient and periodic analyses of such systems. Finite or infinite capacity systems can be modeled. Infinite capacity systems must be approximated by a finite queueing capacity c, chosen large enough that the probability of having c or more customers in the system is smaller than some prespecified level $\epsilon > 0$. Allowing the arrival rate to exceed the service rate exacerbates the issue of choosing c, because the number in the system grows rapidly during this period. When more complicated phase-type distributions are used, the number of states needed to represent a system with queueing capacity c increases. For example, the number of states needed to represent the $M(t)/E_k(t)/1/c$ system is $kc + 1$, versus $c + 1$ in the $M(t)/M(t)/1/c$ case. A large state space greatly increases computation time. This serious drawback prompted development of approximation methods for $M(t)/M(t)/s/c$ systems and its natural generalizations called closure or surrogate distribution approximation (SDA) methods, in which the number of ODE's solved at each iteration is independent of system capacity or utilization.

Choudhury, Lucantoni, and Whitt [6] develop an exact numerical algorithm for calculating the distribution of the workload (virtual waiting time) at an arbitrary time in an $M(t)/G(t)/1$ queue. This is a generalization of the work of Van den Berg and Groenendijk [51] in which they develop a recursive scheme for finding the number of customers as a function of time in an $M/M/1$ system with regularly changing arrival and service intensities. Specifically, Choudhury et al.'s model calculates the transform of the workload distribution at a given time t_i. The arrival rate and service-time distribution change only at finitely many points, i.e., they are piecewise continuous. This model permits the arrival rate to exceed the service rate. The piecewise-stationary time-dependence reduces the $M(t)/G(t)/1$ problem to solving recursively a nested family of problems involving the transient behavior of stationary $M/G/1$ models. The method applies known transform results for the transient workload distribution in a stationary $M/G/1$ model with arbitrary initial workload distribution. To actually compute the transient workload distributions, the method employs a two-dimensional numerical transform technique. There are two major issues in this numerical algorithm: computational time and inversion precision. The computational effort of this method is proportional to the square of the number of stationary (piecewise) intervals, but does not depend on the length of the intervals. In the case of a seven-interval example, in which the arrival rate exceeds capacity during three of the intervals, the method takes 18 minutes to calculate 10 values (5 points in time, two threshold values) in the last interval on a SUN SPARCstation 2 using FORTRAN double precision, resulting in 7-10 digits of accuracy. An equivalent 21-interval example (3 times as many intervals), takes 3 hours to run (about 9 times as long). Choudhury et al.'s algorithm can be used to analyze transient or periodic behavior. Since the model focuses on workload, any work-conserving queue discipline (FCFS, LCFS, processor-sharing) is allowed.

Lemoine [27] develops moment formulas for periodic time-dependent and average asymptotic workload and waiting time in $M(t)/G/1$ queues with periodic Poisson arrival input. The expression for the r^{th} moment of the time-dependent workload distribution involves moments one through $r-1$, in the form of an integral equation. The expression may be explicitly evaluated if the asymptotic probability of an empty system at each t is exactly $1-\rho$, the time-average probability that the system is empty. A sufficient condition for the expressions to be explicitly computable is if service times are discrete and take on only positive integer multiples of the period length. This can sometimes be a serious restriction in applications, for instance, in the airport context.

The difficulty of finding exact solutions for nonstationary queues generated interest in developing approximation methods. There are three main classifications for time-dependent approximations. Surrogate Distribution Approximation (SDA) methods focus on the differential equations for the moments of the number of customers (or phases) in $M(t)/M(t)/s/c$ systems and its generalizations. The Pointwise Stationary Approximation (PSA) and its extensions focus on applying the stationary performance measure formulas for $M(t)/M(t)/s$ at each point in time. A variant of the Markov-modulated approach approximates the time-dependent arrival function and uses known results from Markov-modulated queues to estimate time-dependent workload probabilities in $M(t)/G/1$ queues.

The SDA method is based on the differential equations for the time-dependent mean number of customers in the system, $m(t)$, its second moment, $m_2(t)$, and variance, $v(t)$, initially developed by Clarke [8] for the $M(t)/M(t)/1$ system. These *moment differential equations* (MDE's) can be carefully written for any $Ph(t)/M(t)/s/c$ or $Ph(t)/Ph(t)/1/c$ system. If the unknown quantities appearing in the MDE's can be found or estimated, the MDE's can be integrated, and the time-dependent mean and variance, and higher moments can be found. Hence, SDA methods can approximate the transient behavior of a queue by tracking the time-dependent mean and variance given initial conditions, and, in the case of periodic arrival and service rates (with a common period), it can approximate the limiting periodic mean and variance of the number of customers or phases in the system. For example, let $P_i(t)$ represent the probability of i customers in an $M(t)/M(t)/s$ system at time t. From Rothkopf and Oren [46], the MDE's for the $M(t)/M(t)/s$ system are:

$$m'(t) = \lambda(t) - \mu(t)s + \mu(t) \sum_{n=0}^{s-1} (s-n)P_n(t) \quad (1.1)$$

$$v'(t) = \lambda(t) + \mu(t)s - \mu(t) \sum_{n=0}^{s-1} (2m(t) + 1 - 2n)(s-n)P_n(t) \quad (1.2)$$

Note that, given $\lambda(t)$ and $\mu(t)$ over the interval of interest, the unknowns in equations (1.1) and (1.2) are $P_0(t), \ldots, P_{s-1}(t)$. The SDA use ***surrogate distributions*** to approximate these unknown probabilities, i.e., $P_i(t) \approx P(X = i)$, where X is a random variable associated with a distribution called the ***surrogate***. One finds the parameters of the surrogate by matching the current known values of $m(t)$, $m_2(t)$, and $v(t)$ to the corresponding moments of the surrogate.

Rider [40], Rothkopf and Oren [46], Clark [7], Taaffe and Ong [50], and Ong and Taaffe [36] develop increasingly sophisticated SDA models for queueing models with time-dependent phase-type arrival processes and service-time distributions through the SDA method. This approach offers several significant advantages over solving the entire set of Chapman-Kolmogorov equations. There are few equations to integrate at each time step, as compared to the entire set of Chapman-Kolmogorov equations, resulting in faster solutions to the time-dependent mean and variance for the number in the system. The user is also relieved of the responsibility for selecting a truncation point for the maximum queue length, c, in approximating queues with infinite queueing capacity. Finally, the number of equations to be solved is independent of the system capacity.

The general SDA Algorithm proceeds as follows [49], although the details differ slightly in some cases. For each time t, assume $m(t)$, $m_2(t)$ are known. Assume $P_i(t)$ is approximated by $P(X = i)$, where X is a random variable with an associated distribution, called the surrogate.

1. Solve for the surrogate distribution parameters using $m(t)$, $m_2(t)$, and $v(t)$ to obtain the unknown $P_i(t)$ appearing in the MDE's.

2. Use the $P_i(t)$'s to obtain $m'(t)$, $v'(t)$.

Author	System	Quantities Approximated	Surrogate Distribution	Number of Equations Integrated
Rider	$M(t)/M(t)/1$	$m(t)$	Approximate $P_0(t)$ by modifying exact solution to $P_0(t)$ for constant λ,μ	1
Rothkopf & Oren	$M(t)/M(t)/s$	$m(t)$,$m_2(t)$,$v(t)$	Negative Binomial	2
Clark	$M(t)/M(t)/s$	$m(t)$,$m_2(t)$,$v(t)$, $\delta(t)$, $E[W(t)]$	Conditional PE	5
Taaffe & Ong	$Ph(t)/M(t)/s/c$	$m(t)$,$m_2(t)$,$v(t)$	Conditional PE	$6k_1$
Ong & Taaffe	$Ph(t)/Ph(t)/1/c$	$m(t)$,$m_2(t)$,$v(t)$	Conditional PE	$k_1 + 3k_1k_2$

Table 1.1: Summary of SDA Methods Research, Systems Approximated, and Surrogate Distributions Used.

3. Calculate $m(t + \Delta t)$ and $v(t + \Delta t)$, using numerical integration.

4. $t = t + \Delta t$. Go to Step 1.

Given initial conditions $m(0)$, $m_2(0)$, and $v(0)$, one can find approximations for $m(t)$, $m_2(t)$ and $v(t)$ for any t.

SDA methods differ from each other, depending on which system is being approximated, and how the surrogate distribution is defined. The differences translate into the number of equations to be integrated at each time step, and precisely which probabilities are to be approximated. Let k_1, k_2 be the number of phases in the arrival process and service-time distribution, respectively. Let PE represent the Polya-Eggenberger distribution. (See Johnson and Kotz [21] for more information about PE distributions.) Let $\delta(t)$, $E[W(t)]$ represent the output rate and expected waiting time in the system at time t. Table 1.1 classifies the SDA methods according to system analyzed, surrogate distribution used, number of equations integrated at each time step, and the performance measures collected.

Rothkopf and Oren [46], Clark [7], Taaffe and Ong [50], and Ong and Taaffe [36] all find that their methods approximate $m(t)$ better than $v(t)$ or $\sigma(t)$, the standard deviation for the number of customers in the system. This is consistent with the empirical results of the approximation methods presented in this dissertation. Although it appears that it could easily be done, the researchers who have used the SDA method have not attempted to approximate the actual probability distribution for the number of customers in the system using a surrogate.

The Pointwise Stationary Approximation (PSA) for Markovian systems forms the basis of several approximation schemes for time-average, time-dependent, and peak epoch performance measures. It is appealing because it is the easiest to compute: the basis of the method is the closed-form expressions which exist for stationary parameter systems in steady-state. For this reason, PSA can be used only if the arrival rate never exceeds the service rate. Green and Kolesar [13] originally examined PSA as a long-run (time-averaged) measure. In this case, Green and Kolesar showed empirically that PSA provides an upper-bound on the actual performance measures, such as expected number in the system and probability of delay. Whitt [55] showed that PSA is asymptotically correct in the time-averaged and time-dependent versions, as the service and arrival rates increase with the instantaneous traffic intensity held fixed. Whitt also proved that the Average Stationary Approximation (ASA) is also asymptotically correct under the same conditions as PSA. ASA also uses the stationary $M/M/s$ (and more generally $M/G/s$) formulae with an *averaged* arrival rate over the interval $[t-x, t]$, where x is proportional (or equal) to the mean service time. This averaging allows ASA to be applied to systems in which the arrival rate exceeds the service rate for short periods such that (the time-averaged) utilization does not exceed one.

Green and Kolesar extended PSA to estimate peak epoch and peak hour expected queue lengths and expected delay, in addition to probability of delay [14, 15], in $M(t)/M/s$ and $M(t)/G/\infty$ systems with sinusoidal arrivals. We address the finite-server case here. In the finite-server case, only the PSA approximation to the probability of delay exists when the arrival rate exceeds the service rate. The PSA approximations to the expected queue lengths and expected delay do not exist in this case. The Simple Peak Hour Approximation (SPHA) is an extension of the PSA which uses the average arrival rate over the peak hour to find the peak hour expected queue lengths, expected delay and probability of delay. The SPHA was generally within 10% of the exact system measures when the service rate was at least 2 per hour, and the peak utilization rate was less than 0.83. A service rate of 2 equals a half-hour average service time per call. This order of magnitude in the service rate is frequently experienced in both police patrol and firefighting [14]. For larger service rates (at least 20 per hour), SPHA gave good estimates of exact system parameters at higher maximum utilizations.

Although extremely easy to use, the PSA does have drawbacks for the application in which we are interested. In the finite-server case, the PSA approximation to the expected queue lengths and expected delay does not exist when the arrival rate exceeds the service rate. As mentioned earlier, the infinite-server approximation is not appropriate for the airport context. In both cases, PSA does not exhibit the time-lag between the time at which the system utilization peaks, and the time at which other system measures, such as expected number in the system, peaks. PSA peaks when system utilization peaks.

Rolski [41] develops approximations for time-dependent and time-average workload, mean delay and mean queue size for periodic $M(t)/G/1$ queues under equilibrium conditions. He examines a sequence of Markov-modulated arrival functions which converges in the limit to the exact periodic arrival process. Using theory developed by Regterschot and De Smit [39] for Markov-modulated processes, Rolski states that the time-dependent, and mean periodic workload processes at the k^{th} Markov-modulated queue converge, and proves that they converge in the limit to the exact distributions. Rolski demonstrates the accuracy of his method for the case of periodic queues with deterministic service times. However, it is not clear how these calculations would be performed for a more general service-time distribution.

In conclusion, there exist exact and approximation methods for nonstationary queueing methods. Each method has its strengths and weaknesses. There is room for development of other approximation methods. In this research, we present new, fast and flexible practical methods for approximating the time-dependent probability distribution for the number of customers in nonstationary queueing systems, from which moments for the time-dependent number in the system, and wait in the system can be calculated. The airport application requires the probability distribution to estimate the probability that the wait in queue exceeds any specified amount of time. (This type of information is of great interest to the airlines, their passengers, and the FAA.) One of the methods investigated in this thesis also approximates a system with time-dependent general (non-phase-type) service-time distributions. Furthermore, we develop and test a decomposition method for networks of dynamic single-server queues. With the exception of the marginal decomposition algorithm (MDA) proposed by Schmeiser and Taaffe [48] and the study of the local effects of hubs on hub-and-spoke networks by Peterson et al. [38], we know of no other methods for analyzing such dynamic queueing networks.

1.4 Outline of the Dissertation

This dissertation is organized as follows. Chapters 2 and 3 address approximations for a single-server dynamic queue of great practical importance, the $M(t)/E_k(t)/1$ queue. This infinite-capacity queueing system has a time-varying Poisson arrival process and a time-varying Erlang service-time distribution. An Erlang distribution is very useful for approximating certain common empirical distributions [24], such as the service times observed at airports. Using the Chapman-Kolmogorov forward equations, the $M(t)/E_k(t)/1$ system can be solved exactly by numerical techniques. Unfortunately, the time needed to solve this system exactly can be significant, hence our interest in developing and testing fast, accurate approximations. In Chapter 2, we will derive the five approximation methods examined in this thesis, and, in Chapter 3, we will describe the test cases used to determine the speed and accuracy of these approximations, and critical parameters affecting accuracy. Results show that several of the methods yield excellent approximations in one-third to one-four hundredths of the time it takes to solve the exact system, depending on the order of the Erlang and other parameters. The approximations give accurate results even when demand exceeds capacity for finite periods of time.

We investigate two new approximation methods: the State Probability Vector Approximation (SPVA) and DELAYS. SPVA is the most general of the approximations: it is a computational method for approximating $M(t)/G(t)/1$ systems. DELAYS was developed by Kivestu [23] as a fast approximation to the $M(t)/E_k(t)/1$ system. For completeness, we also investigate $M(t)/M(t)/1$, $M(t)/D(t)/1$, and INTERP [19] as approximations to the $M(t)/E_k/1$ system. The approximations $M(t)/M(t)/1$ and $M(t)/D(t)/1$ are well known. INTERP is a weighted combination of the $M(t)/M(t)/1$ and $M(t)/D(t)/1$ approximations, with the weights depending on the order of the Erlang being approximated. All the methods can approximate finite or infinite queueing-capacity systems and can be used to find transient and/or equilibrium system performance measures. They estimate the time-dependent probability distribution for the number of customers in the system, from which the time-dependent means, variances, and higher moments can be calculated, and similarly for the wait in the system and the probability that a customer is delayed more than a threshold value.

We will demonstrate the flexibility and accuracy of the SPVA approximation to

$M(t)/G(t)/1$ systems. We test the SPVA method empirically for two general service-time distributions. One test case has stationary service rates, the other time-varying. We also test the SPVA approximation to the $M(t)/H_2/1$ system. The hyperexponential distribution has a coefficient of variation greater than or equal to one. By testing the SPVA approximation to the $M(t)/E_k/1$ and $M(t)/H_2/1$ systems, we determine SPVA's ability to approximate queueing systems with small and large service-time coefficients of variation. Finally, we compare the SPVA method to three SDA methods.

Chapter 4 develops a computational approach for approximating dynamic queueing networks. We propose a "disaggregation–aggregation" (DA) approach: analyze the individual queues in the network independently, propagate the time-dependent congestion at individual queues to other queues in the network, update individual queue parameters, and repeat. Therefore, the DA approach consists of two distinct parts:

1. The Queueing Engine: a model to analyze the queues in isolation.
2. The Propagation Algorithm: an algorithm to link the queues together by propagating congestion among queues in the network.

We test computationally the plausibility of a DA approach by examining a tandem-queue (two-queue, acyclic) network. The test cases we examine will be ones for which we can obtain exact, time-dependent solutions to the problem; the exact and approximation solutions are compared. We determine the sensitivity of the DA solution to particular system parameters such as: the fraction of arrivals to a queue in the network which are departures from other queues, as opposed to external arrivals; the average and maximum utilization levels at the queues; the degree of nonstationarity in the arrival processes; and the service-time distributions at queues in the network. We develop rules of thumb for when a DA approach can sensibly be used to analyze dynamic queueing networks. Results indicate that a DA approach is reasonable for modeling any network of "weakly-connected" stations.

Chapter 5 presents observations, results, and conjectures about the general behavior of individual dynamic single-server queues with infinite waiting space. In our extensive computational analysis of the nonstationary queueing systems, we have observed consistent patterns of behavior across all cases examined. For example, the peak in the time-dependent variance for the number of customers in the system occurs strictly later than the peak in the mean number of customers in the system. Based on these types of observations, we will prove several results and state an additional conjecture about the general behavior of dynamic queueing systems.

Chapter 6 summarizes the contributions of this research and indicates possibly fruitful areas of future research.

Concern for developing models to understand and analyze complex real-world dynamic systems motivates this research. Contributions are of both a quantitative and qualitative nature. Quantitatively, this research develops fast, accurate approximation methods for some dynamic queueing systems of significant practical importance. These approximations are flexible and accurate, and it is hoped that they will be used as tools in future analyses. Qualitatively, we hope that the resulting improved understanding of complex dynamic queueing system behavior will provide rules of thumb to help planners and operators of facilities with strongly time-dependent demand and capacity to make better facility management decisions.

Chapter 2

Approximation Methods for Single-Server Queues with Nonstationary Arrivals and/or Services

This chapter presents new approaches for analyzing single-server queueing systems with nonstationary arrivals and/or service-time distributions. The new approaches are the State Probability Vector Approximation (SPVA) and DELAYS, developed by Kivestu [23]. SPVA and DELAYS are computational methods which can be used to analyze the transient or equilibrium behavior of a queueing system. The transient behavior concerns the system evolution from initial conditions to equilibrium. Systems with stationary arrival and service rates as well as those with nonstationary arrival and service rates exhibit transient behavior. One defines equilibrium behavior for systems with nonstationary arrivals or service differently from that of systems with stationary parameters. Equilibrium behavior for nonstationary systems is defined only in the case of stable systems with periodic arrival functions. In these cases, the behavior of the system repeats itself every period. Bambos and Walrand [3], Harrison and Lemoine [17], Heyman and Whitt [18], and Rolski [42, 43] present stability conditions for queueing systems with nonstationary arrivals. A sufficient condition for stability in a single-server queue with periodic Poisson arrival rate is that the time-average arrival rate over the period is strictly less than the service rate. In this research, we assume that all the systems we examine are stable. Chapter 3 examines the accuracy of the approximation methods for both the transient and equilibrium behavior of queueing systems.

SPVA approximates $M(t)/G(t)/1$ systems. Few other methods exist which can model dynamic systems with general service-time distributions exactly or approximately. To our knowledge, none is as computationally efficient as SPVA. SPVA solves for the time-dependent probability distribution for the number of customers in the system. From this probability distribution, moments for the number in the system or waiting time in the system can be found. We demonstrate that SPVA is a fast, flexible, and accurate method in Chapter 3.

DELAYS approximates the $M(t)/E_k(t)/1$ system. It is a fast and accurate method which solves for the probability distribution for the number of customers in the system.

This chapter describes the SPVA and DELAYS methods. We initially focus on these methods as approximations to queueing systems with nonstationary Poisson arrival processes and k^{th}-order Erlang service times, i.e., $M(t)/E_k(t)/1$. Why do we need approximation methods for this system which, after all, can be solved exactly by numerical methods? First, since we can solve the $M(t)/E_k(t)/1$ system exactly, we can compare the results obtained through SPVA and DELAYS to the exact values and assess the quality of the approximations. Second, the $M(t)/E_k(t)/1$ system can be used to approximate a wide variety of service-time distributions. Assessment of the SPVA and DELAYS approximations to the $M(t)/E_k(t)/1$ systems will then give an indication of their flexibility to approximate other queueing systems with service-time distributions having characteristics similar to the k^{th}-order Erlang. Third, finding the exact solution to $M(t)/E_k(t)/1$ system requires a significant amount of computer memory and CPU time. It may not be practical to solve this system exactly under some conditions, such as for large Erlang orders (memory constraints) or if the solution to the $M(t)/E_k(t)/1$ system is not an end in itself but one of many steps to a solution of a larger problem (time constraints). SPVA and DELAYS approximations to the $M(t)/E_k(t)/1$ system use far less memory and CPU time than the exact solutions.

For comparison purposes, we also investigate the $M(t)/M(t)/1$ and and $M(t)/D(t)/1$ methods, and an interpolation method, INTERP, as approximations to the $M(t)/E_k(t)/1$ system. Koopman [26] observed that the $M(t)/E_k(t)/1$ system is "somewhere in between" the $M(t)/M(t)/1$ and $M(t)/D(t)/1$ systems. When $k = 1$, the $M(t)/E_k(t)/1$ system reduces to an $M(t)/M(t)/1$ system, and as $k \to \infty$, it approaches asymptotically the $M(t)/D(t)/1$. However, we would not expect that either the $M(t)/M(t)/1$ or $M(t)/D(t)/1$ approximations are particularly good for $1 < k < \infty$.

INTERP is a linear combination of the $M(t)/M(t)/1$ and $M(t)/D(t)/1$ systems, where the weight associated with each depends on the order of the Erlang being approximated. Chapter 3 shows that INTERP provides an accurate approximation to the $M(t)/E_k(t)/1$ system.

In Chapter 3, we further test SPVA as an approximation to systems with nonstationary Poisson arrivals and general service times.

This chapter is organized as follows. It begins by defining the notation used in this thesis. It then derives the SPVA and DELAYS methods (Sections 2.2.1 and 2.2.2) and finally describes briefly the $M(t)/E_k(t)/1$, $M(t)/M(t)/1$, $M(t)/D(t)/1$, and INTERP methods. Detailed discussions on Markovian queueing systems can be found in many queueing textbooks, e.g., Kleinrock [24].

We note here that we model systems with infinite queueing capacity. In practice, we must, of course, use a finite number of equations to represent the dynamic evolution of these systems. We describe how we choose the number of equations to solve infinite queueing-capacity systems in Chapter 3.

2.1 Notation

We introduce here the notation used in this thesis. We assume the arrival rate to the queueing systems is periodic, with period T. We sometimes refer to the period as a "day," or "twenty-four hours." This is consistent with the airport paradigm described in Chapter 1. Let $\lambda(t)$ be the instantaneous arrival rate to the system at time t, let $\Lambda(t)$ be the cumulative arrival rate over $[0, t]$, and let $\mu(t)$ be the instantaneous service rate at time t. The probability that there are j customers in the system at time t will be denoted by $P_j(t)$. Let $L(t)$ represent the random variable for the number of customers in the system at time t, including the one in service. We denote the first and second moments, the standard deviation, and the variance for the number of customers in the system by $m(t)$, $m_2(t)$, $\sigma(t)$, and $v(t)$, respectively. Peak values (local maxima) will be denoted with asterisks; for example, m^* will denote a peak value of $m(t)$. We let $\alpha_n(j)$ ($\Upsilon_n(j)$) represent the probability that there are j arrivals (at least j arrivals) to the system during the service time of the n^{th} customer.

Let $B_t(x)$ represent the service-time cumulative distribution function (CDF) at time t, let $b_t(x)$ be its probability distribution function (pdf) and $\overline{b}_i(t)$ the i^{th} moment of the service-time distribution, at time t. This implies $\overline{b}_1(t) = \frac{1}{\mu(t)}$. We let $cv_b^2(t)$ represent the squared coefficient of variation, or the variance divided by the squared mean, of the service-time distribution. We use k to represent the order of the Erlang distribution.

We let $\overline{x}_i$ represent the expected service time of the i^{th} customer served in the queueing system. We represent the queueing capacity of the system by c. We assume a FCFS queue discipline for all systems and approximation methods.

2.2 The Approximation Methods

This section describes the methods we examine in this thesis. It begins with the two new methods, SPVA and DELAYS, and then briefly summarizes the $M(t)/E_k(t)/1$, $M(t)/M(t)/1$, $M(t)/D(t)/1$ and INTERP methods.

2.2.1 The State Probability Vector Approximation (SPVA)

SPVA is an $M/G/1$-like approximation to dynamic queueing systems developed by the author. It is the most general of the approximation methods examined in this research in that it makes no assumption about the particular form of the service-time distribution. That is, it can be used as a fast approximation for queueing systems with time-varying Poisson arrivals and general service-time distributions.

Standard analysis of an $M/G/1$ system, as described, e.g., in Kleinrock [24], uses the imbedded Markov Chain at customer departure epochs. The key to this approach is that the number of arrivals to the system during a customer's service time is independent of the customer currently in service. If we allow the arrival rate to vary with time, one must know the time at which the n^{th} departure occurs in order to calculate the number of arrivals during the $(n+1)^{st}$ service time. This added complexity causes the elegant analysis of the standard $M/G/1$ analysis to fail.

SPVA overcomes this problem by assuming that customer departures occur at $t_0, t_1, t_2, \ldots$, which we call the ***customer pseudo-departure epochs***. This assumption is the key to the SPVA approach. The definition of t_n provides a link between departure epochs and the "real" time clock. As a result, the arrival rate of customers immediately after the n^{th} pseudo-departure epoch is set equal to: $\lambda(t_n)$. We can then write the state probability equations for SPVA. The equations assume the imbedded Markov-chain is defined at customer pseudo-departure epochs. These equations look exactly like the equations of the $M/G/1$ system.

$$\begin{aligned} P_j(t_{n+1}) &= P_0(t_n)\alpha_{n+1}(j) + \sum_{i=1}^{j+1} P_i(t_n)\alpha_{n+1}(j-i+1),\ j=0,1,2,\ldots,c-1 \quad (2.1) \\ P_c(t_{n+1}) &= P_0(t_n)\Upsilon_{n+1}(c) + \sum_{i=1}^{c} P_i(t_n)\Upsilon_{n+1}(c-i+1) \quad (2.2) \end{aligned}$$

Given initial conditions for the system at time t_0, we can compute the state probability vector for any customer pseudo-departure epoch.

We define the n^{th} customer pseudo-departure epoch as the sum of the first n expected service times, multiplied by a constant $\beta > 0$. If the service rate does not vary with time, $t_n = \beta n \overline{x}$. If the service rate varies with time,

$$t_n = \beta\left(\sum_{i=1}^{n} \overline{x}_i\right), \quad (2.3)$$

where $\overline{x}_i$ is the expected service time of the i^{th} customer. We use the service rate immediately after the $(i-1)^{st}$ customer pseudo-departure epoch to find $\overline{x}_i$. That is, $\overline{x}_i = \frac{1}{\mu(t_{i-1})}$, where t_{i-1} is defined as in (2.3), and $t_0 \equiv 0$. β is a constant which we initially set to one. We test the effect of varying β on the accuracy of the SPVA method in Section 3.6.4.

Although SPVA assumes t_n depends only on the first moment of the service-time distribution, SPVA uses the entire service-time distribution in the calculation of $\alpha_{n+1}(j)$. The derivation of $\alpha_{n+1}(j)$ for the SPVA Method is similar to the derivation of $\alpha_{n+1}(j)$ in the $M/G/1$ system, except that we have an explicit dependence on time.

$$\begin{aligned} \alpha_{n+1}(j) &= P(j \text{ arrivals during the service time of the } (n+1)^{st} \text{ customer}) \\ &= \int_{x=0}^{\infty} P(j \text{ arrivals in } x \text{ time units, starting at time } t_n \mid \text{ service time } = x) \cdot \\ &\qquad P(\text{service time } = x) \quad (2.4) \\ &= \int_{x=0}^{\infty} \frac{(\lambda(t_n)x)^j}{j!} e^{-\lambda(t_n)x} dB_{t_n}(x) \quad (2.5) \end{aligned}$$

Details on how to calculate $\lambda(t_n)$ appear in Appendix B. In many cases, closed-form expressions for $\alpha_{n+1}(j)$ exist, as shown in Appendix A. In the case of the k^{th}-order Erlang distribution, for example,

$$\alpha_{n+1}(j) = \begin{pmatrix} k-1+j \\ j \end{pmatrix} \frac{(k\mu(t_n))^k[\lambda(t_n)]^j}{(k\mu(t_n)+\lambda(t_n))^{k+j}}$$

In the case that $\mu(t_n) = 0$ or $\lambda(t_n) = 0$ for some t_n, we define $\alpha_{n+1}(j)$ in the obvious way. For example, if $\mu(t) = 0$ for $t \in [t_n, \tau)$, no customers may depart in $t \in [t_n, \tau)$, but customers may arrive during this interval. Define $t_{n+1} = \tau$. Then, $\alpha_{n+1}(j)$ is the probability that j arrivals occur in the interval $[t_n, t_{n+1}]$, where the arrival process is nonstationary Poisson. That is,

$$\begin{aligned} \alpha_{n+1}(j) &= P(j \text{ arrivals in } [t_{n+1} - t_n] \text{ time units}) \\ &= \frac{[\Lambda(t_{n+1}) - \Lambda(t_n)]^j}{j!} e^{-[\Lambda(t_{n+1})-\Lambda(t_n)]}, \end{aligned}$$

where $\Lambda(t)$ is the cumulative arrival rate in $(0, t)$. In this case, we simply calculate the cumulative arrival rate during the period when no customers may depart. Conversely, if $\lambda(t) = 0$ for $t \in [t_n, \tau)$, then no customers may arrive during this interval. Again, let $t_{n+1} = \tau$, and define

$$\alpha_{n+1}(j) = \begin{cases} 1, & \text{if } j = 0 \\ 0, & \text{otherwise} \end{cases}$$

Does PASTA[1] hold for the SPVA Method? One important assumption for PASTA to hold is the "Lack of Anticipation Assumption (LAA)." Intuitively, the LAA requires that future arrivals be independent of the current state of the system. Under the same set of assumptions, PASTA holds for $M(t)/G/1$ systems [56] as well. In this case, Poisson arrivals see the system time averages *over the period*. Furthermore, a transient version of PASTA holds for arrivals to $M(t)/G/1$ systems. In this case, the probability that the system is in state i at time t given at least one arrival occurs in $(t + \Delta t)$ is the same as the probability that the system is in state i at time t. However, Mourtzinou showed that the probability that the system is in state i at time t immediately after a departure occurs is *not* the same

[1] PASTA = Poisson Arrivals See Time Averages. See Wolff [56] or Kleinrock [24].

as the probability that the system is in state i at time t [32]. Since the SPVA Method observes the system at customer pseudo-departure epochs, PASTA does not hold.

We do expect, however, that SPVA, a method which focuses on describing the time-dependent number in the system by looking at customer departure epochs, will be accurate for a system which is "relatively busy" and has frequent departures. Assuming departures occur "frequently enough," we expect that the distribution for the number of customers in the system at departure epochs will be close to the time-dependent number of customers in the system. In contrast, we expect SPVA will be less accurate for a system which is "essentially empty." Such systems have few arrivals, hence few departures, and few opportunities to observe the system. We expect that the number in the system which departures observe on average will be less representative of the actual time-dependent distribution for the number in the system as the system becomes less busy. But when the system is relatively busy, the impact of looking at customer pseudo-departure epochs, instead of pure time averages, is not critical from a practical point of view.

2.2.2 DELAYS

Kivestu [23] developed an approximation he called DELAYS for the $M(t)/E_k(t)/1$ system. It is based on a set of difference equations very similar to those solved in the $M(t)/D(t)/1$ system except that the epochs at which the state probability vector is solved are chosen differently. The epochs are based on ***time constants*** from the transient analysis of stationary queueing systems. For many stationary queueing systems, the rate at which a queue converges to its steady-state characteristics eventually becomes dominated by an exponential term [35]. The time constant we refer to appears in the exponent of the exponential term. The time constant can be used to determine how quickly the system responds to a step function. Therefore, a time-varying arrival process is viewed in this approach as the sum of many step functions. The sum of the responses to each of these step functions describes the dynamic behavior of the system as it evolves over time.

The key concept of DELAYS is the definition of the departure epochs. The DELAYS epochs are the same as those of the $M(t)/D(t)/1$ system multiplied by a constant, r. That is, the epochs are $t_0, t_1, t_2, \ldots$, where $t_j = r\sum_{i=1}^{j} \overline{x}_i$. To find r, Kivestu took the ratio of the time constants of the $M/E_k/1$ and $M/D/1$ systems,

$$r = \frac{\tau_{M/E_k/1}}{\tau_{M/D/1}} = \frac{k+1}{k}, \tag{2.6}$$

where $\tau_{M/G/1} = \frac{\lambda \bar{b}_2}{(1-\lambda \bar{b}_1)^2}$ is a time constant for the $M/G/1$ system found using the diffusion approximation [23, 25]. Reasoning that it takes $\frac{r}{\mu(t_n)}$ time units for the $M(t)/E_k(t)/1$ system to respond to input which the $M(t)/D(t)/1$ system responds to in $\frac{1}{\mu(t_n)}$ time units, Kivestu chose

$$t_n = \frac{k+1}{k} \sum_{i=1}^{n} \overline{x}_i.$$

We use the service rate immediately after the $(i-1)^{st}$ customer departure epoch to find $\overline{x}_i$. That is, $\overline{x}_i = \frac{1}{\mu(t_{i-1})}$. Note that when $k = 1$, then $r = 2$. On the other hand, as $k \to \infty$, $\lim_{k\to\infty}(t_{n+1} - t_n) = \lim_{k\to\infty} \frac{k+1}{k\mu(t_n)} = \frac{1}{\mu(t_n)}$. Therefore, the time step in DELAYS is bounded above and below; as k ranges from 1 to ∞, by

$$\frac{1}{\mu(t_n)} < (t_{n+1} - t_n) \leq \frac{2}{\mu(t_n)}.$$

In DELAYS, the probability of j arrivals in $\frac{k+1}{k}\overline{x}_{n+1}$ units of time, $\alpha_{n+1}(j)$, is calculated using the length of the expected service time, $\frac{1}{\mu(t_n)}$, not the actual time increment, $\frac{k+1}{k\mu(t_n)}$. That is,

$$\alpha_{n+1}(j) = \frac{\left(\frac{\lambda(t_n)}{\mu(t_n)}\right)^j e^{-\frac{\lambda(t_n)}{\mu(t_n)}}}{j!}.$$

The difference equations to be solved at each time epoch are:

$$\begin{aligned} P_j(t_{n+1}) &= P_0(t_n)\alpha_{n+1}(j) + \sum_{i=1}^{j+1} P_i(t_n)\alpha_{n+1}(j-i+1), \; j = 0,1,2,\ldots,c-1 \\ P_c(t_{n+1}) &= P_0(t_n)\Upsilon_{n+1}(c) + \sum_{i=1}^{c} P_i(t_n)\Upsilon_{n+1}(c-i+1) \end{aligned}$$

Given initial conditions for the system at time t_0, we can compute the state probability vector for epoch t_n.

DELAYS expects $\frac{k}{k+1}\Lambda(T)$ arrivals over the interval $[0,T]$. This expectation differs from the expected number of arrivals to the $M(t)/E_k/1$ system over the interval by the factor $\frac{k+1}{k}$. Therefore, DELAYS corrects the calculation for the expected number in the system

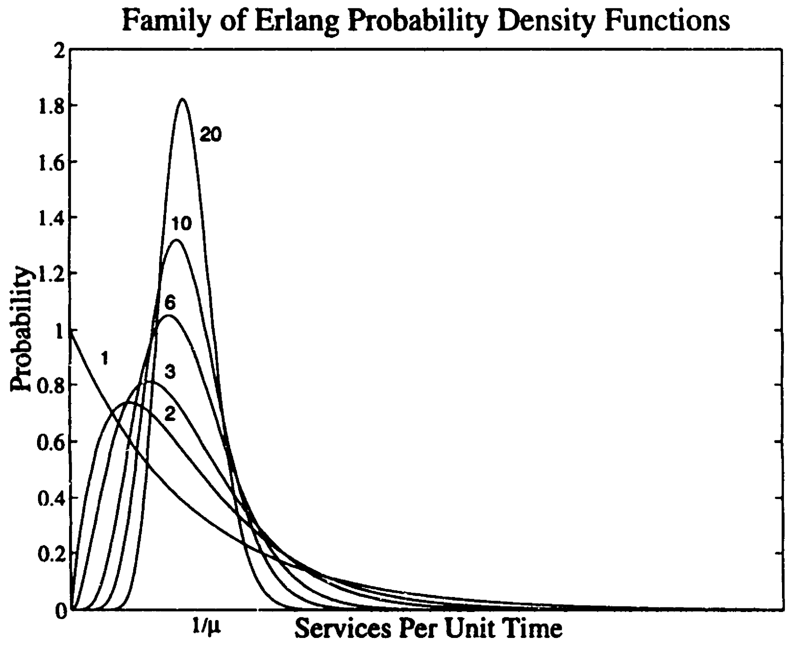


Figure 2-1: Family of Erlang Probability Density Functions

by multiplying by the factor $\frac{k+1}{k}$. Analogously, to correct the i^{th} moment for the number in the system, DELAYS multiplies by the factor $\left(\frac{k+1}{k}\right)^i$.

2.2.3 The M(t)/E$_k$(t)/1 System and Its Wide Applicability

The Erlang distribution has two parameters, $k \in \{1, 2, \ldots\}$ and $\mu(t) \geq 0$. One can write the probability density function (pdf) for the k^{th}-order Erlang in two ways (see Kleinrock [24] and Drake [10]). We use the Kleinrock convention. The probability density function, mean, and variance of the k^{th}-order Erlang are:

$$\begin{aligned} b_t(x) &= \frac{(k\mu(t))^k x^{k-1} e^{-k\mu(t)x}}{(k-1)!}, x \geq 0 \\ E(b_t) &= \frac{1}{\mu(t)} \\ \sigma^2_{b_t} &= \frac{1}{k(\mu(t))^2} \end{aligned}$$

The Erlang distribution can be used to approximate a wide variety of service-time distributions [24]. Figure 2-1 shows the probability distributions for Erlangs of different orders with the same mean. Momentarily disregarding the time-dependence subscripts, if we have sufficient data to obtain estimates of the actual mean, $\overline{x}_e$, and variance, σ^2_e, of a service process, then we can find μ and k for an Erlang distribution to approximate the actual service-time distribution as follows: Let int$(y) \equiv$ the integer nearest to y. Then,

$$\frac{1}{\mu} = \overline{x}_e \Longrightarrow \mu = \frac{1}{\overline{x}_e} \tag{2.7}$$

$$\frac{1}{k\mu^2} = \sigma_e^2 \Longrightarrow k = \text{int}\left(\frac{(\overline{x}_e)^2}{\sigma_e^2}\right) \tag{2.8}$$

If we rewrite equation (2.8) as $\sigma_e^2/(\overline{x}_e)^2 = 1/k$, we see that the squared coefficient of variation of the k^{th}-order Erlang distribution is less than or equal to one.

The k^{th}-order Erlang is the sum of k independently, identically distributed (iid) exponential random variables. This special property allows one to write a Markovian representation of the $M(t)/E_k(t)/1$ queueing system, where the states of the system represent the number of stages of service yet to be completed. In this case, the server provides each customer with k exponential stages of service, and may serve only one customer at a time. Each arrival to the system brings k stages of required service. n customers in the system at time t, with the customer currently in service having $i \leq k$ stages of service yet to complete, corresponds to a total of $(n-1)k + i$ stages of service yet to complete in the system. This relationship allows us to convert from the probability distribution for the number of stages of service in the system to the number of customers in the system at time t as follows. Let $P_i^E(t)$ denote the probability that there are i stages of service in the Erlang system at time t. Then,

$$\begin{aligned} P_0(t) &= P_0^E(t) \\ P_j(t) &= \sum_{i=(j-1)k+1}^{jk} P_i^E(t), j = 1, 2, \ldots, c \end{aligned}$$

Figure 2-2 shows a Markovian system with a Poisson arrival process and third-order Erlang service-time distribution. This figure shows that upward transitions in the state space jump by three, due to the fact that each arrival brings three stages of required service. The downward transitions represent service completions. A customer actually departs the queue after receiving three "stages of service." Downward transitions from states 1, 4, 7, etc., represent customers who physically depart the system.

The Chapman-Kolmogorov (CK) forward equations for the $M(t)/E_k(t)/1/kc$ system

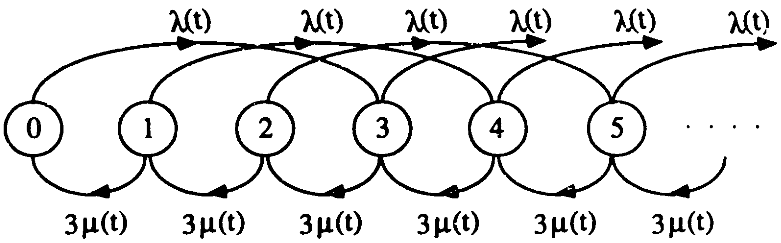


Figure 2-2: State-Transition-Rate Diagram for $M(t)/E_3(t)/1$

are:

$$\begin{aligned}
P_0'(t) &= -\lambda(t)P_0(t) + (k\mu(t))P_1(t) \\
P_i'(t) &= -(\lambda(t) + \mu(t))P_i(t) + (k\mu(t))P_{i+1}(t), \quad 1 \leq i < k \\
P_i'(t) &= \lambda(t)P_{i-k}(t) - (\lambda(t) + k\mu(t))P_i(t) + (k\mu(t))P_{i+1}(t), \quad k \leq i \leq (c-1)k \\
P_i'(t) &= \lambda(t)P_{i-k}(t) + (k\mu(t))(P_{i+1}(t) - P_i(t)), \quad (c-1)k < i \leq kc - 1 \\
P_{kc}'(t) &= \lambda(t)P_{k(c-1)}(t) - k\mu(t)P_{kc}(t))
\end{aligned}$$

Thus, in order to represent an $M(t)/E_k(t)/1$ system with capacity for c customers, kc+1 states (and CK equations) are needed.

2.2.4 M(t)/M/1 Queueing System

Another possible way to approximate an $M(t)/E_k(t)/1$ system is by solving the equations of an $M(t)/M(t)/1$ system. The approximation becomes an exact method when $k = 1$.

We solve the following standard CK equations, representing the number of customers in the system, for the $M(t)/M(t)/1/c$ system:

$$\begin{aligned}
P_0'(t) &= -\lambda(t)P_0(t) + \mu(t)P_1(t) \\
P_i'(t) &= \lambda(t)P_{i-1}(t) - (\lambda(t) + \mu(t))P_i(t) + \mu(t)P_{i+1}(t), \; 1 \leq i \leq c-1 \\
P_c'(t) &= \lambda(t)P_{c-1}(t) - \mu(t)P_c(t)
\end{aligned}$$

2.2.5 M(t)/D(t)/1 Queueing System

As the Erlang order k approaches infinity, the Erlang distribution becomes a unit-impulse function at $x = \frac{1}{\mu(t)}$, the mean of the Erlang distribution [24]. Another possible way to approximate an $M(t)/E_k(t)/1$ system is by solving the equations of an $M(t)/D(t)/1$

system.

We solve the difference equations for the $M(t)/D(t)/1$ system at epochs $t_0, t_1, t_2, \ldots$, where

$$t_n = \sum_{i=1}^{n} \overline{x}_i.$$

We use the service rate immediately after the $(i-1)^{st}$ epoch to find $\overline{x}_i$. That is, $\overline{x}_i = \frac{1}{\mu(t_{i-1})}$. The states of the system represent the number of customers in the system. The equations for the state probability vector at time t_{n+1} are:

$$\begin{aligned}
P_j(t_{n+1}) &= P_0(t_n)\alpha_{n+1}(j) + \sum_{i=1}^{j+1} P_i(t_n)\alpha_{n+1}(j-i+1), \; j = 0, 1, 2, \ldots, c-1 \\
P_c(t_{n+1}) &= P_0(t_n)\Upsilon_{n+1}(c) + \sum_{i=1}^{c} P_i(t_n)\Upsilon_{n+1}(c-i+1),
\end{aligned}$$

where $\alpha_{n+1}(j) = \frac{\left(\frac{\lambda(t_n)}{\mu(t_n)}\right)^j e^{-\frac{\lambda(t_n)}{\mu(t_n)}}}{j!}$. Given initial conditions for the system at time t_0, we can compute the state probability vector for any epoch. We note that solving the above set of equations does not give the exact solution for the time-dependent probability distribution for the number in an $M(t)/D(t)/1$ system. This is because we observe the system at particular epochs.

2.2.6 INTERP: The Interpolation Method

The interpolation method, INTERP, is based on Koopman's observation [26] that the Erlang distribution is "somewhere between" an exponential and a deterministic distribution. INTERP uses a linear combination of the state probability vectors for the $M(t)/M(t)/1$ and $M(t)/D(t)/1$ to approximate the state probability vector for an $M(t)/E_k(t)/1$ system. Hence, INTERP is an approximation method to the $M(t)/E_k(t)/1$ system which uses another approximation, $M(t)/D(t)/1$.

We define the state probability vector for the number of customers in the system for INTERP as:

$$P_j^I(t) = \frac{1}{k}P_j^M(t) + \frac{k-1}{k}P_j^D(t) \tag{2.9}$$

where the I, M, and D superscripts represent the INTERP, $M(t)/M(t)/1$ and $M(t)/D(t)/1$ systems, respectively. This weighting scheme has been used by Horangic [19]. Note that for the case of a stationary $M/E_k/1$ system in equilibrium, INTERP is exact for $k = 1$ and $k \to \infty$, as discussed in Sections 2.2.4 and 2.2.5.

Based on the definition in (2.9), the moments for the number of customers in the system are also linear combinations of the moments for the $M(t)/M(t)/1$ and $M(t)/D(t)/1$ approximations. The i^{th} moment of the number in the system at time t, $m_i(t)$, is:

$$m_i^I(t) = \frac{1}{k} m_i^M(t) + \frac{k-1}{k} m_i^D(t).$$

2.3 Summary

In this chapter, we introduced new computational methods for approximate analysis of nonstationary single-server queues. These methods are the State Probability Vector Approximation (SPVA), developed by this author, and DELAYS, developed by Kivestu [23]. We also described an interpolation method, INTERP, which is a linear combination of the $M(t)/M(t)/1$ and $M(t)/D(t)/1$ systems [26, 19] as well as the $M(t)/E_k(t)/1$, $M(t)/M(t)/1$ and $M(t)/D(t)/1$ systems. SPVA is the most general of the methods discussed; it approximates $M(t)/G(t)/1$ systems. DELAYS and INTERP approximate $M(t)/E_k(t)/1$ systems. All methods can model the transient and equilibrium behavior of these systems and calculate the time-dependent probability distribution for the number of customers in the system.

Chapter 3

Computational Experiments, Results and Comparisons for Dynamic Single-Server Queueing Systems

In this chapter, we determine the quality of new approximation methods for queueing systems with nonstationary arrival and/or service rates. The new methods we investigate are SPVA and DELAYS, derived in Chapter 2. SPVA approximates $M(t)/G(t)/1$ systems. DELAYS approximates $M(t)/E_k/1$ systems. We also investigate the $M(t)/M(t)/1$ and $M(t)/D(t)/1$ systems, and INTERP, which is a linear combination of the $M(t)/M(t)/1$ and $M(t)/D(t)/1$ systems.

We initially examine the accuracy of the approximations to $M(t)/E_k(t)/1$ systems. We test 76 cases ranging from moderately- to heavily-utilized systems. Each case is a combination of five system parameters, which we define in Section 3.3. We define measures of approximation quality in Section 3.1. Based on the quality measures, we identify combinations of parameters for which the approximation methods give good estimates of $M(t)/E_k(t)/1$ system measures.

The 76 test cases involve time-varying arrival rates but stationary service rates. In this way, we gain valuable insights into the behavior of systems with time-varying arrivals. Based on these initial 76 tests, we identify the most promising approximation methods. We test these further in Section 3.5. The additional case we test has both a time-varying arrival and service rate. It is based on the airport paradigm discussed in Chapter 1. We show empirically that the new approximation methods investigated in this research are accurate for systems with time-varying arrival and service rates.

We further test the SPVA approximation for other service-time distributions in Section 3.6. Section 3.6.1 presents SPVA approximation results for $M(t)/G(t)/1$ systems. We test the SPVA method empirically for two general service-time distributions. One test case has stationary service rates, the other time-varying. In both cases, the arrival rate varies with time. In both cases, SPVA gives good results. In Section 3.6.2, we test the SPVA approximation to the $M(t)/H_2/1$ system. The hyperexponential distribution has a coefficient of variation greater than one. By testing the SPVA approximation to the $M(t)/E_k/1$ and $M(t)/H_2/1$ systems, we determine SPVA's ability to approximate queueing systems with small and large service-time coefficients of variation. Section 3.6.2 contains results of test cases with moderate and high service-time coefficient of variation.

We compare the SPVA method with the fast and flexible SDA methods in Section 3.6.3. Descriptions of the SDA methods appear in Section 1.3. The SDA methods apply to systems with phase-type arrival and service processes. They have been shown to give good estimates of the time-dependent mean and variance of the number of customers in the system. We compare SPVA and SDA quality for $M(t)/M(t)/1$ systems.

Finally, we assess the effect of β, a constant used in the SPVA method, on SPVA accuracy, in Section 3.6.4.

This chapter begins with the description of accuracy measures used to assess the approximation methods.

3.1 Measures of Quality

We assess the approximation methods by their speed and accuracy. We measure speed by CPU time. We assess accuracy by comparing approximation estimates of the system measures to the exact values. We first describe accuracy, then speed measures.

An approximation estimate of a system measure is "good" if it is within 5% of the exact measure. Two measures which interest us are the time-dependent mean, $m(t)$, and standard deviation, $\sigma(t)$, for the number in the system. Comparison of the time-dependent measures allows us to determine how accurate the approximation estimates are, point-by-point in time. We measure accuracy for $m(t)$ and $\sigma(t)$ using the Weighted Percentage Error (WPE), which we define as

$$\text{WPE of mean } = \frac{\sum_{i=1}^{I} \left| m^E(t_i) - m^A(t_i) \right|}{\sum_{i=1}^{I} m^E(t_i)} \times 100.0$$

where $m^E(t_i)$ and $m^A(t_i)$ are the exact and approximation values at time t_i. I is the number of times we collect statistics over an interval of time. For example, the interval could be a day, and I could be the 24 statistics collected once an hour over the day. We define a similar measure for $\sigma(t)$, WPE of σ. The WPE has the following desirable properties. It provides

1. a summary statistic which indicates the quality of the approximation estimates of $m(t)$ and $\sigma(t)$ for each test case,
2. a weighted measure in which errors which occur at high values of system measures receive more weight than at low values of system measures, and
3. a measure which can be used to assess transient and equilibrium approximation quality.

Physically, WPE represents the area between the exact and approximation curves as a percentage of the total area under the exact curve.

WPE can measure either transient or equilibrium approximation quality. To assess approximation of transient behavior, we start to measure the WPE at time 0. We set $t_0 = 0$ and select the desired end of the interval of comparison. To assess the approximation of equilibrium behavior, we let the systems run until $m(t)$ and $\sigma(t)$ are no more than 2% apart over two consecutive periods. That is, until

$$\left| \frac{m(t) - m(t-T)}{m(t)} \right| \leq 0.02 \tag{3.1}$$

and

$$\left| \frac{\sigma(t) - \sigma(t-T)}{\sigma(t)} \right| \leq 0.02 \tag{3.2}$$

The two conditions must apply for an entire period T in order to meet our standard for equilibrium. This is the same type of standard as suggested by Rothkopf and Oren [46].

Once we find a value of t, say t_0, satisfying conditions (3.1) and (3.2), we assume the conditions hold for all $t > t_0$.

We also examine plots of $m(t)$ and $\sigma(t)$ over time. These graphs show, for example, whether an approximation method consistently overestimates $m(t)$ as $\lambda(t)$ increases. This information will be valuable in interpreting the approximation measures in the future.

We are also interested in the accuracy of the estimates of the peak expected number of customers in the system, m^*, and peak standard deviation for the number in the system, σ^*. By "peak," we mean the maximum values of $m(t)$ and $\sigma(t)$ achieved over period. We use Relative Error (RE) to assess approximation estimates of peak values. RE is the percentage difference from the exact value:

$$RE \text{ of } m^* = \frac{m^{A*} - m^{E*}}{m^{E*}} \times 100.0.$$

m^{E*} and m^{A*} are the exact and approximation peak expected number in system, respectively. We define a similar measure for the RE of σ^*.

We examine approximation estimates of the probability distribution for the number of customers in the system. This distribution varies with time, so we choose two points in time at which to examine the estimates: the times at which m^* and σ^* are achieved.

We now discuss the other measure of approximation quality, speed. We use one of two methods to measure CPU time. We measure CPU time using the UNIX operating system profiler, **gprof**, in the stationary service rate cases. **gprof** gives the CPU time used by each function called during a process. If several processes are run from a single batch file, the **gprof** output shows the cumulative CPU time used by each function over all the processes. This is a drawback because we would like to have separate CPU times for each function for each case. Fortunately, this only affects SPVA and DELAYS. In these cases, we divide the CPU times given by **gprof** equally over the processes.

When using **gprof**, routines common to two functions also present difficulties. **gprof** does not allocate the CPU time of routines used by two or more functions. For example, **gprof** does not allocate the CPU time of the common Visual Numerics C/Math/Library functions to either $M(t)/E_k(t)/1$ or $M(t)/M(t)/1$, the users of these functions. Therefore, the CPU times listed for the $M(t)/E_k(t)/1$ and $M(t)/M(t)/1$ systems include only the time used solving the actual differential equations, which are unique to each. There are two instances in which a batch file contains a single case: Cases 6 and 32. In these instances, **gprof** gives the exact CPU time needed to solve the $M(t)/E_k(t)/1$ system.

For the $M(t)/H_2/1$ tests and the network analyses of Chapter 4, we use the Visual Numerics C/math/Library utility `imsl_ctime` to measure CPU time. `imsl_ctime` gives precise estimates of CPU time for each function, regardless of the number of cases contained in a single batch file, or whether the functions use common routines.

3.2 Implementation Details

This section describes the computer hardware and software and solution methods used in the computational tests.

3.2.1 Computer Hardware and Software

All computational tests are run on a SUN SPARCstation 10 Model 41. We use the Visual Numerics C/Math/Library ordinary differential equation (ODE) solver to solve the Chapman-Kolmogorov forward (CK) equations of the $M(t)/E_k(t)/1$, $M(t)/M/1$ and $M(t)/H_2/1$ systems, in double precision. This function solves the ODE's using the Runge-Kutta-Verner fifth-order and sixth-order method, with a global error tolerance of 10^{-6} per call to the ODE solver. All other computer programs for the approximation methods and the CK equations were written by this author in the C programming language.

3.2.2 Solution Methods

In this section, we describe the implementation of the methods discussed in Chapter 2. Since the states of the queueing systems represent probabilities, we have the following boundary conditions which hold at all points in time: the probabilities are nonnegative and sum to 1.0. Taaffe [49] experimentally found that the accuracy of the Runge-Kutta numerical integration of the CK equations for an $M/M/1/K$ queueing system improved noticeably by normalizing the probabilities after each step taken by the algorithm. That is, $P_n(t) = \frac{P_n(t)}{\sum_{i=0}^{K} P_i(t)}$. Therefore, in this research, we normalize the solution of the $M(t)/E_k(t)/1$ and $M(t)/M(t)/1$ systems after each algorithm step. We also normalize the SPVA, DELAYS, and $M(t)/D(t)/1$ state probability vectors.

In this research, we approximate $M(t)/E_k(t)/1$ systems with infinite queueing-capacity systems. In practice, we must solve a finite number of equations. Therefore, we must choose the maximum size of the state probability vectors such that the probability of having a greater number of customers in the system is very small. We do this as follows. After each step in the algorithm, we calculate the cumulative probability up to and including the maximum state. We check that this probability is at least as great as $\eta = 0.999999$. We also require the probability mass in the maximum index state to be not "too big," namely, that it is smaller than $\epsilon = 10^{-8}$, similar to the approach in [45] and [14]. If either of these two conditions does not hold, we increase the maximum size of the vector used to store the state probabilities and resolve. The data structures used to store the SPVA, DELAYS, INTERP, $M(t)/M(t)/1$ and $M(t)/D(t)/1$ state probability vectors for a particular test case are all of the same size. As mentioned in Section 2.2.3, the size of the $M(t)/E_k(t)/1$ data structure is k times as large as the other methods.

When we test the SPVA approximation to the $M(t)/H_2/1$ system, we fix the queueing capacities. We note that these capacities do not meet our infinite queueing-capacity standard in some cases. However, the queueing capacities chosen are large from a practical point of view (600, 1200, and 2000).

To speed up the methods, we take advantage of the periodic utilization function by varying the number of equations to be solved at each iteration. Let $K_{\max}(t)$ be the smallest state index at time t such that $P_{K_{\max}}(t) \leq c$, $\sum_{j=0}^{K_{\max}} P_j(t) \geq \eta$ and $P_{K_{\max}}(t) \geq \epsilon$. After the next iteration, the smallest state index at time $t + \tau$ meeting the above criteria is found as follows:

$$K_{\max}(t+\tau) = \min\{i | 0 \leq i \leq c, \quad P_i(t+\tau) \geq \epsilon, \quad \sum_{j=0}^{i} P_j(t+\Delta t) > \eta\}$$

Once $K_{\max}(t + \tau)$ is found, the state probability vector is normalized. This approach succeeds in solving the SPVA, DELAYS, and $M(t)/D(t)/1$ methods. However, when we apply this method to the $M(t)/M(t)/1$ and $M(t)/E_k(t)/1$ systems, they become unstable in heavily-loaded cases. Therefore, we keep the number of equations constant in solving the $M(t)/E_k(t)/1$, $M(t)/M(t)/1$, and $M(t)/H_2/1$ systems, and vary the number of equations in SPVA, DELAYS, and $M(t)/D(t)/1$ methods.

3.3 Parameter Definition

In contrast to queueing systems with stationary parameters in equilibrium, there are additional important parameters which account for behavior of queueing systems with nonstationary arrival and/or service rates. See, for example, Eick et al. [11] and Green et al. [16]. The important parameters are average and maximum system utilization, degree of nonstationarity in the utilization rate, event frequency, and the number of servers. Our research concerns single-server systems, so the last parameter is always one. For single-server systems, average "system utilization" and "traffic intensity" are equivalent. We use the term "system utilization." We now define the four remaining parameters.

We use $\overline{\rho}$ and $\rho_{\max}$ to represent the average and maximum system utilization, respectively, over the period of interest. We define the time-average system utilization to be:

$$\overline{\rho} = \frac{1}{T}\int_{t=0}^{T} \rho(t)dt = \frac{1}{T}\int_{t=0}^{T} \frac{\lambda(t)}{\mu(t)}dt.$$

Maximum utilization is the maximum instantaneous utilization over the period of analysis. That is [16],

$$\rho_{\max} = \max_{0 \le t \le T}\{\rho(t)\}.$$

Green et al. [16] found that systems with $\rho_{\max} < 1$ and $\rho_{\max} \geq 1$ behave differently. Therefore, we include test cases with $\rho_{\max} < 1$ and $\rho_{\max} \geq 1$.

The "degree of nonstationarity" of a queueing system with time-varying arrival and/or service rates is not a well-defined concept. Green et al. [16] use a definition which is sensible for queueing systems with a stationary service rate and sinusoidal Poisson arrival process. In this case, they reason that the larger the amplitude of the sine function, the greater the nonstationarity of the process. We extend this idea to systems in which the service rate varies with time, and the arrival rate is not necessarily sinusoidal. It describes by how much the maximum utilization exceeds the average utilization over the period. We define the degree of nonstationarity, *Relative Amplitude* (RA) [16], to be:

$$RA = \frac{\rho_{\max} - \overline{\rho}}{\overline{\rho}} \tag{3.3}$$

Green et al.'s definition of nonstationarity is a special case of (3.3).

Event frequency is the number of events (arrivals or departures) per cycle. Green et al. [16] showed that event frequency has an effect on queueing systems with nonstationary arrivals and/or departures. We examine two levels of event frequency, high and moderate.

In addition to the four key parameters listed above, we also use the squared coefficient of variation as a parameter. In Sections 3.4.1 and 3.4.2, the squared coefficient of variation is $\frac{1}{k}$, where k is the order of the Erlang. We also define our average arrival rate over the period, $\overline{\lambda}$, such that $\overline{\lambda} = \frac{1}{T}\int_{t=0}^{T} \lambda(t)dt$.

Note that the parameters are not independent of each other. For example, RA and $\overline{\rho}$ determine $\rho_{\max}$. If $\mu(t) = \mu$, then μ and $\overline{\rho}$ determine $\overline{\lambda}$. The four independent parameters, $\overline{\rho}$, μ, RA and cv^2, determine the others.

3.4 Computational Tests and Results for Approximations to M(t)/E$_k$/1 Systems

This section describes the test cases and presents results for approximations to the $M(t)/E_k/1$ system. We first present the stationary service rate cases in Section 3.4.1. We discuss the quality of the approximations in Section 3.4.2. We identify under what conditions the approximation methods give good estimates of $m(t)$, $\sigma(t)$, m^* and σ^*. Finally, we identify the best approximations, based on speed and accuracy.

3.4.1 Stationary Service Rate Test Cases

In the initial 76 test cases we examine, we keep the service rate stationary in order to gain insights into the complex behavior of queueing systems with nonstationary arrivals. For simplicity, we use a sinusoidal Poisson arrival process with amplitude A, similar to Green et al. [16]. The test cases are combinations of the following parameter values, covering moderate to heavy system utilization levels.

- Poisson Arrival Function: $\lambda(t) = \overline{\lambda} + A\sin\left(\frac{2\pi t}{24}\right)$. Since $\lambda(t) \geq 0$, we restrict A to be $0 \leq A \leq \overline{\lambda}$. Note that $\lambda(t)$ is a smooth differentiable function with one peak over each period.

- Average Utilization ($\overline{\rho}$) ranges from moderately- to heavily-loaded systems: 0.5, 0.7, 0.75, 0.9

- Maximum Utilization: $0.67 \leq \rho_{\max} \leq 1.8$. In 60 of the 76 cases, $\rho_{\max} \geq 1$

- Degree of Nonstationarity: $RA = \frac{1}{3}, \frac{2}{3}, 1$. In the case of our sinusoidal arrival function, $RA = \frac{A}{\lambda}$. Note that $0 \leq RA \leq 1$ in this case.

- Event Frequency: high and moderate: $\mu = 100, 10$. We note here that the number of services per hour at busy airports is on the order of one hundred.

- Erlang orders: $k = 1, 3, 6, 10$. This range covers the squared coefficient of variation of service times typically observed at busy US airports.

Tables 3.1 and 3.2 list the parameters for each test case, along with an assigned case number. In the following discussion, we will refer to cases by their case numbers for convenience.

3.4.2 Stationary Service Rate Results

This section begins with an overview of the computational results for stationary service rate cases. We examine the approximations once all methods reach equilibrium. We present the CPU times, and compare estimates of $m(t)$, $\sigma(t)$, m^* and σ^*, as well as of the probability distributions for the number in system. We then assess each approximation method in detail, focusing on the SPVA, DELAYS and INTERP methods. We identify conditions for which the methods give good results.

Tables 3.3 and 3.4 show the CPU times for 61 of the 76 cases analyzed. We did not record CPU times for 15 of the initial test cases. Because one must solve the $M(t)/M/1$ and $M(t)/D/1$ systems to solve INTERP, the INTERP CPU times are the sum of of the CPU times of these two systems. The differences in CPU times among the methods are dramatic. DELAYS is consistently the fastest method, followed by $M(t)/D/1$ and SPVA. The $M(t)/M/1$ and INTERP methods use at least 4 to 14 times as much CPU time as SPVA does. $M(t)/E_k/1$ needs between three and five hundred times as much CPU time to be solved. The amount of time needed to solve the $M(t)/E_k/1$ system increases nonlinearly with k, event frequency, and system utilization. Why? Recall that $kc + 1$ equations are needed to represent the $M(t)/E_k(t)/1$ with queueing capacity c. Therefore, the number of equations grows linearly with k. The nonlinear growth in CPU time is introduced by the fact that the service rates in the $M(t)/E_k(t)/1/kc$ system are multiplied by a factor of k, causing the ODE solver to take smaller time steps at each iteration. High event frequency

Case	$\overline{\rho}$	$\rho_{\max}$	RA	Erlang Order	Event Frequency
1	0.5	0.67	0.33	10	100
2	0.5	0.67	0.33	1	100
3	0.5	0.67	0.33	3	100
4	0.5	0.67	0.33	6	100
5	0.5	0.83	0.67	10	100
6	0.5	0.83	0.67	1	100
7	0.5	0.83	0.67	3	100
8	0.5	0.83	0.67	6	100
9	0.5	1	1	10	100
10	0.5	1	1	1	100
11	0.5	1	1	3	100
12	0.5	1	1	6	100
13	0.5	0.67	0.33	10	10
14	0.5	0.67	0.33	1	10
15	0.5	0.67	0.33	3	10
16	0.5	0.67	0.33	6	10
17	0.5	0.83	0.67	10	10
18	0.5	0.83	0.67	1	10
19	0.5	0.83	0.67	3	10
20	0.5	0.83	0.67	6	10
21	0.5	1	1	10	10
22	0.5	1	1	1	10
23	0.5	1	1	3	10
24	0.5	1	1	6	10
25	0.75	1	0.33	10	100
26	0.75	1	0.33	1	100
27	0.75	1	0.33	3	100
28	0.75	1	0.33	6	100
29	0.75	1.25	0.67	10	100
30	0.75	1.25	0.67	1	100
31	0.75	1.25	0.67	3	100
32	0.75	1.25	0.67	6	100
33	0.75	1.5	1	10	100
34	0.75	1.5	1	1	100
35	0.75	1.5	1	3	100
36	0.75	1.5	1	6	100

Table 3.1: Test Case Parameters for Approximations to $M(t)/E_k/1$ Systems, Cases 1-36

Case	$\overline{\rho}$	$\rho_{\max}$	RA	Erlang Order	Event Frequency
37	0.75	1	0.33	10	10
38	0.75	1	0.33	1	10
39	0.75	1	0.33	3	10
40	0.75	1	0.33	6	10
41	0.75	1.25	0.67	10	10
42	0.75	1.25	0.67	1	10
43	0.75	1.25	0.67	3	10
44	0.75	1.25	0.67	6	10
45	0.75	1.5	1	10	10
46	0.75	1.5	1	1	10
47	0.75	1.5	1	3	10
48	0.75	1.5	1	6	10
49	0.7	1.4	1	10	10
50	0.7	1.4	1	1	10
51	0.7	1.4	1	3	10
52	0.7	1.4	1	6	10
53	0.9	1.2	0.33	10	100
54	0.9	1.2	0.33	1	100
55	0.9	1.2	0.33	3	100
56	0.9	1.2	0.33	6	100
57	0.9	1.5	0.67	10	100
58	0.9	1.5	0.67	1	100
59	0.9	1.5	0.67	3	100
60	0.9	1.5	0.67	6	100
61	0.9	1.8	1	10	100
62	0.9	1.8	1	1	100
63	0.9	1.8	1	3	100
64	0.9	1.8	1	6	100
65	0.9	1.2	0.33	10	10
66	0.9	1.2	0.33	1	10
67	0.9	1.2	0.33	3	10
68	0.9	1.2	0.33	6	10
69	0.9	1.5	0.67	10	10
70	0.9	1.5	0.67	1	10
71	0.9	1.5	0.67	3	10
72	0.9	1.5	0.67	6	10
73	0.9	1.8	1	10	10
74	0.9	1.8	1	1	10
75	0.9	1.8	1	3	10
76	0.9	1.8	1	6	10

Table 3.2: Test Case Parameters for Approximations to $M(t)/E_k/1$ Systems, Cases 37-76

Case	$M(t)/E_k/1^\dagger$	DELAYS	SPVA	INTERP	$M(t)/M/1^\dagger$	$M(t)/D/1$
6	60.8‡	9.7	11.1	–	–	12.2
9	1,158.3	2.5	4.1	25.9	21.3	4.6
10	23.3	2.5	4.1	25.9	21.3	4.6
11	129.8	2.5	4.1	25.9	21.3	4.6
12	510.1	2.5	4.1	25.9	21.3	4.6
13	39.2	0.1	0.1	0.8	0.7	0.1
14	0.9	0.1	0.1	0.8	0.7	0.1
15	3.7	0.1	0.1	0.8	0.7	0.1
16	11.7	0.1	0.1	0.8	0.7	0.1
17	89.6	0.2	0.2	3.1	2.9	0.2
18	2.8	0.2	0.2	3.1	2.9	0.2
19	9.5	0.2	0.2	3.1	2.9	0.2
20	31.0	0.2	0.2	3.1	2.9	0.2
21	210.1	0.3	0.5	6.4	5.9	0.5
22	6.1	0.3	0.5	6.4	5.9	0.5
23	25.6	0.3	0.5	6.4	5.9	0.5
24	76.6	0.3	0.5	6.4	5.9	0.5
25	2,666.0	4.7	7.7	52.0	44.2	7.8
26	48.3	4.7	7.7	52.0	44.2	7.8
27	298.6	4.7	7.7	52.0	44.2	7.8
28	1,111.8	4.7	7.7	52.0	44.2	7.8
29	2,548.0	7.3	13.5	54.5	44.0	10.5
30	50.0	7.3	13.5	54.5	44.0	10.5
31	294.0	7.3	13.5	54.5	44.0	10.5
32	814.6‡	2.3	4.1	–	–	3.9
33	4,226.5	16.5	27.8	123.4	98.7	24.7
34	106.8	16.5	27.8	123.4	98.7	24.7
35	609.3	16.5	27.8	123.4	98.7	24.7
36	1,817.3	16.5	27.8	123.4	98.7	24.7

Table 3.3: CPU Times on SUN SPARCStation 10 Model 41. Columns marked by † only include time solving ODE's, with exception of cases marked by ‡.

Case	$M(t)/E_k/1^\dagger$	DELAYS	SPVA	INTERP	$M(t)/M/1^\dagger$	$M(t)/D/1$
37	232.5	0.5	0.6	5.4	4.9	0.5
38	5.4	0.5	0.6	5.4	4.9	0.5
39	23.9	0.5	0.6	5.4	4.9	0.5
40	78.0	0.5	0.6	5.4	4.9	0.5
41	448.5	0.8	1.2	12.6	11.5	1.1
42	12.3	0.8	1.2	12.6	11.5	1.1
43	47.4	0.8	1.2	12.6	11.5	1.1
44	161.1	0.8	1.2	12.6	11.5	1.1
49	955.8	1.5	2.1	26.9	25.1	1.8
50	26.9	1.5	2.1	26.9	25.1	1.8
51	121.9	1.5	2.1	26.9	25.1	1.8
52	340.7	1.5	2.1	26.9	25.1	1.8
53	4,462.0	14.3	24.6	116.9	95.7	21.2
54	106.9	14.3	24.6	116.9	95.7	21.2
55	692.6	14.3	24.6	116.9	95.7	21.2
56	1,885.7	14.3	24.6	116.9	95.7	21.2
58	149.4	16.4	46.0	168.8	134.4	34.4
59	875.8	16.4	46.0	168.8	134.4	34.4
61	6,566.7	100.7	137.5	570.3	446.9	123.4
63	1,938.5	100.7	137.5	570.3	446.9	123.4
65	476.7	1.1	1.8	11.3	10.0	1.3
66	11.0	1.1	1.8	11.3	10.0	1.3
67	47.9	1.1	1.8	11.3	10.0	1.3
68	163.8	1.1	1.8	11.3	10.0	1.3
69	1,001.6	1.9	3.1	26.8	24.1	2.7
70	26.1	1.9	3.1	26.8	24.1	2.7
71	107.9	1.9	3.1	26.8	24.1	2.7
72	352.1	1.9	3.1	26.8	24.1	2.7
73	1,259.4	3.1	3.8	33.3	29.8	3.6
74	32.3	3.1	3.8	33.3	29.8	3.6
75	142.8	3.1	3.8	33.3	29.8	3.6
76	453.7	3.1	3.8	33.3	29.8	3.6

Table 3.4: CPU Times on SUN SPARCStation 10 Model 41, Continued. Columns marked by † only include time solving ODE's, with exception of cases marked by ‡.

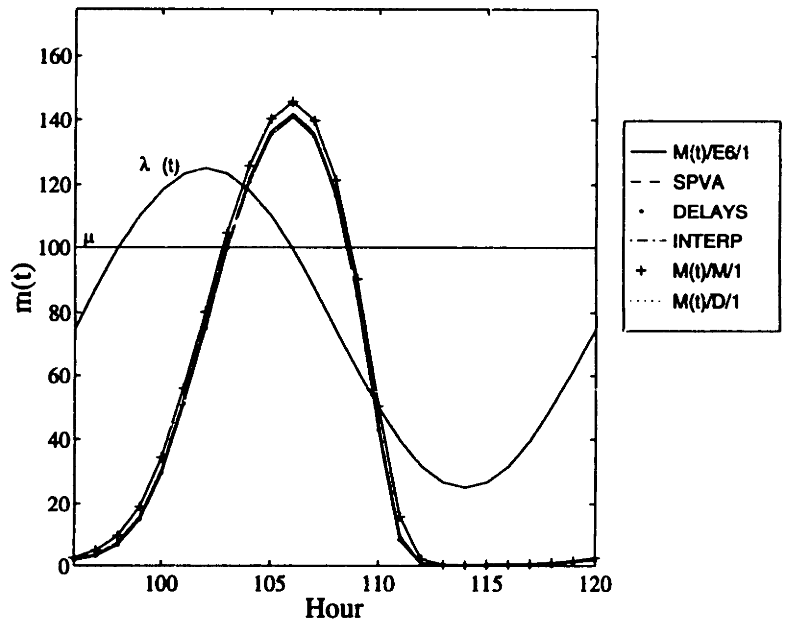


Figure 3-1: Expected Number in the System Over One Period. Case 32.

also increases the coefficients of terms appearing in the ODE's. The combination of these factors causes the nonlinear growth in computation time with increasing k.

SPVA and DELAYS have clear speed advantages over INTERP, $M(t)/M(t)/1$ and $M(t)/E_k(t)/1$. These advantages will become magnified if these methods are used in larger models, e.g., networks or "what-if" scenario analyses. In such cases, the methods may need to be run many times. The result of the speed advantage may be whether or not one can solve a problem in which there are time or memory constraints. Chapter 4 presents a network model in which memory and time constraints are serious issues, even for small problems.

We now make general observations about the estimates of $m(t)$, $\sigma(t)$, m^*, and σ^* by the approximation methods. When discussing the time-dependent accuracy of the methods, we focus our discussion on Case 32. This is a high-frequency case with average utilization of 0.75, maximum utilization of 1.25, Relative Amplitude of 0.67, and Erlang order of 6.

Figures 3-1 and 3-2 show the time-dependent mean and standard deviation for the number of customers in system over one period for Case 32. These figures show typical results. SPVA, DELAYS and INTERP estimates of $m(t)$ and $\sigma(t)$ are very close to that of the exact method. $M(t)/M/1$ and $M(t)/D/1$ over- and underestimate $m(t)$ and $\sigma(t)$, respectively. All methods show the same general shape of the $m(t)$ and $\sigma(t)$ curves. The time lag between the peak in $\lambda(t)$ and the peaks in $m(t)$ and $\sigma(t)$ is similar for all methods.

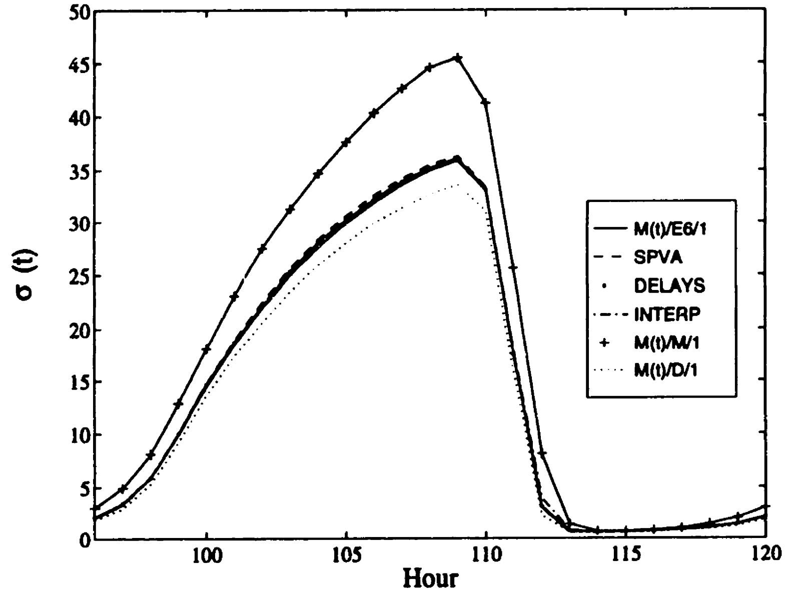


Figure 3-2: Standard Deviation for Number in the System Over One Period. Case 32.

The time lag is about four hours for the mean, and seven hours for the standard deviation.

Figure 3-3 plots the difference between the approximate and exact values for $m(t)$ for Case 32. The top graph shows this difference for all five approximation methods. The $M(t)/M/1$ and $M(t)/D/1$ differences dwarf those of SPVA, DELAYS and INTERP. The bottom graph shows these same differences for the SPVA, DELAYS and INTERP methods only. Over the entire 24-hour period, each of these three methods is within ± 0.4 of the exact value. This is very good, considering $m^* = 141.7$ in this case.

Figure 3-4 plots the difference between the approximate and exact values for $\sigma(t)$ for Case 32. The top graph shows this difference for all five approximation methods. Again, the $M(t)/M/1$ and $M(t)/D/1$ differences dwarf those of SPVA, DELAYS and INTERP. The bottom graph shows these same differences for the SPVA, DELAYS and INTERP methods only. Over the entire 24-hour period, each of these three methods is within ± 1 of the exact value. In this case, $\sigma^* = 35.9$. These results are good. Estimates of $m(t)$ are better than those of $\sigma(t)$ on a case-by-case basis.

Figure 3-4 shows the typical behavior of each of the methods. $M(t)/M/1$ severely overestimates $\sigma(t)$ over a long interval, which makes up most of the period. Likewise, $M(t)/D/1$ severely underestimates $\sigma(t)$ over a long interval of the period. $M(t)/D/1$'s underestimation neither lasts as long nor is as severe as the $M(t)/M/1$ overestimation. This behavior has an impact on INTERP. INTERP gives good estimates of $\sigma(t)$ over most

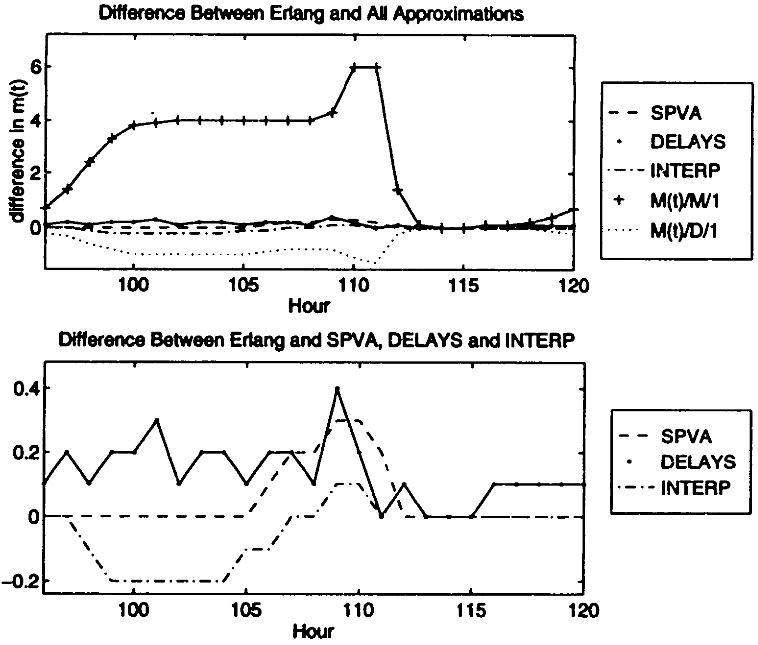


Figure 3-3: Difference between Approximation and Erlang Mean: $m^A(t) - m^E(t)$. Case 32.

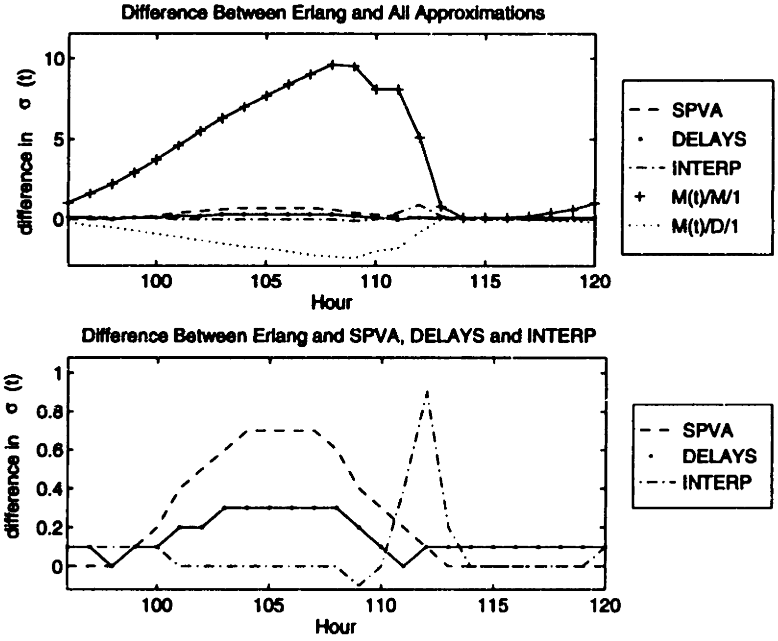


Figure 3-4: Difference between Approximation and Erlang Standard Deviation: $\sigma^A(t) - \sigma^E(t)$. Case 32.

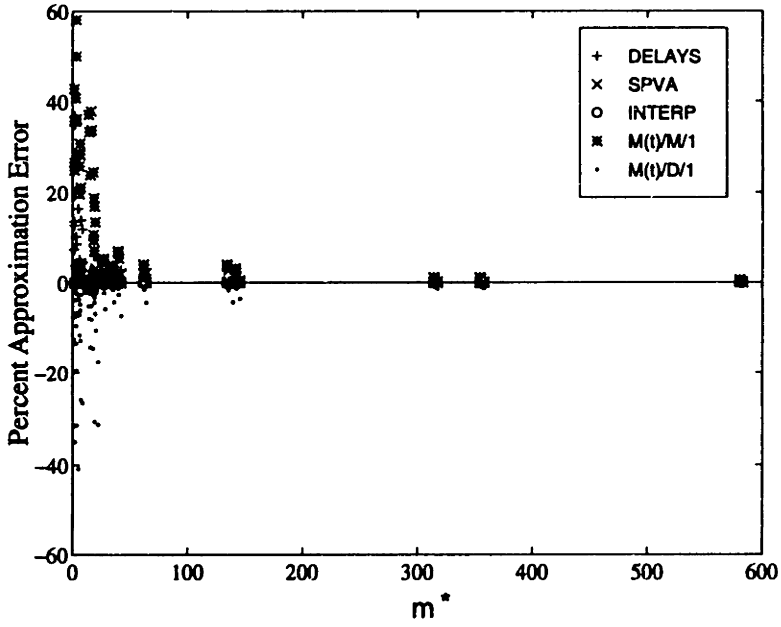


Figure 3-5: Relative Error in Estimating Peak Expected Number in the System, m^*

of the period, except for the "spike" in overestimating $\sigma(t)$, appearing at hour 112. We attribute this spike to $M(t)/M/1$. At hour 112, $M(t)/M/1$ still greatly overestimates $\sigma(t)$. In contrast, $M(t)/D/1$ underestimates $\sigma(t)$, at hour 112, but this underestimation is small. INTERP's definition gives $M(t)/M/1$ $\frac{1}{6}$ of the weight in Case 32, which, combined with the prolonged congestion, results in this spike in $\sigma(t)$ overestimation. INTERP shows this characteristic spike in each case, except for those in which it is exact ($k = 1$).

Figure 3-4 shows that SPVA increasingly overestimates $\sigma(t)$ while $\sigma(t)$ is building to its peak, and falls off thereafter. This behavior appears to varying degrees in all cases examined. Finally, DELAYS shows a somewhat jagged difference from the exact system, sometimes overestimating $\sigma(t)$, sometimes not. Again, this is typical of DELAYS.

Figures 3-5 and 3-6 show plots of the Relative Error (RE) in the approximations of m^* and σ^*, respectively. Most data points corresponding to greater than ±20% RE correspond to the $M(t)/M/1$ and $M(t)/D/1$ methods. Again, $M(t)/M/1$ and $M(t)/D/1$ errors dwarf those of SPVA, DELAYS, and INTERP. Figures 3-7 and 3-8 show the RE in estimating m^* and σ^* for the SPVA, DELAYS, and INTERP methods only. With the exception of a few DELAYS cases, these three approximations give good estimates of m^*. With the exception of a few cases for each approximation, they also give good estimates of σ^*.

Figures 3-9 and 3-10 show the probability distributions for the number in the system for

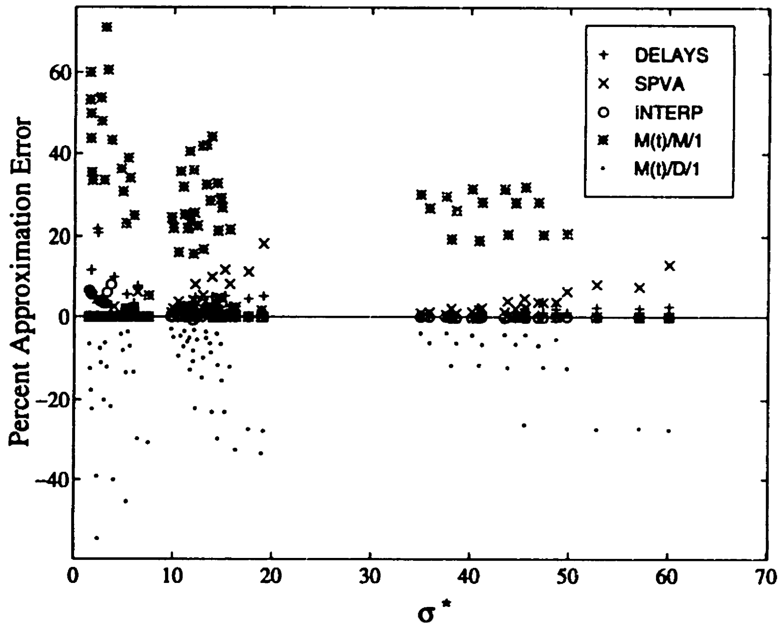


Figure 3-6: Relative Error in Estimating Peak Standard Deviation for Number in the System, σ^*

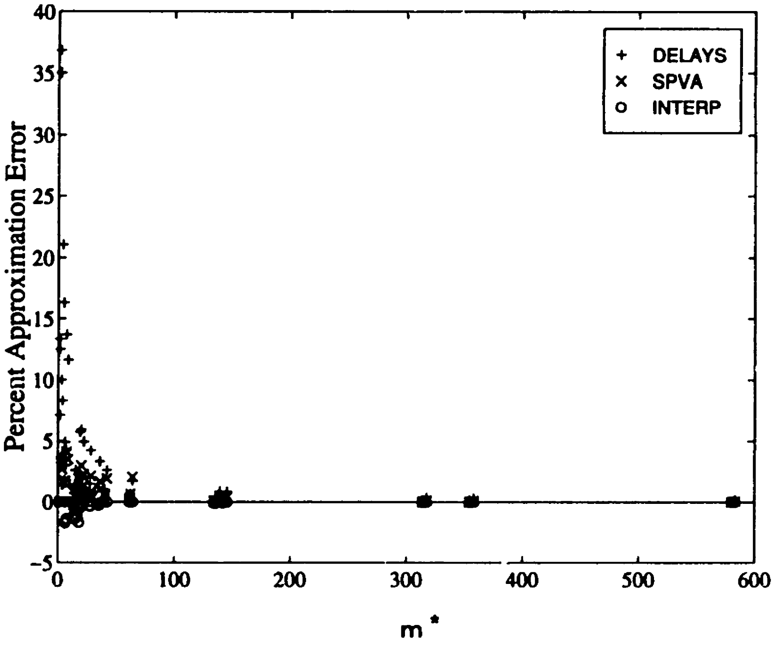


Figure 3-7: Relative Error in Estimating Peak Expected Number in the System, m^*, for SPVA, DELAYS and INTERP Methods

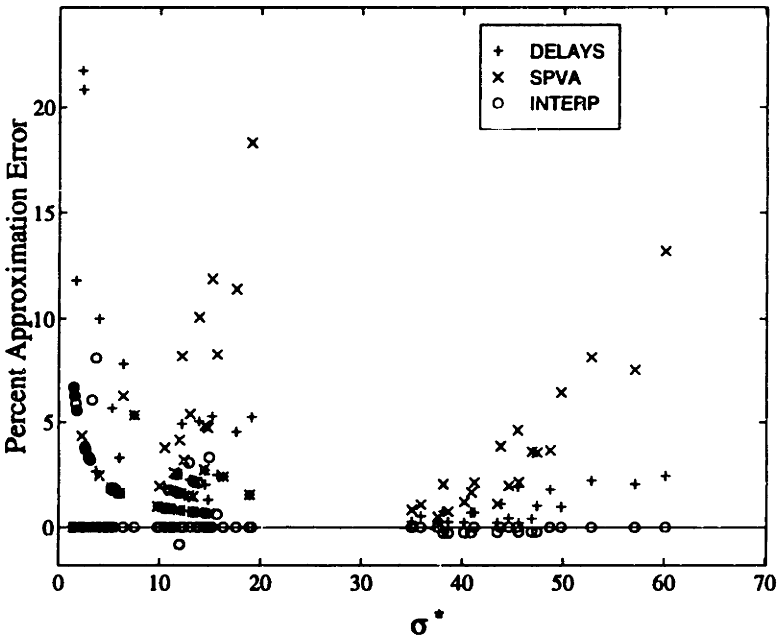


Figure 3-8: Relative Error in Estimating Peak Standard Deviation for Number in System, σ^*, for SPVA, DELAYS and INTERP Methods

Case 32. The distributions are snapshots of the probability distribution at the times at which m^* and σ^* are achieved. The SPVA and INTERP probability distributions approximate the exact $M(t)/E_6/1$ distribution well. Their modes are close to that of the exact, and the difference in area between the approximation and exact curves is small. The probability distribution of the $M(t)/M/1$ system is significantly different from $M(t)/E_6/1$. The mode of the $M(t)/M/1$ distribution is lower than that of the $M(t)/E_6/1$. We know that the variance of the the $M(t)/M/1$ system is larger than that of the $M(t)/E_6/1$ system at the time at which the distributions were recorded. We can see that in Figures 3-9 and 3-10 through the lower mode and longer tails of the $M(t)/M/1$ distribution. In addition, the mode of the $M(t)/M/1$ distribution is to the right of the exact, indicating overestimation of m^*. Likewise, the $M(t)/D/1$ distribution is more narrow and has a higher mode than the exact, indicating a smaller variance for the number in the system. Its mode is to the left of the exact system's. The DELAYS distribution has its modes centered near the exact modes. The DELAYS distribution appears to have a large spread. Recall that the $P_i(t)$ in the DELAYS system corresponds to the $P_{(i(k+1))/k}(t)$ in the system it is approximating. There is a "spreading" effect of the DELAYS distribution in this multiplication. In reality, the DELAYS distribution does not have a large variance for the number of customers in the system.

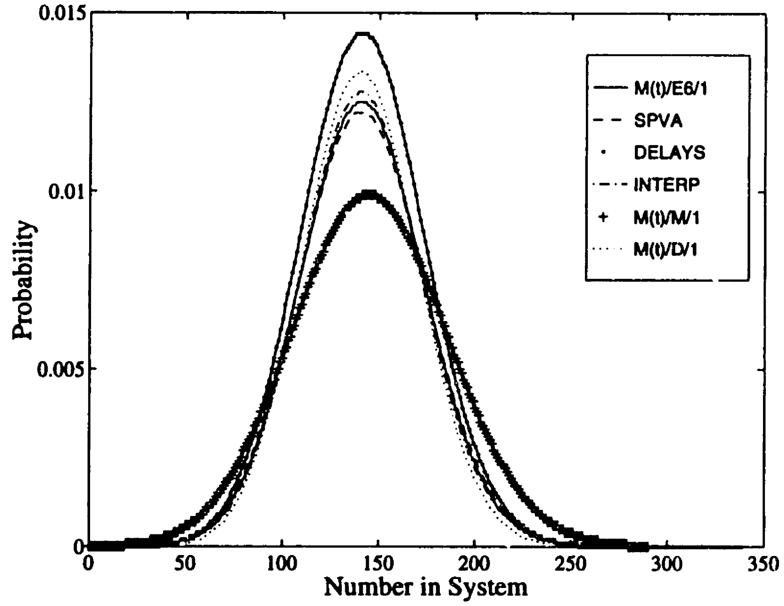


Figure 3-9: Probability Distribution for Number in the System at Time of Maximum Congestion. Case 32.

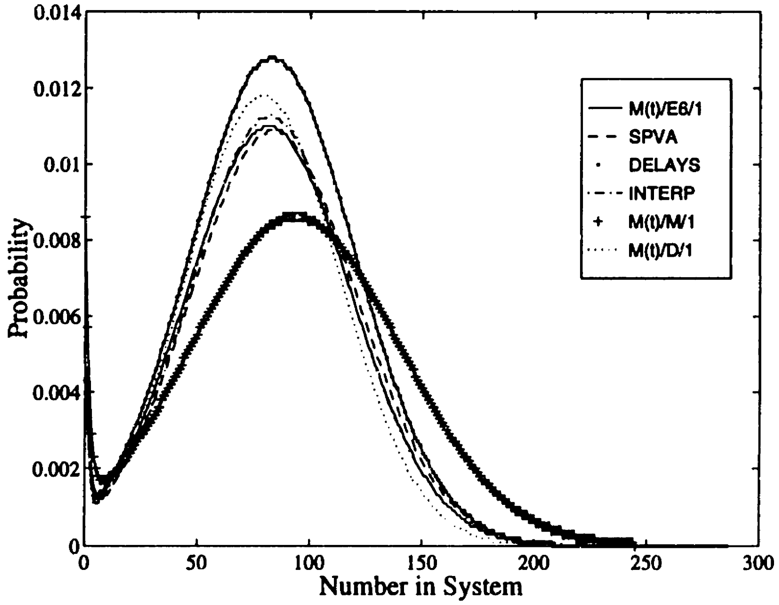


Figure 3-10: Probability Distribution for Number in the System at Time of Maximum Standard Deviation. Case 32.

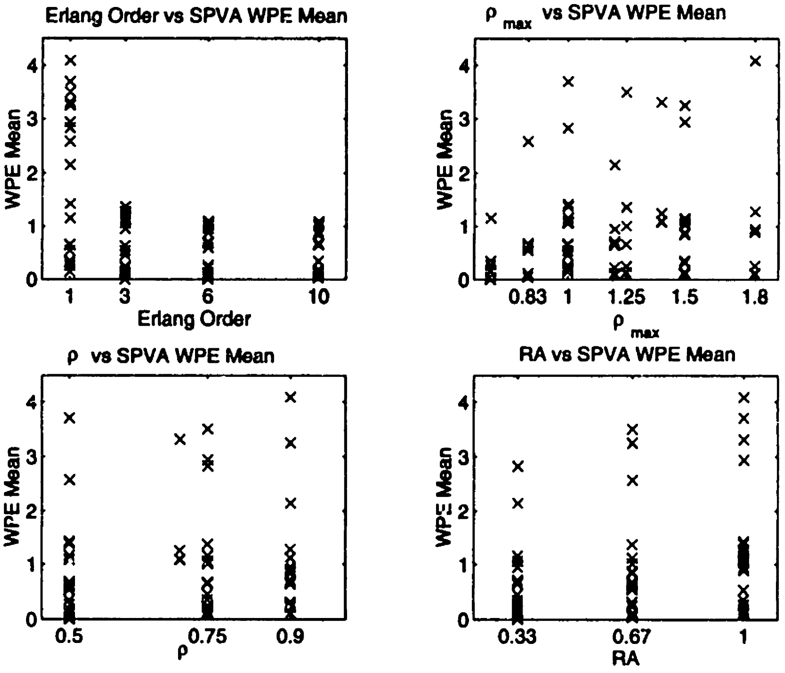


Figure 3-11: SPVA WPE of Mean vs. k, $\rho_{\max}$, $\overline{\rho}$, and RA

The speed and accuracy of the SPVA, DELAYS, and INTERP methods surpass the quality of the $M(t)/M/1$ and $M(t)/D/1$ approximations to the $M(t)/E_k/1$ system. Therefore, we do not consider the $M(t)/M/1$ and $M(t)/D/1$ methods further. We now focus on the SPVA, DELAYS, and INTERP methods. We examine each of these in detail to determine under which conditions they give good estimates of $m(t)$, $\sigma(t)$, m^* and σ^*.

As a general observation, we note that on a case-by-case basis, the errors for $m(t)$ and m^* are on the same order. That is, for each case, WPE of mean $\approx$ RE for m^*. The same holds for $\sigma(t)$ and σ^*. To avoid repetition, we focus our discussion on $m(t)$ and $\sigma(t)$.

Figures 3-11 - 3-16 show the influence of k, $\rho_{\max}$, $\overline{\rho}$, and RA, on the WPE of mean and of standard deviation for each method. We refer to these figures to assess the accuracy and sensitivity of the approximations. Note that the error scale differs in each figure, ranging from 0 - 2% for INTERP to 0 - 50% for DELAYS.

SPVA

SPVA gives good estimates (less than 5% error) of m^* and of $m(t)$ in all 76 cases. Figure 3-11 shows that SPVA accuracy increases with increasing k. The other parameters in isolation do not appear to affect SPVA accuracy.

SPVA gives good estimates of σ^* in 63 of the 76 cases and of $\sigma(t)$ in 56 of 76 cases.

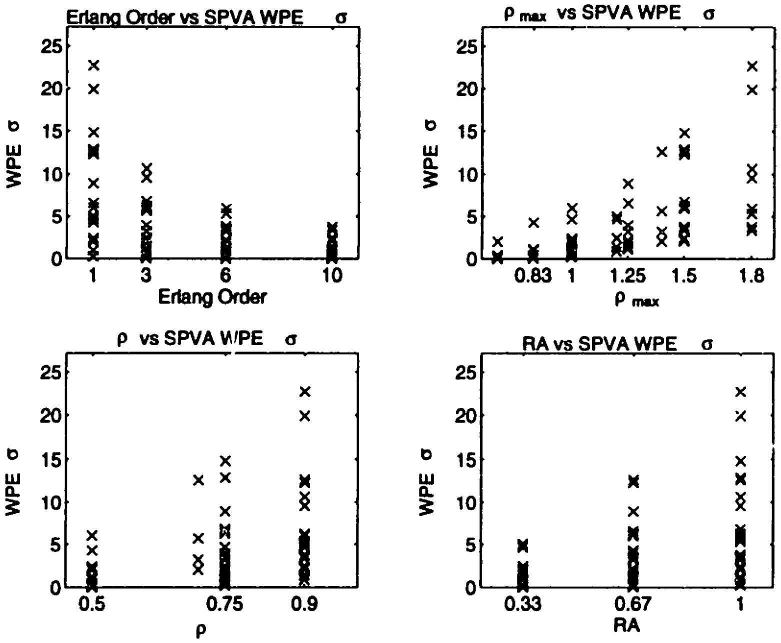


Figure 3-12: SPVA WPE of σ vs. k, $\rho_{\max}$, $\overline{\rho}$, and RA

Figure 3-12 shows that a combination of factors plays a role in the accuracy of $\sigma(t)$ estimates. For example, there are small and large errors for $k = 1$. Likewise for $\overline{\rho} = 0.9$. However, the combination of a low value of k, together with a high RA and a high $\overline{\rho}$, and not any one factor in isolation, decreases accuracy. In general, the WPE of σ is less than 5% for combinations of low k and low RA and $\overline{\rho}$, and of high k and any values of RA and $\overline{\rho}$. As k increases from one, SPVA estimates of $\sigma(t)$ are good for increasingly larger values of RA and $\overline{\rho}$.

In all 76 test cases, SPVA overestimates m^* and σ^*. It also tends to overestimate $m(t)$ and $\sigma(t)$ as $\lambda(t)$ increases. In some cases, SPVA underestimates $m(t)$ and $\sigma(t)$ when $\lambda(t)$ decreases. Estimates of $m(t)$ and m^* are better than those for $\sigma(t)$ and σ^*.

DELAYS

DELAYS gives good estimates of m^* in 60 of the 76 cases, and of $m(t)$ in 52 of 76 cases. Figure 3-13 shows that the DELAYS' estimate of $m(t)$ improves with increasing k, $\rho_{\max}$, $\overline{\rho}$, and RA. A combination of factors contributes to accuracy. It is at first surprising that increasing congestion and nonstationarity yield improvements in approximation accuracy. This can possibly be explained by the fact that DELAYS does not produce exact results for the $M(t)/E_k/1$ system with stationary parameters in equilibrium, and by the fact that for

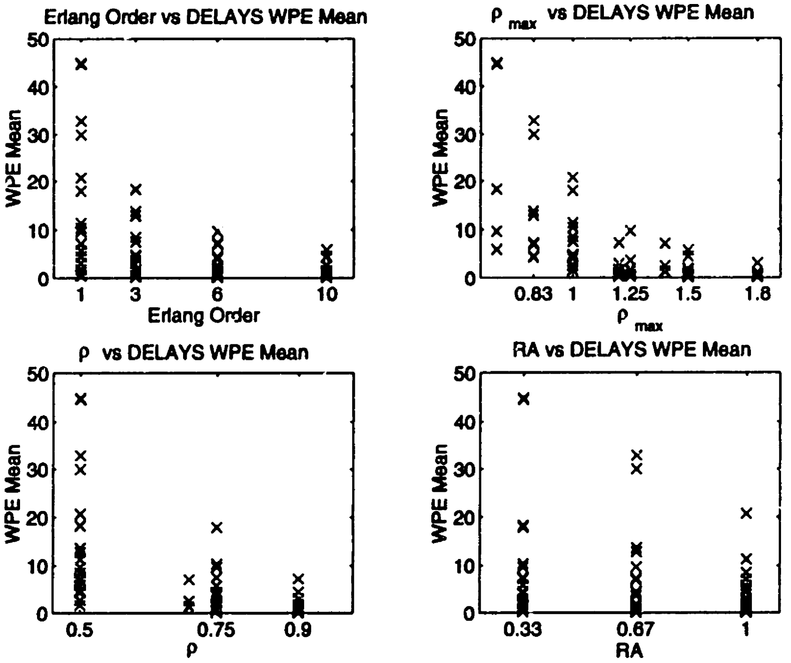


Figure 3-13: DELAYS WPE of Mean vs. k, $\rho_{\max}$, $\overline{\rho}$, and RA

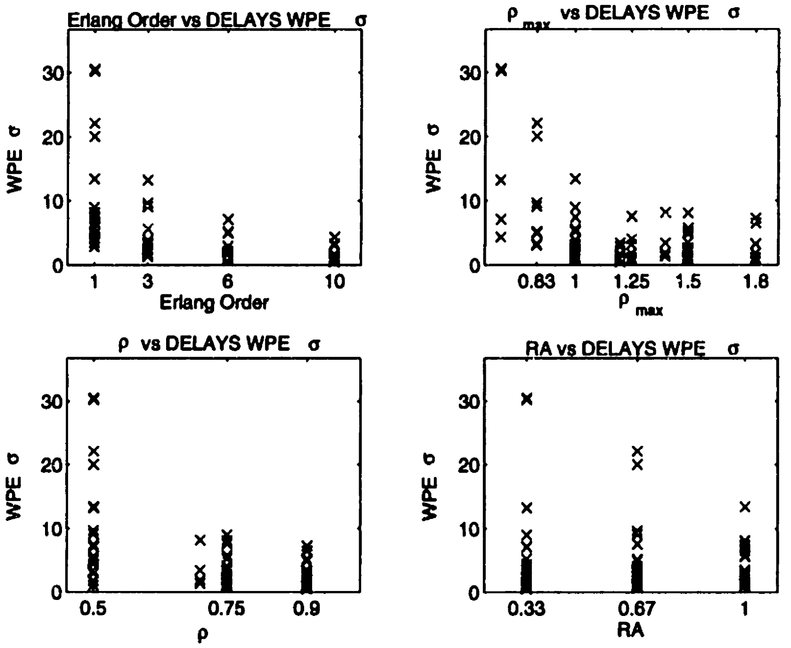


Figure 3-14: DELAYS WPE of σ vs. k, $\rho_{\max}$, $\overline{\rho}$, and RA

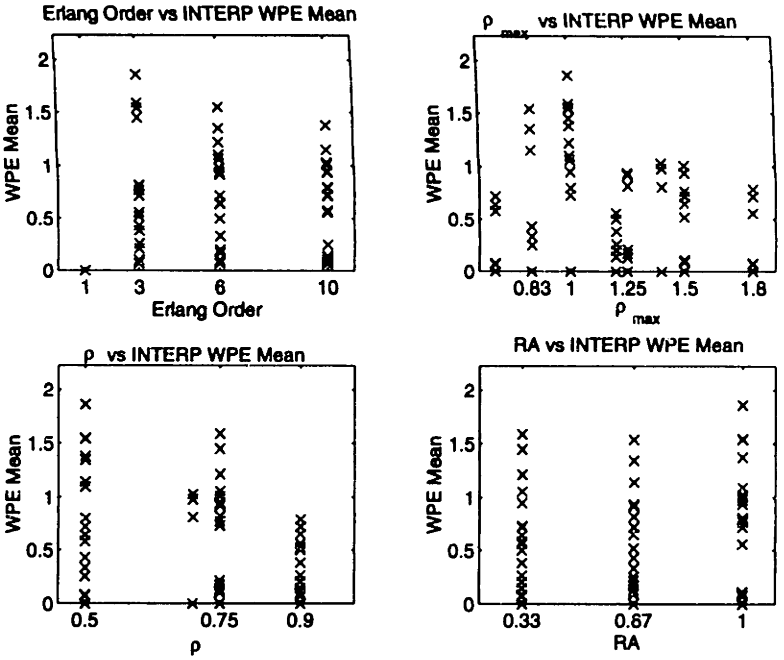


Figure 3-15: INTERP WPE of Mean vs. k, $\rho_{\max}$, $\overline{\rho}$, and RA

these cases the magnitude of the absolute error was very small, but in percentage terms the error was large. DELAYS generally produces good estimates for m^* and $m(t)$ when $k = 1$, and $\overline{\rho}$ is large, or if $\overline{\rho}$ is low, but RA is large. As k increases from one, DELAYS estimates improve for a broader range of the other parameters.

DELAYS gives good estimates of σ^* in 61 of the 76 cases examined, and of $\sigma(t)$ in 60 of 76 cases. Again, a combination of factors contributes to the accuracy, including increasing congestion and nonstationarity. Figure 3-14 shows the WPE of σ plotted against system parameters. Here, we see that the dominant factors which decrease accuracy the accuracy of DELAYS estimates of σ are low $\overline{\rho}$ and k. In generai, DELAYS is accurate for $k > 1$, and $\overline{\rho} > 0.7$

Like SPVA, DELAYS overestimates m^* and σ^*. However, DELAYS gives better estimates of $\sigma(t)$ and σ^* than of $m(t)$ and m^*.

INTERP

INTERP gives excellent estimates for m^*, σ^*, $m(t)$, and $\sigma(t)$. Like SPVA, approximations are better for the mean than for the standard deviation. INTERP gives good estimates of m^* and $m(t)$ in all 76 cases. WPE of mean are within ±2% in all these cases. It gives good estimatęs of σ^* in 70 of the 76 cases, and of $\sigma(t)$ in 73 of 76 cases. Recall that INTERP

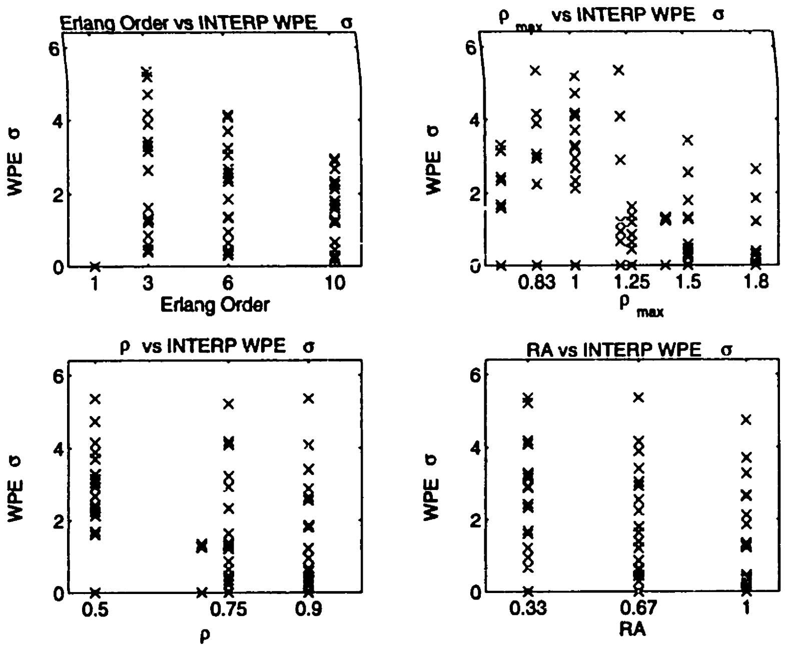


Figure 3-16: INTERP WPE of σ vs. k, $\rho_{\max}$, $\overline{\rho}$, and RA

is exact for $k = 1$ and asymptotically exact for $k \to \infty$. Figure 3-15 shows that k and $\overline{\rho}$ play the largest role in the accuracy of $m(t)$ estimates, and Figure 3-16 shows that k plays the critical role in the accuracy of $\sigma(t)$ estimates. $k = 3$ produces the least accurate results of the 4 values of k tested. Our results indicate that for $k \geq 6$, and of course for $k = 1$, INTERP gives good estimates for $\sigma(t)$, regardless of other system parameters. Note that the largest WPE of σ is less than 6%.

INTERP can both over- and underestimate system measures. INTERP underestimates m^* when not exact. It can over- or underestimate σ^*, when not exact. INTERP approximates $m(t)$ and $\sigma(t)$ well as $\lambda(t)$ increases but overestimates $m(t)$ and $\sigma(t)$ after $\lambda(t)$ peaks. In our test cases, INTERP produces a characteristic "spike" in overestimating $\sigma(t)$, as shown in Figure 3-4. We hypothesize the spike is due to the influence of $M(t)/M/1$ congestion in the system long after the congestion in the exact system has dissipated.

3.4.3 Summary of Approximations to M(t)/E$_k$/1 System

We have presented results of testing five approximation methods for the $M(t)/E_k/1$ system. The 76 test cases cover a wide variety of average and maximum system utilizations, degrees of nonstationarity, event frequencies, and Erlang orders. According to our measures of quality, three methods emerge as good approximations: SPVA, DELAYS, and INTERP.

	$m(t)$					$\sigma(t)$				
Approximation	maximum	Case Parameters				maximum	Case Parameters			
Method	WPE	k	$\overline{\rho}$	μ	RA	WPE	k	$\overline{\rho}$	μ	RA
SPVA	4%	1	0.9	10	1	23%	1	0.9	10	1
DELAYS	45%	1	0.5	10	1/3	31%	1	0.5	10	1/3
INTERP	2%	3	0.5	100	1	5%	3	0.9	10	1/3
$M(t)/M/1$	48%	10	0.75	100	1/3	56%	10	0.75	100	2/3
$M(t)/D/1$	36%	1	0.75	100	1/3	40%	1	0.75	100	1/3

Table 3.5: Mean and Standard Deviation Weighted Percentage Error (WPE) Maximum Range over 76 Cases of Stationary Service-Time Parameter

When does one choose to use an approximation method? Which one should be chosen? We have seen that there are large time and memory requirements to solve the $M(t)/E_k/1$ system exactly. When it is not practical to do so, one may choose to use an approximation method. There is a tradeoff between accuracy and speed in choosing between INTERP and SPVA or DELAYS. INTERP offers superior accuracy. However, one must solve both the $M(t)/M/1$ and $M(t)/D/1$ systems to use the INTERP method. This costs 4 to 14 times as much time as SPVA. SPVA and DELAYS are significantly faster than INTERP. If speed is critical, and accuracy within 3 – 5% of the exact value is sufficient, then SPVA and DELAYS offer good alternatives to solving the $M(t)/E_k/1$ system. We show examples of systems in which memory and speed are critical considerations in Chapter 4.

We conclude this section with some final remarks about the methods.

None of the approximations show sensitivity to the two event frequencies which we test. We believe that for even higher event frequencies, the approximation quality will remain unaffected or improve. However, we cannot conclude from our 76 tests that the same holds for lower event frequencies, say for $0 < \mu \leq 1$.

For completeness, we include Tables 3.5 and 3.6. They show the worst case performance of each method for estimating $m(t)$ and $\sigma(t)$, and m^* and σ^*, respectively. The tables show the typical combinations of parameters which yield the worst case performance.

We note that, with the exception of DELAYS, the approximation methods give better estimates of m^* and $m(t)$ than of σ^* and $\sigma(t)$. This observation is consistent with those made by other researchers. Rothkopf and Oren [46], Clark [7], Taaffe and Ong [50], and Ong and Taaffe [36] also produced better estimates of $m(t)$ than of $\sigma(t)$ in the systems they analyzed.

	m^*					σ^*				
Approximation	maximum	Case Parameters				maximum	Case Parameters			
Method	RE	k	$\bar{\rho}$	μ	RA	RE	k	$\bar{\rho}$	μ	RA
SPVA	4%	1	0.5	10	1	18%	1	0.9	10	1
DELAYS	37%	1	0.5	10	1/3	22%	1	0.5	10	1/3
INTERP	-2%	3	0.75	100	1/3	8%	3	0.5	100	2/3
$M(t)/M/1$	58%	10	0.5	100	2/3	71%	10	0.5	100	2/3
$M(t)/D/1$	-41%	1	0.5	100	2/3	-54%	1	0.5	10	1/3

Table 3.6: Maximum Percentage Error in Estimating m^* and σ^* over 76 Cases of Stationary Service-Time Parameter

3.5 Approximations to $\mathbf{M(t)/E_k(t)/1}$ System: Time-Varying Arrival and Service Rates

In this section, we further test two approximation methods for systems in which the arrival and service rates vary with time. We test the SPVA and DELAYS approximation methods to the $M(t)/E_k(t)/1$ system. We choose a scenario to test SPVA and DELAYS based on our airport application. We estimate the time-dependent mean and standard deviation for the number in system at Boston Logan International Airport.

We determine the arrival and service rates at the airport as follows. We use an hourly demand rate at Logan from May 6, 1993, measured in operations per hour. The rates include scheduled and GA operations. The number of operations ranges from 0 to over 100 per hour. We choose a 9^{th}-order Erlang to model the service-time distribution, corresponding to the squared service-time coefficient of variation of $\frac{1}{9}$. The capacity scenario chosen is one in which Logan experiences a decrease in capacity during the morning hours of operation, possibly due to morning fog. The fog causes capacity to drop from 120 operations per hour to 60 from 6 am to 9 am. This period of decreased capacity corresponds to Logan's morning peak demand period. The time-average utilization over the 24-hour period is approximately 0.63 and $\rho_{\max} \approx 1.66$, implying that $RA \approx 1.63$. We assume the system starts out empty.

Figure 3-17 shows the arrival and service rates over the 24 hours, the time-dependent mean number in the system for the $M(t)/E_9(t)/1$ system, and the SPVA and DELAYS approximations to the $M(t)/E_9(t)/1$ system. There is no discernible difference between the exact and approximation curves! The SPVA and DELAYS approximations to $m(t)$ are good in this case.

Figure 3-18 shows the time-dependent standard deviation of the number in the system

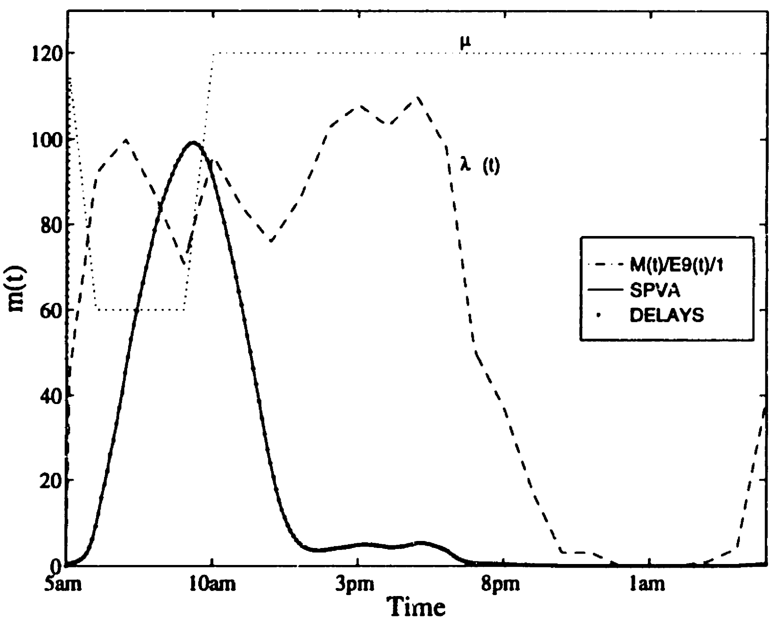


Figure 3-17: SPVA and DELAYS Approximations to the Expected Number in the $M(t)/E_9(t)/1$ System. Boston Logan International Airport Scenario.

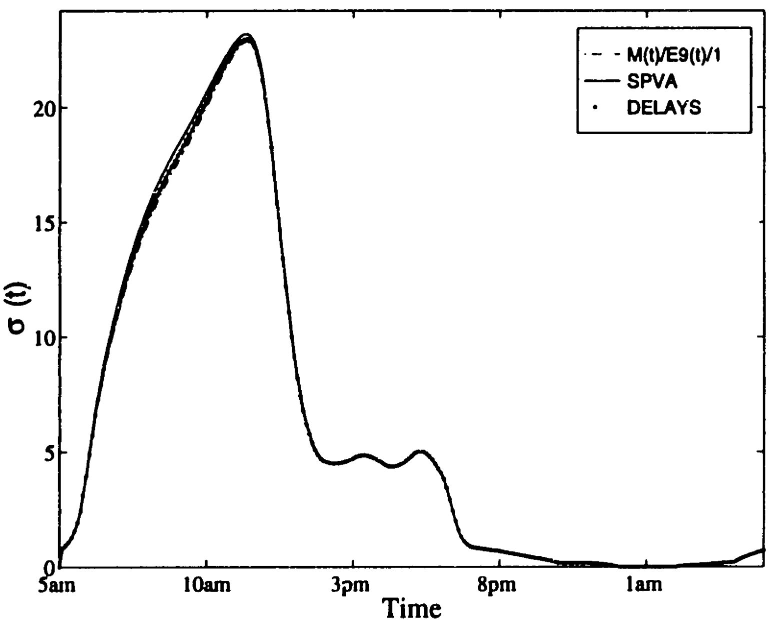


Figure 3-18: SPVA and DELAYS Approximations to Standard Deviation for the Number in the $M(t)/E_9(t)/1$ System. Boston Logan International Airport Scenario.

for the $M(t)/E_9(t)/1$ system, and the SPVA and DELAYS approximations. Again the SPVA and DELAYS approximations are good. The differences between the exact and approximation curves are small. The largest difference between DELAYS and the exact value is 0.3; the exact value at this time is about 18. The largest absolute difference between SPVA and the exact value is 0.7, when the exact value is about 17.

It is striking that the approximation methods capture the behavior of the exact $m(t)$ and $\sigma(t)$ well over the entire period. This includes, for example, the small local peaks in $\sigma(t)$ occurring at about 3:30 pm and 5:30 pm. There is no indication in this test case that a time-varying service rate adversely affects the accuracy of the SPVA and DELAYS methods. More empirical testing is necessary to draw conclusions on whether SPVA and DELAYS are accurate for other types of time-dependent service-rate functions.

Finally, SPVA and DELAYS need only a few seconds to determine $m(t)$ and $\sigma(t)$ for this airport scenario. In contrast, the ODE solver needs more than 40 minutes to solve the exact system. The approximation methods' performance in this scenario indicate that they give good estimates of $m(t)$ and $\sigma(t)$ when both the arrival rate and service-time distribution vary with time, in much less time than required for the exact solution.

3.6 Further Testing of SPVA

In this section, we test SPVA further. In Section 3.6.1 we use SPVA to approximate $M(t)/G(t)/1$ systems. We demonstrate how easily SPVA can be used to approximate $M(t)/G(t)/1$ systems. The results indicate that SPVA estimates of $m(t)$ and $\sigma(t)$ show behavior and values we expect. Section 3.6.2 assesses SPVA method accuracy for $M(t)/H_2/1$ systems. The hyperexponential distribution has a coefficient of variation greater than or equal to one. By testing the SPVA approximation to both $M(t)/E_k(t)/1$ and $M(t)/H_2/1$ systems, we obtain an indication of how well SPVA can approximate service-time distributions with virtually any coefficient of variation. Section 3.6.3 compares the quality of the SPVA method to the SDA methods of Rider, Rothkopf and Oren, and Clark. Finally, we determine SPVA sensitivity to the departure epoch stepsize. We vary the β discussed in Section 2.2.1 to determine its effect on SPVA estimates of m^*, σ^*, $m(t)$ and $\sigma(t)$.

3.6.1 SPVA Approximation to M(t)/G(t)/1 Systems

In this section, we apply the SPVA method to $M(t)/G(t)/1$ systems. What methods are available to analyze such systems? The exact method of Choudhury et al. [6] applies to systems with piecewise linear arrival rates. Since the computational time is proportional to the square of the number of stationary intervals, this method may not be practical for the types of arrival rate functions we want to model. The exact method of Lemoine [27] applies to $M(t)/G/1$ systems which have special types of service-time distributions. Exact, equilibrium solutions can be found if the service-time distributions are discrete, with probability mass only at integer multiples of the arrival function period. Rolski's [41] approximation method applies to $M(t)/G/1$ systems in equilibrium. There is room for development of more methods, which can be used for either transient or equilibrium analysis of systems with time-varying arrival and/or service rates. SPVA is such a method. The derivation of the SPVA approximation in Section 2.2.1 makes no assumption about the specific form of the service-time distribution. Recall also that SPVA is exact for stationary parameter $M/G/1$ systems in equilibrium. We now take advantage of these properties to examine two queueing systems with service-time distributions which might be considered as candidates for approximating service-time distributions at some airports: the uniform and triangular service-time distributions.

We illustrate the application of SPVA to $M(t)/G(t)/1$ systems using the same airport arrival rates as in Section 3.5. These rates are measured in number of operations per hour. Our first test case has stationary service rates, the second, nonstationary. In the stationary case, we assume that the expected service rate is 120 operations per hour, with $cv^2 = \frac{1}{9}$.

The uniform and triangular distributions each have two parameters. Their service-time distributions, means and variances are:

Uniform Service Times

$$\begin{aligned} f(x) &= \begin{cases} \frac{1}{b-a}, & 0 \leq a \leq x \leq b, \\ 0, & \text{otherwise} \end{cases} \\ \bar{b}_1 &= \frac{a+b}{2} \\ \sigma^2 &= \frac{(b-a)^2}{12} \end{aligned}$$

Triangular Service Times

$$\begin{aligned}
f(x) &= \begin{cases} \frac{4}{(d-c)^2}(x-c), & 0 \le c \le x \le \frac{d+c}{2} \\ \frac{4}{(d-c)^2}(d-x), & \frac{d+c}{2} \le x \le d \\ 0, & \text{otherwise} \end{cases} \\
\bar{b}_1 &= \frac{c+d}{2} \\
\sigma^2 &= \frac{(c-d)^2}{24}
\end{aligned}$$

Given the mean, $\bar{b}_1$, and squared coefficient of variation, cv^2, of the airport service times, we fit the uniform and triangular parameters as follows. For the uniform distribution, we find a and b in terms of $\bar{b}_1$ and cv^2:

$$\begin{aligned}
a &= \bar{b}_1(1 - cv\sqrt{3}) \\
b &= \bar{b}_1(1 + cv\sqrt{3})
\end{aligned}$$

Similarly, for the triangular distribution:

$$\begin{aligned}
c &= \bar{b}_1(1 - cv\sqrt{6}) \\
d &= \bar{b}_1(1 + cv\sqrt{6})
\end{aligned}$$

To apply the SPVA method, we use expression (2.5) to find the $\alpha_{n+1}(j)$'s for the uniform and triangular service-time distributions. Plugging in the appropriate service-time distribution in each case, we find the $\alpha_{n+1}(j)$'s to be:

Uniform Service Times

$$\alpha_{n+1}(j) = \frac{1}{(\lambda(t_n))(b-a)} \left\{ e^{-a\lambda(t_n)} \left[\sum_{i=0}^{j} \frac{(\lambda(t_n)a)^i}{i!} \right] - e^{-b\lambda(t_n)} \left[\sum_{i=0}^{j} \frac{(\lambda(t_n)b)^i}{i!} \right] \right\}$$

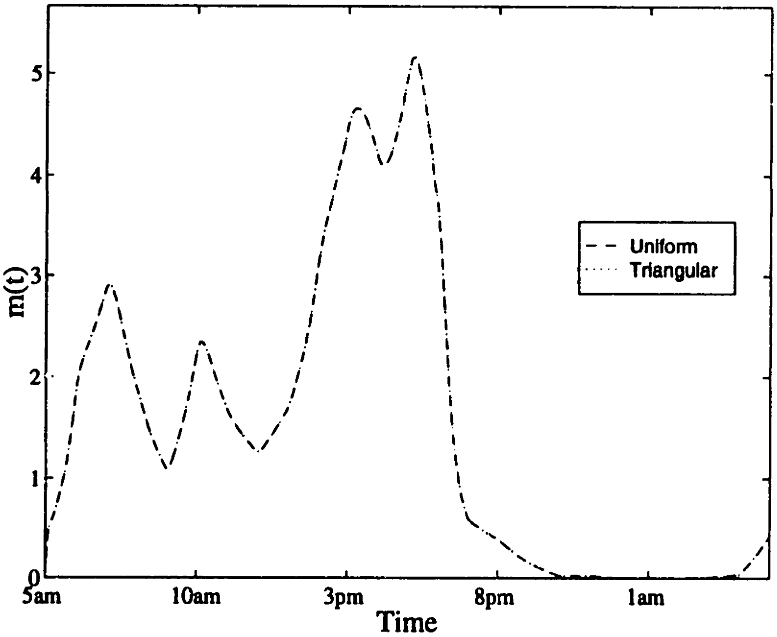


Figure 3-19: SPVA Approximation to Time-Dependent Mean of Uniform and Triangular Service-Time Distributions

Triangular Service Times

$$\begin{aligned}
\alpha_{n+1}(j) \quad = \quad & \frac{4}{((d-c)^2)(\lambda(t_n))} \cdot \\
& \left\{ e^{-(\frac{c+d}{2})\lambda(t_n)} \left\{ \left(c + d - 2\left(\frac{j+1}{\lambda(t_n)}\right)\right) \left[\sum_{i=0}^{j} \frac{(\lambda(t_n)\left(\frac{c+d}{2}\right))^i}{i!} \right] \right.\right. \\
& \qquad \left. -2 \left[\frac{(\lambda(t_n))^j \left(\frac{c+d}{2}\right)^{j+1}}{j!} \right] \right\} \\
& +e^{-d\lambda(t_n)} \left\{ \frac{(\lambda(t_n)^j c^{j+1}}{j!} + \left(\frac{j+1}{\lambda(t_n)} - d \right) \left[\sum_{i=0}^{j} \frac{(\lambda(t_n)d)^i}{i!} \right] \right\} \\
& \left. -e^{-c\lambda(t_n)} \left\{ - \left(\frac{(\lambda(t_n)^j c^{j+1}}{j!} \right) + \left(c - \frac{j+1}{\lambda(t_n)} \right) \left[\sum_{i=0}^{j} \frac{(\lambda(t_n)c)^i}{i!} \right] \right\} \right\}
\end{aligned}$$

We use the SPVA algorithm to find the time-dependent mean and standard deviation for the number in the system. We start the system from rest, and observe the transient behavior over one period, a day.

Figure 3-19 shows the time-dependent mean number in the system predicted by the SPVA approximation to the queueing systems with uniform and triangular service-time distributions for the stationary service-rate case. It is striking that the curves are indistin-

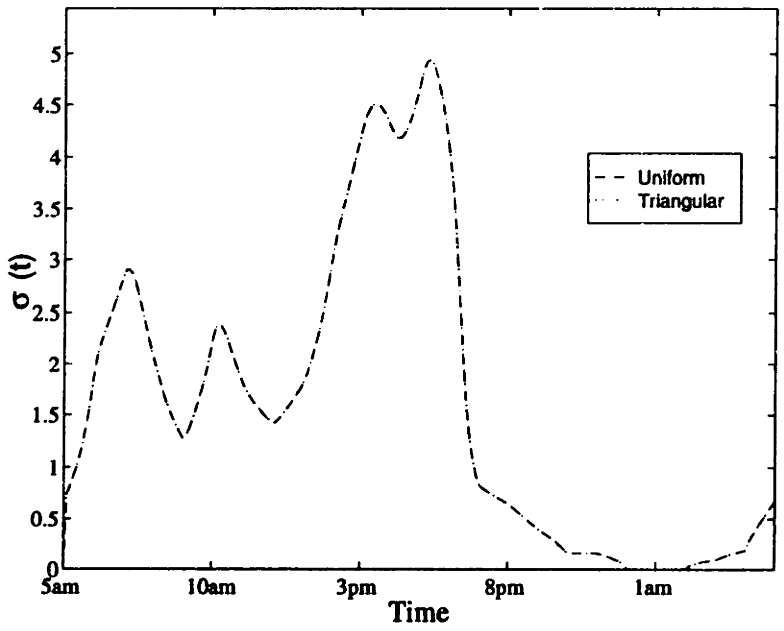


Figure 3-20: SPVA Approximation to Time-Dependent Standard Deviation of Uniform and Triangular Service-Time Distributions

guishable. They also behave as we expect queues with time-varying arrival rates to behave. The arrival rate to the system is as shown in Figure 3-17. There are four local peaks in the arrival rate over the 24 hours. Figure 3-19 also shows four local peaks in $m(t)$. They all occur after the corresponding peaks in the arrival rate.

Figure 3-20 shows the time-dependent standard deviation for the number in the system predicted by the SPVA approximation to the queueing systems with uniform and triangular service-time distributions. Again, the curves are indistinguishable. The standard deviation behaves as we expect, which in this case is similar to the mean. It has four local peaks, which occur after the four corresponding local peaks in the arrival rate.

SPVA gives estimates of $m(t)$ and $\sigma(t)$ which behave as we expect. However, we cannot make precise statements about SPVA quality in these cases without comparing these estimates to exact values. Since no techniques exist to model transient $M(t)/G(t)/1$ system behavior, these exact values must be found using simulation.

We now use SPVA to approximate $m(t)$ and $\sigma(t)$ for the same queueing systems with a time-varying service rate. The scenario is the same as in Section 3.5. Figure 3-21 shows the arrival rate and service capacity over the 24 hours, and the time-dependent mean number in the system predicted by the SPVA approximation to the uniform and triangular service-time systems. It also shows $m(t)$ predicted by the DELAYS and SPVA approximations

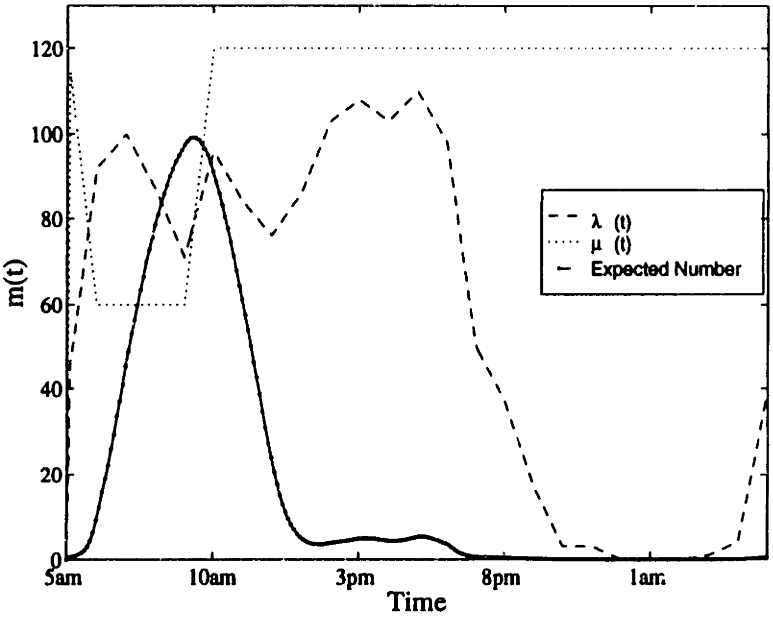


Figure 3-21: $m(t)$ of the $M(t)/E_9(t)/1$ System, the DELAYS Approximation to the $M(t)/E_9(t)/1$ System, and the SPVA Approximations to Systems with the 9th-order Erlang, Uniform and Triangular Service-Time Distributions

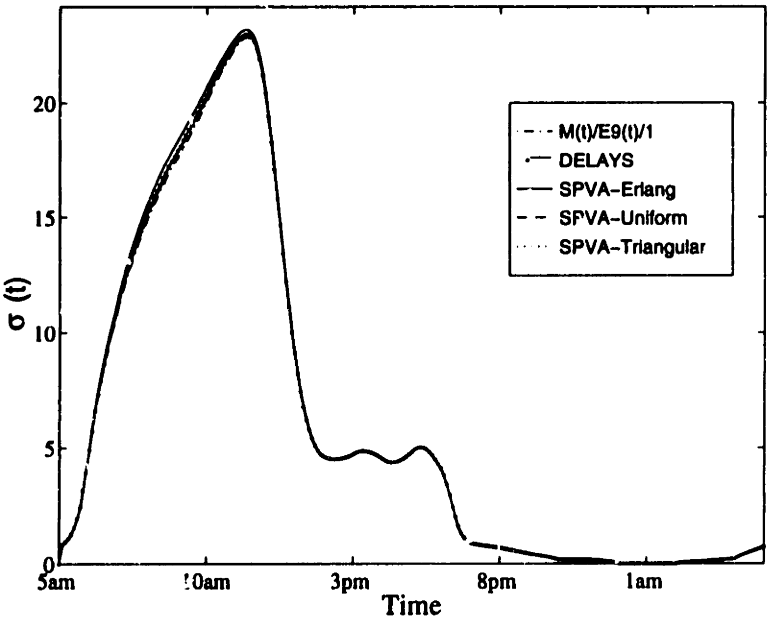


Figure 3-22: $\sigma(t)$ for $M(t)/E_9(t)/1$ system, the DELAYS Approximation to the $M(t)/E_9(t)/1$ System, and the SPVA Approximations to Systems with 9th-order Erlang, Uniform and Triangular Service-Time Distributions

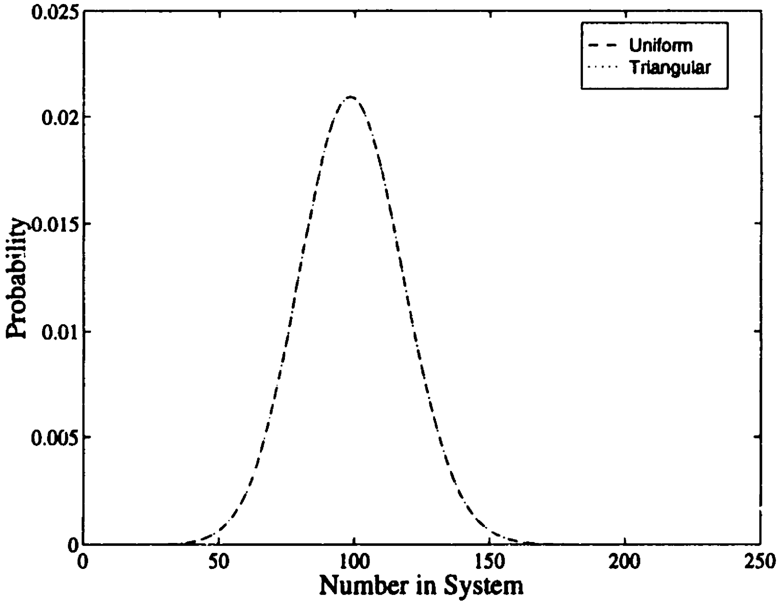


Figure 3-23: SPVA Approximation to Probability Distribution at Time of Maximum Congestion for Systems with Uniform and Triangular Service-Time Distributions

to the $M(t)/E_9(t)/1$ system, as well as the exact $M(t)/E_9(t)/1$ mean. These last three curves are repeated from Figure 3-17. One cannot distinguish the five curves from one another! This leads us to the following two hypotheses. First, the SPVA approximation for $M(t)/G(t)/1$ systems with uniform and triangular service-time distributions is good. It behaves as we expect. Second, the first two moments of a service-time distribution drive the behavior of $m(t)$. This hypothesis is the time-varying counterpart of the Pollaczek-Khintchin (PK) formula for $M/G/1$ systems. The PK formula for the expected number in the system contains only the first two moments of the service-time distribution, in addition to the arrival rate and system utilization.

Figure 3-22 shows $\sigma(t)$ for the number of customers in system, for the same airport scenario discussed above. Again, the behavior of the SPVA approximation to the uniform and triangular service-time systems is strikingly similar to that of the exact $M(t)/E_9(t)/1$ system.

Recall that SPVA estimates the entire probability distribution for number in system for $M(t)/G(t)/1$ systems. Figure 3-23 shows the SPVA approximation to the distributions for number in the system for the two $M(t)/G(t)/1$ systems at the time of maximum congestion for the second scenario. The distributions are virtually identical!

Up to now, few methods have been proposed to model transient or equilibrium behavior of $M(t)/G(t)/1$ systems. SPVA is such a method. It is easy to implement. The only customization necessary for a service-time distribution is that one must determine the form of the $\alpha_{n+1}(j)$'s to plug into the general SPVA algorithm. Once the $\alpha_{n+1}(j)$'s have been identified, the SPVA system is as easy to solve as an $M(t)/D(t)/1$ system. Furthermore, SPVA produces results for $M(t)/G(t)/1$ systems in seconds on a SUN SPARCstation 10 Model 41. This is three to five hundred times faster than solving the $M(t)/E_k(t)/1$ CK equations. In the case of $M(t)/G(t)/1$ systems with non-phase-type distributions, one must use simulation to find the "exact" solution, a time-consuming method. The complex definition of the $\alpha_{n+1}(j)$'s for $M(t)/G(t)/1$ systems with uniform and triangular service-time distributions does not have an impact on the CPU times of SPVA.

We do not compare the SPVA estimates of $m(t)$ and $\sigma(t)$ to simulation results. We leave this to the future, at which time we hope to examine SPVA accuracy for a wide variety of service-time distributions. The results of this section indicate, however, that SPVA gives estimates which we expect. The behavior of $m(t)$ and $\sigma(t)$ is consistent with other systems with the same mean and cv^2 for the service-time distribution. We conclude that SPVA has the potential to be a fast, flexible approximation to a broad range of $M(t)/G(t)/1$ systems.

3.6.2 SPVA Approximation to M(t)/H$_2$/1 Systems

This section examines the SPVA approximation to the $M(t)/H_2/1$ system. Why do we need an approximation for this system, which, after all, can be solved exactly by numerical methods? First, since we can solve the $M(t)/H_2/1$ system exactly, we can compare the results obtained through SPVA to the exact values and assess the quality of the approximation. Second, the hyperexponential distribution can be used to approximate a wide variety of service-time distributions with squared coefficient of variation greater than or equal to one. It is in some sense the "complement" of the Erlang distribution, which can approximate service-time distributions with squared coefficient of variation less than or equal to one. We assess the SPVA approximation to $M(t)/E_k(t)/1$ systems in Sections 3.4.2 and 3.5. Assessment of the SPVA approximation to the $M(t)/H_2/1$ system will then give an indication of SPVA's flexibility to approximate queueing systems with service-time distributions with coefficients of variation greater than or equal to one. Third, finding the exact solution to the $M(t)/H_2/1$ system requires a significant amount of computer memory and CPU time. It may not be practical to solve this system under some conditions, such as high squared coefficients of variation (memory constraints) or if the solution to the $M(t)/H_2/1$ system is not an end in itself but one of many steps to a solution of a larger problem (time constraints). The SPVA approximation to the $M(t)/H_2/1$ system uses far less memory and CPU time than the exact solution.

This section is organized as follows. We first describe the hyperexponential distribution and the $M(t)/H_2/1$ queueing system. We then present the test cases used to assess the quality of the SPVA approximation to the $M(t)/H_2/1$ system, followed by the results. Finally, we assess the SPVA approximation for the range of service-time coefficients of variation we examine in this thesis.

The hyperexponential distribution is a mixture of exponential distributions. We consider here a mixture of two exponentials, and denote this particular hyperexponential distribution as "H_2." H_2 has three parameters, $0 \leq p \leq 1$ and $\mu_1, \mu_2 \geq 0$. The probability density function, mean, second moment, and variance of the H_2 are:

$$\begin{aligned}
h(x) &= p\mu_1 e^{-\mu_1 x} + (1-p)\mu_2 e^{-\mu_2 x}, x \geq 0 \\
E[H] &= \frac{p}{\mu_1} + \frac{1-p}{\mu_2} \\
E[H^2] &= 2\left(\frac{p}{(\mu_1)^2} + \frac{1-p}{(\mu_2)^2}\right) \\
cv^2 &= \frac{E[H^2] - (E[H])^2}{(E[H])^2} = \frac{2\left(\frac{p}{(\mu_1)^2} + \frac{1-p}{(\mu_2)^2}\right) + \left(\frac{p}{\mu_1} + \frac{1-p}{\mu_2}\right)^2}{\left(\frac{p}{\mu_1} + \frac{1-p}{\mu_2}\right)^2}
\end{aligned}$$

H_2 has coefficient of variation greater than or equal to one.

We now show how we can use the H_2 to obtain a two-parameter fit to the first two moments of a service-time distribution. We assume balanced means, i.e., $\frac{p}{\mu_1} = \frac{(1-p)}{\mu_2}$. Given the mean, $\bar{b}_1$ and $cv^2 \geq 1$, then we can find p, μ_1 and μ_2 as in [52],

$$\begin{aligned}
p &= \frac{1 + \sqrt{\frac{cv^2-1}{cv^2+1}}}{2} \\
\mu_1 &= \frac{2p}{\bar{b}_1} \\
\mu_2 &= \frac{2(1-p)}{\bar{b}_1}
\end{aligned}$$

Intuitively, what does a H_2 service-time distribution mean in a queueing system context? It means that with probability p, a customer requires a service time distributed as a negative exponential random variable with mean $\frac{1}{\mu_1}$, and with probability $(1-p)$ a service time distributed as a negative exponential random variable with mean $\frac{1}{\mu_2}$. The $M(t)/H_2/1$ system would have the state-transition-diagram shown in Figure 3-24. [24]. If the system is in state s_i, then there are s customers in the system, and the customer currently in service has a service time distributed as a negative exponential random variable with mean $\frac{1}{\mu_i}$, $i = 1, 2$. The number of states needed to represent a system with queueing capacity c is $2c + 1$.

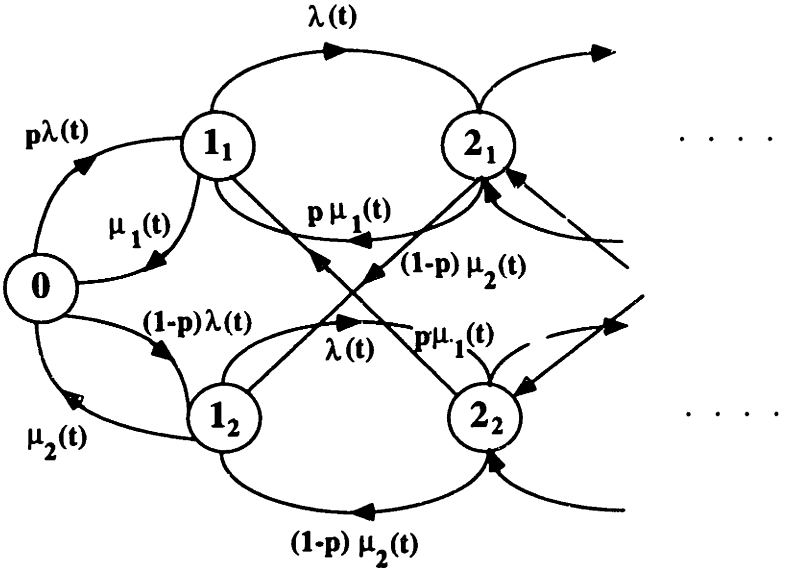


Figure 3-24: State-Transition-Rate Diagram for $M(t)/H_2(t)/1$

To apply the SPVA method, we use expression (2.5) to find the $\alpha_{n+1}(j)$'s for the $M(t)/H_2/1$ system. It is:

$$\alpha_{n+1}(j) = \lambda(t_n)^j \left[\frac{p\mu_1}{(\mu_1 + \lambda(t_n))^{j+1}} + \frac{(1-p)\mu_2}{(\mu_2 + \lambda(t_n))^{j+1}} \right]$$

We examine 13 test cases to obtain an indication of the quality of the SPVA approximation to the $M(t)/H_2/1$ system. We use a set of parameters similar to those described in Section 3.3. The difference is that we explicitly identify the coefficient of variation of H_2 as a parameter, instead of k for the Erlang distribution. Recall that $\frac{1}{k} = cv^2$. We keep the service rate stationary in order to gain insights into the complex behavior of queueing systems with nonstationary arrivals. For simplicity, we use a sinusoidal Poisson arrival process with amplitude A, similar to Green et al. [16]. The test cases are combinations of the following parameter values, covering moderate to heavy system utilization levels.

Case	cv^2	$\overline{\rho}$	$\rho_{\max}$	RA	μ	Queue Capacity
1	2	0.5	1.0	1.0	100	600
2	2	0.75	1.125	0.5	100	1200
3	2	0.9	1.35	0.5	100	1200
4	2	0.9	1.20	0.33	100	1200
5	2	0.5	1.0	1.0	10	1200
6	2	0.75	1.125	0.5	10	1200
7	2	0.9	1.35	0.5	10	1200
8	10	0.5	1.0	1.0	100	2000
9	10	0.75	0.5	0.125	100	2000
10	10	0.9	1.20	0.33	100	2000
11	10	0.5	1	1	10	2000
12	10	0.75	1.125	0.5	10	2000
13	10	0.9	1.35	0.5	10	2000

Table 3.7: Test Case Parameters for Approximations to $M(t)/H_2/1$ Systems

- Poisson Arrival Function: $\lambda(t) = \overline{\lambda} + A \sin\left(\frac{2\pi t}{24}\right)$. Since $\lambda(t) \geq 0$, we restrict A to be $0 \leq A \leq \overline{\lambda}$. Note that $\lambda(t)$ is a smooth differentiable function with one peak over each period.

- Average Utilization ($\overline{\rho}$) ranges from moderately- to heavily-loaded systems: 0.5, 0.75, 0.9

- Maximum Utilization: $1.0 \leq \rho_{\max} \leq 1.35$.

- Degree of Nonstationarity: $RA = \frac{1}{3}, \frac{1}{2}, 1$. In the case of our sinusoidal arrival function, $RA = \frac{A}{\overline{\lambda}}$. Note that $0 \leq RA \leq 1$ in this case.

- Event Frequency: high and moderate: $\mu = 100, 10$.

- Squared Coefficient of Variation: moderate and high: $cv^2 = 2, 10$.

Table 3.7 lists the parameters for each test case, along with an assigned case number. We use the case number for convenience. The last column of Table 3.7 lists the queueing capacity for each case. These capacities are large from a practical point of view. However, some cases, e.g., Cases 9 and 10, do not meet our criteria for infinite queueing-capacity systems, discussed in Section 3.2.2.

Table 3.8 lists the CPU times for the exact solution to the $M(t)/H_2/1$ system and for the SPVA approximation. The exact system requires 4 to 70 times as much CPU time to solve than SPVA does. It is striking that SPVA requires significantly more time to

Case	$M(t)/H_2/1$	SPVA
1	412.1	56.8
2	864.4	133.5
3	895.3	153.0
4	899.1	148.8
5	785.3	11.0
6	786.8	13.3
7	753.3	15.6
8	1,612.4	408.5
9	1,682.3	554.3
10	1,764.5	658.3
11	1,307.1	39.9
12	1,320.3	55.0
13	1,300.7	65.2

Table 3.8: CPU Times on SUN SPARCStation 10 Model 41 for Exact $M(t)/H_2/1$ System and SPVA Approximation

approximate the $M(t)/H_2/1$ systems than the $M(t)/E_k/1$ systems. For example, compare Case 9 of Table 3.3 with Case 8 of Table 3.8. The former case requires 4.1 seconds of CPU time whereas the latter requires 408.5. Why is there such a large difference in the SPVA CPU times between the two systems? In both cases, $\overline{\rho} = 0.5$, $\rho_{\max} = 1.0$, RA=1.0, and $\overline{\mu} = 100$. The difference between the two cases is the coefficient of variation: it is $\frac{1}{10}$ in the former, and 10 in the latter. There are on average more customers in the $M(t)/H_2/1$ system than in the $M(t)/E_k/1$ system with the same $\overline{\rho}$, $\rho_{\max}$, RA, and $\overline{\mu}$. The probability distribution for the number in the system has a significant probability mass in high-index states. SPVA calculates the probability distribution from state 0 to the highest index state such that the cumulative probability is greater than a threshold value, and that probability mass is greater than a small positive number ϵ. Hence, SPVA solves more equations at each iteration to approximate the $M(t)/H_2/1$ system than it does to approximate the $M(t)/E_k/1$ system, resulting in higher CPU time.

Table 3.9 lists the SPVA approximation errors, as well as m^* and σ^*. In general, it appears that SPVA errors are larger for the $M(t)/H_2/1$ systems than for the $M(t)/E_k/1$ systems. Seven of the cases have $cv^2 = 2$, and six have $cv^2 = 10$. In the cases in which $cv^2 = 2$, the worst errors occur in Cases 5 and 7, both of which have the moderate event frequency. In Case 5, the WPE of mean is 6.8 and the RE of RE of m^* is 6.9%. In Case 7, the WPE of σ is 11.5 and the RE of RE of σ^* is 11.3%. These errors are not

Case	WPE of Mean	WPE of σ	m^*	RE of m^*	σ^*	RE of σ^*
1	2.40	4.08	23.76	2.45%	20.81	3.72%
2	1.81	5.41	75.87	1.48%	42.94	4.59%
3	0.78	11.70	250.78	0.40%	65.19	7.78%
4	1.12	6.83	143.95	0.73%	58.00	5.31%
5	6.81	10.70	8.56	6.89%	8.06	10.35%
6	5.36	9.90	15.33	5.53%	12.08	10.27%
7	4.68	11.53	34.87	3.49%	20.34	11.27%
8	6.68	10.55	46.27	6.19%	44.66	9.08%
9	5.10	10.69	103.24	4.61%	74.89	9.79%
10	3.13	9.13	175.21	2.66%	103.51	9.17%
11	21.34	29.41	13.6	20.90%	15.75	29.97%
12	9.91	15.92	24.96	11.51%	25.09	19.65%
13	11.08	19.03	54.67	9.59%	44.08	19.91%

Table 3.9: Errors of SPVA approximation of $M(t)/H_2/1$ Systems

large, but they are larger than those seen for the $M(t)/E_k/1$ system. Of the cases with $cv^2 = 10$, the largest errors occur in Case 11, which has moderate event frequency. In this case, WPE of mean and of σ are 21.3 and 29.4, respectively, and RE of m^* and of σ^* are 20.9% and 30%, respectively. It appears that increasing cv^2 contributes to increased SPVA approximation error. However, Case 8, with $cv^2 = 10$ and $\mu = 100$, has errors in the range of 6-10%. Although these are outside our range of good (less than 5%), they are not bad. We hypothesize that for cases with high cv^2, combinations of moderate ρ, moderate RA, and high frequency, or high ρ and low to moderate RA, SPVA will yield estimates of $m(t)$ and $\sigma(t)$ in the 6-10% range. We hypothesize that for moderate cv^2, combinations of moderate ρ, moderate RA, and high frequency, or high ρ and low to moderate RA, SPVA will yield good estimates of $m(t)$ and $\sigma(t)$.

Table 3.10 bears out SPVA's trend for increasing error with increasing cv^2. We compare two cases in which we allows the cv^2 to vary from $\frac{1}{10}$ to 10. We examined these cv^2's for the $M(t)/E_k/1$ and $M(t)/H_2/1$ systems. The two cases have $\bar{\rho}$, $\rho_{\max}$, RA and event frequency held constant. The four error measures, WPE of mean and of σ, and RE of m^* and of σ^* increase monotonically with increasing cv^2. The increase in error is approximately linear for the cases shown. WPE of σ and RE of σ^* increase faster than their equivalents for the mean, but not significantly.

Figure 3-25 plots the time-dependent mean number in the system for the exact system and the SPVA approximation to the $M(t)/H_2/1$ system. SPVA overestimates $m(t)$ as it

Case Parameters				Error	cv^2					
$\overline{\rho}$	$\rho_{\max}$	RA	μ	Measure	$\frac{1}{10}$	$\frac{1}{6}$	$\frac{1}{3}$	1	2	10
0.5	1	1	100	WPE of Mean	0.18	0.26	0.53	1.43	2.42	6.68
				WPE of σ	0.27	0.44	0.93	2.40	4.08	10.55
				RE of m^*	0	0	0.65	1.56	2.45	6.19
				RE of σ^*	0	0.83	1.55	2.45	3.72	9.08
0.5	1	1	10	WPE of Mean	1.09	1.08	1.38	3.70	6.81	21.34
				WPE of σ	0.89	1.20	2.15	6.00	10.70	29.41
				RE of m^*	0	0.90	1.64	4.11	6.89	20.90
				RE of σ^*	0	0	1.92	6.25	10.35	29.97

Table 3.10: Trend of Increasing SPVA Estimation Error with Increasing cv^2, with All Other Parameters Held Fixed

grows to its peak. The SPVA estimate of $m(t)$ drops off more quickly than the exact curve after the peak. The SPVA curve has approximately the correct shape, and peaks at approximately the same time as the exact system.

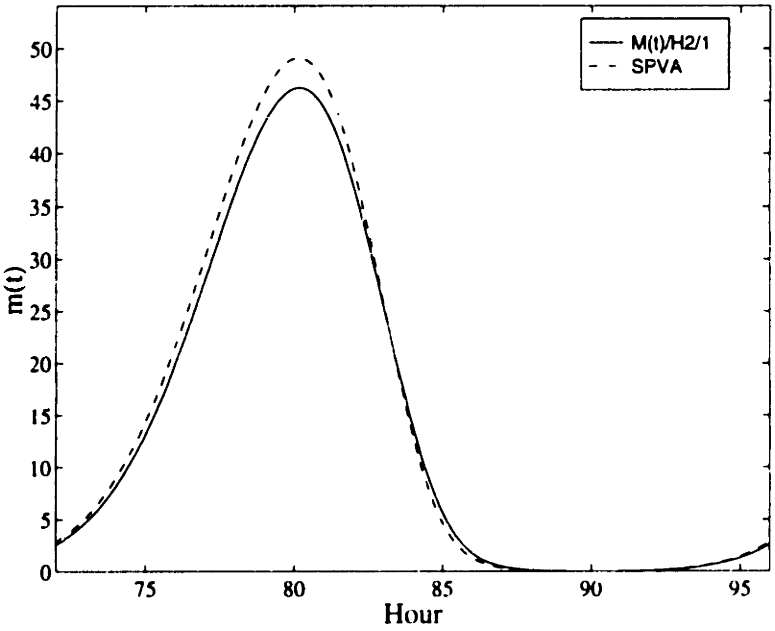


Figure 3-25: Expected Number in the $M(t)/H_2/1$ System Over One Period. Case 8.

Figure 3-26 plots the time-dependent standard deviation for the number in system for the exact system and the SPVA approximation to $M(t)/H_2/1$ system. SPVA overestimates $\sigma(t)$ as it grows to its peak. After the peak, though, the SPVA estimate of $\sigma(t)$ drops off more quickly than the exact curve after the peak. This behavior is similar to the SPVA estimate of $m(t)$. SPVA gives better estimates of $m(t)$ than of $\sigma(t)$.

What can we conclude about the SPVA approximation to $M(t)/H_2/1$ systems? SPVA

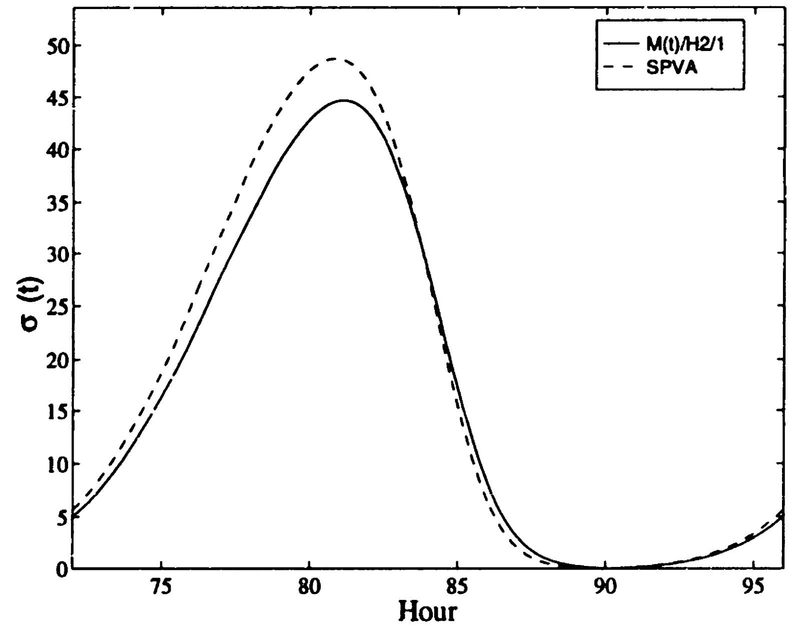


Figure 3-26: Standard Deviation for the Number in the $M(t)/H_2/1$ System Over One Period. Case 8.

can be used to give good estimates of $m(t)$, $\sigma(t)$, m^* and σ^* for $M(t)/H_2/1$ systems in many cases. The primary parameter which determines SPVA accuracy is cv^2. With all other parameters fixed, SPVA error increases with increasing cv^2. We already know from Section 3.4.2 that as cv^2 increases, decreases in $\overline{\rho}$ and RA, and high event frequency may keep the error from growing. We see that even for extremely high $cv^2 = 10$, SPVA still gives estimates of $m(t)$, $\sigma(t)$, m^* and σ^* with errors in the 6-10% range. We note that $\rho_{\max} \geq 1$ in every case we examined. This indicates that we stressed the SPVA approximation, and it still gave good results. Furthermore, SPVA requires significantly less CPU time than solving the exact $M(t)/H_2/1$ system. SPVA also requires half as much memory to represent a system with queueing capacity c as does the exact system.

3.6.3 A Comparison With Other Researchers' Methods

In this section, we compare the SPVA method to Surrogate Distribution Approximation (SDA) methods. We briefly described the methods in Section 1.3. Here we supplement Section 1.3 by giving more details of the SDA methods, then compare the SPVA and SDA in two test cases. In both cases, we approximate $M(t)/M(t)/1$ systems. The first test case is one that appears in Rider [40] and Rothkopf and Oren [46]. They compare the exact time-dependent mean number in the system to that predicted using Rider's and Rothkopf and Oren's methods. We extend the comparison by examining the time-dependent standard deviation for the number in the system. We also compare the exact probability distribution for the number in the system to that predicted by the SDA method of Rothkopf and Oren and to the SPVA method. The second case we examine appears in Clark [7]. Clark compares his method to the exact solution and to the solution using Rothkopf and Oren's method. We compare the SPVA time-dependent mean and variance for the number in the system in a case appearing in Clark's paper.

To avoid confusion, we refer to Rothkopf and Oren's SDA method as "SDA-RO", to Rider's as "SDA-R", and to Clark's as "SDA-C."

The general SDA method is based on the differential equations for the time-dependent mean number in the system, $m(t)$, its second moment, $m_2(t)$, and variance, $v(t)$, initially developed by Clarke [8] for the $M(t)/M(t)/1$ system. These *moment differential equations* (MDE's) can be carefully written for any $Ph(t)/M(t)/s/c$ or $Ph(t)/Ph(t)/1/c$ system. The exact form of the MDE's depend on the system approximated and the definition of the state space. SDA methods also differ from one another in which surrogate distribution is used to estimate the time-dependent probabilities which appear in the MDE's.

Comparison of SPVA and Rider's and Rothkopf and Oren's SDA Methods

The MDE of the SDA-R method is the time-dependent mean for the number in system. The MDE's of the SDA-RO method are for the time-dependent mean and variance for the number in the system. For the $M(t)/M(t)/1$ system, these are:

$$\begin{aligned} m'(t) &= \lambda(t) - \mu(t)(1 - P_0(t)) \qquad (3.4) \\ v'(t) &= \lambda(t) + \mu(t) - \mu(t)P_0(t)(2m(t) + 1). \qquad (3.5) \end{aligned}$$

The probability needed to close the above system is $P_0(t)$. This is the key to the SDA methods: estimate the unknown quantities appearing in the MDE's using a surrogate distribution. Then, solve the MDE's to find updated values for $m(t + \Delta t)$ and $v(t + \Delta t)$, and iterate. SDA-R estimates this quantity by modifying the exact transient solution for P_0 for stationary systems. SDA-R simplifies the exact expression by using the current value of the mean number in the system to estimate future values of $P_0(t)$. In contrast, SDA-RO estimates $P_0(t)$ in terms of $m(t)$ and $v(t)$ using the negative binomial distribution. The negative binomial distribution has two parameters, r and p, and is fully described by its mean, $\frac{r(1-p)}{p}$ and variance, $\frac{r(1-p)}{p^2}$:

$$P_n = \begin{pmatrix} r+n-1 \\ n \end{pmatrix} p^r(1-p)^n \tag{3.6}$$

Specifically, $P_0 = p^r$. The SDA-RO method matches the time-dependent mean and variance of the number in the system to the moments of the negative binomial to find estimates of p and r, and thus $P_0(t)$: $p = \frac{m(t)}{v(t)}$, $r = \frac{m(t)^2}{v(t)-m(t)}$, and $P_0(m(t), v(t)) = p^r$. We solve for $m(t)$ and $v(t)$ using the general SDA algorithm described in Section 1.3.

The test case in which we compare SPVA with SDA-R and SDA-RO is an $M(t)/M(t)/1/30$ system in equilibrium. Both the arrival and service rates vary with time over a 24-hour period. The data are rates per hour. The arrival rate is interpolated between hours, and the service-rate changes as a step-function. When we carry out the computation, we allow the system to run for 96 hours to achieve equilibrium. Table 3.11 lists the hourly arrival and service rates, the exact values of $m(t)$, and approximations for $m(t)$. SDA-R has two sets of approximation values for this case, depending on how the method's parameter, T, is set. We list the one for $T = \frac{0.4}{\mu(t)}$, which is the more accurate one. Table 3.11 also shows the equilibrium solution for an $M/M/1$ system, applied hour-by-hour to this case. It performs poorly in comparison to the other methods. Only when the system operates for a relatively long period of time with the same parameters does the PSA give good estimates of $m(t)$.

SPVA, SDA-RO, and SDA-R give good estimates of $m(t)$. The maximum absolute difference between the exact and each approximation method over the 24 hours is 0.10 for SDA-RO, 0.22 for SPVA, and 0.23 for SDA-R. The WPE of mean is 1.48 for SDA-RO, 2.46 for SPVA, and 2.87 for SDA-R. These errors are small. SDA-RO gives slightly better estimates of $m(t)$ than do SPVA and SDA-R. Figure 3-27 plots the time-dependent mean number of customers over the period. The differences between the methods are negligible. The three methods approximate the time-dependent behavior of $m(t)$ well.

Table 3.12 lists the SPVA and SDA-RO estimates of the standard deviation for the number in the system for the $M(t)/M(t)/1$ case. We do not have SDA-R estimates of the standard deviation in this case. Table 3.12 shows that SPVA and SDA-RO give comparable

Hour	Data		Expected Number in System				
	Arrival Rate	Service Rate	Exact	Steady State Approx.	Rider	Rothkopf & Oren	SPVA
1	12.0	15.0	4.52	4.00	4.65	4.47	4.62
2	10.0	13.0	3.63	3.33	3.86	3.67	3.68
3	7.0	13.0	2.66	1.17	2.80	2.76	2.67
4	5.0	13.0	1.29	0.63	1.15	1.39	1.20
5	4.5	7.0	0.69	1.80	0.58	0.66	0.64
6	4.5	7.0	1.39	1.80	1.28	1.33	1.46
7	5.0	10.0	1.74	1.00	1.67	1.69	1.80
8	5.5	10.0	1.28	1.22	1.28	1.28	1.26
9	6.5	13.0	1.49	1.00	1.46	1.49	1.50
10	7.0	13.0	1.17	1.17	1.16	1.18	1.15
11	7.0	15.0	1.17	0.88	1.16	1.17	1.17
12	7.0	15.0	0.90	0.88	0.89	0.90	0.89
13	7.0	15.0	0.88	0.88	0.88	0.88	0.88
14	7.0	15.0	0.88	0.88	0.88	0.88	0.88
15	7.0	8.0	0.88	7.00	0.88	0.88	0.88
16	7.0	8.0	2.40	7.00	2.27	2.34	2.61
17	7.0	12.0	3.14	1.40	3.05	3.06	3.37
18	10.0	12.0	2.81	5.00	2.84	2.81	2.87
19	10.0	12.0	3.41	5.00	3.44	3.40	3.53
20	10.0	15.0	3.77	2.00	3.83	3.74	3.89
21	10.0	15.0	2.70	2.00	2.83	2.77	2.70
22	13.0	15.0	3.18	6.50	3.12	3.22	3.18
23	13.0	15.0	4.03	6.50	3.98	3.99	4.11
24	13.0	15.0	4.53	6.50	4.54	4.47	4.63

Table 3.11: Comparison of SPVA and SDA Methods: Estimates of $m(t)$.

Hour	Data		Standard Deviation for Number in the System		
	Arrival Rate	Service Rate	Exact	Rothkopf & Orer.	SPVA
1	12.0	15.0	4.69	4.91	4.82
2	10.0	13.0	4.36	4.55	4.39
3	7.0	13.0	3.73	3.83	3.69
4	5.0	13.0	2.46	2.34	2.26
5	4.5	7.0	1.41	1.13	1.23
6	4.5	7.0	1.77	1.66	1.84
7	5.0	10.0	2.05	2.01	2.14
8	5.5	10.0	1.78	1.74	1.74
9	6.5	13.0	1.87	1.86	1.89
10	7.0	13.0	1.63	1.61	1.59
11	7.0	15.0	1.60	1.59	1.60
12	7.0	15.0	1.34	1.32	1.31
13	7.0	15.0	1.29	1.28	1.29
14	7.0	15.0	1.28	1.28	1.28
15	7.0	8.0	1.28	1.28	1.28
16	7.0	8.0	2.36	2.43	2.70
17	7.0	12.0	3.01	3.14	3.37
18	10.0	12.0	3.04	3.13	3.20
19	10.0	12.0	3.49	3.62	3.69
20	10.0	15.0	3.85	4.00	4.04
21	10.0	15.0	3.39	3.48	3.40
22	13.0	15.0	3.44	3.46	3.49
23	13.0	15.0	4.02	4.14	4.16
24	13.0	15.0	4.47	4.65	4.63

Table 3.12: Comparison of SPVA and SDA Methods: Estimates of $\sigma(t)$.

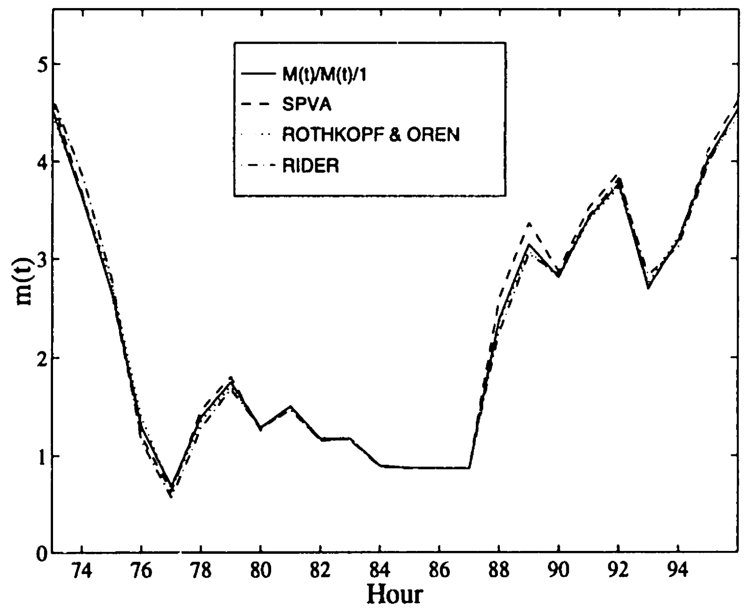


Figure 3-27: Comparison of SPVA and SDA Estimates of the Time-Dependent Expected Number in the System.

estimates of $\sigma(t)$. The estimates of $\sigma(t)$ are not as good as those for $m(t)$. The maximum absolute difference between the exact and approximation methods is 0.36 for SPVA, and 0.29 for SDA-RO. The WPE of mean for SPVA is 3.94, and for SDA-RO is 3.38. Figure 3-28 shows the time-dependent estimates of the standard deviation for the number in the system. Both SPVA and SDA-RO over- and underestimate $\sigma(t)$ during some time intervals. The approximations are good.

Figure 3-29 shows the probability distribution for the number of customers in the system predicted by SPVA and SDA-RO. This probability distribution is at hour 96, the hour at which the system achieves the maximum number of customers in the system. The SPVA method must estimate this distribution at each time step. SDA-RO does not; it needs only the value of $P_0(t)$ at each step in the algorithm. However, we use the negative binomial distribution from expression (3.6) to estimate the entire distribution for SDA-RO. The difference between the exact and SPVA estimate of the probability distribution can barely be seen. SPVA gives a better estimate of the probability distribution than does SDA-RO. However, the difference between the SPVA and SDA-RO probability distributions is small. SDA-RO's estimate is good, although it overestimates the probability mass in the lower index states, and underestimates it in the higher index states. SDA-RO's estimate of $P_0(t)$ appears to be virtually exact. This, of course, is the key to the method.

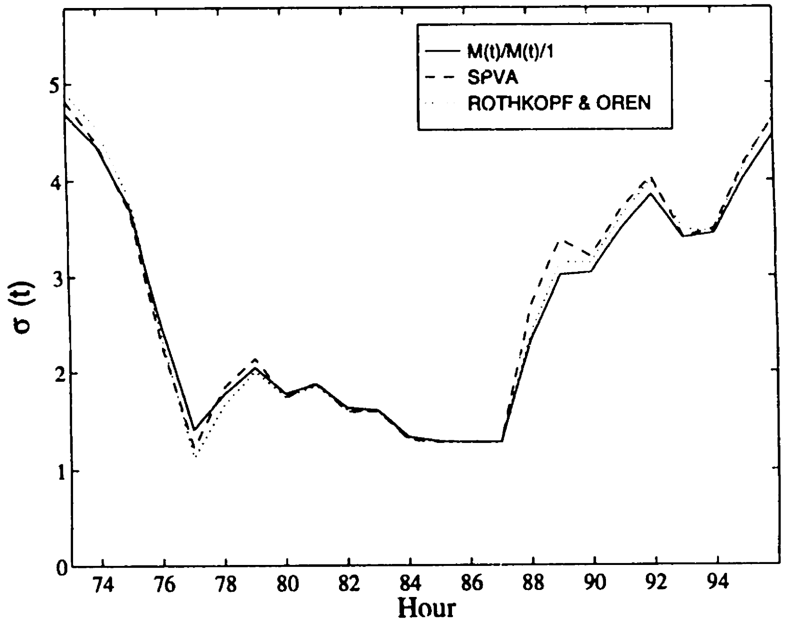


Figure 3-28: Comparison of SPVA and SDA-RO Estimates of the Time-Dependent Standard Deviation for the Number in the System.

We now compare the CPU times of the SPVA and SDA-RO methods. SDA-RO has fewer numerical computations than SPVA does. We do not expect that SPVA will be faster than SDA-RO. This is indeed the case, but the difference is smaller than expected. When we run the above test case and do not calculate the entire probability distribution for SDA-RO, it requires 0.14 seconds of CPU time. SPVA requires 0.35 seconds, and the exact method 2.02 seconds. SDA-RO is 2.5 times faster than SPVA in this case. When we do calculate the entire probability distribution for SDA-RO, it uses 0.32 seconds. In this case, the difference in CPU time between SPVA and SDA-RO is negligible.

In conclusion, SPVA gives estimates of $m(t)$ and $\sigma(t)$ comparable to the SDA methods of Rider and Rothkopf and Oren. If the entire probability distribution for the number in the system is not needed, SDA-RO is approximately 2.5 times faster than SPVA in the case we examine. In cases in which there is a very large number of customers in the system, there may be a larger difference in the CPU times. However, if the entire probability distribution for the number in the system is needed, the speed advantage of SDA-RO diminishes greatly.

Rothkopf and Oren mention that errors occasionally build up during the convergence to periodic limiting results. When working with the SDA-RO method, this did occur in some of the preliminary test cases we examined. However, we note that we wrote the SDA-RO program ourselves. It may not be identical to the program written by the original

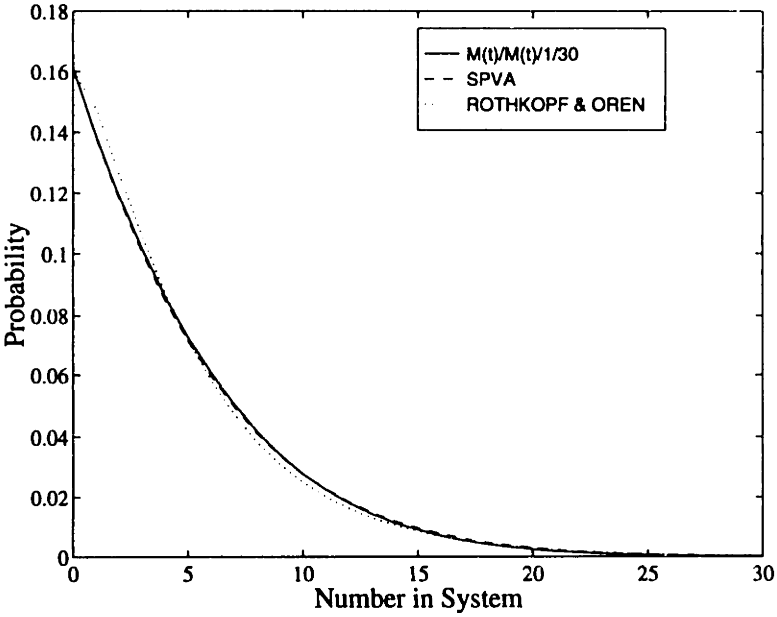


Figure 3-29: Comparison of SPVA and SDA-RO Probability Distributions for the Number in the System.

SDA-RO method programmers. However, our computational results for SDA-RO for the test case described in this section match those published by Rothkopf and Oren extremely well. Therefore, it is likely that our program is correct. Rothkopf and Oren note that this buildup cannot occur if sometime during the period the system empties of customers. We contrast this with SPVA. SPVA behaves in a stable manner in all the computational results presented in this chapter. We examine cases in which the queue both empties and does not empty during each period in Section 3.4.2. Some of the cases in which the queue does not empty have extremely high average utilization ($\overline{\rho} = 0.9$) and maximum utilization ($\rho_{\max} > 1$). In one case, the mean number in the system ranges from 2 to 140. SPVA is stable even under these extreme conditions.

Comparison of SPVA with Clark's and Rothkopf and Oren's SDA Methods

In this section, we compare the SPVA method to SDA-C and SDA-RO. The case we compare is one which appears in [7]. It is for an $M(t)/M(t)/1$ system. We begin with a brief description of the SDA-C method.

The SDA-C method is a refinement of Rothkopf and Oren's SDA method for $M(t)/M(t)/s$ systems. SDA-RO requires the estimation of s probabilities: $P_0(t)$, $P_1(t), \ldots, P_{s-1}(t)$ (see expressions 1.1 and 1.2). The SDA-C methods differs from SDA-RO in three ways. First, the surrogate distribution for the number of customers in the system is represented by two conditional distributions. These conditional distributions depend on whether there are less than or equal to s customers in the system, or greater than s customers in the system. These conditional distributions reflect the form of the exact steady-state distribution for the number of customers in an $M/M/s$ system: the form of P_i depends on whether $i \leq s$ or $i \geq s$. The second difference between SDA-C and SDA-RO is the surrogate distribution used to approximate the probabilities which appear in the MDE's. The SDA-C method uses the Polya-Eggenberger (PE) distribution instead of the negative binomial. Third, the SDA-C method requires the solution of five MDE's, instead of just two in the SDA-RO method. However, these MDE's require the estimation of only two probabilities: $P_s(t)$ and $P_{s+1}(t)$.

The test case used to compare SPVA with SDA-C and SDA-RO has a sinusoidal-Poisson arrival rate with parameter $\lambda(t) = 1 + \sin(\frac{2\pi}{24})$. The service rate is constant: $\mu = 1.67$. Note that $\rho_{\max} > 1$. This is the first test of SPVA with service rate on the order of one. We compare $m(t)$ and $v(t)$ over the period for the SPVA, SDA-RO and SDA-C methods. The actual numerical results unfortunately do not appear in [7]. Therefore, we estimate the errors from the hand-drawn graphs which appear in [7].

The time-dependent mean in the case which we test ranges from a minimum of 0.4 to a maximum of 4.9 over the period. In Figure 3 of [7], the maximum absolute error of SDA-C is approximately 0.05, which is 3% of $m(t) \approx 1.5$, and that of SDA-RO is 0.22, which is 28% of $m(t) \approx 0.8$. SPVA's maximum error is 0.53, which is 13% of $m(t) \approx 4$. SDA-C and SDA-RO both over- and underestimate $m(t)$ over the period. SDA-RO's absolute error is greatest when $m(t)$ is at its minimum. SPVA overestimates $m(t)$ over the period. Its maximum absolute error occurs just after the peak in $m(t)$.

The time-dependent variance ranges from a minimum of 1.3 to a maximum of 16.3 over the period. In Figure 6 of [7], the maximum absolute error of SDA-C is approximately 1, corresponding to 10% of $v(t) \approx 10$, and that of SDA-RO is 4, which is 40% of $v(t) \approx 10$. SPVA's maximum error is 5.4, which is 35% of $v(t) \approx 15$.

In this case, SDA-C gives better estimates of $m(t)$ and $v(t)$ than do SPVA or SDA-RO. However, the magnitude of the absolute errors is small in this case. In all the tests which appear in [7], SDA-C gives better estimates for $m(t)$ and $v(t)$ than does SDA-RO. SPVA's errors are on the same order as those of SDA-RO. However, results from Section 3.6.2

indicate that SPVA performs less well for low event frequencies, which is the case here. This may explain why SPVA does not perform as well as SDA-C.

Summary of SPVA and SDA Methods

We have compared SPVA to three approximation methods. In our test cases, SPVA gives estimates of $m(t)$ which are as good as those of the SDA-R and SDA-RO methods, and of $\sigma(t)$ which are as good as the SDA-RO method. SDA-C gives better estimates of $m(t)$ and $v(t)$ than do SDA-RO and SPVA. However, the magnitude of the absolute errors for $m(t)$ is small. The absolute errors of $v(t)$ are larger than for $m(t)$. However, the test case for comparing SDA-C and SPVA is for a low event frequency. We hypothesize that SPVA performs less well for systems with low event frequencies than for moderate or high event frequencies.

The SDA methods require less CPU time than SPVA. In our test case, the SPVA method required 2.5 times as much CPU time as did SDA-RO. However, if the probability distribution for the number in the system is needed, the speed advantage of the SDA methods diminish greatly.

All SDA methods solve a constant number of MDE's for a given system at each time step, regardless of system queueing capacity. In contrast, the number of equations SPVA solves at each time step varies. The number depends on the number of customers in the system. More equations must be solved during periods of high traffic intensity than during periods of low traffic intensity.

SPVA is a more general method than the SDA methods examined. It can be used to approximate $M(t)/G(t)/1$ systems. Although it is not specifically designed to approximate $M(t)/M(t)/1$ systems like the SDA methods examined, it does give good estimates of $m(t)$ and $\sigma(t)$ for such systems.

SPVA is a stable method. Both Clark and Rothkopf and Oren mention that cases exist in which there is a buildup of error as the system evolves to its equilibrium (time-dependent) solution.

SDA methods exist for systems with more complicated phase-type arrival and service-time distributions. Specifically, Taaffe and Ong [50] and Ong and Taaffe [36] developed SDA methods for the $Ph(t)/Ph(t)/1/c$ and $Ph(t)/M(t)/s/c$ queueing systems. In these cases, the number of MDE's which must be solved depend on the the number of phases in the arrival and service-time distributions. Most importantly, this number is independent of system capacity, i.e., it is fixed. The probabilities which must be estimated to close the MDE's depend on the definition of the state space. To apply the SDA methods, one must carefully describe the state space, and then write the MDE's for the state space. The set of equations differs from system to system, and can be complicated. SPVA, in contrast to the SDA methods, is simple to implement. The only customization necessary to model an $M(t)/G(t)/1$ system is to find the $\alpha_{n+1}(j)$'s. Appendix A shows the $\alpha_{n+1}(j)$'s for some service-time distributions. Once the form of the $\alpha_{n+1}(j)$'s is known, the method is as simple to implement as an $M(t)/D(t)/1$ system. We intend to compare the SPVA approximation to the SDA methods of Ong and Taaffe for the $M(t)/E_k(t)/1$ and $M(t)/H_2(t)/1$ systems in the future.

3.6.4 The Choice of β

Recall that β appears in the SPVA expression (2.3) which determines the time of the n^{th} customer pseudo-departure epoch. In all the computational results presented in Sections 3.4.1 and 3.5, we use $\beta = 1$. This means that the SPVA time step equals the expected value of the service time. One might reason, though, that a more judicious choice of β may exist which accounts for the second moment of the service-time distribution. In this section, we test different values of β which account for the 2^{nd} moment of the service time.

To test the effect of the choice of β, we experiment with one of SPVA's "worst case" approximations to the $M(t)/E_k/1$ system: Case 58, with parameters $k = 1$, $\overline{\rho} = 0.90$, $RA = \frac{2}{3}$, and $\mu = 100$. We choose a worst case because we expect the benefit of changing the stepsize, if any, will manifest itself here.

Table 3.13 shows the SPVA estimates and errors for m^*, $m(t)$, σ^*, and $\sigma(t)$ for values of β both smaller and larger than 1. Cases in which $\beta \neq 1$ yield results significantly worse than those for $\beta = 1$. We can possibly explain this result by noticing that SPVA already accounts for the entire service-time distribution in the calculation of the $\alpha_{n+1}(j)$ in expression (2.5). Since we obtain the best results using $\beta = 1$ for Case 58, we conjecture that $\beta = 1$ is the best choice for all cases.

Case	measure	$M(t)/M/1$	SPVA $\beta = 1$	SPVA β=2	SPVA $\beta = 1.6$	SPVA $\beta = 0.5$
Case 58	m^*	357.97	358.51	183.43	227.30	706.78
	% diff in m^*		0.15%	−48.76%	−36.50%	97.44%
	WPEMean		0.33%	47.49%	35.48%	94.65%
	σ^*	57.12	61.38	43.02	48.25	87.37
	% diff in σ^*		7.46%	−24.68%	−15.53%	52.96%
	WPEσ		12.24%	21.39%	12.04%	58%

Table 3.13: SPVA Approximation Quality with various β's

3.7 Summary

In this chapter, we examined approximation methods for queueing systems with time-varying arrival and service rates. We investigated the SPVA, DELAYS, and INTERP methods. All three methods generate the probability distribution for the number of customers in the system. The new SPVA method approximates $M(t)/G(t)/1$ systems. The DELAYS and INTERP methods approximate $M(t)/E_k/1$ systems. The Erlang distribution has $cv^2 \leq 1$. We showed that SPVA, DELAYS and INTERP approximate the $M(t)/E_k/1$ system well. Furthermore, they offer huge time and memory savings over solving the exact system in many cases. INTERP is the most accurate method. However, it requires 4 to 14 as much CPU time as the SPVA method, and has more memory requirements than do SPVA and DELAYS. If estimates within 3 – 5% accuracy are sufficient for estimates of the mean and standard deviation for the number in the system, then the SPVA and DELAYS methods are good options. They are fast, and have small memory requirements.

We further showed that SPVA and DELAYS approximate $M(t)/E_k(t)/1$ systems well. We tested a scenario for Boston Logan International Airport with time-varying arrival rates and capacities. The SPVA and DELAYS estimates of the time-dependent mean and standard deviation for the number in the system were virtually exact.

We further tested SPVA as an approximation to $M(t)/G(t)/1$ systems. We demonstrated that SPVA is a fast, flexible approximation method for queueing systems with non-phase-type service-time distributions. We examined the uniform and triangular service-time distributions. The SPVA estimates of the mean and standard deviation of the number in the system is what we expected for systems with nonstationary arrival and service rates. Based on the results, we formed two conjectures. First, the SPVA approximation to $M(t)/G(t)/1$ systems is good. Second, the first two moments of the service-time distribution are the primary drivers of the time-dependent mean number in the system. Furthermore, SPVA requires a few seconds of CPU time to find the mean and standard deviation for the $M(t)/G(t)/1$ systems, as well as the probability distribution for the number in the system. This makes it one of the fastest methods currently available for approximate analysis of $M(t)/G(t)/1$ systems.

We tested the SPVA approximation to the $M(t)/H_2/1$ system. H_2 has $cv^2 \geq 1$. Our computational tests indicate that SPVA approximations to the $M(t)/H_2/1$ system will be good in some cases. These cases include moderate cv^2 and moderate and high event frequencies. Holding all parameters constant, SPVA error increases as cv^2 increases. However, even for high cv^2 ($cv^2 = 10$), SPVA errors are in the 6 – 10% range for high event frequencies and heavy utilization. We expect that errors will be smaller for systems with high cv^2 and high event frequency, and moderate system utilization and/or lower degrees of system nonstationarity.

Finally, we compared the SPVA approximation to three SDA methods for $M(t)/M(t)/1$ systems. We showed that SPVA gave estimates of the time-dependent mean and standard deviation are comparable to the Rider and Rothkopf and Oren methods. The results of Clark's method were slightly better than those of SPVA for a low event frequency system. The SDA methods require less CPU time than SPVA to estimate the mean and standard deviation. However, this time advantage diminishes if the SDA methods estimate the probability distribution for the number in the system as well. Clark and Rothkopf and Oren stated that the SDA methods can develop a buildup of error as the system evolves to its equilibrium solution in some cases. This buildup of error did not occur for SPVA in any of the 92 cases we examined.

Up to now, few methods for approximating $M(t)/G(t)/1$ systems have been developed. The SPVA method approximates transient and equilibrium behavior of $M(t)/G(t)/1$ systems. We showed that SPVA performs extremely well under many system conditions. It is as easy to solve. It is fast, requiring seconds of CPU time on a SUN SPARCstation 10 Model 41. This is significantly faster than simulation methods and exact methods, if they exist. We conclude that SPVA is potentially a quite useful tool for approximating $M(t)/G(t)/1$ systems.

Chapter 4

A Decomposition Method for Networks of Queues with Strongly Time-Varying Arrival Rates

This chapter presents a new decomposition method for approximate analysis of networks of queues with time-varying arrival rates. We focus on applying this method to a network of airports, although this model is applicable to other systems for which the assumptions are valid. No model exists today for analyzing such a network. The ability to analyze the network of airports is especially crucial given the current airline hub-and-spoke flight patterns. Congested hub airports propagate delay to other airports, even when those other airports may not themselves be congested.

One uses such a model strategically for airport capacity planning from a system-wide perspective. We want to answer questions such as, "What is the system-wide benefit of increased capacity at Chicago O'Hare Airport?" This tool can also be used to perform "what-if" analyses. For example, "How much would delay increase if airport demand increases system-wide by 5%, without any capacity expansion?", or "Would a $200M investment at Boston or Los Angeles yield a greater decrease in delays system-wide?" Airlines may also use such a model to develop more robust flight schedules. They can determine the amount of "slack" or "buffer" time built into the schedule needed to minimize the "network effects" of local delay, i.e., the propagation throughout their route network of delays occurring at airports which experience congestion.

An important attribute of strategic models such as the one we describe here is speed. This virtually rules out simulation as a tool for such high-level network-wide analysis, because of the number of simulations necessary to obtain statistically valid results, and the computation time per simulation.

Modeling a network of queues with nonstationary arrival rates is very difficult, although some exact models do exist. One can write and solve the CK equations describing an open network of $M(t)/M(t)/s$ queues with probabilistic routing and instantaneous travel time between queues in the network. If there are K queues in the network each with capacity N_i, $i = 1, 2, \ldots, K$, then the number of equations needed to describe such a system is $\prod_{i=1}^{K} N_i$, which can be very large. The same type of analysis can be extended to phase-type arrival processes and/or service times, but this results in a further increase in the number of states needed to describe the system. Hence, even in these relatively simple cases, the computational effort to analyze the queueing network is extremely heavy. The issue of numerical stability also arises.

Massey and Whitt [30] developed exact results for networks of $M(t)/G/\infty$ queues. In this case, customers move independently of each other through the system because there is no queueing. For certain initial conditions, there is a time-dependent product-form solution for number in the system. As mentioned in Section 1.3, infinite-server approximations to finite-server queueing systems may not be good in cases in which the arrival rate is close to or exceeds the service rate. Massey and Whitt are currently exploring an infinite-server approximation to a network of finite-server queues.

Because few options exist to analyze exactly dynamic queueing networks, we develop an approximation method. A natural first choice is to decompose a K-queue network into K individual queues for which we do have solution tools, then use a propagation algorithm to link the individual queues together. This approach allows analysis of networks for which no tools currently exist, or which would otherwise require more memory or CPU time than practically available to solve exactly. For example, a decomposition approach reduces the number of equations to be solved for a network of K $M(t)/M/s$ queues from $\prod_{i=1}^{K} N_i$ to $\sum_{i=1}^{K} N_i$. This results in great memory and CPU time savings and may make analyzing the network possible. The decomposition approach we describe here is similar to the approach of the Queueing Network Analyzer (QNA) for steady-state analysis of networks of queues with stationary parameters [53].

The decomposition method we investigate consists of two components:

1. The Queueing Engine: a model to analyze the queues in isolation. The engine gives statistics of interest for the individual queues, which may depend on the system being modeled, as well as the information needed by the propagation algorithm. The Queueing Engine should be extremely fast.

2. The Propagation Algorithm: an algorithm to propagate the congestion among queues in the network. This algorithm should use the information about local delays obtained at each of the individual queues by the Queueing Engine.

As a result, there are two distinct areas of investigation in decomposition methods: developing fast, accurate approximation models for analyzing individual dynamic queues, and identifying effective algorithms for propagating congestion. Chapters 2 and 3 of this thesis have investigated the former. This chapter examines combinations of queueing engines and propagation methods, and assesses their effectiveness.

The decomposition method must account for the dynamic behavior of the individual queues and their effect on each other. One of the unknowns in the system is the time-dependent arrival rate to each of the queues. The primary task of the propagation method is to estimate this unknown. The arrival rate to each queue in the nonstationary network is partially a sum of departure rates from other queues, which are unknown. These departure rates are time-dependent and reflect the time lag between the arrival rate and levels of congestion at each queue. Note that this contrasts with a stationary, stable network of queues in equilibrium in which we do know the arrival (and departure) rates for each queue. This demonstrates the complexity that nonstationary arrivals introduce to networks of queues.

There is a large body of literature for modeling departure processes for queues with stationary parameters in equilibrium. See, for example, Albin [1], Albin and Kai [2], Daley [9], and Whitt [53, 54], and references therein. However, these methods are not directly applicable to systems with nonstationary arrivals. For example, the variance of the departure processes is used to determine the variance of the superimposed arrival stream of downstream queues. However, finding the variance of the departure process from a system with nonstationary arrivals is problematic. The variance depends in part on the probability the system is idle, and busy. In general, P(system idle at time t) $\neq 1 - \rho(t)$, and P(system busy at time $t) \neq \rho(t)$ in queues with nonstationary arrival or service rates (see Chapter 5). Furthermore, the probabilities are time-dependent quantities for which no closed form expressions exist, or, when they do exist, they may not be tractable. Hence, we cannot find the variance of the departure process, or use the results from systems with stationary arrivals. The only nonstationary system for which we do have exact information about the departure process is for a network of $M(t)/G/\infty$ queues, where this process is nonstationary Poisson [12, 30].

In this chapter, we test a particular decomposition approach for modeling networks of queues with nonstationary arrivals. Arrivals to the individual queues originate both from inside and outside the network. The arrivals to each queue from outside the network form a nonstationary Poisson arrival process. The arrivals to each queue from inside the network are formed by departures from other queues, which have an unknown distribution. The total arrival stream to each queue is the sum of two streams: a stream which is a Poisson arrival process, and a stream which has an unknown distribution. Hence, the total arrival process may not be Poisson. In the decomposition approach we pursue, we assume that the total arrival process at each individual queue forms a nonstationary Poisson process. The main question we want to answer is, "How critical is the Poisson arrival assumption in our decomposition approach?" We shall test the sensitivity of the Poisson arrival process assumption to the following factors: the fraction of total arrivals to each queue which are departures from other queues in the network, the average utilization at each queue, the degree of nonstationarity in the arrival process, and the service-time distributions at the queues. We hypothesize that the degree of nonstationarity in the arrival process may "randomize" the arrival and departure processes, making the downstream Poisson arrival assumption plausible.

The rest of this chapter is organized as follows. We describe the decomposition method, then the network and test cases used to assess its accuracy. Section 4.3 presents the results. We also briefly discuss the Approximate Network Delays (AND) Model, developed by Malone and Odoni [29] to perform "what-if" analyses for strategic planning in the US airport system.

4.1 The Decomposition Methods

The decomposition method estimates the time-dependent probability distribution for the number in the system in each queue in the network. To avoid confusion, we refer to the individual queues in the network as "stations." The method is based upon calculating the probability distribution for the number in the system at each station independently and then estimating the network effects by calculating departure rates from each station. The queueing engine calculates the probability distributions for the number in the system, and the propagation method calculates the departure rates. In this section we will first describe the mechanics of the calculations of the decomposition algorithm, then the queueing engines and the propagation methods used.

Thc decomposition algorithm implemented here assumes that we know the initial probability distribution for number in the system at $t = 0$, the exogenous time-dependent arrival rate, the service rate, the queueing capacity, and the interstation routing probabilities for each of the stations. The algorithm operates in discrete time intervals advancing from $t = 0$ to the end of the interval of analysis. At each time increment, the algorithm uses the probability distribution for the number in the system as input into a propagation algorithm which calculates the departure rates from each station. These rates are then used as input into a queueing engine to calculate the probability distribution for the number in the system for a future time interval.

With this scheme in mind, we describe the decomposition method in detail. We must first introduce the following special notation for networks of stations, to supplement that of Chapter 2.

Notation:

K = number of stations in the network

$\lambda_i(t)$ = arrival rate to station i from outside the network at time t

$\lambda_{ij}(t)$ = departure rate from station i to station j at time t

$\hat{\lambda}_i(t)$ = total arrival rate to station i at time $t = \lambda_i(t) + \sum_{j=1}^{K} \lambda_{ji}(t)$

$S_{i,t}$ = the expected system time of a customer arriving to station i at time t

μ_i = service rate at station i

p_{ij} = probabilistic routing matrix

$P_{i,l}(t)$ = $P(l$ customers at station i at time $t)$

c_i = queueing capacity at station i

T = time interval of analysis

We note here that the service rate and probabilistic routing matrix can also be time-dependent. In our experiments, we keep them stationary, but the model that has been implemented does accommodate both possibilities.

The Decomposition Algorithm

In this chapter we will consider two propagation methods, the Disaggregation-Aggregation (DA) and Schmeiser-Taaffe (ST) [48] propagation methods. The DA method calculates the departure rate from a station from the point of view of arrivals. In contrast, ST uses the instantaneous departure rate. Because of the different perspectives of the two approaches, we present the decomposition algorithms separately below. We give full descriptions of each in Section 4.1.2.

Decomposition Algorithm for the DA Propagation Method:

Given $\lambda_i(t)$, μ_i, c_i, p_{ij}, and $P_{i,l}(0)$ for all $i, j \in \{1, \ldots K\}$, and $t \in [0, T]$,

1. Set $t = 0$.

2. Call the Propagation Algorithm to calculate $\lambda_{ij}(t + S_{i,t})$ from $P_{i,l}(t)$ for each i, j. Set $\hat{\lambda}_i(t + S_{i,t}) = \lambda_i(t + S_{i,t}) + \sum_{j=1}^{K} \lambda_{ji}(t + S_{i,t})$.

3. Call the Queueing Engine to calculate $P_{i,l}(t + \Delta t)$ using $\hat{\lambda}_i(t + \Delta t)$ for each i.

4. $t = t + \Delta t$. If $t \leq T$, go to step 2.

The DA algorithm will actually calculate the departure rates λ_{ij} different lengths of time into the future depending upon the expected number in the system at the time t.

Decomposition Algorithm for ST Propagation Method:

Given $\lambda_i(t)$, μ_i, c_i, p_{ij}, and $P_{i,l}(0)$ for all $i, j \in \{1, \ldots K\}$, and $t \in [0, T]$,

1. Set $t = 0$.

2. Call the Propagation Algorithm to calculate $\lambda_{ij}(t)$ from $P_{i,l}(t)$ for each i, j. Set $\hat{\lambda}_i(t) = \lambda_i(t) + \sum_{j=1}^{K} \lambda_{ji}(t)$.

3. Call the Queueing Engine to calculate $P_{i,l}(t + \Delta t)$ using $\hat{\lambda}_i(t)$ for each i.

4. $t = i + \Delta t$. If $t \leq T$, go to step 2.

We now describe the Queueing Engine and Propagation Methods.

4.1.1 The Queueing Engines

The decomposition method breaks up the K-station network into K independent stations. The queueing engine calculates the time-dependent probability distribution for the number at each station in the network, as if the stations were operating in isolation. The queueing engine is an analytical model, chosen on the basis of the network examined. For example, if we examine a network of single-server stations each with exponential service, we may choose the queueing engine to be the CK equations describing an $M(t)/M/1$ station. Alternatively, we may choose the queueing engine to be an approximation method described in Chapter 2, such as SPVA. The choice of queueing engine will depend on the network examined and the goals of the analysis. In this research, we have modeled a network composed of single-server stations each of which has an exponential or k^{th}-order Erlang service-time distribution. The queueing engines we use are SPVA and the CK equations for the $M(t)/M/1$ and $M(t)/E_k/1$ systems. Finally, We assume a FCFS queue discipline at each station.

4.1.2 The Propagation Algorithms

The propagation algorithm determines the time-dependent departure rate of each station in the network. As we mentioned in the description of the decomposition algorithm we test the Disaggregation-Aggregation (DA) and Schmeiser-Taaffe (ST) [48] propagation methods. We now describe each in detail.

The DA Propagation Method

The DA method calculates the departure rate from station i to j for the point in time when a customer arriving to station i at time t will depart in the future. Conceptually, an arrival to station i at time t must wait for all customers already at station i to be served, plus for its own service time, before it can depart station i. DA uses the expected value of the system time to estimate system time at station i at time t. We denote the expected system time as $S_{i,t}$. DA estimates that an arrival to station i at time t will depart at time

$\hat{t}_{i,t} = t + S_{i,t}$. At time t, DA estimates the future departure rate from station i to station j at time $\hat{t}_{i,t}$ to be $\hat{\lambda}_i(t)p_{ij}$.

DA stores this future information in a look-up table, which we call λ_{ij}. This table stores departure rates for time intervals of length h. Thus, entry m of the look-up table represents the departure rate from station i to station j during the time interval $(h(m-1), hm)$. To store the future departure information from station i to station j, DA must find the appropriate entry in the look-up table in which to store $\hat{\lambda}_i(\hat{t}_{i,t})p_{ij}$. DA translates the continuous variable $\hat{t}_{i,t}$ to an integer value which is the index of the look-up table. This is accomplished as follows:

$$\text{index} = \left\lceil \frac{\hat{t}_{i,t}}{h} \right\rceil$$

Conversely, when the Queueing Engine retrieves the information in the look-up table to find the total arrival rate to station i at time t, it must find the look-up table index which represents t. This is done as follows:

$$\text{index} = \left\lceil \frac{t}{h} \right\rceil$$

Note that if the server at station i is busy, $\mu_i p_{ij}$ is the expected departure rate from station i to j. Therefore, DA assumes each entry in the look-up table has a maximum capacity of $\mu_i p_{ij}$. DA stores the departure rate from station i to j, $\hat{\lambda}_i(\hat{t}_{i,t})p_{ij}$, in the look-up table entry `index` if the current amount in `index`, plus $\hat{\lambda}_i(\hat{t}_{i,t})p_{ij}$, does not exceed $\mu_i p_{ij}$. If this is not the case, the amount by which the capacity is exceeded is distributed in consecutive bins. In summary, the DA algorithm applied to station i is:

```
Given $\hat{\lambda}_i(t)$, $\mu_i$, $p_{ij}$, and $P_{i,l}(t)$,
$S_{i,t} = [E[$number of customers at station i at time t$] + 1]/\mu_i$
$\hat{t}_{i,t} = t + S_{i,t}$
index=$\left\lceil \frac{\hat{t}_{i,t}}{h} \right\rceil$
departure rate = $\hat{\lambda}_i(t)p_{ij}$
while( departure rate $>$ 0.0 )
    temp = $\min[(\mu_i p_{ij}), (\lambda_{ij}$[index] + departure rate)]
    excess = [($\lambda_{ij}$[index] + departure rate) - $(\mu_i p_{ij})$]
    $\lambda_{ij}$[index] = temp
    index = index + 1
    departure rate = excess
end while
```

When using the DA Method, the arrival rate to station i at time t is found as follows: Let $\texttt{index} = \lceil \frac{t}{h} \rceil$. Then,

$$\hat{\lambda}_i(t) = \lambda_i(t) + \sum_{j=1}^{K} \lambda_{ji}[\texttt{index}]$$

In all experiments, we choose h to be the expected service time. For models in which the data is less refined, h may be chosen to be a larger number. For example, our data may be in the form of hourly arrival rates. In this case, we may choose h to represent one-half hour or one-hour intervals.

The ST Propagation Method

Schmeiser and Taaffe [48] developed a propagation method for networks where each station is approximated by an $M(t)/M(t)/s$ system. We call this method "ST." ST uses the expected departure rate from station i to j to approximate $\lambda_{ij}(t)$. We extend the method to the general approximation method, SPVA, and to $M(t)/E_k/1$ stations.

The expression for expected departure rate depends on the state definition used by the queueing engine. If the states of station i represent the number of customers, the expected departure rate from i to j is:

$$\begin{aligned} \lambda_{ij}(t) &= p_{ij}\,[0 \cdot P(\text{server at station } i \text{ is idle at time } t) + \\ &\qquad \mu_i \cdot P(\text{server at station } i \text{ is busy at time } t)] \\ &= p_{ij}\mu_i(1 - P_{i,0}(t)) \qquad (4.1) \end{aligned}$$

We use expression (4.1) when we model station i by SPVA or the CK equations for an $M(t)/M/1$ station.

In contrast, the CK equations for the $M(t)/E_k/1$ system represent the number of stages of service in the system still to complete. A customer needs $k\mu_i$ stages of service. A departure can occur only from states 1, $k+1$, $2k+1$, $3k+1$, ... Therefore, the expected customer departure rate at time t from an $M(t)/E_k/1$ station is $k\mu_i \sum_{\ell=0}^{c_i} P_{i,\ell k+1}(t)$. The

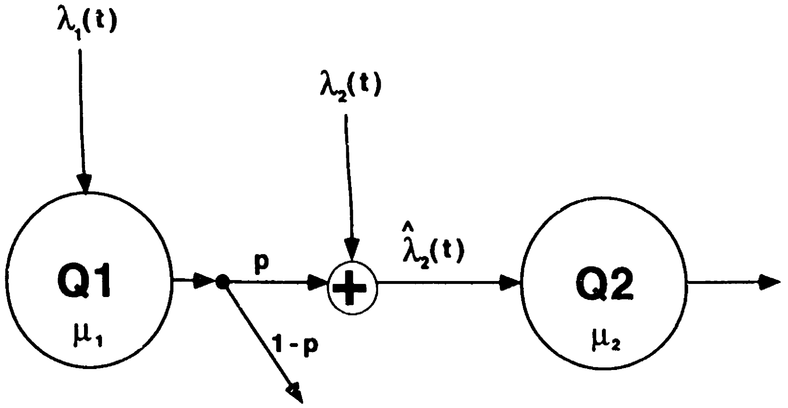


Figure 4-1: The Tandem Queue Network Model to Test the Decomposition Method

expected customer departure rate from station i to j is:

$$\lambda_{ij}(t) = p_{ij} k \mu_i \sum_{\ell=0}^{c_i} P_{i,\ell k+1}(t) \tag{4.2}$$

We use expression (4.2) when we model station i by the CK equations for an $M(t)/E_k/1$ station.

In total, we test combinations of two queueing engines, SPVA and the CK equations, and two approximation methods, DA and ST, resulting in four approximation methods. The CK+DA and CK+ST combinations, we call the "first-level" decompositions because they are based on an exact (CK) approach for computing the queueing statistics at each individual station. The SPVA+DA and SPVA+ST are "second-level" decompositions because they use an approximate method (SPVA) for computing queueing statistics. We compare these results to the exact network in which we solve large sets of CK equations describing the joint state probabilities for all the stations in the network.

4.2 Network Description and Test Case Parameters

We now describe the network used to test the decomposition approach, and the corresponding parameters and their values.

4.2.1 The Network

We test the four decomposition methods using the tandem-queue network illustrated in Figure 4-1. This network is simple. This simplicity will allow us to observe the performance of the decomposition method and to isolate the factors which affect its accuracy.

The tandem-queue network consists of two stations labeled Q1 and Q2. All Q1 arrivals come from outside the network. A departure from Q1 goes to Q2 with probability $p_{12} = p$, or leaves the network with probability $1-p$. Arrivals to Q2 come from outside the network and from Q1. The "+" sign between Q1 and Q2 indicates that the two streams, $\lambda_2(t)$ and $\lambda_{12}(t)$, are added to form the total arrival stream to Q2. Departures from Q2 leave the network after completing service. There is no feedback in this network.

The decomposition method will reduce the analysis of this network to analysis of individual stations. Since all of Q1's arrivals come from outside the network, the decomposition method will give the same time-dependent probability distribution at Q1 as the individual queue analysis. Q2, however, depends on the time-dependent output of Q1. Therefore, the decomposition method will be approximate for Q2. We measure the accuracy of the decomposition method at Q2. We compare the exact network values $m(t)$, $\sigma(t)$, m^*, and σ^* at Q2 to those predicted by the decomposition method.

4.2.2 Parameters

We parameterize the test cases in a manner similar to Section 3.3. However, we differentiate between the two stations. The test cases are combinations of the following parameter values.

Average Utilization, $\overline{\rho}_i$, ranges from moderate to heavy at both stations: 0.5 – 0.9. $\overline{\rho}_1$ and $\overline{\rho}_2$ are input data for the model.

The service rates at Q1 and Q2, μ_1 and μ_2, are input data. The service-time distributions at Q1 and Q2 are also input data. Note that the service-time distribution at Q1 affects the estimates of $m(t)$, $\sigma(t)$, m^*, and σ^* at Q2. We model the service-time distributions at Q1 as exponential and 3^{rd}-order Erlang, and Q2 as exponential. Note also that the departure process from Q1 is not Poisson if the arrival rate varies with time, even if the service times are exponential.

We model the external Poisson arrival process as a sinusoidal function: $\lambda_i(t) = \overline{\lambda}_i + RA_i\overline{\lambda}_i \sin\left(\frac{2\pi t}{24}\right)$. RA_i and $\overline{\lambda}_i$ are the relative amplitude and average external arrival rate to station i, respectively. $\overline{\overline{\lambda}}_2$ represents the average total arrival rate to Q2. We define RA_i below. RA_1 and RA_2 are input data. $\overline{\lambda}_i$ and $\overline{\overline{\lambda}}_2$ are calculated from other input data:

$$\overline{\lambda}_1 = \overline{\rho}_1\mu_1$$

$$\begin{aligned}\bar{\hat{\lambda}}_2 &= \bar{\rho}_2\mu_2 \\ \bar{\lambda}_2 &= \bar{\hat{\lambda}}_2 - p\bar{\lambda}_1\end{aligned}$$

We define the degree of stationarity as in Section 3.3, except we differentiate between RA_2, the relative amplitude of the external arrival rate, and $\widehat{RA}_2$, the relative amplitude of the total arrival rate to Q2. RA_1 and RA_2 are input data, and $\widehat{RA}_2$ is calculated from model outputs: $\widehat{RA}_2 = \frac{\rho_{\max,2}-\bar{\rho}_2}{\bar{\rho}_2}$. $RA_1 = \frac{1}{3}, \frac{2}{3}, 1$, $RA_2 = 0, \frac{1}{3}, \frac{2}{3}, 1$, and $\widehat{RA}_2$ ranges from 0.02 to 0.98.

The fraction of arrivals to Q2 which are departures from Q1, f, ranges from 10% to 90%. We examine f =0.1, 0.3, 0.4, 0.5, 0.9. In each case, $(1-f)$ of the arrival stream to Q2 forms a nonstationary Poisson process. f is a user-specified input. p, the probability that a random departure from Q1 goes to Q2, is derived from other input data: $p = \frac{f\bar{\hat{\lambda}}_2}{\bar{\lambda}}$.

We represent maximum system utilizations at Q1 and Q2 by $\rho_{\max,1}$ and $\rho_{\max,2}$. $\rho_{\max,1}$ is calculable from other input data. It ranges from 0.83 to 1.8. In 49 of 56 test cases, $\rho_{\max,1} \geq 1$. $\rho_{\max,2}$ is a model output and depends on $\hat{\lambda}_2(t)$, another model output. In our test cases, $\rho_{\max,2}$ ranges from 0.55 to 1.7, and $\rho_{\max,2} \geq 1$ in 26 of 56 cases.

We would like to model networks with infinite queueing capacity at each station, as well as Erlang service-time distributions of higher-orders. However, memory constraints dictate the maximum queueing capacity and the Erlang orders. We showed in Chapter 3 that increasing the order of the Erlang increases the number of states, and therefore memory requirements, to model a system with queueing capacity c. Therefore, we strike a balance between Erlang orders modeled and queueing capacity. We model Q1 and Q2 with queueing capacity 450 when Q1's service-time distribution is exponential, and with capacity 250 when Q1 has a 3^{rd}-order Erlang distribution.

The order of the Erlang distribution also affects the complexity of the CK equations describing the exact system. Even for this very simple tandem-queue network, the CK equations are extremely complex for both service-time distributions as Q1. We show the equations for the case in which Q1 has a 3^{rd}-order Erlang service-time distribution. Let $c_1 = c_2 = c =$ queueing capacity at Q1 and Q2, and $Q_{i,j}(t) =$ Prob(i stages of service at Q1 and j customers at Q2 at time t). Then the CK equations for the network are:

$$Q'_{0,0}(t) = -(\lambda_1(t) + \lambda_2(t))Q_{0,0}(t) + 3(1-p)\mu_1 Q_{1,0}(t) + \mu_2 Q_{0,1}(t)$$

$$
\begin{aligned}
Q'_{0,j}(t) &= -(\lambda_1(t)+\lambda_2(t)+\mu_2)Q_{0,j}(t)+\lambda_2(t)Q_{0,j-1}(t)+3p\mu_1Q_{1,j-1}(t)\\
&\quad +3(1-p)\mu_1Q_{1,j}(t)+\mu_2Q_{0,j+1}(t), 1\leq j\leq c-1\\
Q'_{0,c}(t) &= -(\lambda_1(t)+\mu_2)Q_{0,c}(t)+\lambda_2(t)Q_{0,c-1}(t)\\
&\quad +3p\mu_1Q_{1,c-1}(t)+3\mu_1Q_{1,c}(t)\\
Q'_{i,0}(t) &= -(\lambda_1(t)+\lambda_2(t)+3\mu_1)Q_{i,0}(t)+3\mu_1Q_{i+1,0}(t)+\mu_2Q_{i,1}(t),\\
&\quad 1\leq i\leq 2\\
Q'_{i,0}(t) &= -(\lambda_1(t)+\lambda_2(t)+3\mu_1)Q_{i,0}(t)+\lambda_1(t)Q_{i-3,0}(t)+3(1-p)\mu_1Q_{i+1,0}(t)+\\
&\quad \mu_2Q_{i,1}(t), i\in\{3n\}, n=1,2,\ldots,c-1\\
Q'_{i,0}(t) &= -(\lambda_1(t)+\lambda_2(t)+3\mu_1)Q_{i,0}(t)+\lambda_1(t)Q_{i-3,0}(t)+3\mu_1Q_{i+1,0}(t)+\mu_2Q_{i,1}(t),\\
&\quad i\in\{3n+1,3n+2\}, n=1,2,\ldots,c-1\\
Q'_{i,0}(t) &= -(\lambda_2(t)+3\mu_1)Q_{i,0}(t)+\lambda_1(t)Q_{i-3,0}(t)+3\mu_1Q_{i+1,0}(t)+\mu_2Q_{i,1}(t),\\
&\quad 3c-2\leq i\leq 3c-1\\
Q'_{3c,0}(t) &= -(\lambda_2(t)+3\mu_1)Q_{3c,0}(t)+\lambda_1(t)Q_{3c-3,0}(t)+\mu_2Q_{3c,1}(t)\\
Q'_{i,j}(t) &= -(\lambda_1(t)+\lambda_2(t)+\mu_2+3\mu_1)Q_{i,j}(t)+\lambda_2(t)Q_{i,j-1}(t)+3\mu_1Q_{i+1,j}(t)\\
&\quad +\mu_2Q_{i,j+1}(t), 1\leq i<3, 1\leq j\leq c-1\\
Q'_{i,c}(t) &= -(\lambda_1(t)+\mu_2+3\mu_1)Q_{i,c}(t)+\lambda_2(t)Q_{i,c-1}(t)+3\mu_1Q_{i+1,c}(t), 1\leq i\leq 2\\
Q'_{i,c}(t) &= -(\lambda_1(t)+\mu_2+3\mu_1)Q_{i,c}(t)+\lambda_2(t)Q_{i,c-1}(t)+3\mu_1Q_{i+1,c}(t)\\
&\quad +\lambda_1(t)Q_{i-3,c}(t)+3p\mu_1Q_{i+1,c-1}(t), i\in\{3n\}, n=1,2,\ldots,c-1\\
Q'_{i,c}(t) &= -(\lambda_1(t)+\mu_2+3\mu_1)Q_{i,c}(t)+\lambda_2(t)Q_{i,c-1}(t)+3\mu_1Q_{i+1,c}(t)\\
&\quad +\lambda_1(t)Q_{i-3,c}(t), i\in\{3n+1,3n+2\}, n=1,2,\ldots,c-1\\
Q'_{i,c}(t) &= -(\mu_2+3\mu_1)Q_{i,c}(t)+\lambda_2(t)Q_{i,c-1}(t)+3\mu_1Q_{i+1,c}(t)\\
&\quad +\lambda_1(t)Q_{i-3,c}(t), 3c-2\leq i\leq 3c-1\\
Q'_{3c,c}(t) &= -(\mu_2+3\mu_1)Q_{3c,c}(t)+\lambda_2(t)Q_{3c,c-1}(t)+\lambda_1(t)Q_{3c-3,c}\\
Q'_{i,j}(t) &= -(\lambda_1(t)+\lambda_2(t)+\mu_2+3\mu_1)Q_{i,j}(t)+\lambda_1(t)Q_{i-3,j}(t)+\lambda_2(t)Q_{i,j-1}(t)\\
&\quad +3p\mu_1Q_{i+1,j-1}(t)+3(1-p)\mu_1Q_{i+1,j}(t)+\mu_2Q_{i,j+1}(t),\\
&\quad i\in\{3n\}, n=1,2,\ldots,c-1, 1\leq j\leq c-1\\
Q'_{i,j}(t) &= -(\lambda_1(t)+\lambda_2(t)+\mu_2+3\mu_1)Q_{i,j}(t)+\lambda_1(t)Q_{i-3,j}(t)+\lambda_2(t)Q_{i,j-1}(t)+\\
&\quad +3\mu_1Q_{i+1,j}(t)+\mu_2Q_{i,j+1}(t),
\end{aligned}
$$

$$
\begin{aligned}
& i \in \{3n+1, 3n+2\}, n = 1, 2, \ldots, c-1, 1 \le j \le c-1 \\
Q'_{i,j}(t) &= -(\lambda_2(t) + \mu_2 + 3\mu_1)Q_{i,j}(t) + \lambda_1(t)Q_{i-3,j}(t) + \lambda_2(t)Q_{i,j-1}(t) + \\
& +3\mu_1 Q_{i+1,j}(t) + \mu_2 Q_{i,j+1}(t), \\
& 3c-2 \le i \le 3c-1, 1 \le j \le c-1 \\
Q'_{3c,j}(t) &= -(\lambda_2(t) + \mu_2 + 3\mu_1)Q_{3c,j}(t) + \lambda_1(t)Q_{3c-3,j}(t) + \lambda_2(t)Q_{3c,j-1}(t) + \\
& +\mu_2 Q_{3c,j+1}(t), 1 \le j \le c-1
\end{aligned}
$$

The complexity of these equations will increase with more stations in the network.

We analyze the systems over a 24-hour period. Again, this choice is dictated by the CPU time needed to solve the exact system. Tables 4.1 and 4.2 list the CPU times needed to solve the exact system, and each of the decomposition methods. All cases were run on a SUN SPARCstation 10 Model 41 in double precision. The "CK+DA" and "CK+ST" columns indicate the CK equation queueing engine with the DA and ST propagation methods, respectively. The "SPVA+DA" and "SPVA+ST" columns indicate the SPVA queueing engine with the DA and ST approximation methods, respectively. Note that the exact solution CPU time in Table 4.2 is at least 12 hours, as compared to 3 minutes for CK+DA and CK+ST, and 10 seconds for the SPVA+DA and SPVA+ST methods.

Tables 4.3 and 4.4 list the parameters associated with each of the 57 cases. We examined 24 cases in which Q1 had an exponential distribution, and 33 cases in which Q1 had a 3^{rd}-order Erlang distribution. We investigated 9 extra cases of the 3^{rd}-order Erlang distribution for $f = 0.3$ and 0.4 to assess decomposition method accuracy for more values between $f = 0.1$ and 0.5.

4.3 Performance Measures, Test Case Results and Discussion

This section presents performance measures used to assess decomposition accuracy, and presents and discusses results. We examine decomposition quality in several ways. First, we examine the "first level" approximations, CK+DA and CK+ST, which we refer to as DA and ST. These first-level approximations give us insight into the effect of decomposing the network. We also compare the quality of the propagation methods to each other. We

Case	Exact	CK+DA	CK+ST	SPVA+DA	SPVA+ST
1	32,829.0	110.5	84.7	21.1	20.8
2	33,108.4	97.1	75.5	20.6	20.3
3	27,211.2	124.8	106.9	22.5	20.1
4	26,555.7	132.5	97.7	21.0	20.7
5	27,774.2	124.1	94.5	20.5	20.2
6	25,203.8	138.6	103.4	24.8	20.7
7	29,158.8	100.5	92.0	19.5	19.6
8	25,886.2	125.4	105.9	20.9	20.4
9	22,716.7	113.8	95.3	18.9	18.4
10	22,315.7	130.7	103.0	20.3	20.3
11	31,832.7	85.9	85.0	19.9	21.6
12	25,652.2	100.7	96.7	21.4	22.8
13	28,214.5	123.8	100.3	18.9	18.7
14	26,570.5	111.7	102.1	21.0	21.6
15	27,284.6	107.3	92.7	18.7	18.4
16	28,405.9	94.0	84.6	19.0	18.7
17	33,542.1	77.3	75.7	20.7	21.8
18	26,825.3	111.1	93.2	21.8	22.9
19	25,768.5	149.7	112.8	20.4	21.1
20	29,186.5	118.8	73.5	19.6	19.1
21	27,987.4	147.2	90.7	20.1	19.1
22	26,673.3	165.0	104.1	20.0	19.2
23	31,942.2	86.9	86.6	20.1	21.1
24	26,193.6	91.3	90.5	20.8	21.6

Table 4.1: CPU Times (Seconds) on SUN SPARCstation 10 Model 41: Q1 $M(t)/M/1$

Case	Exact	CK+DA	CK+ST	SPVA+DA	SPVA+ST
1	55,120.2	174.7	164.5	10.3	10.2
2	53,417.5	167.1	158.5	9.4	9.4
3	46,481.6	173.3	166.0	10.1	10.2
4	46,876.8	187.1	173.0	10.2	10.2
5	46,530.6	179.3	167.8	9.6	9.6
6	47,592.2	191.2	175.3	10.3	10.3
7	54,558.1	168.5	164.6	9.8	9.9
8	47,106.2	175.9	170.0	10.4	10.5
9	45,843.3	190.0	179.2	9.1	9.2
10	45,939.2	181.6	170.2	9.5	9.2
11	52,022.8	151.0	145.4	10.2	10.4
12	47,546.5	160.6	150.3	10.1	10.6
13	52,222.7	159.4	147.6	10.0	10.0
15	46,431.8	180.5	172.2	9.1	9.1
16	53,102.6	173.4	167.5	9.3	9.3
17	54,427.3	160.2	155.4	10.2	10.2
18	47,306.5	175.8	165.8	10.3	10.3
19	47,568.5	194.9	171.6	10.3	10.1
20	53,796.2	182.9	163.6	9.6	9.6
21	47,742.2	197.0	173.6	9.6	9.6
22	45,289.1	200.9	175.9	9.6	10.5
23	51,629.3	148.8	144.7	10.3	10.3
24	48,081.6	164.1	151.5	10.1	10.1
25	48,766.6	149.9	133.2	8.3	8.2
26	48,258.0	155.6	142.8	8.4	8.4
27	44,500.0	165.4	149.4	8.2	8.2
28	49,024.2	182.9	168.4	10.6	11.2
29	53,766.1	192.9	175.9	11.3	11.6
30	42,738.4	168.6	151.9	8.2	8.2
31	47,194.3	192.6	176.4	11.6	11.2
32	47,478.2	169.9	164.0	10.1	10.1

Table 4.2: CPU Times (Seconds) on SUN SPARCstation 10 Model 41: Q1 $M(t)/E_3/1$

Case	f	$\overline{\rho}_1$	$\overline{\rho}_2$	RA_1	RA_2	$\widehat{RA}_2$	$\rho_{\max,1}$	$\rho_{\max,2}$
1	0.1	0.75	0.75	0.67	0.67	0.63	1.25	1.23
2	0.1	0.90	0.50	0.67	0.67	0.61	1.50	0.81
3	0.5	0.75	0.75	1.00	0.00	0.17	1.50	0.88
4	0.5	0.75	0.75	0.67	0.67	0.50	1.25	1.13
5	0.5	0.90	0.50	0.67	0.67	0.39	1.50	0.69
6	0.9	0.75	0.75	0.67	0.67	0.37	1.25	1.02
7	0.1	0.75	0.75	1.00	0.00	0.03	1.50	0.78
8	0.9	0.75	0.75	1.00	0.00	0.30	1.50	0.98
9	0.9	0.90	0.50	1.00	0.00	0.10	1.80	0.55
10	0.9	0.90	0.50	0.67	0.67	0.17	1.50	0.58
11	0.1	0.50	0.90	0.67	0.67	0.67	0.83	1.50
12	0.5	0.50	0.90	0.67	0.67	0.66	0.83	1.50
13	0.1	0.50	0.90	1.00	0.00	0.09	1.00	0.98
14	0.5	0.50	0.90	1.00	0.00	0.46	1.00	1.32
15	0.5	0.90	0.48	1.00	0.00	0.09	1.80	0.53
16	0.1	0.90	0.50	1.00	0.00	0.02	1.80	0.51
17	0.1	0.75	0.75	0.33	1.00	0.93	1.00	1.45
18	0.5	0.75	0.75	0.33	1.00	0.64	1.00	1.23
19	0.9	0.75	0.75	0.33	1.00	0.36	1.00	1.02
20	0.1	0.90	0.50	0.33	1.00	0.91	1.20	0.96
21	0.5	0.90	0.50	0.33	1.00	0.56	1.20	0.78
22	0.9	0.90	0.50	0.33	1.00	0.20	1.20	0.60
23	0.1	0.50	0.90	0.33	1.00	0.93	0.83	1.74
24	0.5	0.50	0.90	0.33	1.00	0.67	0.83	1.50

Table 4.3: Case Parameters for Decomposition Test Cases in which Q1 has Exponential Distribution.

Case	f	$\bar{\rho}_1$	$\bar{\rho}_2$	RA_1	RA_2	$\widehat{RA}_2$	$\rho_{\max,1}$	$\rho_{\max,2}$
1	0.1	0.75	0.75	0.67	0.67	0.63	1.25	1.23
2	0.1	0.90	0.50	0.67	0.67	0.62	1.50	0.81
3	0.5	0.75	0.73	1.00	0.00	0.19	1.50	0.88
4	0.5	0.75	0.75	0.67	0.67	0.50	1.25	1.13
5	0.5	0.90	0.49	0.67	0.67	0.43	1.50	0.69
6	0.9	0.75	0.75	0.67	0.67	0.37	1.25	1.03
7	0.1	0.75	0.75	1.00	0.00	0.04	1.50	0.78
8	0.9	0.75	0.72	1.00	0.00	0.35	1.50	0.98
9	0.9	0.90	0.43	1.00	0.00	0.28	1.80	0.55
10	0.9	0.90	0.48	0.67	0.67	0.23	1.50	0.58
11	0.1	0.50	0.90	0.67	0.67	0.67	0.83	1.50
12	0.5	0.50	0.90	0.67	0.67	0.67	0.83	1.50
13	0.1	0.50	0.90	1.00	0.00	0.09	1.00	0.98
14	0.5	0.50	0.90	1.00	0.00	0.47	1.00	1.32
15	0.5	0.90	0.46	1.00	0.00	0.15	1.80	0.53
16	0.1	0.90	0.49	1.00	0.00	0.03	1.80	0.51
17	0.1	0.75	0.75	0.33	1.00	0.93	1.00	1.45
18	0.5	0.75	0.75	0.33	1.00	0.65	1.00	1.24
19	0.9	0.75	0.75	0.33	1.00	0.37	1.00	1.02
20	0.1	0.90	0.50	0.33	1.00	0.91	1.20	0.96
21	0.5	0.90	0.50	0.33	1.00	0.56	1.20	0.78
22	0.9	0.90	0.50	0.33	1.00	0.20	1.20	0.60
23	0.1	0.50	0.90	0.33	1.00	0.93	0.83	1.74
24	0.5	0.50	0.90	0.33	1.00	0.67	0.83	1.50
25	0.1	0.50	0.50	0.67	0.67	0.67	0.83	0.83
26	0.1	0.50	0.50	1.00	1.00	0.99	1.00	1.00
27	0.3	0.50	0.50	1.00	1.00	0.98	1.00	0.99
28	0.3	0.75	0.75	0.67	0.67	0.57	1.25	1.18
29	0.3	0.90	0.90	0.33	0.33	0.27	1.20	1.14
30	0.4	0.50	0.50	1.00	1.00	0.97	1.00	0.99
31	0.4	0.75	0.75	0.67	0.67	0.53	1.25	1.15
32	0.4	0.75	0.74	1.00	1.00	0.76	1.50	1.30
33	0.4	0.90	0.90	0.33	0.33	0.25	1.20	1.12

Table 4.4: Case Parameters for Decomposition Test Cases in which Q1 has 3^{rd}-order Erlang Distribution.

then assess the "second level" approximations, which are SPVA+DA and SPVA+ST. We call these second level because they have two layers of approximation: the network has been decomposed, necessitating the use of a propagation method, and we model individual queues with the SPVA method. We compare the error introduced by SPVA by comparing the SPVA+DA with DA results, and SPVA+ST with ST results.

We begin the discussion with performance measures.

4.3.1 Network Performance Measures

We assess the decomposition methods by their speed and accuracy. Tables 4.1 and 4.2 show that solving the network exactly requires 2 and 3 order of magnitude more CPU time than solving the decomposition methods. Our test cases examined the simplest network possible. There are only two queues, with no feedback. From our experience, we conclude that it may be practically infeasible to model and solve larger and more complicated networks using the exact solution approach.

The methods using the SPVA queueing engine use far less CPU time than the CK queueing engines. Tables 3.3 and 3.4 show similar results. The SPVA method is faster by an order of magnitude. There is a one to 30 second difference between the DA and ST methods using the CK queueing engine. This difference is not evident when the SPVA queueing engine is used.

We use the same accuracy measures as in Section 3.1. Accuracy for m^* and σ^* at Q2 is measured by the Relative Error (RE). We measure accuracy for $m(t)$ and $\sigma(t)$ at Q2 by the Weighted Percentage Error (WPE). We now discuss the results.

4.3.2 Test Case Results and Discussion

We present the first-level decomposition results first. We isolate through them the error introduced by decomposing the network. We identify the parameters (service-time distribution, f, $\overline{\rho}_1$, $\overline{\rho}_2$, $\rho_{\max,1}$, $\rho_{\max,2}$) which affect this error. We then compare the first and second level approximations to identify the change in error introduced by the SPVA queueing engine.

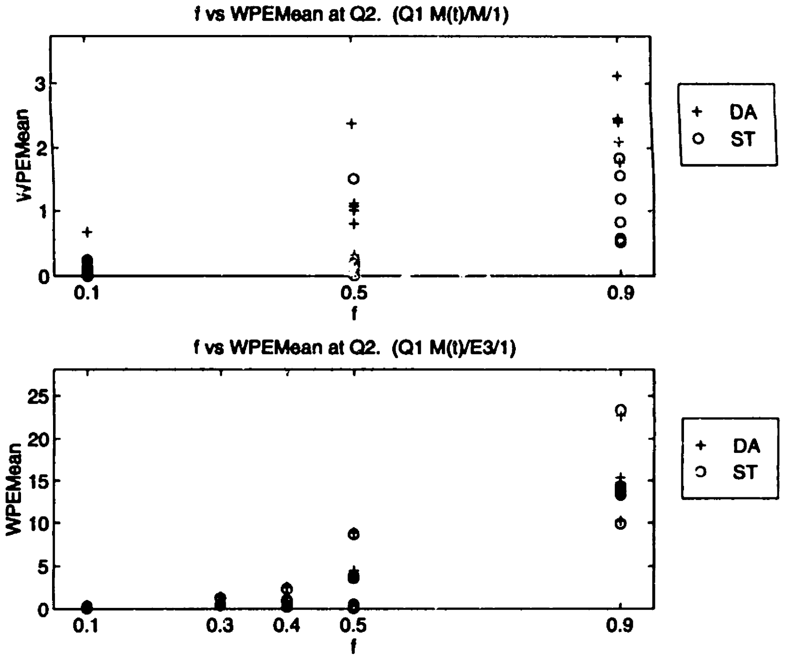


Figure 4-2: f vs. WPE of Mean at Q2. Q1 with Exponential (Top) and with 3^{rd}-order Erlang (Bottom) Service-Time Distribution

Propagation Method Quality

Our results show that the service-time distribution at Q1, as well as f, the fraction of arrivals to Q2 which are departures from Q1, are the primary factors which affect the accuracy of our decomposition estimates of $m(t)$, $\sigma(t)$, m^*, and σ^*. Figure 4-2 shows the plots of f vs WPE of mean for Q1 with exponential (top) and 3^{rd}-order Erlang (bottom) service-time distributions. Within each plot, the error increases as f increases from 0.1 to 0.9. The WPE of mean is less than 5% in all 24 cases when Q1 has an exponential distribution. In contrast, the WPE of mean reaches 25% when f = 0.9 and Q1 has a 3^{rd}-order Erlang distribution. WPE of mean is less than 3% for f= 0.3 or 0.4. When f=0.5, the WPE of mean is less than 9%. The WPE of σ, and RE of m^* and σ^* show the same increasing error as f increases. They show errors on the same order as those shown for WPE of mean for the two service-time distributions, respectively.

Each of the twenty-four cases listed in Table 4.3 has the same input data as the corresponding case in Table 4.4 except that the service-time distribution at Q1 in the former is exponential and in the latter, 3^{rd}-order Erlang. Associated with each case in the tables are the four error measures. We compare the effect of the change in service-time distribution for each case by examining the change in the error measure on a case-by-case basis. In

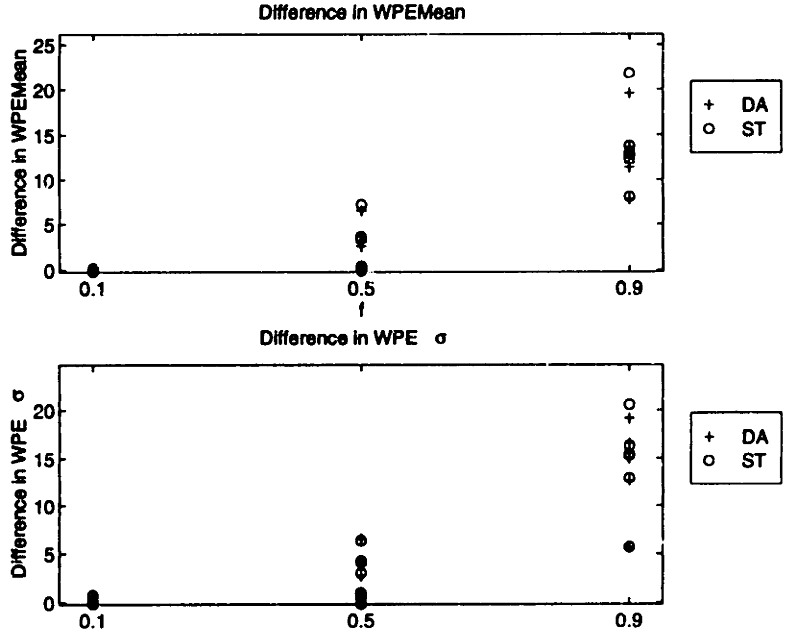


Figure 4-3: Case-By-Case Difference between WPE of Mean (top) and WPE of σ (Bottom) for 3^{rd}-order Erlang and Exponential Service-Time Distributions.

Figure 4-3, we plot the difference in WPE of mean on a case-by-case basis. That is, each point in the graph is `(WPE of mean with 3^{rd}-order Erlang distribution – WPE of mean with exponential service-time distribution)` for each of the 24 cases. We plot these differences vs. f. This plot isolates the effect of changing the service-time distribution, while keeping all other parameters the same. When $f = 0.9$, the increase in error is between 5 and 25%. Strikingly, the increase in error is negligible when $f = 0.1$. Even when $f = 0.5$, the increase in error is less than 10%.

What effect do the other parameters, RA_1, $\widehat{RA}_2$, $\overline{\rho}_1$, $\overline{\rho}_2$, $\rho_{\max,1}$ and $\rho_{\max,2}$ have on decomposition accuracy? Our test cases indicate that they do not have as much effect as f and the service-time distribution. Figure 4-4 shows the WPE of σ plotted against these parameters. Q1 parameters appear in the left column, Q2 in the right. Q1 parameters result in increasing errors with increasing arrival rate nonstationarity, and maximum and average utilization. But there are also cases which have high RA_1, $\rho_{\max,1}$, and $\overline{\rho}_1$, but small WPE. None of these factors in isolation decreases appreciably the accuracy of the decomposition. The Q2 parameters, $\widehat{RA}_2$ and $\rho_{\max,2}$ show some large errors at values in the middle of their ranges. However, there are also small errors in these same middle ranges. None of these factors in isolation decreases decomposition accuracy. We conclude that relative amplitude, average and maximum utilization are secondary factors in determining

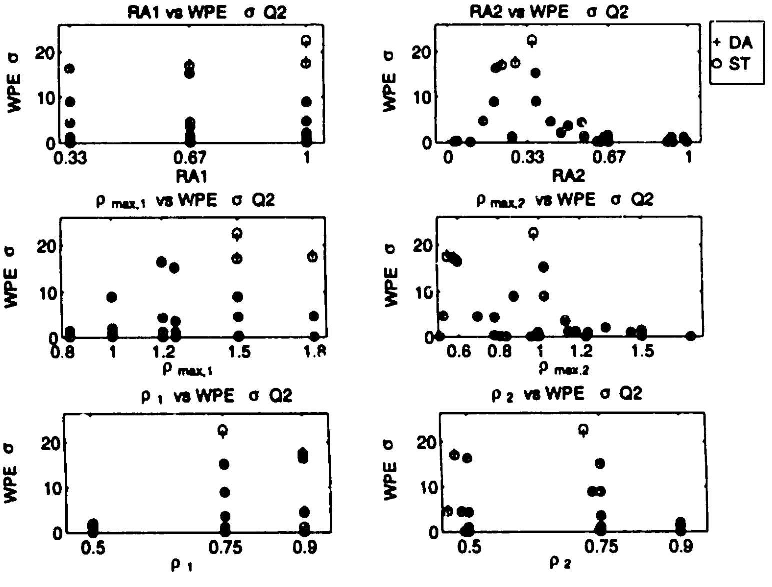


Figure 4-4: WPE of σ vs. Relative Amplitude and Maximum and Average Utilization at Q1 and Q2

decomposition accuracy.

We fix $f = 0.5$ to see if we can isolate secondary effects. Figure 4-5 plots WPE of σ vs. secondary factors for the 3^{rd}-order Erlang distribution. The secondary effects are clear. For a fixed f, decomposition accuracy decreases with increasing relative amplitude and maximum and average utilization at Q1. However, the opposite is true for Q2 parameters. Decomposition method accuracy increases with increasing relative amplitude in the total arrival process, and in maximum and average utilization at Q2, for a fixed f.

Some of our results are similar to Albin [1] in which she examined queues with stationary arrival rates in equilibrium. She examined the difference between the expected number in a $\sum_{i=1}^{n} GI/M/1$ system and in an $M/M/1$ system. Albin's arrival processes had squared coefficients of variation of $\frac{1}{2}$, 2, and 9. She found that the difference between the expected number in the $\sum_{i=1}^{n} GI/M/1$ and $M/M/1$ systems decreases primarily as the number of superposition arrival processes, n, increases. Our result, that as f increases, accuracy decreases, is analogous. Albin found that the secondary factors were average utilization and the squared coefficient of variation of the component processes. For a given n, the difference increases as system utilization increases, and the absolute difference of the squared coefficient of variation of the component processes from 1.0 grows. Our analogous result is that for a fixed f, decomposition accuracy decreases as RA_1, $\overline{\rho}_1$, and $\rho_{\max,1}$ increase. In contrast, for a fixed f, decomposition accuracy increases as $\widehat{RA}_2$, $\overline{\rho}_2$, and $\rho_{\max,2}$ increase.

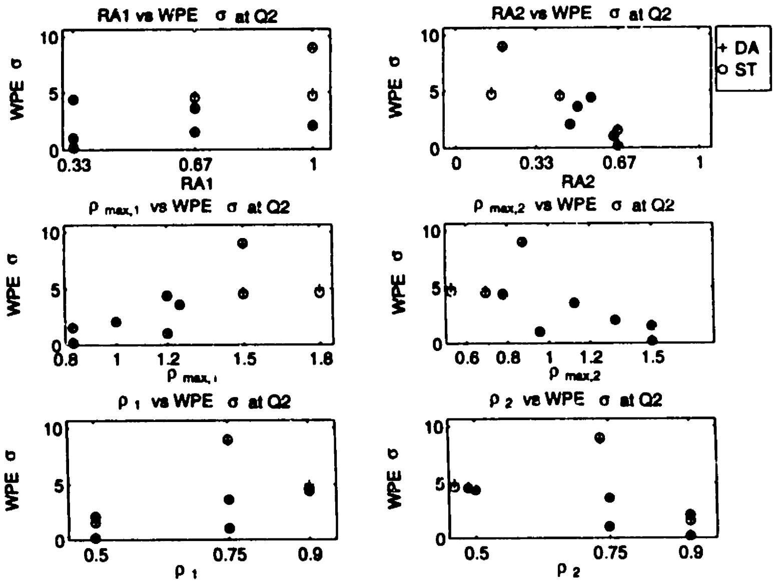
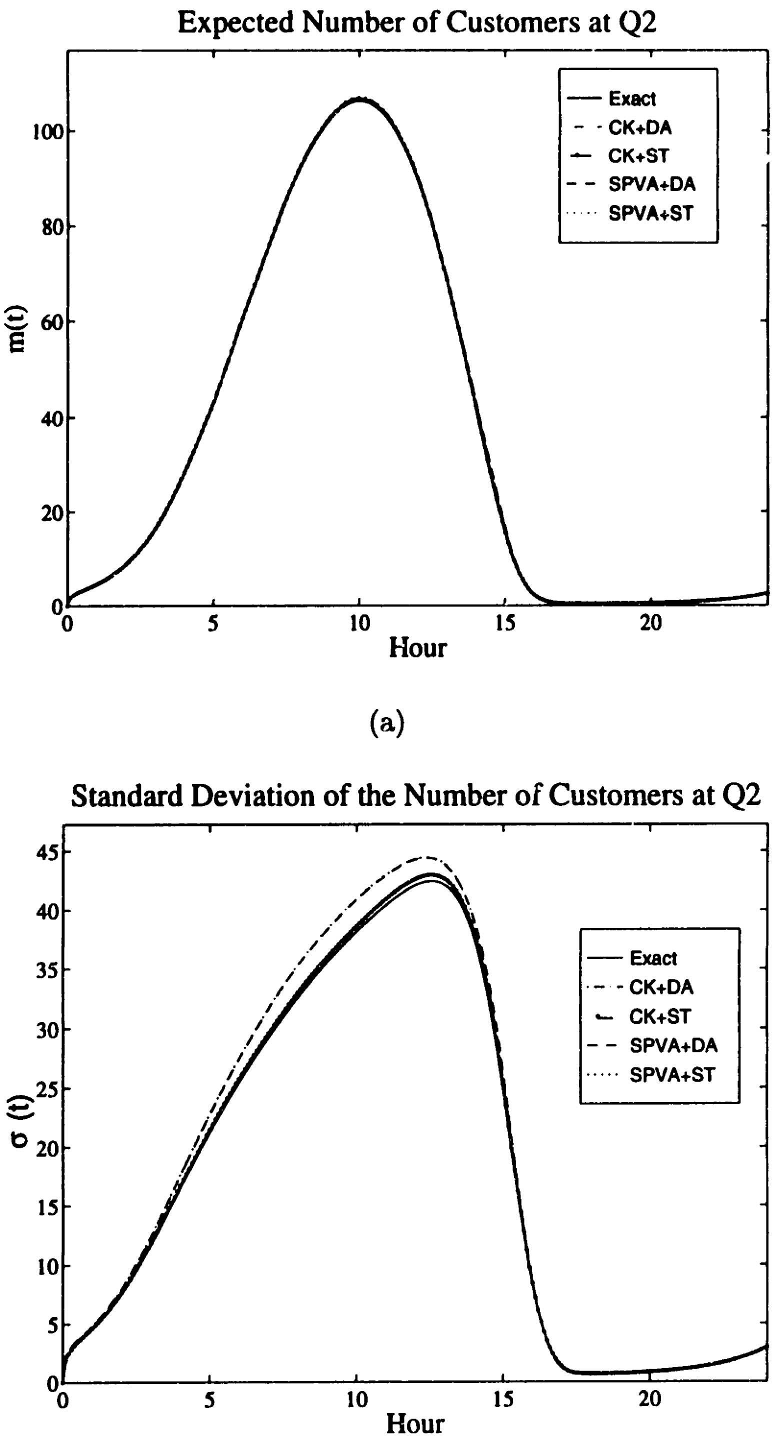


Figure 4-5: WPE of σ vs. Relative Amplitude and Maximum and Average Utilization at Q1 and Q2, for fixed $f = 0.5$.

We note here that Albin's experiments differ from ours in that she examined superimposed arrival processes composed of n streams, each having the specified squared coefficient of variation. In contrast, our experiments have a fraction f of total arrivals which come from one source with an unknown distribution, and $1 - f$ which have a nonstationary Poisson distribution.

The size of our errors are smaller in many cases than Albin's. As an example, we examine Case 14 of Table 4.4 with Q1 having a 3^{rd}-order Erlang distribution, f=0.5 and $\bar{\rho}_2 = 0.9$. The squared coefficient of variation of the interdeparture times from Q1 is unknown. DA and ST estimate $m(t)$ at Q2 within ±0.4. WPE of mean is 0.17. In contrast, the error in Albin's stationary systems with $\rho = 0.9$, $cv^2 = 0.5$ and $n = 2$ is about 2.5, or 38.5%. This leads us to hypothesize that the nonstationarity of the arrival processes **does** dampen the error. It is difficult to understand why, although Koopman [26] observed that the nonstationarity in the arrival process does not allow the full impact of differences which exist for stationary systems in equilibrium to take hold. Chapter 5 presents some observations and conjectures about the behavior of queueing systems with nonstationary arrival and/or service rates.

The difference between the propagation methods DA and ST is not significant. DA and ST produce similar error measures for $m(t)$, $\sigma(t)$, m^* and σ^*. The choice of queueing engine affects RE and WPE in the same way for both propagation methods. Figure 4-6 shows the four decomposition method estimates of $m(t)$ and $\sigma(t)$ at Q2, in addition to



(a)

(b)

Figure 4-6: Exact and Estimates of the Mean and Standard Deviation of the Number of Customers at Q2. Q1 with 3^{rd}-order Erlang Service-Time Distribution, Case 28.

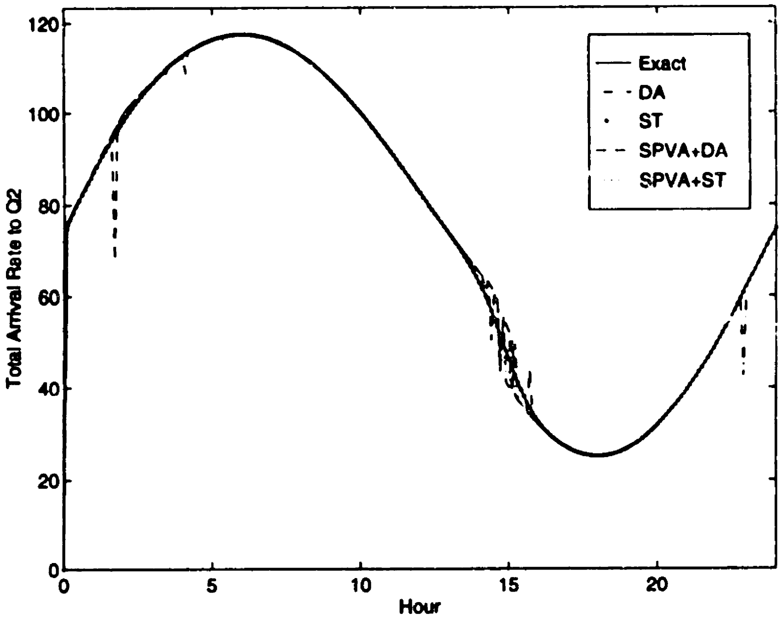


Figure 4-7: Exact and Estimates of $\hat{\lambda}_2(t)$. Q1 with 3^{rd}-order Erlang Service-Time Distribution, Case 28.

the exact values, for Case 28 with 3^{rd}-order Erlang distribution and $f = 0.3$. All four decomposition estimates appear virtually exact. DA and ST give better estimates of $m(t)$ and $\sigma(t)$ than do SPVA+DA and SPVA+ST. Figure 4-6 (b) shows that SPVA+DA and SPVA+ST overestimate $\sigma(t)$, similar to results in Chapter 3. Results from Chapter 3 show that SPVA overestimates $\sigma(t)$ in the case of $M(t)/M/1$ systems. Q2 is not an $M(t)/M/1$ system, but SPVA models it as one. DA and ST also overestimate $\sigma(t)$. One explanation is that DA and ST assume the arrival process to Q2 has squared coefficient of variation equal to one. However, we do not know what the actual squared coefficient of variation of the arrival process is. Q1 has a 3^{rd}-order Erlang service-time distribution, which probably means that the interdeparture times from Q1 to Q2 have squared coefficient of variation less than one. In this case, DA and ST assume a larger squared coefficient of variation in the arrival process, which would explain the overestimation of $\sigma(t)$.

Figure 4-7 shows the exact and decomposition method estimates for the total arrival rate to Q2. In this and all other cases tested, ST gives exact estimates of $\hat{\lambda}_2(t)$. SPVA+ST also produces good estimates of $\hat{\lambda}_2(t)$. The errors in estimating $m(t)$, $\sigma(t)$, m^* and σ^* at Q2 occur because the distribution of the arrival process is not Poisson. What is striking about Figure 4-7 are the jagged lines at around the 4-hours and 15-hours regions. These lines belong to the DA propagation method. This behavior manifests itself in both the

CK+DA and SPVA+DA methods in all cases examined. It results from the expansion and contraction of the expected system time at Q1 over the interval of analysis.

The relationship between the expansion of system time and arrival estimates in the DA Method can be explained as follows. As the expected number of customers at Q1 grows over an interval, so does the expected system time. As the system time grows, the spacing between the DA Method's estimates of future departure times grows. As the DA Method assigns future departure times to the look-up table λ_{ij}, the increased spacing between departure times may result in an entry of the look-up table being skipped, i.e., no departures are assigned to a particular look-up table entry. Therefore, an entry may remain empty. The jagged effect on the arrival rate comes about when an empty entry in the look-up table is located between two non-empty entries. In this case, there is a drop in the total departure rate from station i to station j for the interval of time represented by the empty entry. Since λ_{ij} is a component of the total arrival stream to station j, this drop will manifest itself as jags in the total arrival rate to station j. An equal and opposite effect occurs as the expected number in the system contracts after it reaches its peak. In this case, system time decreases over the interval. The result is that DA sometimes "doubles up" on assignments of departures to a particular entry in the look-up table. In summary, he effect of this jagged behavior on system statistics $m(t)$, $\sigma(t)$, m^* and σ^* appears insignificant; however, the effect on the time-dependent arrival rate is noticeable.

The Queueing Engine Effect on Decomposition Method Accuracy

How well do the SPVA+DA and SPVA+ST methods perform in our test cases? Clearly, they estimate $m(t)$ as well as the CK+DA and CK+ST methods, but they generally overestimate $\sigma(t)$. This is consistent with the behavior of SPVA observed in the individual queue results of Chapter 3. In those tests, SPVA gave good estimates of $m(t)$ in all 76 cases. However, SPVA overestimated $\sigma(t)$. This overestimation is particularly noticeable for $M(t)/M/1$ systems with high arrival rate nonstationarity, and high average and maximum utilization. The overestimation decreases rapidly for $M(t)/E_k/1$ as k increases, and for lower arrival rate nonstationarity, and lower average and maximum utilization. In our tandem-queue network, we model Q2 as an $M(t)/M/1$ station. Therefore, it is not surprising that SPVA should overestimate $\sigma(t)$ for Q2.

The service-time distribution and f are the primary factors which determine the decom-

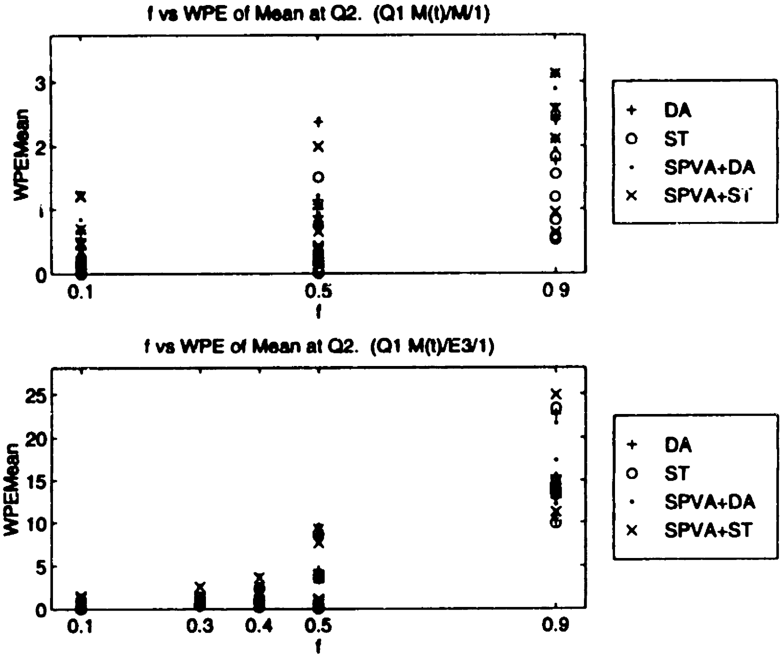


Figure 4-8: f vs. WPE of Mean at Q2. Q1 with Exponential (Top) and with 3^{rd}-order Erlang (Bottom) Service-Time Distribution

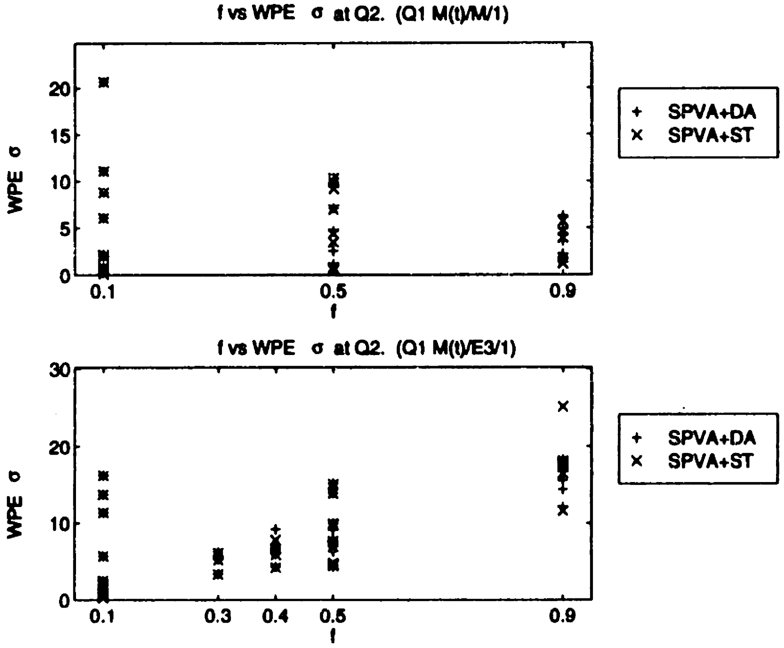


Figure 4-9: f vs. WPE of σ at Q2 for the SPVA+DA and SPVA+ST Decomposition Methods. Q1 with Exponential (Top) and with 3^{rd}-order Erlang (Bottom) Service-Time Distribution

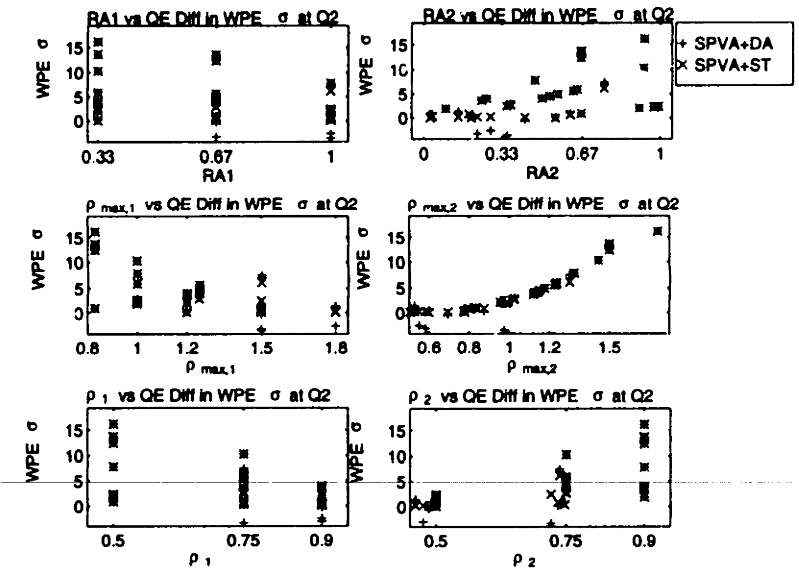


Figure 4-10: Case-By-Case Difference in Queueing Engine WPE of σ. Q1 had 3^{rd}-order Erlang Distribution.

position method accuracy when SPVA is used as the queueing engine. Figure 4-8 shows the WPE of mean for all four decomposition methods. The methods with SPVA as the queueing engine show errors on the same order as those with the CK queueing engine. Larger f results in a decrease in accuracy. Figure 4-9 shows the WPE of SPVA methods for σ vs. f. f does not affect the decomposition accuracy in isolation. In fact, the top plot shows the opposite trend from that of the CK+DA and CK+ST methods. However, we point out two things. First, in the top graph, SPVA models both Q1 and Q2 as $M(t)/M/1$ stations, hence overestimation of $\sigma(t)$ is expected. Second, the points represented by the four greatest errors for each SPVA method in both graphs correspond to cases 1, 11, 17 and 23. These systems have high average utilization ($\bar{\rho}_2 \geq 0.75$) and degree of nonstationarity ($0.67 \leq \widehat{RA}_2 \leq 0.93$) at Q2. In Chapter 3, we showed that the accuracy of SPVA estimates of $\sigma(t)$ decrease for $M(t)/M/1$ systems as average utilization and degree of nonstationarity increase. It is by chance that the choice of case parameters does not contain a case with high f and high $\bar{\rho}_2$ and $\widehat{RA}_2$, which might show a different trend in Figure 4-9. Figure 4-10 shows the secondary parameters vs. case-by-case differences between the WPE of σ for SPVA+DA with DA, and SPVA+ST with ST, for the 3^{rd}-order Erlang distribution at Q2. It clearly shows that increasing $\rho_{\max,2}$ increases the WPE of the SPVA method as compared to CK methods. Combinations of high $\widehat{RA}_2$ and high $\bar{\rho}_2$ also yield higher SPVA WPE than CK. These observations are consistent with the SPVA decrease in $\sigma(t)$ quality for $M(t)/M/1$

systems. We conclude that in both cases the problem is due to the queueing engine error, and not propagation method error.

The SPVA methods give good estimates of $m(t)$, $\sigma(t)$, m^* and σ^* for Q2 when the parameter choice for the queues in isolation will yield good results. With respect to the decision about whether to use CK or SPVA, SPVA enjoys a speed advantage. It gives excellent estimates of $m(t)$, $\sigma(t)$, m^* and σ^* when the Erlang order is greater than or equal to three. Therefore, use SPVA when possible.

4.4 The Approximate Network Delays Model

One of the motivations for this work is to validate some of the premises of a model that we have developed in parallel with this research over the past three years, the Approximate Network Delays (AND) Model [29]. The AND Model employs a decomposition approach for approximate analysis of the national airport system. The AND Model is macroscopic and is best-suited for use in strategic, or "policy analysis," studies in which the primary objective is to assess the relative performance of a wide range of alternatives. For this reason, it aspires to be very fast, in terms of both input preparation and execution times, so it can be used to explore a large number of "scenarios." The model can also be used as a screening device to identify the few most promising among many alternative courses of action, which can then be studied in detail through more "microscopic" models. Specifically, AND quantifies the system-wide changes in delay as a result of local changes at airports in the network. These local changes may take the form of significant expansion of capacity at a particular airport, or of a forecasted growth in operations.

As mentioned in Chapter 1, the network of airports is weakly connected. The fraction of arrivals to any airport which are departures from any other single airport is less than 0.1. A stronger link exists between landings and takeoffs at a single airport. In this case, the fraction is about 0.3–0.4. However, one can argue that variable gate times, airline scheduling practices, and late arrivals and departures lead to a very substantial decoupling of this link. Our tandem-queue results indicate that these attributes of the network of airports satisfy the "weakly-connected" assumption of a decomposition method, and thus the assumptions of the AND Model. We conclude the decomposition approach of AND is appropriate for a network of airports.

The AND model was conceived at MIT where a prototype involving 3 airports was developed [29]. In 1994, under a co-operative research project between MIT and the MITRE Corporation, the AND model was generalized to accept any specified number of interconnected airports in the range from 2 to 58. In fact, a current implementation of AND models the principal 58 commercial airports in the United States as the elements of the network. Other airport networks can be modeled, if the appropriate data are available.

The AND Model differs from another airport model developed at MIT by Peterson et al. [38], which we refer to as the "Peterson Model" in this discussion. The Peterson Model is a quasi-deterministic multi-class queueing model which examines hub airports in a hub-and-spoke network. Specifically, it tries to capture the effect of extreme peaks in demand at a hub airport and its effect on the spokes and other hub airports with which it is tightly connected. The special structure of the Peterson Model limits its applicability on a national scale, which indicates the difference between the AND and Peterson Models. Another difference between AND and the Peterson Models is the assumptions of the queueing model used by each. The Peterson Model assumes arrivals are time-varying but deterministic. Airport capacity varies according to an underlying discrete-state discrete-transition Markov model, but the capacity associated with each state of the Markov Chain is deterministic. In contrast, the AND Model assumes an arrival process which is nonstationary Poisson. Airport capacity in the AND Model also varies with time and is stochastic.

The AND model employs a combination of analytical and algorithmic approaches to estimate local and system-wide delays in a network. The model iterates between the analytical and algorithmic parts. Figure 4-11 illustrates the relationship between the analytical and algorithmic parts of AND. The analytical part employs the DELAYS model to compute delays at individual airports. (See Section 2.2.2 for a description of DELAYS, and Chapter 3 for performance analysis.) The delays propagation part "propagates" delays from each individual airport to the rest of the network by tracing how individual aircraft are affected by local delays. The current implementation of AND uses the DA propagation method. As an example, consider three airports, A, B and C, in a network. If an aircraft flying from A to B to C suffers a serious delay on landing at B on a particular day, this may also affect its expected time of departure from B, as well as its expected time of arrival at C late in that day. The analytical part of AND will compute the delay suffered on landing at B by solving the dynamic queueing model that describes congestion at B, based on the

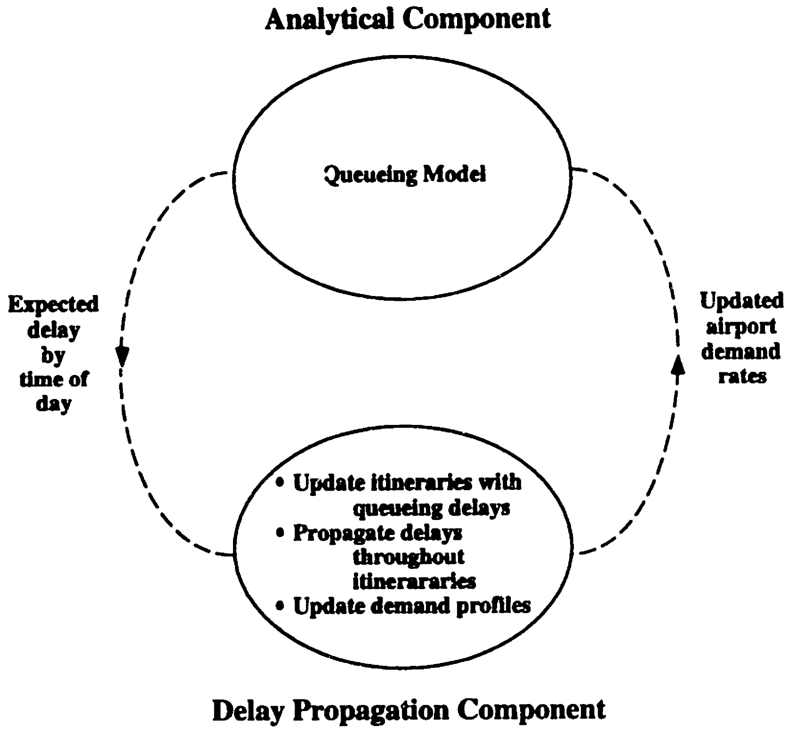

Analytical Component
Queueing Model
Expected delay by time of day
Updated airport demand rates
• Update itineraries with queueing delays
• Propagate delays throughout itinerararies
• Update demand profiles
Delay Propagation Component

Figure 4-11: Overview of AND Model Components

(dynamic) demand and capacity profiles at B. The delay propagation part will then revise the expected departure time of the aircraft from B and its expected arrival time at C, based on

1. the amount of delay the aircraft suffers upon arrival at B,
2. the amount of that delay that can be "absorbed" on the ground at B due to any "slack" in the turn-around time allotted to that aircraft by the airline's schedule, and
3. the flight time required to go from B to C.

It can be seen that adjustments of this kind to the expected arrival and departure times will affect demand profiles at the "downstream" airports later in the day. This may necessitate revising the delay estimates at individual airports by going back to the DELAYS model. The AND model thus iterates between revisions of the demand profiles at individual airports and computation of delays at the airports until the entire time period of interest (one or more days of operations) is completed.

The model itself actually consists of: an AND pre-processor, which uses airline schedules to prepare full itineraries for every aircraft in the network along with dynamic (i.e., by time-of-day) demand and capacity profiles for all the individual airports in the system; and of the AND Analytic/Delay Propagation Engine that computes and propagates delays in the manner outlined in the previous paragraph.

Two versions of AND exist, a serial model and a parallel model. Both versions run on SUN SPARCstation 10 workstations. The parallel version exploits networks of workstations or multiprocessor workstations to speed up model execution by a factor of approximately 2 at this point. This is accomplished primarily by taking advantage of the decomposition approach to distribute among the parallel processors the task of computing delays at the individual airports. Further improvements along these lines may be achievable in the future.

The computational performance of the AND model is very efficient, compared to that of large-scale simulations modeling network delay propagation in the ATM/airport sector. For example, the execution of a recent run involving a complete day of operations at the 58 principal airports in the United States took about 20 minutes on the serial version of the model and about half this time in the parallel version. A total of about 50,000 landings and takeoffs took place at these airports during the day in question. It should be noted that the model computes the entire probability distribution for the number of aircraft at each airport at any time t. In other words the fundamental quantity of interest is $P_{i,l}(t)$, i.e., the probability that there will be l aircraft waiting to land or takeoff at airport i at time t, for all possible values of l, i and t.

Figure 4-12 shows the CPU usage of a parallel implementation version of AND across the real time axis. This graph shows three distinct behaviors. First, the series of low bumps at the beginning of the computation correspond to the AND pre-processor preparing the aircraft itineraries and airport demand and capacity as input to the model. The middle series of high, telescoping peaks corresponds to the CPU effort in solving the queueing engines for the individual airports. These telescope because, with each call to the queueing engine, the delay at airports is propagated through the network later in the day, until the very end of the period of analysis. Hence, only the first call to the queueing engines starts from time 0. The subsequent calls start at later points in time corresponding to how far into the day the delay has been propagated. The dips between the telescoping peaks correspond to the propagation algorithm. The final peak at the end of the graph corresponds to the calculation of statistics collected by AND.

The AND model generates a variety of output statistics, which can be aggregated for the entire network or be specific by airport and even by aircraft. Its principal emphasis is

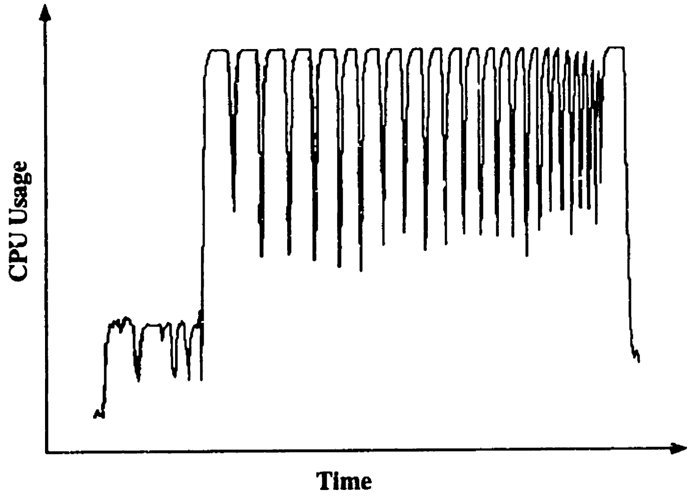


Figure 4-12: CPU Usage for an AND Parallel Model Run

on delay statistics, both in terms of additional time needed to complete a flight leg ("true delay") and in terms of the deviation from the flight's schedule ("effective delay"). Note that true delay may be different from effective delay because airlines often include beforehand in their published flight schedules an allowance for potential deiays.

The AND model runs with a mouse-driven graphical user interface (GUI). Through the GUI the user can select different scenarios for execution, create new scenarios or modify existing ones. A "Capacity Editor" is included with the GUI to facilitate the modification of the capacity profiles of the airports in the network as desired. A map display allows the user to add or subtract airports to/from the network.

Over the past few months, the MITRE Corporation has tried to compare AND Model results with outputs of simulation models of the National Airspace System [4]. The simulation models compared include the National Airspace System Performance Capability (NASPAC) Simulation Modeling System (SMS). The NASPAC SMS was developed by MITRE in the late 1980's at the request of the FAA. The model is now being used as a benchmark against which other models are compared. Preliminary results indicate that when NASPAC is used with all of its features, NASPAC and AND give comparable results.

In conclusion, the AND model represents an important new development in the area of system-wide modeling of airport operations. It could also be extended in the future to include delay analyses that would also consider selected congested en route sectors.

By itself, AND can be a viable alternative to simulation models for approximate policy analyses that must explore a large number of alternatives. It may in fact be far preferable to simulations, because of its speed, robustness and, most important, its computation of probability distributions, not point estimates. For detailed design studies that aim at a microscopic level of detail and high accuracy, a model such as AND can still be highly valuable as a screening tool or pre-processor that would help identify a small number of alternatives that deserve further evaluation with a more detailed simulation model such as NASPAC.

4.5 Summary

In this chapter we investigated a decomposition method for approximate analysis of networks of queues with time-varying arrival rates. This approach is an alternative to simulation and to modeling the system as a network of $M(t)/M(t)/1$ stations. Both of these methods are infeasible for almost all cases with the exception of the simplest networks. To simulate a network, one must perform many simulations to obtain statistically valid results. This, combined with the computation time per simulation, make it an infeasible approach for high-level, network-wide analysis. If we model the network as an open network of $M(t)/M(t)/1$ stations with probabilistic routing and instantaneous travel times between stations, we can write down the CK equations describing the network and solve the system exactly using numerical methods. However, we showed that the CK equations are complex even for a simple tandem-queue network. Furthermore, the memory requirements of such a system, and the CPU time needed to solve it, make this approach infeasible for almost all cases.

The decomposition methods investigated in this chapter offer good alternatives for approximate analysis of networks of queues with time-varying arrival rates. A decomposition approach may offer the only feasible method of analysis of such a network. First, the memory requirements are linear in the number of stations in the network. This attribute permits analysis of networks with a large number of stations. Second, the CPU time required to solve the decomposition methods is two to three orders of magnitudes less than the systems mentioned above, if they can be solved. Third, the equations describing the relationships among the queues are simple. This reduces the complexity of the network.

When will a decomposition method be accurate? It is difficult to generalize based on our tandem-queue results, but we get at least an indication of the most important factors affecting decomposition accuracy. The primary factors determining the decomposition accuracy at a station are f, the fraction of arrivals to a station which are departures from one single other station, and the service-time distributions at the stations. Larger f leads to decreased accuracy. As the coefficient of variation of the service-time distribution becomes smaller, the accuracy also decreases.

Specifically, even for $f \approx 0.5$, most cases of CK+DA and CK+ST have RE and WPE of the mean and of σ which are less than 5%. The same is true of SPVA+DA and SPVA+ST for WPE and RE of mean. When the service-time distribution changes from exponential to 3^{rd}-order Erlang, the increase in WPE of mean for $f = 0.5$ is less than 10% for all four decomposition combinations. For $f = 0.1$, the increase is negligible. But for $f = 0.9$, the increase in WPE is at least 10% when the service-time distribution changes.

How does one choose a method to analyze a network of queues with nonstationary arrivals? If $f = 0.9$, one should clearly not use a decomposition method. If $f \leq 0.5$, the first and second level decomposition methods are good. How does one choose which combination to use? The choice of Queueing Model depends on the system being modeled. It should be based on the knowledge of the circumstances under which the Queueing Engine performs within the desired accuracy, and the types of statistics desired. One of the key attributes of the Queueing Engine is that it be extremely fast. In our research, we used the CK equations and the SPVA approximation method. SPVA enjoys speed advantages over the CK equations. We showed that SPVA is very accurate for approximating many systems in this chapter and in Chapter 3. The two propagation methods investigated in this chapter, DA and ST, also produced estimates of the same level of quality. The ST method produces smoother estimates of departure rates from the stations in the network than does the DA method. DA's estimates of departure rates from the stations are not as good, but that does not seem to affect the system measures in a significant way.

Chapter 5

On the Timing of the Peak Mean and Variance for the Number of Customers in an M(t)/M(t)/1 System

The research in this thesis focuses on systems with arrival and/or service rates which vary with time. In practice, when managers design or analyze facilities with strongly time-varying demand and capacity, they commonly focus on the performance of the facility during *peak utilization:* that period during which the facility is busiest. For example, airport authorities strive to have their facilities designed so that aircraft and passenger delays in the peak periods are within tolerable limits. Therefore it is important to develop an understanding of when a peak in congestion (= the expected number of customers in the system) will occur in relation to a peak in the system utilization. The difference between the time at which the system utilization peaks and the time at which a system performance measure (for example, number in the system) reaches its highest value is called the *time lag.* The time lag can be a matter of minutes or hours, depending on the type of queueing system, the average utilization rate, and how much the utilization peak rises over the average utilization level. In addition, insight into the relationship between the time-dependent mean, $m(t)$, and variance, $v(t)$, for the number of customers in the system may allow more effective management of congestion at facilities where demand and capacity vary strongly with time.

Researchers have already begun addressing this time lag issue in queueing systems with nonstationary arrival and/or service rates. In the case of oversaturated queues, Newell [33] conjectured that the peak mean number of customers in the system should occur at about the end of the period of oversaturation, without making assumptions about the particular form of the arrival process or service-time distribution. Using a diffusion approximation to a nonstationary queue, he observed that the maximum variance for the number of customers in the system occurs later than the maximum mean number in the system, based on numerical calculations. Green and Kolesar [14] and Green et al. [16] addressed the behavior of several performance measures of periodic $M(t)/M/s$ queueing systems. They noted that not only does the arrival rate peak *not* coincide with peaks in other measures such as expected queue length and probability of delay, but the measures also behave differently from one another. They also noted that as the *event frequency* (the number of arrivals or service completions per cycle) increases, the lag between the peak in the arrival rate and the mean number in queue *decreases*. Eick et al. [11, 12] examined $M(t)/G/\infty$ queueing systems. In their breakthrough research, they used exact results for $m(t)$ derived by Palm [37] and Khintchine [22] to find exact, closed-form expressions for $m(t)$, the extreme values of $m(t)$, and for the time lag between the peak in the arrival rate and the number of customers in the system, in the case of a sinusoidal arrival rate function. These results can also be used to approximate a finite-server system if the arrival rate does not approach or exceed the service rate. This is not the case for some real-world systems with strongly time-varying arrival and service rates. For such systems, an approximation based on $M(t)/G/\infty$ would likely underestimate the actual time lag. Eick et al. also prove that the mean number of customers in an $M(t)/G/\infty$ system with a sinusoidal arrival rate is symmetric about its extremes, i.e., if an extreme occurs at time t_m^*, then $m(t_m^* - t) = m(t_m^* + t)$ for all t. In contrast, Green et al. showed that no system performance measures for finite server systems (1 - 12 servers) are symmetric about their extremes.

Our research into $M(t)/M(t)/1$ systems is strongly motivated by real-world applications. Common characteristics of such systems are arrival and service rates that are highly time-dependent, but do typically change in a periodic and smooth manner, with regularly occurring intervals during which the arrival rate is low enough that the queue usually empties. Thus, we are interested in the systems that are stable, in the sense that the system returns virtually to rest at some point during every period. It will be convenient to think of the period length as 24 hours, but it could of course be different. The assumption of a nonstationary Poisson arrival process is reasonable for many real-world applications. The assumption of exponentially distributed service times is typically less reasonable; we make this assumption for analytical tractability. However, our computational results lead us to believe that our results also hold for certain more realistic service-time distributions.

The purpose of this chapter is to provide theoretical insight and computational results for the time lag between a peak in the system utilization and corresponding peaks in the mean and variance (and, of course, the standard deviation) of the number in the system in stable, periodic $M(t)/M(t)/1$ systems. Specifically, we provide a necessary condition for when local extremes of the mean number in the system will occur, and a condition for the local peak in the variance to occur strictly after a corresponding local mean peak, in Section 5.1. Depending on the utilization function, there may not be a one-to-one correspondence between peaks in the system utilization and peaks in the mean and variance, over a single period. To avoid this complication, we will interpret our results in Section 5.1 under the assumption that the instantaneous utilization peaks exactly once per period. In Section 5.1, we show that if the instantaneous utilization during some interval exceeds one, then the next local peak in the mean number in the system will occur strictly after the end of that interval. In Section 5.2, we present computational results supporting a conjecture: in a periodic, stable $M(t)/M(t)/1$ queue with a smooth utilization function that peaks once per period, the mean peaks after the utilization and the variance peaks after the mean. In Section 5.3, we discuss the issue of multiple utilization peaks over a single period, corresponding to the multiple rush hours encountered in some real-world systems.

5.1 Extremal Conditions for the Mean and Variance of the Number of Customers in the System

A time lag between the peak in the arrival rate and the peak in the system congestion has been observed in $M(t)/M/s$, $M(t)/E_k/1$, $M(t)/G/\infty$, and other types of nonstationary queueing systems. The top graph in Figure 5-1 shows an example of this lag for the mean of an $M(t)/M/1$ queueing system in which $\lambda(t) = 75 + 50\sin(2\pi/24)$ and $\overline{\mu} = 100$. The bottom graph in Figure 5-1 shows the standard deviation for the number of customers in the system for an $M(t)/M/1$ queueing system with the same parameters as in the top

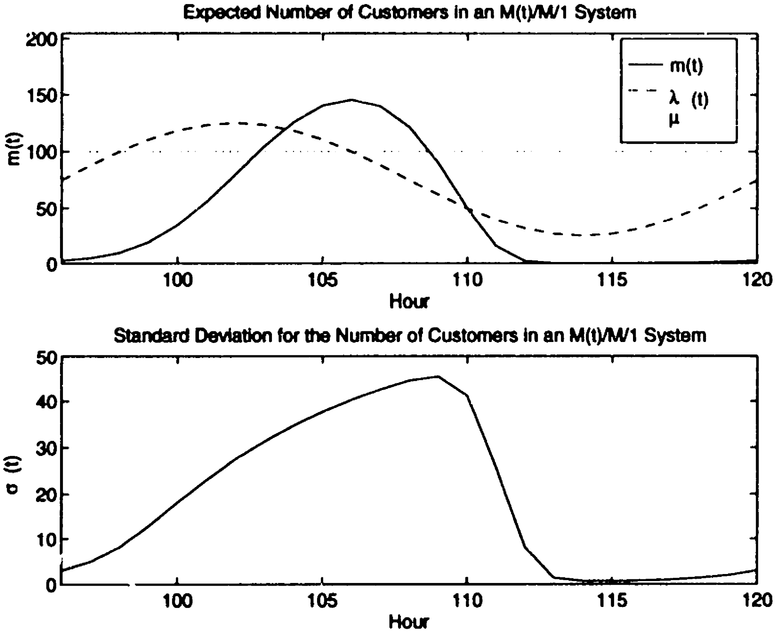


Figure 5-1: Example of Time Lag in the Mean and Standard Deviation for the Number in the System of an $M(t)/M/1$ System

graph. The peak in the arrival rate occurs at $t = 102$, the peak in the mean at about $t = 106$, and the peak in the standard deviation at about $t = 109$. Our computational results indicate that the standard deviation for the number in the system peaks later than the mean number in the system does for all the nonstationary single-server queueing systems with infinite queueing capacity which we have examined. In this section, we focus on the $M(t)/M(t)/1$ system and establish conditions for the peak in the mean and variance for the number of customers in the system, as well as the relationship between the two.

Notation: We use the same notation as appears in Section 2.1. We supplement it with the following. Primes will be used to denote derivatives, e.g., $m'(t) = dm(t)/dt$. The time at which m^* is achieved will be denoted t^*_m. We define t^*_σ and t^*_v similarly.

After proving the following preliminary lemma, we will derive conditions for when the expected number of customers in the system peaks.

Lemma 1 *In an $M(t)/M(t)/1$ queueing system, if $P_0(0) > 0$, then $P_0(t) > 0$ for all $t \geq 0$.*

Proof: The familiar Chapman-Kolmogorov forward equation for state 0 in an $M(t)/M(t)/1$ system is

$$P_0'(t) = -\lambda(t)P_0(t) + \mu(t)P_1(t).$$

Let $\hat{P}_0(0) = P_0(0)$, where $\hat{P}_0(t)$ satisfies

$$\hat{P}_0'(t) = -\lambda(t)\hat{P}_0(t). \tag{5.1}$$

The quantity $\hat{P}_0(t)$ will be no greater than $P_0(t)$ for all $t \geq 0$, as we now show. Since $\mu(t)P_1(t) \geq 0$ it follows that $P_0'(t) \geq \hat{P}_0'(t)$ for all $t \geq 0$. Integrating on both sides of $P_0'(t) \geq \hat{P}_0'(t)$ we obtain

$$P_0(t) \geq \hat{P}_0(t) \text{ for all } t \geq 0. \tag{5.2}$$

The solution to equation (5.1) is (see, e.g., Luenberger [28]):

$$\hat{P}_0(t) = P_0(0) \exp\left[-\int_{\tau=0}^{t} \lambda(\tau)d\tau\right]$$

Since $P_0(0) > 0$ and $\exp\left[-\int_{\tau=0}^{t} \lambda(\tau)d\tau\right] > 0$ for all $t \geq 0$, we have that $\hat{P}_0(t) > 0$ for all $t \geq 0$. Finally, inequality (5.2) implies that $P_0(t) > 0 \; \forall t \geq 0$. ■

Theorem 1 *In an $M(t)/M(t)/1$ Queueing System, a necessary condition for the times at which the expected number of customers in the system $m(t)$ takes on its local extreme values is:*

$$\frac{\lambda(t)}{\mu(t)} = 1 - P_0(t) \tag{5.3}$$

Proof: We use the expression for $m'(t)$ derived by Clarke [8] in 1956 and used by Rothkopf and Oren [46] in the derivation of their closure approximation for the nonstationary $M/M/s$ queue[1]:

$$m'(t) = \lambda(t) - \mu(t)\left[1 - P_0(t)\right] = \mu(t)\left[\rho(t) - (1 - P_0(t))\right] \tag{5.4}$$

To find when $m(t)$ achieves its extreme values, we simply set $m'(t) = 0$ and find the following condition:

$$m'(t) = 0 \;\Leftrightarrow\; \frac{\lambda(t)}{\mu(t)} = 1 - P_0(t) \text{ (if } \mu(t) > 0) \tag{5.5}$$

Equation (5.5) must hold for $m(t)$ to achieve a local maximum, m^*, or local minimum, m_*. ■

[1] The expression for $m'(t)$ which appears in Clark [8] requires that the operations of differentiation and infinite summation be switched. A proof of this appears in Appendix C.

Eick et al. [11] specialized their $M(t)/G/\infty$ results to the case of exponential service-times and proved a result for $M(t)/M/\infty$ systems which looks similar to that of Theorem 1: $m'(t) = \lambda(t) - \mu m(t)$. In the case of sinusoidal-Poisson arrivals, a closed-form expression for $m(t)$ exists (in the infinite-server case), and the exact time at which m^* occurs , and the value of m^*, can be found.

Theorem 1 has the following important corollary.

Corollary 1 *Suppose that $P_0(0) > 0$ and $\rho(t) > 1$ for $t \in (t_1, t_2)$ for an $M(t)/M(t)/1$ system. Then the first congestion peak m^* after t_1 will occur after t_2, i.e., $t_m^* > t_2$.*

Proof: From (5.4), we see that when $\rho(t) > 1$, $m'(t) = \mu(t)\left[\rho(t) - (1 - P_0(t))\right] \geq \mu(t)(\rho(t) - 1) > 0$. Therefore, $m(t)$ does not peak while $\rho(t) > 1$, i.e., $t_m^* \geq t_2$. By Lemma 1, $P_0(t_2) > 0$, which implies that $1 - P_0(t_2) < 1 = \rho(t_2)$. Thus, the condition of equation (5.5) is not met, so m^* does not occur at t_2. We conclude that $t_m^* > t_2$. ■

Figures 5-2(a), 5-2(b), and 5-2(c) illustrate Corollary 1. They correspond to an $M(t)/M/1$ system with $\lambda(t) = 90 + 30\sin(2\pi/24)$ and $\overline{\mu} = 100$. Figures 5-2(a), 5-2(b), and 5-2(c) depict $\rho(t), m'(t)$, and $m(t)$, respectively, over one period of $\lambda(t)$. The times t_1 and t_2 mark the beginning and the end of the period during which $\rho(t) > 1$, t_3 is the time t_m^* at which $m(t)$ peaks, and t_4 is the time at which the minimum value m_* of $m(t)$ is achieved. Note that $t_3 - t_2$ is very small in this particular case, but positive nevertheless.

In our work with nonstationary queueing systems, we have also observed that the behavior of the variance and standard deviation for the number of customers in the system can be quite different from that of the mean. One of the most salient differences is that the variance and standard deviation peak later – sometimes much later – than the mean. Theorem 2 establishes a condition under which the variance peak v^* occurs later than the peak in the mean m^*. Let t_v^* be the time at which v^* is achieved.

Theorem 2 *In an $M(t)/M(t)/1$ system, $t_v^* > t_m^*$ iff*

$$\frac{1}{m^* + 1} > P_0(t_m^*). \tag{5.6}$$

We remark that in all our numerical computations to date, $t_v^* > t_m^*$.

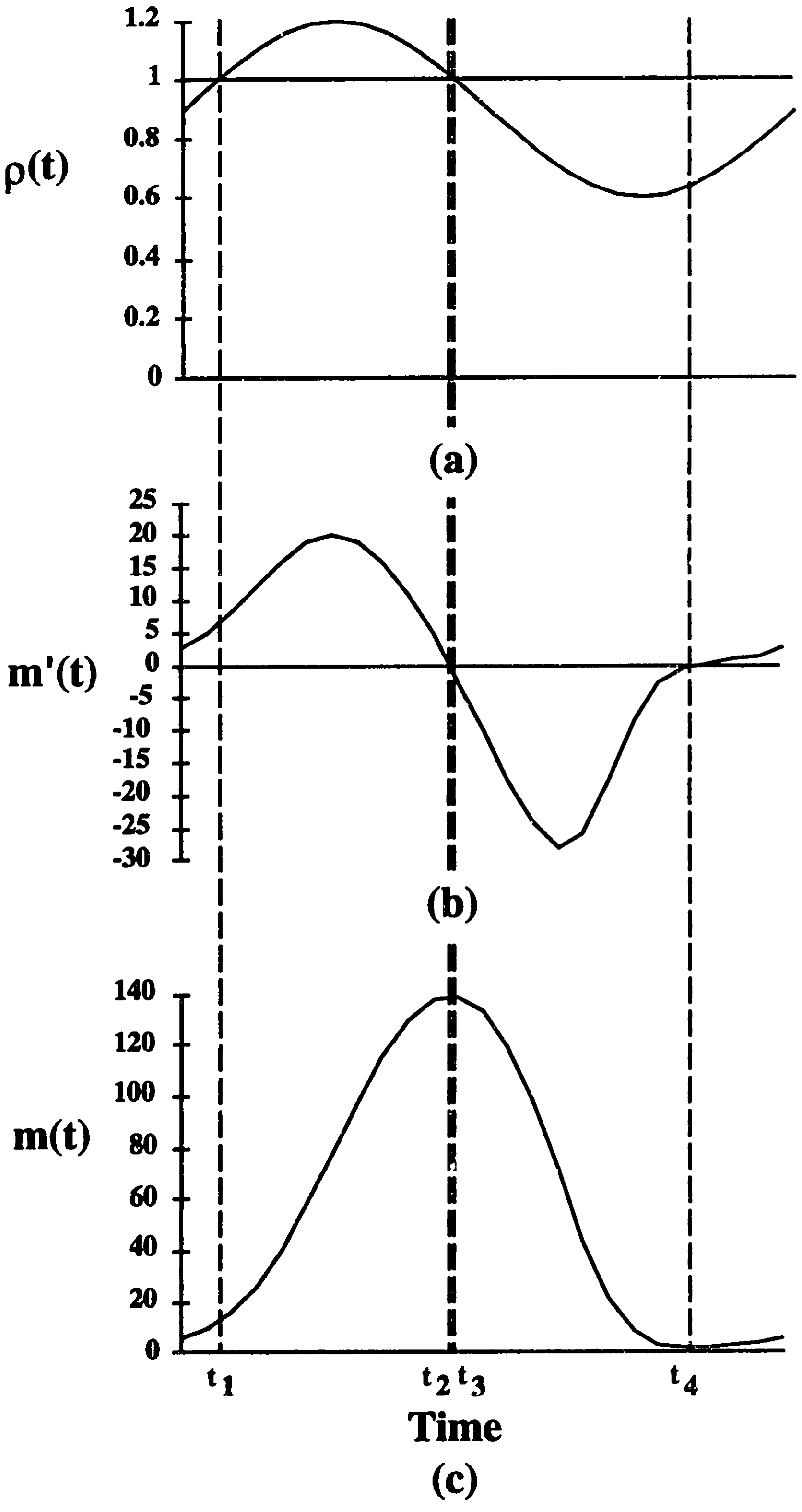


Figure 5-2: $\rho(t)$, $m'(t)$, and $m(t)$ for an $M(t)/M/1$ System

Proof: The proof consists of showing that $v'(t_m^*) > 0$ iff condition (5.6) holds. We use the expression for $v'(t)$ derived by Clarke [8],

$$v'(t) = \mu(t)\left[\frac{\lambda(t)}{\mu(t)} + 1 - P_0(t)[1 + 2m(t)]\right] \tag{5.7}$$

At the time t_m^*, when $m(t)$ achieves its peak m^*, the relation $\frac{\lambda(t_m^*)}{\mu(t_m^*)} = 1 - P_0(t_m^*)$ will hold, by Theorem 1. Substituting this relation into equation (5.7), we get:

$$v'(t_m^*) \quad = \quad 2\mu(t_m^*)\,(1 - P_0(t_m^*)[m^* + 1])\,. \tag{5.8}$$

Assuming that $\mu(t_m^*) > 0$, the right-hand-side of equation (5.8) will be positive iff $1 - P_0(t_m^*)[m^* + 1] > 0$, that is, iff $\frac{1}{m^*+1} > P_0(t_m^*)$ ■

We note that $\frac{1}{m^*+1}$ is never greater than one so condition (5.6) is not trivially true. We also note that for stationary $M/M/1$ systems in steady state, $m^* = \rho/(1-\rho)$. Therefore, in this case, $\frac{1}{m^*+1} = 1 - \rho = P_0$, i.e., the two sides of the inequality (5.6) are equal. We note that a completely equivalent condition to inequality (5.6) is: $\rho(t_m^*) > \frac{m^*}{m^*+1}$.

In this section, we analyzed the $M(t)/M(t)/1$ queueing system. We obtained results consistent with the Theorems 1 and 2 in Section 3.4.2 for the $M(t)/E_k/1$ system and for the SPVA, DELAYS, INTERP and $M(t)/D/1$ methods.

5.2 Computational Results for M(t)/M/1 Systems

In this section we present some of the computational results for $M(t)/M/1$ queueing systems which led us to investigate the time lags for the mean and variance for the number of customers in the system and to derive the results of Section 5.1. We cull these results from the test cases of Section 3.4.2 which contained 19 different cases for the $M(t)/M/1$ system. Each case had $\mu(t) = \overline{\mu}$. We also present results supporting the following conjecture:

Conjecture 1 *Suppose that the utilization function $\rho(t) = \lambda(t)/\mu(t)$ for a stable $M(t)/M(t)/1$ system is smooth, continuous, and periodic, with one peak per period. Then that peak will induce peaks in $m(t)$ and $v(t)$, with the mean peak occurring strictly later than the utilization peak and the variance peak occurring strictly later than the mean peak.*

In Table 5.1, we provide data which confirm Theorems 1 and 2 and Corollary 1. Note that because the ODE solver takes discrete time steps, $m'(t^*_m)$ is not exactly equal to 0. In Table 5.1, column 7, we show how small the difference $t^*_m - t_2$ is for the cases in which $\rho_{\max} > 1$, where (t_1, t_2) is the interval during which $\rho_{\max}$ exceeds 1.0. For the cases in which $\rho_{\max} \leq 1$, N/A (= Not Applicable) appears in this column. Note that in the cases in which $\overline{\mu} = 100$ and $\rho_{\max} > 1$, which correspond to heavily-stressed systems, there was no discernible difference (to two decimal places) between t^*_m and t_2. Although $P_0(t^*_m)$ is positive, it is extremely small in these cases, as can be seen in column 9 of Table 5.1. Based on condition (5.4) we expect $t^*_m - t_2$ to be very small when $P_0(t^*_m) \approx 0$.

In Tables 5.1 and 5.2, we provide support for Conjecture 1. Our computational results are for a stationary service rate, so only the arrival rate changes with time, but this is immaterial as far as the validity of the conjecture[2]. There are three sets of values which support Conjecture 1. First, condition (5.6) of Theorem 2 holds in all 19 cases examined. Second, also in all 19 cases examined, $v'(t^*_m) > 0$, sometimes $v'(t^*_m) \gg 0$, indicating that $v(t)$ is still increasing at the moment that $m(t)$ peaks. Third, $\sigma(t)$ peaked later than $m(t)$, sometimes by several hours, in all 19 cases. Table 5.2 lists the time lag between the peak in the arrival rate and the mean in column 4, and between the peak in the arrival rate and standard deviation in column 5. We note that in all the computational results for the other nonstationary single-server systems we mentioned at the end of Section 5.1, $\sigma(t)$ peaked later than $m(t)$, leading us to believe that this behavior may be typical of a broader class of nonstationary queueing systems, with service-time distributions which are not necessarily exponential.

Figures 5-3, 5-4 and 5-5 are graphical representations of Table 5.2 for the 19 cases examined. Figures 5-3 and 5-4 plot $\rho_{\max}$ vs. the time lag between the peak in the arrival rate and the times at which m^* and σ^* occur, respectively. The size of the time lag of m^* is smaller than that of σ^*. Figure 5-5 plots $\rho_{\max}$ vs. $t^*_\sigma - t^*_m$ and shows that t^*_σ exceeded t^*_m in our test cases. Furthermore, the time lag of σ^* increased significantly faster than that of m^* for cases in which $\rho_{\max} > 1$. This observation again suggests that $m(t)$ behaves differently from $v(t)$ and $\sigma(t)$, in significant ways.

[2] An $M(t)/M(t)/1$ system can be transformed into an $M(\tau)/M/1$ system, via the time transformation $\tau(t) = \int_0^t \mu(s)ds$ (see [8]). The ordering of the peaks in $\rho(t)$, $m(t)$, and $v(t)$ is not affected by this time transformation.

$\bar{\mu}$	$\bar{\rho}$	RA	t_m^*	$m'(t_m^*)$	$v'(t_m^*)$	$t_m^* - t_2$	$\frac{1}{m^*+1}$	$P_0(t_m^*)$
100	0.5	$\frac{1}{3}$	30.2	0.0038	0.03	N/A	0.33352	0.33346
		$\frac{2}{3}$	30.5	-0.0006	0.11	N/A	0.1695	0.1694
		1	103.5	0.0196	43.21	N/A	0.0494	0.0388
	0.75	$\frac{1}{3}$	104.0	-0.0022	42.49	N/A	0.0429	0.0338
		$\frac{2}{3}$	106.0	0.0014	199.61	0.00	0.0068	0.0000
		1	154.7	0.0355	200.04	0.00	0.0031	0.0000
	0.9	$\frac{1}{3}$	154.7	0.0196	198.50	0.00	0.0071	0.0001
		$\frac{2}{3}$	155.4	0.0061	200.01	0.00	0.0028	0.0000
		1	227.6	-0.0068	199.99	0.00	0.0171	0.0000
10	0.5	$\frac{1}{3}$	55.1	-0.0032	0.14	N/A	0.3424	0.3399
		$\frac{2}{3}$	103.7	-0.0008	1.09	N/A	0.2105	0.1991
		1	152.4	0.0013	4.15	N/A	0.1207	0.0956
	0.7	1	346.3	0.0050	17.59	0.02	0.0340	0.0041
	0.75	$\frac{1}{3}$	153.2	0.0006	4.17	N/A	0.1046	0.0828
		$\frac{2}{3}$	226.1	0.0008	13.66	0.13	0.0472	0.0150
		1	274.7	-0.0023	19.06	0.01	0.0269	0.0013
	0.9	$\frac{1}{3}$	226.9	0.0014	12.72	0.21	0.0431	0.0157
		$\frac{2}{3}$	347.4	-0.0073	19.33	0.01	0.0231	0.0008
		1	347.6	0.0111	20.00	0.00	0.0154	0.0000

Table 5.1: Numerical results: derivatives of $m(t)$ and $v(t)$ at the time t_m^* when $m(t)$ peaks.

$\bar{\mu}$	$\bar{\rho}$	RA	lag in m^*	lag in σ^*
100	0.5	$\frac{1}{3}$	0.15	0.22
		$\frac{2}{3}$	0.49	0.72
		1	1.51	2.21
	0.75	$\frac{1}{3}$	2.01	2.94
		$\frac{2}{3}$	4.00	6.88
		1	4.70	9.10
	0.9	$\frac{1}{3}$	4.70	8.12
		$\frac{2}{3}$	5.36	10.89
		1	5.57	12.12
10	0.5	$\frac{1}{3}$	1.10	1.50
		$\frac{2}{3}$	1.70	2.50
		1	2.40	3.40
	0.7	1	4.33	6.80
	0.75	$\frac{1}{3}$	3.20	4.70
		$\frac{2}{3}$	4.13	6.00
		1	4.71	7.70
	0.9	$\frac{1}{3}$	4.91	7.69
		$\frac{2}{3}$	5.37	9.10
		1	5.60	10.20

Table 5.2: Time lags in hours between the peak in arrival rate and peaks in the mean and standard deviation of the number of customers in the system.

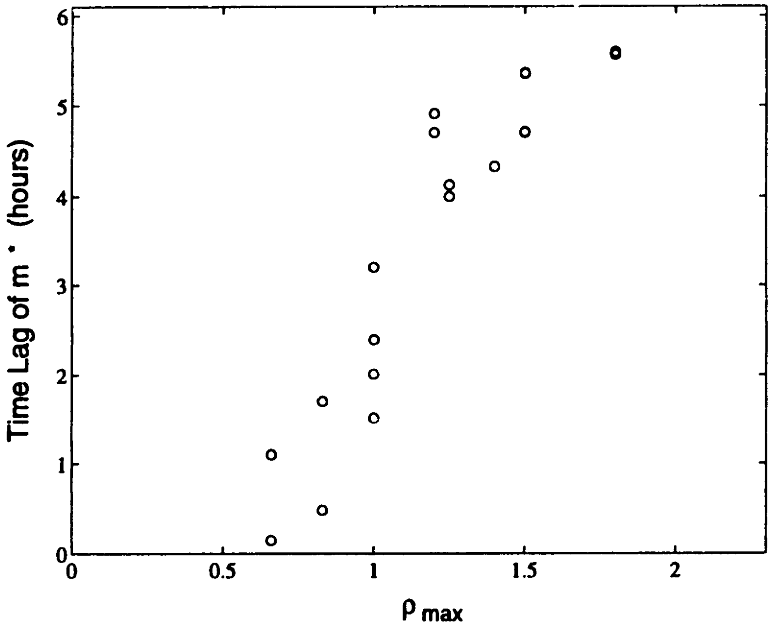


Figure 5-3: Time Lag for Peak in the Expected Number in the System

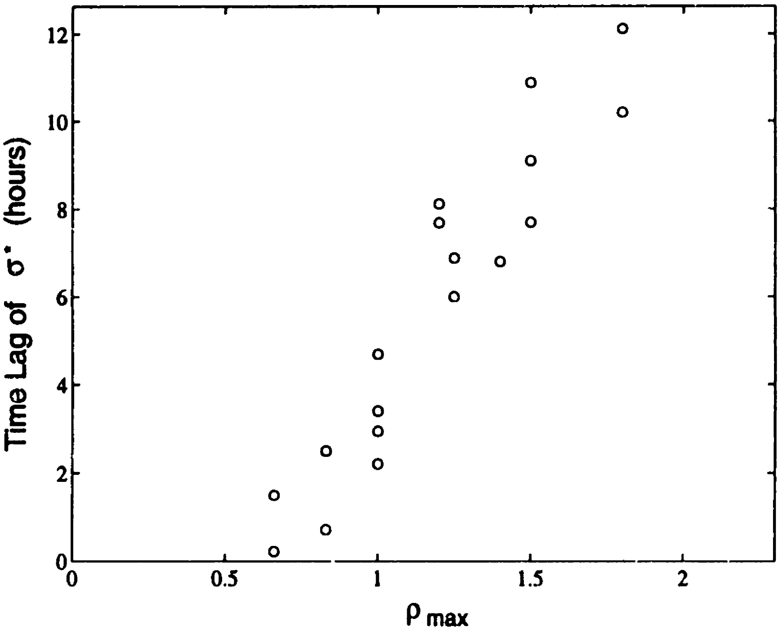


Figure 5-4: Time Lag for Peak in the Standard Deviation of the Number in the System

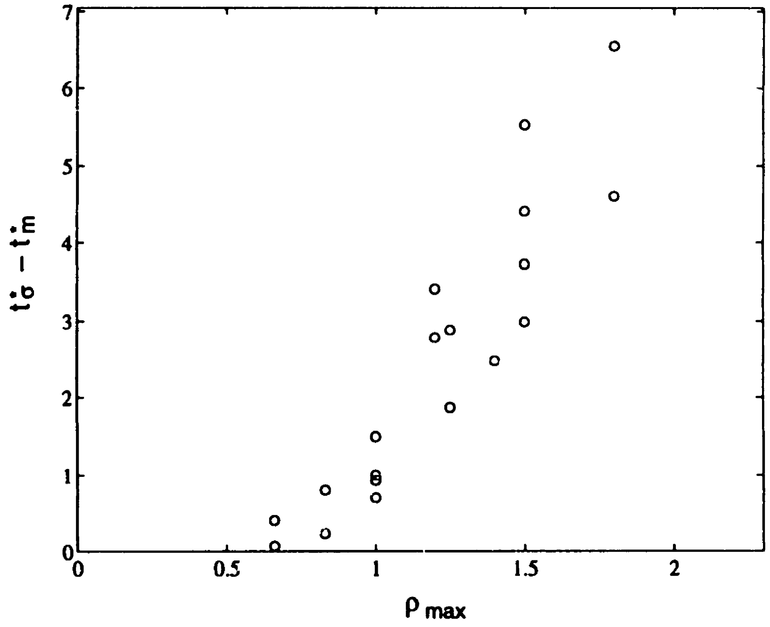


Figure 5-5: Difference Between the Time Lags for the Peak in the Standard Deviation and the Expected Number in the System

5.3 Summary

This chapter began to explore the time lag between the peak in the utilization function and the times at which local peaks m^*, v^*, and σ^* for the mean, variance, and standard deviation of the number of customers in the system occur. We conjecture that for stable, periodic $M(t)/M(t)/1$ queueing systems with smooth, continuous, utilization functions that peak once per period, the mean peak m^* will occur later than the system utilization peak and the variance peak v^* will occur later than its corresponding mean peak (Conjecture 1). We suspect that the same is true of $M(t)/G/1$ systems.

We demonstrated analytically, for a periodic $M(t)/M(t)/1$ queue, that if $\rho_{\max} > 1$, then the mean number in the system is increasing when the utilization peaks. Thus, any local peak in the mean number in the system induced by a utilization peak will occur strictly later than the utilization peak. Computational results for $M(t)/M(t)/1$ and other periodic single-server queueing systems confirmed our analytical result and support Conjecture 1.

Our results in Theorems 1 and 2, and Corollary, do not depend on specific arrival, service-rate, or utilization functions; however they are most easily interpreted for the case when the utilization function is periodic and peaks once per period. In the case of more general utilization functions with multiple maxima over a single day, there may not be a one-to-one correspondence between utilization peaks and mean peaks. Consider, for example, a utilization function which exceeds one throughout somı interval of the day, during which there are two local utilization peaks. After the end of this interval, the utilization will drop below one, and the mean will then reach a local maximum if the utilization stays below one long enough. In this case, two utilization peaks correspond to one mean peak. Thus, the practical implication of our findings is that if there are multiple utilization peaks over the day, and these peaks are "well-spaced," meaning that the periods of system utilization exceeding one are separated by sufficiently long periods of time during which system utilization is strictly less than one, then there will be a one-to-one correspondence between local utilization maxima and local mean maxima.

Chapter 6

Conclusions and Future Research

In this thesis, we developed approximate, computational methods for the analysis of dynamic queues and for networks of dynamic queues. Specifically, we investigated three facets of these problems. First, we developed and tested new approximate methods for analyzing individual queues with time-varying arrival and/or service rates. These methods are efficient and have limited computer memory requirements, and model a variety of queues with different service-time distributions. Second, we proposed and tested approximate decomposition methods for analyzing open, weakly-connected networks of dynamic queues. These methods allow analysis of systems for which exact methods of analysis do not exist, or are infeasible to implement for practical reasons. The decomposition approach is computationally and computer-memory efficient. Third, we investigated aspects of the behavior of queueing systems with time-varying arrival and service rates. The observations we make for these systems provide some rules-of-thumb that should help planners and operators of facilities with strongly time-dependent demand and capacity to make better facility management decisions.

Our focus on networks of dynamic single-server queues was motivated by a particular system of great practical importance: the national network of airports. We used the paradigm of the national network of airports to motivate the problem, and to choose scenarios for numerical tests.

We discuss the main results of the thesis and their implications in Section 6.1. We discuss future research directions in Section 6.2.

6.1 Results of the Thesis

The main contribution of this thesis is the development of fast, flexible and easy-to-implement computational methods for approximate analysis of dynamic single-server queueing systems and networks of dynamic queues. In Chapter 2, we introduced new computational methods for approximate analysis of nonstationary single-server queues. These methods include the State Probability Vector Approximation (SPVA), developed by this author, and DELAYS, developed by Kivestu [23]. We also investigated an interpolation method, INTERP [26, 19]. SPVA is the most general of the three methods tested; it approximates $M(t)/G(t)/1$ systems. DELAYS and INTERP approximate $M(t)/E_k(t)/1$ systems. All methods can model the transient and equilibrium behavior of these systems and calculate the time-dependent probability distribution for the number of customers in the system. The computational results of Chapter 3 indicated that SPVA, DELAYS and INTERP approximate $M(t)/E_k/1$ systems well under conditions encountered in real-world systems. Specifically, they gave estimates of the quantities of interest, i.e., the time-dependent mean and standard deviation, and their peak values, within 5% of exact values in many cases. These approximation methods required significantly less CPU time and computer memory than solving the exact $M(t)/E_k/1$ system.

Further computational testing of SPVA indicated that it is potentially a good approximation for single-server queueing systems with time-dependent general service-time distributions and Poisson arrival processes. In all cases tested, SPVA solved for the quantities of interest in a matter of seconds. Our observations of SPVA are the following. First, the quality of SPVA estimates of the quantities of interest were within 5% of exact values when both the arrival and service rates varied with time in the two cases we tested. Second, we showed that SPVA is fast, flexible, and easy-to-implement in approximating two particular $M(t)/G(t)/1$ systems. SPVA gave results for these systems which were consistent with the behavior of nonstationary queueing systems with similar parameters. Third, we investigated SPVA accuracy for approximating $M(t)/H_2/1$ systems. The H_2 distribution, in contrast to the k^{th}-order Erlang, has a coefficient of variation which is greater than or equal to one. The SPVA approximation to the $M(t)/H_2/1$ system revealed that it gave good approximations of the quantities of interest under certain combinations of parameters, even when the coefficient of variation was extremely high. These tests also indicated that as the coefficient of variation of the service-time distribution grew, the quality of the SPVA estimates decreased, when all other parameters were held constant. The decrease in quality was most noticeable for cases of moderate event frequency. Finally, we compared SPVA to the Surrogate Distribution Approximation (SDA) methods of Rider, Clark, and Rothkopf and Oren. SPVA gave estimates of the time-dependent mean which were comparable to those of Rider and Rothkopf and Oren. Clark's SDA method was superior to SPVA in the case we examined. We note, however, that the case was for low event frequency.

In Chapter 4 we began to investigate a decomposition approach for approximate analysis of a weakly-connected network of nonstationary single-server queues, or stations. Few results exist for these complicated systems. The decomposition method we proposed provides an alternative to simulating networks of nonstationary queues and to the exact modeling of a system as a Markovian network. Both simulation and the Markovian representation of the network are often infeasible, the exception being cases of very small and simple networks. In contrast, the decomposition method we investigated has memory requirements which are linear in the number of stations, requires little CPU time, and easily captures the relationships among stations in the network. We performed computational tests on a two-queue tandem network for which we could compare the decomposition results to the exact values. Memory restrictions limited the choice of the service-time distributions and queueing capacities we could model, even for this simple network. The solution times for the exact system were on the order of 6 and 12 hours for the two sets of cases we examined using this simple network. In contrast, the four versions of the decomposition method we examined required between 10 and 200 seconds to solve the same sets of systems. Our results indicate that there are two primary factors which affect the accuracy of the decomposition method: the fraction of arrivals to a station which are departures from any other single station, and the service-time distributions at the stations. Larger fractions lead to decreased accuracy. As the coefficient of variation of the service-time distribution becomes smaller, the accuracy also decreases. Specifically, for fractions of less than 0.5, the decomposition error is less than 5% of the exact value when the coefficient of variation of the service-time distribution is 1.0. When the coefficient of variation of the service-time distribution is 0.58, the decomposition method errors are less than 10%. For smaller fractions, the errors were smaller for both service-time coefficients of variation. When the fraction is 0.1, for example, the errors were negligible for service-time coefficients of variation of 1.0 and 0.58. It is difficult to generalize this conclusion for more networks with more stations or different service-time distributions, though, based on the results for our particular network configuration.

The combination of fast, accurate approximation methods for dynamic queues and of a decomposition method which uses them provides a potentially useful tool for approximate, macroscopic analysis of a network of airports. The network of airports is weakly connected. The fraction of arrivals to any airport which are departures from any other single airport is less than 0.1. A stronger link exists between landings and takeoffs at a single airport. In this case, the fraction is about 0.3–0.4. Our tandem-queue results indicate that these attributes of the network of airports satisfy the "weakly-connected" assumption of a decomposition method. We conclude a decomposition approach is appropriate for a network of airports.

Finally, in Chapter 5 we investigated some aspects of the behavior of $M(t)/M(t)/1$ systems. We found a necessary condition for the time when the extremes in the time-dependent mean of $M(t)/M(t)/1$ systems occur. The necessary condition depends only on the time-dependent arrival and service rates and the probability of an empty system. More importantly, the corollary of this condition states that if the system utilization exceeds 1.0 for some interval in an $M(t)/M(t)/1$ system, the peak in the mean will occur strictly after the end of this interval of system oversaturation. These results do not depend on the exact form of the arrival and service rate functions. Our computational results indicated that this principle may hold for other nonstationary queues. The corollary also provides a rule-of-thumb for determining when the peak in the expected number of customers occurs in real-world systems, assuming the arrival and service rates are known. Several other conjectures regarding the behavior of $M(t)/M(t)/1$ systems which appear important from the practical point of view were also presented.

In carrying out this research, we identified new research opportunities. We now briefly discuss them.

6.2 Future Research

Based on the research presented in this thesis, we have identified many potentially fruitful areas of research. These fall into two categories: extensions of the SPVA approximation method, and further exploration of the decomposition method to networks of dynamic queues.

We have identified four areas of further research involving SPVA. They are:

1. Assess the accuracy of the SPVA method for $M(t)/G(t)/1$ systems in a more precise manner.

2. Compare the SPVA method to more general SDA methods which approximate $Fh(t)/Ph(t)/1/c$ systems. These methods were developed by Ong and Taaffe [36] and are currently the benchmark against which new approximation methods must be compared.

3. Explore the possibility of extending SPVA to systems with Phase-type arrival distributions.

4. Determine if SPVA can accurately model multi-server systems.

The area of networks of dynamic single-server queues offers many possibilities for research. In this thesis, we solved exactly only a two-station tandem network. Research on larger and more complicated networks is necessary in order to refine our understanding of the factors affecting decomposition accuracy. For example, our network examined the effect of a single station feeding a second station. When we assessed the accuracy of the decomposition method, one of the two primary factors affecting the accuracy was the fraction of total arrivals to the second station which were departures from the first station. A natural question to ask next is, "What is the effect of having two stations feeding a single station while holding the fraction of arrivals to the second station which are departures from the first two stations, fixed? What effect does the superposition of several departure streams have on the accuracy of the decomposition method?" We could not address this important question in our two-station tandem network.

Research on networks of dynamic queues with feedback and different service-time distributions is also necessary. However, CPU time and computer memory limit the size and complexity of networks which we can solve exactly. Simulation of such systems is also impractical as far as obtaining levels of accuracy adequate for analysis. Therefore, we suggest exploring the effect of these attributes by extending our two-queue network in such a way that exact results can still be found. That is, keep the network small, and add attributes one-at-a-time. In this way, the impact of these attributes on decomposition accuracy can be assessed in isolation and in combination with others.

Appendix A

Examples of α_{n+1}(j) for Specific Service-Time Distributions

A.1 Exponential Service Times

$$\begin{aligned}
\text{If } dB(x) &= \mu e^{-\mu x} dx, x \geq 0, \\
\text{then } \alpha_{n+1}(j) &= \int_{x=0}^{\infty} \frac{(\lambda(t_n)x)^j}{j!} e^{-\lambda(t_n)x} \mu e^{-\mu x} dx \\
&= \frac{\mu[\lambda(t_n)]^j}{(\mu + \lambda(t_n))^{j+1}}
\end{aligned}$$

A.2 k^{th}-order Erlang Service Times

$$\begin{aligned}
\text{If } dB(x) &= \frac{(k\mu)^k x^{k-1} e^{-k\mu x}}{(k-1)!} dx, x \geq 0, \\
\text{then } \alpha_{n+1}(j) &= \int_{x=0}^{\infty} \frac{(\lambda(t_n)x)^j}{j!} e^{-\lambda(t_n)x} \frac{(k\mu)^k x^{k-1} e^{-k\mu x}}{(k-1)!} dx \\
&= \begin{pmatrix} k-1+j \\ j \end{pmatrix} \frac{(k\mu)^k [\lambda(t_n)]^j}{(k\mu + \lambda(t_n))^{k+j}}
\end{aligned}$$

A.3 Hyperexponential Service Times

$$\begin{aligned}
\text{If } dB(x) &= \sum_{i=1}^{m} p_i\mu_i e^{-\mu_i x}dx, x \geq 0, \sum_{i=1}^{m} p_i = 1, p_i \geq 0 \text{ for } i = 1, 2, \ldots, m, \\
\text{then } \alpha_{n+1}(j) &= \int_{x=0}^{\infty} \frac{(\lambda(t_n)x)^j}{j!} e^{-\lambda(t_n)x} \sum_{i=1}^{m} p_i\mu_i e^{-\mu_i x}dx \\
&= \lambda(t_n)^j \left[\sum_{i=1}^{m} \frac{p_i\mu_i}{(\mu_i + \lambda(t_n))^{j+1}}\right]
\end{aligned}$$

A.4 Uniform Service Times

$$\begin{aligned}
\text{If } dB(x) &= \frac{1}{b-a}dx, 0 \leq a \leq x \leq b, \\
\alpha_{n+1}(j) &= \frac{1}{(\lambda(t_n))(b-a)} \left\{ e^{-a\lambda(t_n)} \left[\sum_{i=0}^{j} \frac{(\lambda(t_n)a)^i}{i!}\right] - e^{-b\lambda(t_n)} \left[\sum_{i=0}^{j} \frac{(\lambda(t_n)b)^i}{i!}\right] \right\}
\end{aligned}$$

A.5 Triangular Service Times

$$\begin{aligned}
\text{If } dB(x) &= \left\{ \begin{array}{ll} \frac{4}{(d-c)^2}(x-c), & 0 \leq c \leq x \leq \frac{d+c}{2} \\ \frac{4}{(d-c)^2}(d-x), & \frac{d+c}{2} \leq x \leq d \end{array} \right. , \\
\alpha_{n+1}(j) &= \frac{4}{((d-c)^2)(\lambda(t_n))} \cdot \\
& \left\{ e^{-(\frac{c+d}{2})\lambda(t_n)} \left\{ \left(c + d - 2\left(\frac{j+1}{\lambda(t_n)}\right)\right) \left[\sum_{i=0}^{j} \frac{(\lambda(t_n)\left(\frac{c+d}{2}\right))^i}{i!}\right] \right.\right. \\
& \qquad \left. -2\left[\frac{(\lambda(t_n))^j \left(\frac{c+d}{2}\right)^{j+1}}{j!}\right] \right\} \\
& \quad +e^{-d\lambda(t_n)} \left\{ \frac{(\lambda(t_n)^j c^{j+1}}{j!} + \left(\frac{j+1}{\lambda(t_n)} - d\right) \left[\sum_{i=0}^{j} \frac{(\lambda(t_n)d)^i}{i!}\right] \right\} \\
& \quad \left. -e^{-c\lambda(t_n)} \left\{ -\left(\frac{(\lambda(t_n)^j c^{j+1}}{j!}\right) + \left(c - \frac{j+1}{\lambda(t_n)}\right) \left[\sum_{i=0}^{j} \frac{(\lambda(t_n)c)^i}{i!}\right] \right\} \right\}
\end{aligned}$$

Appendix B

The Arrival Process

In all of the queueing systems, we assume a nonstationary Poisson arrival process with instantaneous rate parameter $\lambda(t)$ at time t. Assuming $\lambda(s)$ is an integrable function, let

$$\Lambda(t) \;=\; \int_{s=0}^{t} \lambda(s)ds.$$

Then (see Ross [44]),

$$\text{Probability of } n \text{ arrivals in } (t_1, t_2) = \begin{cases} \frac{[\Lambda(t_2)-\Lambda(t_1)]^n e^{-(\Lambda(t_2)-\Lambda(t_1))}}{n!}, & n = 0, 1, 2, \ldots \\ 0, & \text{ctherwise} \end{cases}$$

If $\lambda(s)$ is an integrable function over the interval of interest, then $\Lambda(t_2) - \Lambda(t_1) = \int_{s=t_1}^{t_2} \lambda(s)ds$ exists. In our research with sinusoidal arrival functions, one can find $\Lambda(t+\bar{b}_1) - \Lambda(t)$ exactly. However, in the case of general arrival functions, it may be significantly easier to use $\lambda(t)\bar{b}_1$ instead of $\int_{s=t}^{t+\bar{b}_1} \lambda(s)ds$. For example, if we attempt to evaluate expression (2.4) for $\lambda(t) = \bar{\lambda} + \bar{\lambda}RA\sin\left(\frac{2\pi}{24}\right)$, we get the following difficult-to-evaluate expression:

$$\alpha_{n+1}(j) \;=\; \int_{x=0}^{\infty} \frac{\left[\bar{\lambda}x^2 + \bar{\lambda}RA\left(\frac{24}{2\pi}\right)x\left[\sin\left(\frac{\pi}{24}(2t_n+x)\right)\sin\left(\frac{\pi x}{24}\right)\right]\right]^j}{j!} \cdot \exp\left\{-\left[\bar{\lambda}x^2 + \bar{\lambda}RA\left(\frac{24}{2\pi}\right)x\left[\sin\left(\frac{\pi}{24}(2t_n+x)\right)\sin\left(\frac{\pi x}{24}\right)\right]\right]\right\} dB_{t_n}(x)$$

However, we know that the time clock advances $\bar{b}_1$ time units in the SPVA, DELAYS, and $M(t)/D(t)/1$ methods. If $\int_{s=t_n}^{t_n+\bar{b}_1} \lambda(s)ds \approx \lambda(t_n)\bar{b}_1$, we may use $\lambda(t_n)$ in the complicated expression above, and thus obtain simpler expressions, such as (2.5). This approximation is

	Linear Case		$\bar{\lambda} = 50$		$\bar{\lambda} = 5$	
	Exact	Approx	Exact	Approx	Exact	Approx
parameter value	0.30000275	0.3	0.5007	0.5	0.5065	0.5
$a_0(t, t + \bar{b}_1)$	0.7408	0.7408	0.6061	0.6065	0.6026	0.6065
$a_1(t, t + \bar{b}_1)$	0.2223	0.2222	0.3035	0.3033	0.3052	0.3033
$a_2(t, t + \bar{b}_1)$	0.0333	0.0333	0.0760	0.0759	0.0773	0.0758
$a_3(t, t + \bar{b}_1)$	0.0034	0.0033	0.0127	0.0126	0.0131	0.0126

Table B.1: Exact and Approximate Values for Nonstationary Poisson Arrival Process Parameter Cases

good if $\lambda(t)$ changes slowly relative to the length of the average service time. To illustrate that this is a reasonable assumption, we give an example from our airport application where $\bar{b}_1 \approx 0.01$ hour (assuming a service rate of 100 per hour). In Table B.1, we show how small the difference is between the exact parameter $(\Lambda(t + \bar{b}_1) - \Lambda(t))$, labelled "Exact", and the approximate parameter $(\lambda(t)\bar{b}_1)$, labelled "Approx," for several cases. We examine the following functions of the nonstationary Poisson process parameter:

1. A linear function. This function describes the greatest change in airport arrival rate during the course of a day at Logan International Airport. The slope of this function is 55. $\bar{b}_1 = 0.01$.

2. A high-frequency sinusoidal function with parameter $\lambda(t) = 50 + 50 \sin\left(\frac{2\pi t}{24}\right)$. This function produces values of $\lambda(t)$ comparable to the range of arrival rates at Logan over a 24-hour period. $\bar{b}_1 = 0.01$.

3. A low-frequency sinusoidal function with parameter $\lambda(t) = 5 + 5 \sin\left(\frac{2\pi t}{24}\right)$. This case was chosen to determine if approximation accuracy is sensitivity to the frequency of events. $\bar{b}_1 = 0.1$.

In Table B.1, we also show the probability of $0, 1, 2,$ and 3 arrivals in $\bar{b}_1$ time units using the exact and approximate parameters. The notation used is $a_i(t, t+\bar{b}_1) \equiv P(i \text{ arrivals in } (t, t+\bar{b}_1))$.

In the three examples shown in Table B.1, the difference between the probabilities for the number of arrivals in the interval calculated using the exact and approximate parameter is, at worst, visible only in the third decimal place. Therefore, it is reasonable to approximate $\Lambda(t+\bar{b}_1) - \Lambda(t)$ by $\lambda(t)\bar{b}_1$. We define the probability of k arrivals in a time interval of length

τ to be:

$$a_n(t, t+\tau) = \begin{cases} \frac{[\lambda(t)\tau]^n e^{-\lambda(t)\tau}}{n!}, & n = 0, 1, 2, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{B.1}$$

Appendix C

Supplementary Proof

Consider an $M(t)/M(t)/1$ queue and its evolution over some finite time interval $[a, b]$, as governed by the forward differential equations:

$$\begin{aligned} P_i'(t) &= \lambda(t)P_{i-1}(t) - (\lambda(t) + \mu(t))P_i(t) + \mu(t)P_{i+1}(t) \text{ for } i = 1, 2, \ldots \\ P_0'(t) &= -\lambda(t)P_0(t) + \mu(t)P_1(t) \end{aligned} \tag{C.1}$$

Given initial conditions $\{P_i(a) \geq 0\}$ where $\sum_{i=0}^{\infty} P_i(a) = 1$, we are assured of the existence of a unique, nonnegative continuously differentiable solution to (C.1).

Assume that the arrival and service rate functions λ and μ are nonnegative and that λ is bounded on $[a, b]$. For $t \in [a, b]$, define $m(t) \equiv \sum_{i=0}^{\infty} iP_i(t)$ and assume that $m(t)$ is finite and continuous[1].

Proposition: $m(t)$ is differentiable on $[a, b]$, with derivative

$$m'(t) \equiv \frac{d}{dt}\left\{\sum_{i=0}^{\infty} iP_i(t)\right\} = \sum_{i=0}^{\infty} iP_i'(t)$$

i.e., $m'(t)$ exists on $[a, b]$ and can be computed by interchanging the infinite summation and the differentiation.

Proof: We define the following functions:

[1] This is really a condition on the arrival and service rate functions: we require them to b: sufficiently well behaved so that $m(t)$ is continuous.

$$f(t) \equiv \sum_{i=0}^{\infty} iP_i'(t)$$
$$f_n(t) \equiv \sum_{i=0}^{n} iP_i'(t)$$
$$g_n(t) \equiv \sum_{i=0}^{n} iP_i(t)$$

We need to show that $f_n(t)$ converges uniformly on $[a,b]$ to $f(t)$ (see theorem 7.17 in [47]). Due to the form of $|f_n(t) - f(t)|$, it turns out that this amounts to showing that $g_n(t)$ converges uniformly on $[a,b]$ to $m(t)$ and we therefore show this first.

For any n, the function $g_n(t)$ is continuous, since it is the sum of continuous functions. By definition, $g_n(t)$ converges pointwise to $m(t)$ and it does so in a monotonic manner, since all the terms in the summation are nonnegative. Since $m(t)$ is assumed continuous, the assumptions of theorem 7.13 in [47] are satisfied (the theorem is stated for the case of a monotonically nondecreasing sequence of functions, but is easily seen to hold also for monotonically non-increasing sequences) and we are assured that $g_n(t)$ converges uniformly to $m(t)$ on $[a,b]$. A similar application of the same theorem shows that $\{\sum_n^\infty P_i(t)\}_{n=0}^\infty$ converges uniformly to 0 on $[a,b]$. A bounding argument, where $|g_n(t) - m(t)|$ is used as an upper bound on $|(n+1)P_{n+1}(t)|$, shows that $\{nP_n(t)\}_{n=0}^\infty$ also converges uniformly to 0 on $[a,b]$.

Next, we will compute $|f_n(t) - f(t)|$, using the forward equations (C.1):

$$\begin{aligned}
|f_n(t) - f(t)| &= \left| \sum_{i=n+1}^{\infty} iP_i'(t) \right| \\
&= \left| \sum_{i=n+1}^{\infty} i\left\{\lambda(t)P_{i-1}(t) - (\lambda(t)+\mu(t))P_i(t) + \mu(t)P_{i+1}(t)\right\} \right| \\
&= \left| \lambda(t) \sum_{i=n+1}^{\infty} i\left\{P_{i-1}(t) - P_i(t)\right\} + \mu(t) \sum_{i=n+1}^{\infty} i\left\{-P_i(t) + P_{i+1}(t)\right\} \right| \\
&= \left| \lambda(t)\left\{ \sum_{i=n}^{\infty} (i+1)P_i(t) - \sum_{i=n+1}^{\infty} iP_i(t) \right\} + \mu(t)\left\{ -\sum_{i=n+1}^{\infty} iP_i(t) + \sum_{i=n+2}^{\infty} (i-1)P_i(t) \right\} \right| \\
&= \left| \lambda(t)\left\{ nP_n(t) + \sum_{i=n}^{\infty} P_i(t) \right\} - \mu(t)\left\{ (n+1)P_{n+1}(t) + \sum_{i=n+2}^{\infty} P_i(t) \right\} \right| \\
&\leq \lambda(t)\left\{ nP_n(t) + \sum_{i=n}^{\infty} P_i(t) \right\} \leq \lambda^*\left\{ nP_n(t) + \sum_{i=n}^{\infty} P_i(t) \right\}
\end{aligned}$$

where $\lambda^* = \max_{a \leq t \leq b} \lambda(t)$. Next,

$$\lim_{n\to\infty}\sup_{a\leq t\leq b}\lambda^*\left\{nP_n(t)+\sum_{i=n}^{\infty}P_i(t)\right\}\leq\lambda^*\lim_{n\to\infty}\sup_{a\leq t\leq b}\{nP_n(t)\}+\lambda^*\lim_{n\to\infty}\sup_{a\leq t\leq b}\left\{\sum_{i=n}^{\infty}P_i(t)\right\}=0$$

since both $\{\sum_n^\infty P_i(t)\}_{n=0}^\infty$ and $\{nP_n(t)\}_{n=0}^\infty$ converge uniformly to zero on $[a,b]$. This completes the proof. ■

Bibliography

[1] S.L. Albin. On Poisson Approximations for Superposition Arrival Processes in Queues. *Management Science*, 28(2):126–137, 1982.

[2] S.L. Albin and S.R. Kai. Approximation for the Departure Process of a Queue in a Network. *Naval Research Logistics Quarterly*, 33:129–143, 1986.

[3] N. Bambos and J. Walrand. On Queues With Periodic Inputs. *Journal of Applied Probability*, 26:381–389, 1989.

[4] E.L. Blair, A.L. Haines, J.H. Hoffman, D.C. Millner, M.J. White, and A.E. Zukas. Comparison of Analytic System-Level Models. Technical Report MP 95W0000081, The Center for Advanced Aviation System Development, The MITRE Corporation, Mclean, VA 20591, 1995.

[5] Volpe National Transportation Systems Center, MITRE Corporation, and JIL Systems. 1994 Aviation Capacity Enhancement Plan. Technical Report DOT/FAA/ASC-94-1, U.S. Department of Transportation, Federal Aviation Administration, Office of System Capacity and Requirements, Washington, D.C. 21591, 1993.

[6] G.L. Choudhury, D.M. Lucantoni, and W. Whitt. Numerical Solution of $M(t)/G(t)/1$ Queues. AT&T Bell Laboratories. Submitted to *Operations Research*, 1993.

[7] G.M. Clark. Use of Polya Distributions in Approximate Solutions to Nonstationary $M/M/s$ Queues. *Communications of the Association of Computing Machinery*, 24(4), 1981.

[8] A.B. Clarke. A Waiting Line Process of the Markov Type. *Annals of Mathematical Statistics*, 27:452–459, 1956.

[9] D.J. Daley. Queueing Output Processes. *Advances in Applied Probability*, 8:395–415, 1976.

[10] A. Drake. *Fundamentals of Applied Probability.* McGraw-Hill, New York, 1967.

[11] S.G. Eick, W.A. Massey, and W. Whitt. $M_t/G/\infty$ Queues with Sinusoidal Arrival Rates. *Management Science*, 39(2):241–252, 1993.

[12] S.G. Eick, W.A. Massey, and W. Whitt. Physics of the $M_t/G/\infty$ Queue. *Operations Research*, 41(4):731–742, 1993.

[13] L.V. Green and P.J. Kolesar. The Pointwise Stationary Approximation for Queues with Nonstationary Arrivals. *Management Science*, 37(1):84–97, 1991.

[14] L.V. Green and P.J. Kolesar. On the Accuracy of the Simple Peak Hour Approximation for Markovian Queues. Graduate School of Business, Columbia University, New York, NY. Submitted to *Management Science*, 1993.

[15] L.V. Green and P.J. Kolesar. Simple Peak Congestion in $M(t)/G\infty$ Queues with Sinusoidal Arrivals. Graduate School of Business, Columbia University, New York, NY., 1994.

[16] L.V. Green, P.J. Kolesar, and A. Svoronos. Some Effects of Nonstationarity on Multiserver Markovian Queueing Systems. *Operations Research*, 39(3):502–511, 1991.

[17] J.M. Harrison and A.J. Lemoine. Limit Theorems for Periodic Queues. *Journal of Applied Probability*, 14:566–576, 1977.

[18] D.P. Heyman and W. Whitt. The Asymptotic Behavior of Queues with Time-Varying Arrival Rates. *Journal of Applied Probability*, 21:143–156, 1984.

[19] B. Horangic. Some Queueing Models of Airport Delays. Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 1990.

[20] O.B. Jennings, A. Mandelbaum, W.A. Massey, and W. Whitt. Server Staffing to Meet Nonstationary Demand. AT&T Bell Labroratories. Submitted to *Management Science*, 1994.

[21] N.L. Johnson and S. Kotz. *Urn Models and Their Applications.* John Wiley & Sons, 1977.

[22] A.Y. Khintchine. *Mathematical Models in the Theory of Queueing.* Charles Griffin and Co., London, 1960. (Translation of 1955 Russian book).

[23] P. Kivestu. Alternative Methods of Investigating the Time-Dependent $M/G/k$ Queue. Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 1976.

[24] L. Kleinrock. *Queueing Systems: Theory*, volume 1. John Wiley & Sons, New York, 1975.

[25] L. Kleinrock. *Queueing Systems: Computer Applications*, volume 2. John Wiley & Sons, New York, 1976.

[26] B. Koopman. Air Terminal Queues Under Time-Dependent Conditions. *Operations Research*, 20:1089–1114, 1972.

[27] A. Lemoine. Waiting Time and Workload in Queues with Periodic Poisson Input. *Journal of Applied Probability*, 26:390–397, 1989.

[28] D.G. Luenberger. *Introduction to Dynamic Systems. Theory, Models, and Applications.* John Wiley & Sons, 1979.

[29] K. Malone. Modeling a Network of Queues Under Nonstationary and Stochastic Conditions. Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 1993.

[30] W.A. Massey and W. Whitt. Networks of Infinite-Server Queues with Nonstationary Poisson Input. *Queueing Systems*, 13:183–250, 1993.

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

[32] G. Mourtzinou. *An Axiomatic Approach to Queueing Systems.* PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 1995.

[33] G.F. Newell. Queues With Time-Dependent Arrival Rates I – III. *Journal of Applied Probability*, 5:436–606, 1968.

[34] G.F. Newell. *Applications of Queueing Theory.* Chapman and Hall, London, 1971.

[35] A.R. Odoni and E. Roth. An Empirical Investigation of the Transient Behavior of Stationary Queueing Systems. *Operations Research*, 31(3):432–455, 1983.

[36] K.L. Ong and M.R. Taaffe. Approximating Nonstationary $Ph(t)/Ph(t)/1/c$ Queueing Systems. *Mathematics and Computers in Simulation*, 30:441–452, 1988.

[37] C. Palm. Intensity Variations in Telephone Traffic. *North-Holland*, 1988. (Translation of 1943 article in *Ericsson Technics*, 44, 1–189).

[38] M.D. Peterson, D.J. Bertsimas, and A.R. Odoni. Decomposition Algorithms for Analyzing Transient Phenomena in Multi-Class Queuing Networks in Air Transportation. Technical Report OR 278-93, Operations Research Center, Massachusetts Institute of Technology, 77 Massachusetts Avenue, E40-149, Cambridge, Massachusetts 02139, 1993.

[39] G.J.K. Regterschot and J.H.A. De Smit. The Queue $M/G/1$ with Markov Modulated Arrivals and Services. *Mathematics of Operations Research*, 11(3):465–483, 1986.

[40] K.L. Rider. A Simple Approximation to the Average Queue Size in the Time-Dependent $M/M/1$ Queue. *Journal of the Association for Computing Machinery*, 23(2):361-367, 1976.

[41] T. Rolski. Approximation of Periodic Queues. *Advances in Applied Probability*, 19:691–707, 1987.

[42] T. Rolski. Queues With Nonstationary Inputs. *Queueing Systems*, 5:113–130, 1989.

[43] T. Rolski. Relationships Between Characteristics in Periodic Poisson Queues. *Queueing Systems*, 4:17–26, 1989.

[44] S.M. Ross. *Stochastic Processes.* John Wiley & Sons, Inc., 1983.

[45] E. Roth. *An Investigation of the Transient Behavior of Stationary Queuing Systems.* PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 1981.

[46] M.H. Rothkopf and S.S. Oren. A Closure Approximation for the Nonstationary $M/M/s$ Queue. *Management Science*, 25(6):522–534, 1979.

[47] W. Rudin. *Principles of Mathematical Analysis.* McGraw-Hill, New York, third edition, 1976.

[48] B.W. Schmeiser and M.R. Taaffe. Time-dependent queueing network approximations as simulation external control variates. *Operations Research Letters*, 16:1–9, 1994.

[49] M.R. Taaffe. *Approximating Nonstationary Queueing Models.* PhD thesis, The Ohio State University, Columbus, Ohio, USA, 1982.

[50] M.R. Taaffe and K.L. Ong. Approximating Nonstationary $Ph(t)/M(t)/s/c$ Queueing Systems. *Annals of Operations Research*, 8:103–116, 1987.

[51] J.L. van den Berg and W.P. Groenendijk. Transient Analysis of an $M/M/1$ Queue with Regularly Changing Arrival and Service Intensities. *Teletraffic and Datatraffic in a Period of Change, Proceedings of ITC 13*, pages 677–681, 1991. A. Jensen and V.B. Iversen, editors.

[52] W. Whitt. Approximating a Point Process by a Renewal Process, I: Two Basic Methods. *Operations Research*, 30(1):125–147, 1982.

[53] W. Whitt. The Queueing Network Analyzer. *The Bell System Technical Journal*, 62(9):2779–2815, 1983.

[54] W. Whitt. A Light Traffic Approximation for Single-Class Departure Processes from Mulit-Class Queues. *Management Science*, 34(11):1333–1346, 1988.

[55] W. Whitt. The Pointwise Stationary Approximation for $M_t/M_t/s$ Queues is Asymptotically Correct as the Rates Increase. *Management Science*, 37(3):307–315, 1991.

[56] R.W. Wolff. Poisson Arrivals See Time Averages. *Operations Research*, 30(2):223–231, 1982.