NEW DIRECTIONS IN QUEUE INFERENCE FOR MANAGEMENT IMPLEMENTATION

by

Susan Aileen Hall

A.B., Oberlin College, 1976

S.M., M.I.T., 1982

SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY
IN OPERATIONS RESEARCH

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1992

Signature of Author _____
Operations Research Center
May 15, 1992

Certified by _____
Richard C. Larson
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by _____
Richard C. Larson
Co-Director, Operations Research Center

New Directions in Queue Inference for Management Implementation

by

Susan Aileen Hall

Submitted to the Department of Electrical Engineering and Computer Science on May 15, 1992 in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Operations Research

Abstract

Queue inference is a method of utilizing service-time data, available from ATM's, point-of-sale terminals, and other sources of transactional data, in order to estimate queue parameters. Three basic limitations to the standard method of queue inference are: the statistics generated by the algorithm may not be extensive enough; the running time for the algorithm is too long for large congestion periods or real-time applications; and the algorithm fails to take advantage of partial queue-length information which may be available. This thesis addresses these three limitations, presenting several new results in the area, which expand the utility and applicability of the method.

First, two theorems are presented, which pertain to the density function for the arrival time of the k-th customer out of the N customers who must wait in queue during a given congestion period. These theorems are used to find an algorithm which generates all N of these density functions in $O(N^4)$ time. The density function for the arrival time of a random (unordered) customer is also found, as are the density functions for the queue waiting time of the k-th customer, or a random customer, assuming a first-come-first-served queueing discipline. An airline industry application for using these density functions is presented in some detail.

Second, several alternative algorithms are found which give bounds and approximations to the expected cumulative number of arrivals to the queue as a function of time, a function which, as generated by the original algorithm, is concave and piecewise-linear. Specifically, first a simple lower bound is found, which takes advantage of the concavity of the function. Then an upper bound is found, which approximates all of the ordered arrival-time densities as being truncated uniforms. Finally, an approximation is presented, which uses trapezoidal functions as approximations for these densities. Although none of these performs well by itself, computational results

show that combinations of some of these algorithms give queue-length estimates very close to that of the original algorithm. Next, modification of the set of conditioning inequalities which are used to calculate the arrival-time probabilities (the probability that the k-th customer arrives before the i-th service completion) in the original algorithm, is considered. A general theorem is proved which describes how the arrival-time probabilities are changed when the conditioning inequalities are modified. This theorem is used to find two more lower-bound algorithms: the first of these only includes a subset of the conditioning inequalities when calculating the arrival-time probabilities; the second considers adding the constraint that the maximum queue length over the duration of the congestion period be below some specified threshold. Sample runs of all of these algorithms are provided to demonstrate typical performance and reduced runtimes.

Third, the incorporation of partial queue-length information into the estimates of queue length is considered in two cases. First, in the case that the queue length actually did not exceed some specified value during a given congestion period, the lower bound algorithm described above may be used to give better queue-length estimates in less running time. Second, if the times of all $(M-1)$-to-M and M-to-$(M-1)$ queue-length transitions are known, then that exact information allows partitioning of the congestion period and separate analysis of the partitions, giving faster runtimes and better estimates. Sample runs of these algorithms are also provided.

Finally, practical applications for this work are summarized, and suggestions for future work, especially in the areas of heuristics and statistical analysis, are presented.

Thesis Supervisor: Richard C. Larson
Title: Professor of Electrical Engineering and Computer Science

Acknowledgements

This research was supported in part by the following grants and fellowships:

- NSF Grant Number 8714649, "Design and Control of Integrated Logistics Distribution Systems"
- NSF Grant Number SES 9119962, "Inferring Queue Performance from Transactional Data"
- Vinton Hayes Fellowship, through the Department of Electrical Engineering and Computer Science

There are many people whom I would like to thank for their help, direct and indirect, in the creation of this thesis. First and foremost, my thanks to my advisor, Dick Larson, a truly remarkable man, for whom I feel privileged to have worked. He has been an endless source of enthusiasm, drive, and ideas, all of which helped to spur me on and get me excited about doing my research. I also owe my deepest gratitude to Al Drake, who took me under his wing some 13 years ago and first got me interested in probability. He has shown me over the years what makes a truly great teacher, and it has been an honor to have spent seven semesters working with him, teaching his wonderful course, 6.041/6.431. Thanks also to the other members of my thesis committee, Arnie Barnett and John Little, who, on short notice, were able to accommodate my fairly hectic schedule, as I attempted to defend my thesis before delivering my second child. (I made it with six days to spare!) Finally, thanks to Amedeo Odoni, who, as my first advisor at the ORC, got me started on this long journey to my Ph.D.

Of course, the students and staff of the OR Center were one major reason I decided to do my doctoral work at MIT. I have thoroughly enjoyed all of the friends I have made there over the years. I thank Marcia Chapman and Paulette Mosley for their patience with my crazy life and growing family, and Laura Terrell and Michelle Brodeur for all their endless help with the myriad details of being a graduate student.

A special thank you to my three babysitters. Frauke came in from Germany and took my little Robert from being an infant of 7 months to a toddler of 19 months, and gave me my first taste of independence after childbirth. Becky had Robert for the next year and turned him into a sociable little boy, meanwhile allowing me to get a big chunk of my thesis work done. And then Robin came in with me eight months pregnant with two months to go on the thesis, and calmly took over our childcare, housekeeping, cooking, and landscaping, as I went through two of the hardest months of my life.

Of course, I never would have entered the field of operations research if I hadn't been interested in math and science in the first place. For sparking that interest, and for showing me that I could do with my life whatever I wanted, I thank my father. I thank my mother for her boundless patience and love over the years, with all the various paths I have chosen, from as early as I can remember until now. My two sisters, Mindy and Leslie, have always been close and supportive, and I thank them for that, as well as for showing me that a Ph.D. thesis can actually be completed!

Finally, a special thank you to my immediate family. To Robert, my dearest little boy, thank you for your beautiful smile and love, even when Mommy had to spend so much time upstairs working. To Lonnie, my sweet little girl, thank you for sharing the first month of your precious life with my thesis. And to David, my devoted and wonderful husband: for the backrubs, the healthy dinners, the adventures with Robert, your sense of humor (you made me laugh in spite of myself, even in the toughest of times), and your love, I thank you from the bottom of my heart.

Contents

List of Figures

List of Tables

Chapter 1

Background and Motivation

The quantity of data available to operations analysts has been burgeoning over the past decade, due in large part to the advent and growing omnipresence of computers. Finding ways to utilize these data to improve customer service and to optimize manufacturing and other operations is one of the current challenges for operations managers. One familiar example of such data streams is given by those generated by Automatic Teller Machines (ATM's), which currently record the times of all ATM card insertions (service commencement times) and card removals (service completion times). However, until recently, there was no way to estimate the queue lengths behind the ATM's from the service time data.

In 1990, Larson introduced the Queue Inference Engine (QIE) [Lars 90], an algorithm which uses these service-time data to generate estimates of queue parameters during a congestion period. (A congestion period occurs when all servers are busy, and service completions are followed almost immediately by service commencements.) After some pre-processing of the data to determine when the system is in a congestion period, the QIE operates on a single congestion period to calculate estimates of the queue length as a function of time, the time-averaged queue length, the expected wait in queue, and the probability distribution of queue length as experienced by a randomly-arriving customer. With these estimates, the bank may decide to add ATM's to or remove ATM's from an over-utilized or under-utilized site, respectively.

Potential applications of transactional data analysis algorithms, like the QIE algorithm, abound in industries other than the banking example cited. Airports, urban transportation, hotels, fast-food restaurants, supermarkets, and telecommunications are all areas in which huge amounts of data are being, or could be, collected. However, in many of these applications, implementation of the QIE algorithm may not be practical or desirable.

First, there may be a desire to use different, more extensive performance measures than those generated by the QIE. For example, the QIE generates only the expected wait in queue for a random customer during a congestion period. But some industries may wish to perform off-line analyses of customer arrival times, in order to generate arrival-time distributions for future customer service staffing. Others may wish to use performance measures other than expected waiting time, such as the probability that any customer, or some specific customer, had to wait longer than five minutes in queue (see the paper by Jones and Larson [Jone 91] for an approach to this problem).

Second, when congestion periods are large or analysis needs to be near real-time, there is a need for faster algorithms. In some cases, congestion periods may be enormous: the Citibank ATM's located at City Hall in New York City routinely have lunchtime congestion periods of hundreds of people [Lars 92]. The QIE algorithm has a computational complexity of $O(N^3)$ where N is the number of customers who waited in queue during the given congestion period (this was demonstrated both by Larson—see [Lars 91], and by Bertsimas and Servi, in a multidimensional integration approach—see [Bert 91]). Running the full $O(N^3)$ QIE algorithm on such a large congestion period would be impossible in any sort of desktop or real-time environment, so that some kinds of approximation techniques are called for. Daley and Servi have begun to address this need through the development of an $O(N^2(\ln N))$ algorithm that can approximate the QIE calculations with any prespecified level of precision [Dale 91]. However, even this algorithm may be overly cumbersome for some instances. For example, the banking industry (and potentially others, includ-

ing the supermarket industry) would like to have near real-time analysis available at their ATM sites, so that decisions could be made to switch some ATM's from being full-service to being "express" machines for cash advances only. In this real-time application, some very simple and fast approximation techniques are called for.

Finally, in some environments, additional data which provide partial queue-length information are also available to the analyst. The current QIE algorithm has no way to take advantage of these data. Customer-tracking technologies (pressure-sensitive mats, ultrasonic detectors, etc.) are currently under development and in use in some industries. These can provide information as to whether a queue exists or not, or whether some finite waiting-room capacity has been exceeded. This information can be used to supplement the transactional data upon which the QIE operates. Algorithms which take advantage of these data to provide more accurate queue-length estimates, as well as faster runtimes, are needed, especially when large congestion periods are to be analyzed.

This thesis addresses all of these needs in the following ways. First, we provide an algorithm to determine the exact density functions for both the ordered arrivals and the unordered arrivals in a congestion period. These functions can then be used to provide more extensive estimates of other queue quantities of interest, such as the density of the wait in queue either for a random arrival or for the k-th person to arrive during the congestion period, and the tail probabilities, i.e., the probability that a random arrival waited more than i minutes in queue during a given congestion period. Second, we provide several alternative algorithms which give bounds and approximations to the QIE's exact estimate of the time-dependent number of customers in queue. These algorithms typically run in $O(N)$ or $O(N^2)$ time; and some of them are much less complex than the exact QIE, so provide a realistic option when computing time is a constraint. Finally, we consider an environment in which partial queue-length information is available and develop algorithms to take advantage of the additional information. These algorithms typically give better estimates of the

quantities calculated by the exact QIE and also run in less time.

The problem of queueing inference is relatively new, so there is not a lot of literature directed to this specific area. The work that has been done in the area over the last five years has been based on three approaches, all of which are rooted in familiar and standard mathematical techniques. The approach taken by Larson (see [Lars 90], [Lars 91]) and by much of this thesis is one based on order statistics (see [Barl 72] or [Davi 81]). A second approach, taken by Bertsimas and Servi (see [Bert 91]) and also taken in Chapter 3 of this thesis, is based on multidimensional integration. Finally, a third approach, taken by Daley and Servi (see [Dale 91]), is based on the theory of Markov chains with taboo probabilities, in which specified states of the chain are disallowed at certain times during the process (see [Chun 60]).

After this introduction, the thesis continues in Chapter 2 with an overview of how the original QIE algorithm works, what assumptions it is based upon, and what specific statistics it provides the user. Also covered in Chapter 2 is much of the basic notation that is used in the thesis. Although there is significant overlap with the notation used in [Lars 90], there are also some departures, which should be noted. At the end of Chapter 2, an example of a typical QIE run is provided.

In Chapter 3, a deeper look into the arrival-time and waiting-time densities, both for the k-th customer and for a random customer in a congestion period, is undertaken. Simple examples of congestion periods with $N = 2$ and $N = 3$ are presented to motivate the subsequent generalizations to N customers. Two theorems, which establish the polynomial nature of the ordered arrival-time densities, are proved. Then, a general algorithm, to find the probability density function for the arrival-time of the k-th customer to arrive during an N-customer congestion period, is derived. This algorithm provides all of the arrival-time densities in $O(N^4)$ time. The density function for the arrival-time of a random (unordered) customer is then found. The densities for the queue waiting times, both for ordered and unordered customers, are easily found as a byproduct of the arrival-time densities, under the assumption of a first-

come-first-served queue. Finally, an interesting application for these types of densities from the airline industry is presented in some detail.

Chapter 4 begins to explore the issue of bounds and approximations to the original QIE algorithm. As its primary output, the original algorithm generates the expected cumulative number of arrivals to the system, as a function of time, a function which was shown in [Lars 90] to be concave and piecewise-linear. Hence, first we find a simple lower bound to this function, based on its concavity, which is easily demonstrated to be a lower bound and to be concave. Next, we find an upper bound to the function, based on approximating the ordered arrival-time densities by uniform densities. This is also demonstrated to be an upper bound and to be concave. Then, we find an approximation to the function, which is based on approximating the ordered arrival-time densities by trapezoidal densities. This is demonstrated to be neither an upper nor lower bound, nor a concave nor convex function. However, it is demonstrated to be a lower bound to the uniform upper bound. Finally, computational results are presented from simulation runs, which look at typical output from these algorithms. We also look at output which combines weighted components of pairs of these algorithms to see how these combinations perform in approximating the exact QIE.

Chapter 5 introduces a more general way of framing the problem presented by the original QIE algorithm, which is how to calculate arrival-time probabilities (the probabilities that the k-th customer arrives before the i-th service completion) under a set of very specific conditioning inequalities. In this chapter, the set of conditioning inequalities is greatly generalized. First, the motivation for changing the conditioning inequalities is presented. Next, a theorem is proved, which provides the means to construct a set of stochastically dominant lower bounds to the function generated by the QIE. Finally, a generalized algorithm is derived, which allows calculation of the arrival-time probabilities under (almost) any set of conditioning inequalities.

Chapter 6 takes the generalized algorithm which was found at the end of Chapter

5 and specializes it to two cases, both of which result in lower bounds to the original QIE algorithm. In the first case, we consider simply omitting some of the conditioning inequalities in calculation of the arrival-time probabilities. We look at sample runs of two implementations of this lower bound algorithm, which is considerably faster than the original QIE. We then consider a second approach: namely, adding the constraint that the maximum queue length during the given congestion period remain below some threshold. This allows us to neglect large-queue events in calculation of the arrival-time probabilities and hence saves computation time. Of course, in the case that we actually know that the queue length did not exceed the given value, the algorithm actually gives better queue estimates than the original QIE algorithm, which does not take advantage of this information. We also present sample runs for this lower bound algorithm.

The notion that we might know that some maximum queue length was not exceeded during a congestion period leads to the ideas presented in Chapter 7: namely, that partial queue-length information can improve estimates and simultaneously reduce runtimes. Specifically, we consider a congestion period in which we know the times of all $(M-1)$-to-M and all M-to-$(M-1)$ queue-length transitions. In this case, since we have perfect information as to the state of the system at these instants, we may partition the congestion period and analyze the partitions separately. We present an even more general algorithm which allows calculation of arrival-time probabilities, even when the system does not start empty, and when there is a different number of arrivals than departures during some partition. We also show how to reconstruct all of the queue statistics of interest from these partition analyses. Finally, we present sample runs which demonstrate the reduced runtimes and improved accuracies of the partition analysis over the original QIE analysis.

In Chapter 8, we summarize the results of the thesis and the practical applications of the work. We describe work that still remains to be done, both direct extensions of this thesis, mostly in the areas of heuristics and statistical analysis, and more general

ideas that need to be explored. We continue now with a review of the original QIE algorithm and an introduction to notation.

Chapter 2

Review of the QIE Algorithm and Notation

In this chapter, we review the basic QIE algorithm. We use some of the same notation as that used in [Lars 90], but we have also adapted some new notation where we believe that it clarifies concepts. Hence, as we review the QIE, we also review notation and introduce any modifications.

As already mentioned, the QIE algorithm operates on a single congestion period. Therefore, one issue that arises when dealing with real data is how to determine when the system is in congestion. It is possible to have "gaps" between service completions and the following service commencement, even when all servers are busy, due to such things as time to walk to the next available machine, time to put one's ATM card back in one's wallet, etc. Hence, some preprocessing of the data is required, before the QIE is utilized, to determine exactly when the congestions periods are. Techniques that are employed include: having a set cutoff time for gap lengths within congestion periods; making probabilistic decisions as to whether a gap indicates the end of a congestion period; using historical data for that ATM at that time of day to perform a Bayesian analysis on the gaps; and using partial queue-length information (queue/no queue) to supplement the decision process (this is discussed briefly in Chapter 8). We now

continue with a description of the QIE algorithm.

The QIE model assumes Poisson arrivals to a queueing system, which may comprise a single, multiple, or changing number of servers. The Poisson arrival rate, λ, is assumed to be constant over the duration of a single congestion period. The service distribution may be completely general, the only assumption being that whenever a server becomes available and a queue exists, that server will be utilized immediately by one of the waiting customers. The data that are provided to the QIE include N, the total number of customers who waited in queue during a given congestion period; t_0, the start time of the congestion period, when the last idle server becomes busy, which here we assume to be 0; and $\mathbf{t} \equiv (t_1, t_2, \ldots, t_N)$, the times of service commencement for the queued customers. (We assume $t_0 < t_1 < \cdots < t_N$.) Note that $t_1, t_2, \ldots, t_N$ may also be thought of as service completion or departure times, but, because we allow multiple servers, the customers who depart at these times may or may not be different from customers 1 through N. (Certainly the departure at t_1 cannot be one of these N customers.) For example, in an M/M/s queue with 1000 servers and a 10-customer congestion period, it is quite unlikely that any of the 10 departures during the congestion period would correspond to customers who had just arrived during that congestion period.

Customer 1 is the first customer to arrive and find all servers busy. Customer N is the last customer to commence service immediately after a departure: i.e., the departure at t_{N+1} creates an idle server. So the period during which all servers are busy is actually $[0, t_{N+1}]$. However, because we know that a server was made idle at t_{N+1}, we also know that there must have been exactly zero arrivals to the system during the interval $(t_N, t_{N+1}]$. Therefore, the interval of interest, during which there is some uncertainty as to when arrivals occurred, is the interval $(0, t_N]$; and this is the interval which is analyzed by the QIE algorithm (see Figure 2.1). It is also the only time during the congestion period during which it is possible to have a positive queue length.

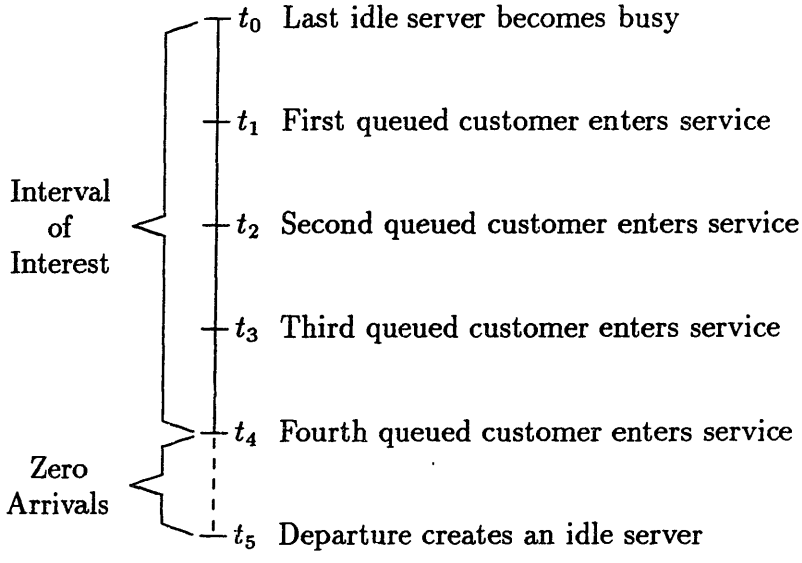


Figure 2.1: Congestion Period Description When $N = 4$

$A(t)$ denotes the counting process (see [Ross 83] for the formal definition of a counting process) which represents the cumulative number of arrivals to the system over the time period $(0, t]$, where $t \leq t_N$. Here, we do not include the arrival that initiated the congestion period in $A(t)$, since we know with certainty when that arrival occurred. Also, we define $A(0) \equiv 0$. The key attribute of a counting process which we will use is that for $s \leq t$, we also have that $A(s) \leq A(t)$. We also let $A(t_1, t_2)$ denote the cumulative number of arrivals to the system over the time period $(t_1, t_2]$, where $t_1 < t_2$. Finally, $Q(t)$ denotes the number of customers in queue at time t. The unordered arrival times of customers 1 to N are denoted by $U_1, U_2, \ldots, U_N$. When these arrival times are ordered, they are denoted by $X_1, X_2, \ldots, X_N$, rather than the more standard $U_{(1)}$, etc. Note that both the U's and the X's are unobserved quantities, while N and the t_i's are observed quantities.

If we let $D(t)$ represent the cumulative number of departures which have occurred during a congestion period, then we have that

$$Q(t) = A(t) - D(t) \tag{2.1}$$

Also, we adopt the following convention:

$$\begin{aligned} D(t) &= i, \quad t_i \le t < t_{i+1}, \quad i = 0, 1, \ldots, N-1 \\ D(t_N) &= N \\ \Longrightarrow \quad Q(t_i) &= Q(t_i^-) - 1 \\ A(t) &= i, \quad X_i \le t < X_{i+1}, \quad i = 0, 1, \ldots, N-1 \\ A(t) &= N, \quad X_N \le t \le t_N \\ \Longrightarrow \quad Q(X_i) &= Q(X_i^-) + 1 \end{aligned} \tag{2.2}$$

that is, the functions A, D, and Q are all right-continuous functions [Rudi 76]. Figure 2.2 depicts a congestion period with $N = 12$ queued customers and shows the relationship between the three functions. This figure does not depict the right-continuous nature of the three functions: were it to do so, the horizontal line segments in the figure would have closed (filled) left endpoints and open right endpoints.

In order for customers 1 through N to comprise a congestion period, it must be the case that $X_1 \le t_1, X_2 \le t_2, \ldots, X_N \le t_N$. Otherwise, if the i-th arrival after all servers became busy occurred after t_i, then a server would have been made idle at time $t = t_i$, and the congestion period would have ended. Note that the above condition required for the congestion period to continue may also be represented by $A(t_1) \ge 1, A(t_2) \ge 2, \ldots, A(t_N) \ge N$. When we combine these conditions with the boundary conditions of the process, i.e. $A(0) = 0$ and $A(t_N) = N$, we get the following set of events which all must occur:

$$\begin{aligned} 0 &\le A(t_0) \le 0 \\ 1 &\le A(t_1) \le N \\ 2 &\le A(t_2) \le N \\ &\vdots \\ N &\le A(t_N) \le N \end{aligned} \tag{2.3}$$

In order to get a more compact notation for this set of events, we make the following

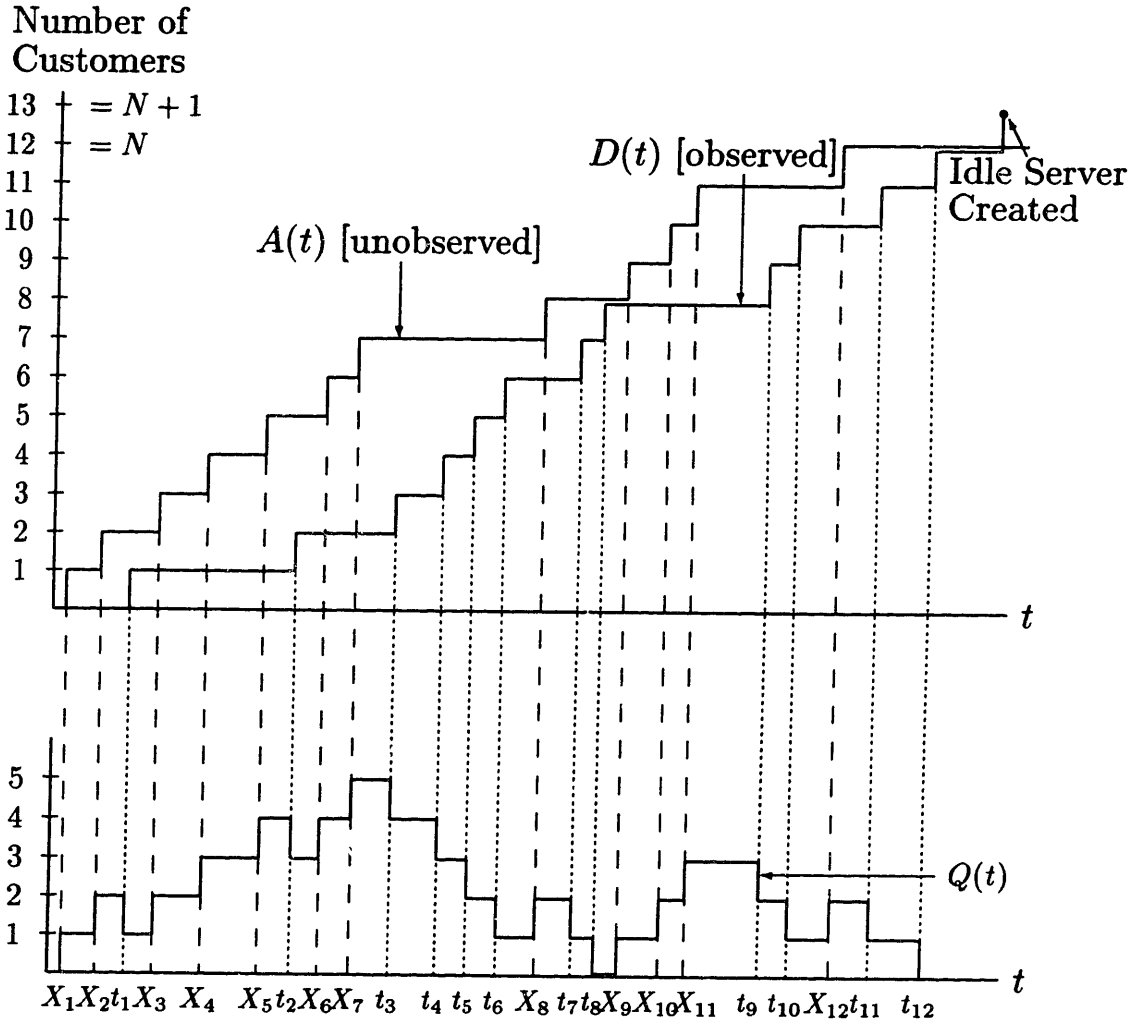


Figure 2.2: $A(t)$, $D(t)$, and $Q(t)$ for a Congestion Period with $N = 12$

general definition:

$$\mathcal{E}^B(\mathbf{t}) \equiv \bigcap_{j=0}^{N} \{l_j \leq A(t_j) \leq u_j\}$$

where B is a general set of lower and upper bounds on the $A(t_j)$'s, given by:

$$B \equiv \{l_0, l_1, l_2, \ldots, l_N, u_0, u_1, u_2, \ldots, u_N\} \tag{2.4}$$

So, by defining the set S to be the standard set of bounds on the $A(t_j)$'s, i.e.

$$S \equiv \{0, 1, 2, \ldots, N, 0, N, N, \ldots, N\} \tag{2.5}$$

then $\mathcal{E}^S(\mathbf{t})$ represents all of the information that we are given about the system during a given congestion period. We also define $\mathcal{E}^{0,N}(\mathbf{t})$ to be the boundary conditions for the process, i.e.

$$\mathcal{E}^{0,N}(\mathbf{t}) \equiv \{A(t_0) = 0\} \bigcap \{A(t_N) = N\}$$

The quantities calculated by the QIE include:

- $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$, the expected cumulative number of arrivals to the system, up to and including time t, conditioned on $\mathcal{E}^S(\mathbf{t})$;
- $\mathrm{E}[Q(t)|\mathcal{E}^S(\mathbf{t})]$, the expected number of customers in queue at time t, also conditional on $\mathcal{E}^S(\mathbf{t})$;
- $\mathrm{E}[L_Q|\mathcal{E}^S(\mathbf{t})]$, the time-averaged queue length over a congestion period;
- $\mathrm{E}[W_Q|\mathcal{E}^S(\mathbf{t})]$, the expected wait in queue over a congestion period;
- $\Pi[k|\mathcal{E}^S(\mathbf{t})]$, the probability that a random arriving customer finds k customers in queue ($k = 0, 1, \ldots, N-1$);
- and $\mathrm{E}[\ell_Q|\mathcal{E}^S(\mathbf{t})]$, the expected queue length experienced by a random arriving customer.

(Note that the PASTA result [Wolf 82] would, in the case of Poisson arrivals, tell us that $\mathrm{E}[L_Q] = \mathrm{E}[\ell_Q]$: however, the conditioning information in $\mathcal{E}^S(\mathbf{t})$ effectively negates the Poisson arrival assumption.) All of these quantities may be calculated as functions of the following quantities:

$$\begin{aligned}\beta_{ki}(\mathbf{t}) &\equiv \Pr[X_k \le t_i | \mathcal{E}^S(\mathbf{t})] \\ &= \Pr[A(t_i) \ge k | \mathcal{E}^S(\mathbf{t})], \quad k = 1, 2, \ldots, N, \quad i = 1, 2, \ldots, N\end{aligned}$$

Note that $\beta_{ki}(\mathbf{t}) = 1$ for $k \le i$. We now describe the method for calculating the other values for $\beta_{ki}(\mathbf{t})$. First, we need the following two definitions:

$$\mathcal{E}^{S \le i}(\mathbf{t}) \equiv \bigcap_{j=0}^{i} \{l_j \le A(t_j) \le u_j\} \tag{2.6}$$

$$\mathcal{E}^{S \ge i}(\mathbf{t}) \equiv \bigcap_{j=i}^{N} \{l_j \le A(t_j) \le u_j\} \tag{2.7}$$

where the l_j's and u_j's are assumed to come from the set S. Now we may begin:

$$\begin{aligned}\beta_{ki}(\mathbf{t}) &= \Pr[A(t_i) \ge k | \mathcal{E}^S(\mathbf{t})] \\ &= \Pr[A(t_i) \ge k+1 | \mathcal{E}^S(\mathbf{t})] + \Pr[A(t_i) = k | \mathcal{E}^S(\mathbf{t})]\end{aligned}$$

Recognizing the first term above as $\beta_{(k+1),i}(\mathbf{t})$ when $k < N$ and zero when $k = N$, we get that:

$$\beta_{ki}(\mathbf{t}) = \begin{cases} \beta_{(k+1),i}(\mathbf{t}) + \Pr[A(t_i) = k | \mathcal{E}^S(\mathbf{t})], & k = 0, 1, \ldots, N-1 \\ \Pr[A(t_i) = k | \mathcal{E}^S(\mathbf{t})], & k = N \end{cases}$$

So clearly the term of interest to calculate is $\Pr[A(t_i) = k | \mathcal{E}^S(\mathbf{t})]$, which we do as follows:

$$\begin{aligned}\Pr[A(t_i) = k | \mathcal{E}^S(\mathbf{t})] &= \frac{\Pr[\mathcal{E}^S(\mathbf{t}) | A(t_i) = k, \mathcal{E}^{0,N}(\mathbf{t})] \times \Pr[A(t_i) = k | \mathcal{E}^{0,N}(\mathbf{t})]}{\Pr[\mathcal{E}^S(\mathbf{t}) | \mathcal{E}^{0,N}(\mathbf{t})]} \\ &= \frac{1}{\Pr[\mathcal{E}^S(\mathbf{t}) | \mathcal{E}^{0,N}(\mathbf{t})]} \Big\{ \Pr[\mathcal{E}^{S \le i}(\mathbf{t}) | A(t_i) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\ &\quad \times \Pr[\mathcal{E}^{S \ge i}(\mathbf{t}) | A(t_i) = k, \mathcal{E}^{0,N}(\mathbf{t})] \times \Pr[A(t_i) = k | \mathcal{E}^{0,N}(\mathbf{t})] \Big\} \\ &= \frac{1}{\tilde{\alpha}_{NN}(\mathbf{t})} \left\{ \tilde{\alpha}_{ki}(\mathbf{t}) \times \tilde{\eta}_{ki}(\mathbf{t}) \times \binom{N}{k} \left(\frac{t_i}{t_N}\right)^k \left(\frac{t_N - t_i}{t_N}\right)^{N-k} \right\}\end{aligned}$$

Note that we may break up $\Pr[\mathcal{E}^S(\mathbf{t})|A(t_i)=k,\mathcal{E}^{0,N}(\mathbf{t})]$ into $\tilde{\alpha}_{ki}(\mathbf{t})\times\tilde{\eta}_{ki}(\mathbf{t})$, because, given the value of $A(t_i)$, events prior to t_i are conditionally independent of events subsequent to t_i. The last term in the braces above is due to the fact that the number of arrivals by time t in a Poisson process that started at time $t=0$ and had N arrivals by time t_N is a binomial random variable with "p" equal to $\frac{t}{t_N}$. In the above, we have used the following definitions for $\tilde{\alpha}_{ki}(\mathbf{t})$ and $\tilde{\eta}_{ki}(\mathbf{t})$:

$$\tilde{\alpha}_{ki}(\mathbf{t}) \equiv \Pr[\mathcal{E}^{S\leq i}(\mathbf{t})|A(t_i)=k,\mathcal{E}^{0,N}(\mathbf{t})] \tag{2.8}$$

$$\tilde{\eta}_{ki}(\mathbf{t}) \equiv \Pr[\mathcal{E}^{S\geq i}(\mathbf{t})|A(t_i)=k,\mathcal{E}^{0,N}(\mathbf{t})] \tag{2.9}$$

Note that:

$$\begin{aligned}\Pr[\mathcal{E}^S(\mathbf{t})|\mathcal{E}^{0,N}(\mathbf{t})] &= \Pr[\mathcal{E}^{S\leq N}(\mathbf{t})|\mathcal{E}^{0,N}(\mathbf{t})] = \tilde{\alpha}_{NN}(\mathbf{t})\\ &= \Pr[\mathcal{E}^{S\geq 0}(\mathbf{t})|\mathcal{E}^{0,N}(\mathbf{t})] = \tilde{\eta}_{00}(\mathbf{t})\end{aligned}$$

The next task is to determine the values in the $\tilde{\alpha}$-matrix for $i=1,2,\ldots,N$ and for $k=1,2,\ldots,N$. First, it should be obvious that:

$$\begin{aligned}\tilde{\alpha}_{k1}(\mathbf{t}) &= \Pr[\mathcal{E}^{S\leq 1}(\mathbf{t})|A(t_1)=k,\mathcal{E}^{0,N}(\mathbf{t})] = 1, \quad k=1,2,\ldots,N\\ \tilde{\alpha}_{ki}(\mathbf{t}) &= 0, \quad k=1,2,\ldots,i-1,\ \ i=2,3,\ldots,N-1\end{aligned}$$

We also define $\tilde{\alpha}_{kN}(\mathbf{t})\equiv 0$ for $k=1,2,\ldots,N-1$. Now consider the following:

$$\begin{aligned}\tilde{\alpha}_{ki}(\mathbf{t}) &= \Pr[\mathcal{E}^{S\leq i}(\mathbf{t})|A(t_i)=k,\mathcal{E}^{0,N}(\mathbf{t})]\\ &= \sum_{j=0}^{k-i+1}\Pr[\mathcal{E}^{S\leq i-1}(\mathbf{t}), i\leq A(t_i)\leq N, A(t_{i-1},t_i)=j|A(t_i)=k,\mathcal{E}^{0,N}(\mathbf{t})]\\ &= \sum_{j=0}^{k-i+1}\Pr[\mathcal{E}^{S\leq i-1}(\mathbf{t})|A(t_{i-1},t_i)=j, A(t_i)=k,\mathcal{E}^{0,N}(\mathbf{t})]\\ &\quad\times\Pr[A(t_{i-1},t_i)=j|A(t_i)=k,\mathcal{E}^{0,N}(\mathbf{t})]\\ &= \sum_{j=0}^{k-i+1}\Pr[\mathcal{E}^{S\leq i-1}(\mathbf{t})|A(t_{i-1})=k-j,\mathcal{E}^{0,N}(\mathbf{t})]\\ &\quad\times\Pr[A(t_{i-1},t_i)=j|A(t_i)=k,\mathcal{E}^{0,N}(\mathbf{t})]\\ &= \sum_{j=0}^{k-i+1}\tilde{\alpha}_{(k-j),(i-1)}(\mathbf{t})\times\binom{k}{j}\left(\frac{t_{i-1}}{t_i}\right)^{k-j}\left(\frac{t_i-t_{i-1}}{t_i}\right)^{j},\\ &\quad k=i,i+1,\ldots,N, \quad i=2,3,\ldots,N\end{aligned} \tag{2.10}$$

So we first fill in the first column of the matrix with ones and the upper half of the matrix with zeroes; then we proceed to the second column, and the third column, etc., each time calculating the unknown values using the values from the previous column.

Next, we must determine the values in the $\tilde{\eta}$-matrix for $i = 0, 1, \ldots, N-1$ and for $k = 0, 1, \ldots, N$. Again, it should be obvious that:

$$\begin{aligned}
\tilde{\eta}_{Ni}(\mathbf{t}) &= \Pr[\mathcal{E}^{S\geq i}(\mathbf{t}) | A(t_i) = N, \mathcal{E}^{0,N}(\mathbf{t})] = 1, \quad i = 1, 2, \ldots, N-1 \\
\tilde{\eta}_{k,(N-1)}(\mathbf{t}) &= \Pr[\mathcal{E}^{S\geq N-1}(\mathbf{t}) | A(t_{N-1}) = k, \mathcal{E}^{0,N}(\mathbf{t})] = \begin{cases} 0, & k = 0, 1, \ldots, N-2 \\ 1, & k = N-1 \end{cases} \\
\tilde{\eta}_{ki}(\mathbf{t}) &= 0, \quad k = 0, 1, \ldots, i-1, \quad i = 1, 2, \ldots, N-2
\end{aligned}$$

We also define $\tilde{\eta}_{k0}(\mathbf{t}) \equiv 0$ for $k = 1, 2, \ldots, N$. Now consider the following:

$$\begin{aligned}
\tilde{\eta}_{ki}(\mathbf{t}) &= \Pr[\mathcal{E}^{S\geq i}(\mathbf{t}) | A(t_i) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\
&= \sum_{j=0}^{N-k} \Pr[\mathcal{E}^{S\geq i+1}(\mathbf{t}), i \leq A(t_i) \leq N, A(t_i, t_{i+1}) = j | A(t_i) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\
&= \sum_{j=0}^{N-k} \Pr[\mathcal{E}^{S\geq i+1}(\mathbf{t}) | A(t_i, t_{i+1}) = j, A(t_i) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\
&\quad \times \Pr[A(t_i, t_{i+1}) = j | A(t_i) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\
&= \sum_{j=0}^{N-k} \Pr[\mathcal{E}^{S\geq i+1}(\mathbf{t}) | A(t_{i+1}) = k + j, \mathcal{E}^{0,N}(\mathbf{t})] \\
&\quad \times \Pr[A(t_i, t_{i+1}) = j | A(t_i) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\
&= \sum_{j=0}^{N-k} \tilde{\eta}_{(k+j),(i+1)}(\mathbf{t}) \times \binom{N-k}{j} \left(\frac{t_{i+1} - t_i}{t_N - t_i}\right)^j \left(\frac{t_N - t_{i+1}}{t_N - t_i}\right)^{N-k-j}, \\
&\quad k = i, i+1, \ldots, N-1, \; i = 1, 2, \ldots, N-2, \text{ and } k=0, i=0 \qquad (2.11)
\end{aligned}$$

So we first fill in the last column and bottom row of the matrix with zeroes and ones as specified above; and the upper half of the matrix with zeroes. Then we proceed to the second-to-last column, and the third-to-last column, etc., each time calculating the unknown values using the values from the column to the right.

We now present the following definitions, which make all of the above equations

simpler and follow exactly the equations given in [Lars 90]:

$$\alpha_{ki}(\mathbf{t}) \equiv \tilde{\alpha}_{ki}(\mathbf{t}) \times \left(\frac{t_i}{t_N}\right)^k$$
$$\eta_{ki}(\mathbf{t}) \equiv \tilde{\eta}_{ki}(\mathbf{t}) \times \left(\frac{t_N - t_i}{t_N}\right)^{N-k}$$

With these definitions, we have the following for the definitions of the α-matrix:

$$\begin{aligned}
\alpha_{k1}(\mathbf{t}) &= \left(\frac{t_1}{t_N}\right)^k, \quad k = 1,2,\ldots,N \\
\alpha_{ki}(\mathbf{t}) &= 0, \quad k = 1,2,\ldots,i-1, \quad i = 2,3,\ldots,N \\
\alpha_{ki}(\mathbf{t}) &= \sum_{j=0}^{k-i+1} \alpha_{(k-j),(i-1)}(\mathbf{t}) \times \binom{k}{j} \left(\frac{t_i - t_{i-1}}{t_N}\right)^j, \\
&\quad k = i, i+1,\ldots,N, \quad i = 2,3,\ldots,N
\end{aligned}$$

The η-matrix is defined by the following:

$$\begin{aligned}
\eta_{Ni}(\mathbf{t}) &= 1, \quad i = 1,2,\ldots,N-1 \\
\eta_{k,(N-1)}(\mathbf{t}) &= \begin{cases} 0, & k = 0,1,\ldots,N-2 \\ \dfrac{t_N - t_{N-1}}{t_N}, & k = N-1 \end{cases} \\
\eta_{ki}(\mathbf{t}) &= 0, \quad k = 0,1,\ldots,i-1, \; i = 1,2,\ldots,N-2 \\
&\quad \text{and} \quad k = 1,2,\ldots,N-1, \; i = 0 \\
\eta_{ki}(\mathbf{t}) &= \sum_{j=0}^{N-k} \eta_{(k+j),(i+1)}(\mathbf{t}) \times \binom{N-k}{j} \left(\frac{t_{i+1} - t_i}{t_N}\right)^j, \\
&\quad k = i, i+1,\ldots,N-1, \; i = 1,2,\ldots,N-2, \quad \text{and} \quad k = 0, \; i = 0
\end{aligned}$$

Finally, we have for the β-matrix:

$$\begin{aligned}
\beta_{ki}(\mathbf{t}) &= 1, \quad k = 1,2,\ldots,i, \; i = 1,2,\ldots,N \\
\beta_{Ni}(\mathbf{t}) &= \Pr[A(t_i) = N | \mathcal{E}^S(\mathbf{t})] \\
&= \frac{\tilde{\alpha}_{Ni}(\mathbf{t})\tilde{\eta}_{Ni}(\mathbf{t})}{\tilde{\alpha}_{NN}(\mathbf{t})} \left(\frac{t_i}{t_N}\right)^N \\
&= \frac{\alpha_{Ni}(\mathbf{t})}{\alpha_{NN}(\mathbf{t})}, \quad i = 1,2,\ldots,N-1 \\
\beta_{ki}(\mathbf{t}) &= \beta_{(k+1),i}(\mathbf{t}) + \frac{1}{\alpha_{NN}(\mathbf{t})} \left\{ \binom{N}{k} \alpha_{ki}(\mathbf{t})\eta_{ki}(\mathbf{t}) \right\}, \\
&\quad k = i+1, i+2,\ldots,N-1, \; i = 1,2,\ldots,N-2
\end{aligned}$$

Hence, we begin by filling in the upper right triangle of the β-matrix with ones; then we calculate the bottom row of the matrix; finally we calculate each column, from the bottom up, by a multiplication of elements from the α-matrix and the η-matrix, which we then add to the element of the β-matrix just below the one being calculated.

The method just presented for calculating the values in the β-matrix is very similar to that presented in [Lars 90], except that here the derivations are in terms of the quantity $A(t_i)$, while in [Lars 90], they are in terms of the quantity X_k. We will be focussing on the quantity $A(t_i)$ in much of what is to come: hence, the presentation here in those terms.

Finally, we review the method by which the queue statistics are calculated from the $\beta_{ki}(\mathbf{t})$'s. The quantities that we will deal with the most are $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$ and $\mathrm{E}[Q(t)|\mathcal{E}^S(\mathbf{t})]$. Many of the approximations and bounds in the later chapters of the thesis are found in terms of $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$, but most of the figures that we present are comparisons of $\mathrm{E}[Q(t)|\mathcal{E}^S(\mathbf{t})]$. Note that we may find $\mathrm{E}[Q(t)|\mathcal{E}^S(\mathbf{t})]$ from $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$ as follows (see Equations 2.1 and 2.2):

$$\begin{aligned} Q(t) &= A(t) - i, \quad t_i \le t < t_{i+1}, \quad i = 0, 1, \ldots, N-1 \qquad (2.12) \\ Q(t_N) &= 0 \\ \Longrightarrow \quad \mathrm{E}[Q(t)|\mathcal{E}^S(\mathbf{t})] &= \mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})] - i, \quad t_i \le t < t_{i+1}, \quad i = 0, 1, \ldots, N-1 \end{aligned}$$

We know that $\mathrm{E}[A(t_0)|\mathcal{E}^S(\mathbf{t})] = 0$ and that $\mathrm{E}[A(t_N)|\mathcal{E}^S(\mathbf{t})] = N$. Further, Larson showed that $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$ is a concave, piecewise-linear function, with breakpoints at the t_i's [Lars 90]. Hence, we need only find $\mathrm{E}[A(t_i)|\mathcal{E}^S(\mathbf{t})]$ for $i = 1, 2, \ldots, N-1$ to determine the entire function. We find $\mathrm{E}[A(t_i)|\mathcal{E}^S(\mathbf{t})]$ from the following:

$$\begin{aligned} \mathrm{E}[A(t_i)|\mathcal{E}^S(\mathbf{t})] &= \sum_{k=1}^{N} \Pr[A(t_i) \ge k|\mathcal{E}^S(\mathbf{t})] \\ &= \sum_{k=1}^{N} \beta_{ki}(\mathbf{t}), \quad i = 1, 2, \ldots, N-1 \end{aligned}$$

The quantity $\mathrm{E}[L_Q|\mathcal{E}^S(\mathbf{t})]$ is easily found from the following:

$$\mathrm{E}[L_Q|\mathcal{E}^S(\mathbf{t})] = \frac{1}{t_N} \int_0^{t_N} \mathrm{E}[Q(t)|\mathcal{E}^S(\mathbf{t})]dt$$

$$= \frac{1}{t_N}\sum_{i=1}^{N}(t_i - t_{i-1}) \times \frac{1}{2}\left\{\mathrm{E}[Q(t_i)|\mathcal{E}^S(\mathbf{t})] + \mathrm{E}[Q(t_{i-1})|\mathcal{E}^S(\mathbf{t})] + 1\right\}$$

We also have:

$$\mathrm{E}[W_Q|\mathcal{E}^S(\mathbf{t})] = \left(\frac{t_N}{N}\right)\mathrm{E}[L_Q|\mathcal{E}^S(\mathbf{t})]$$

We can find $\Pi[k|\mathcal{E}^S(\mathbf{t})]$ from the following:

$$\Pi[k|\mathcal{E}^S(\mathbf{t})] = \frac{1}{N}\left[\sum_{j=1}^{N-k-1}\left(\beta_{(k+j),j}(\mathbf{t}) - \beta_{(k+j+1),j}(\mathbf{t})\right) + \beta_{N,(N-k)}(\mathbf{t})\right],$$
$$k = 0, 1, \ldots, N-1$$

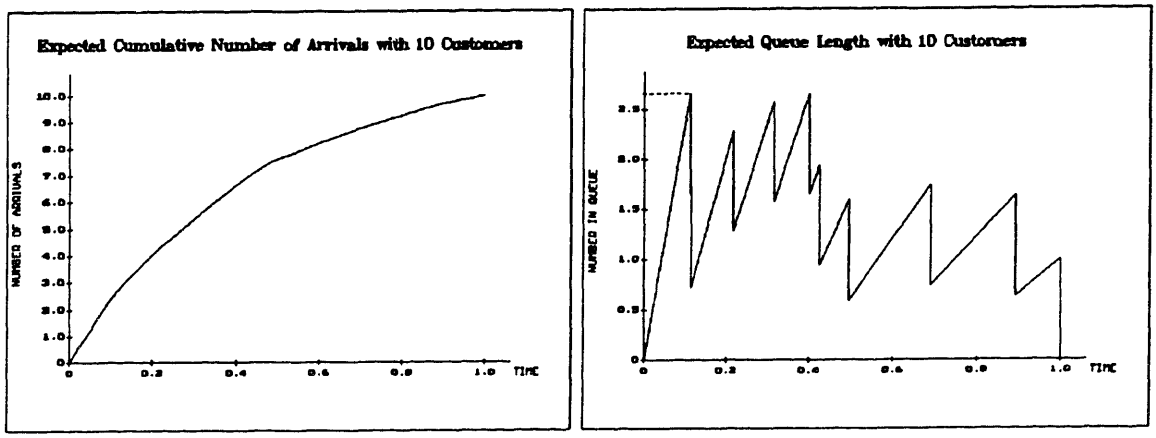
Finally $\mathrm{E}[\ell_Q|\mathcal{E}^S(\mathbf{t})]$ is found from:

$$\mathrm{E}[\ell_Q|\mathcal{E}^S(\mathbf{t})] = \sum_{k=0}^{N-1} k \times \Pi[k|\mathcal{E}^S(\mathbf{t})]$$

For details of any of the above derivations, see [Lars 90].

Figure 2.3 depicts both $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$ and $\mathrm{E}[Q(t)|\mathcal{E}^S(\mathbf{t})]$ for a congestion period with $N = 10$ customers and $\mathbf{t}$-vector as given. Also presented are the β-matrix and all of the above statistics for this congestion period ($\Pi[k|\mathcal{E}^S(\mathbf{t})]$ is abbreviated Π_k).

Having now reviewed all of the notation and concepts from the original QIE algorithm which are pertinent to this thesis, we continue with an analysis of the densities of the ordered and unordered arrival times.



Number of Customers = 10
t-vector:

0.1132	0.1157	0.2185	0.3150	0.4011	0.4248	0.4938	0.6906	0.8936	1.0000

Matrix of the Betas:

1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.9790	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.4934	0.5159	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.1557	0.1669	0.7504	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.0321	0.0353	0.3820	0.8447	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.0043	0.0048	0.1244	0.5113	0.9235	1.0000	1.0000	1.0000	1.0000	1.0000
0.0004	0.0004	0.0241	0.1826	0.5457	0.6797	1.0000	1.0000	1.0000	1.0000
0.0000	0.0000	0.0027	0.0346	0.1577	0.2196	0.4696	1.0000	1.0000	1.0000
0.0000	0.0000	0.0002	0.0036	0.0242	0.0374	0.1093	0.5930	1.0000	1.0000
0.0000	0.0000	0.0000	0.0002	0.0016	0.0028	0.0108	0.1494	0.6392	1.0000

Vector of Incidence Probabilities:

$\Pi_0 = 0.3605$ $\Pi_1 = 0.3817$ $\Pi_2 = 0.1874$ $\Pi_3 = 0.0581$ $\Pi_4 = 0.0110$

$\Pi_5 = 0.0012$ $\Pi_6 = 0.0001$ $\Pi_7 = 0.0000$ $\Pi_8 = 0.0000$ $\Pi_9 = 0.0000$

Expected Cumulative Number of Arrivals By t_i:

$t_1 : 2.6648$ $t_2 : 2.7234$ $t_3 : 4.2838$ $t_4 : 5.5770$ $t_5 : 6.6528$

$t_6 : 6.9395$ $t_7 : 7.5898$ $t_8 : 8.7424$ $t_9 : 9.6392$ $t_{10} : 10.0000$

Other statistics:

$$\mathrm{E}[L_Q|\mathcal{E}^S(\mathrm{t})] = 1.3663 \quad \mathrm{E}[W_Q|\mathcal{E}^S(\mathrm{t})] = 0.1366 \quad \mathrm{E}[\ell_Q|\mathcal{E}^S(\mathrm{t})] = 0.9813$$

Figure 2.3: Queue Statistics for $N = 10$ Congestion Period

Chapter 3

Arrival and Wait Time Densities

It is useful to consider the probability density functions of the times of customer arrivals over a single congestion period, conditioned on the arrival-time inequalities, for two primary reasons. First, there are occasions on which we might require more extensive performance measures than those generated by the β-matrix. An example of such an application is provided at the end of this chapter. Second, by obtaining insight into these densities, we may determine some useful ways of approximating them. We first consider the two specific cases of $N = 2$ and $N = 3$, and then we proceed to make some inferences about the distributions for general N. We describe a simple way to determine the density function for the time of arrival of the last customer in the congestion period, and then we give an $O(N^4)$ algorithm to determine the density functions for all N customers. We briefly describe the density function for the *unordered* times of customer arrivals, conditioned on the arrival-time inequalities (without the conditioning, these densities are just uniform over the duration of the congestion period). Finally, a derivation for waiting time densities under the assumption of a first-come-first-served (FCFS) queue is presented. At the end of the chapter, two applications in which these density functions could be utilized directly, are explored. In Sections 3.1 and 3.2, we normalize the duration of the congestion

period to be 1 time unit, i.e. we consider departure times $t'_1, t'_2, \ldots, t'_N = 1$, where

$$t'_i = t_i/t_N, \quad i = 1, 2, \ldots, N$$

Note that this engenders no loss of generality. As mentioned in Chapter 2, we also define $t_0 \equiv 0$.

3.1 Congestion Periods with $N = 2$ Queued Customers

It is well-known that without the arrival-time inequality conditions, the joint density function for X_1 and X_2 during a congestion period with $N = 2$ queued customers is uniform over the triangular region defined by $0 < X_1 < X_2 \leq 1$. Consequently, the marginal densities for X_1 and X_2 are both linear and given by:

$$\begin{aligned} f(X_1) &= 2 - 2X_1, \quad 0 < X_1 \leq 1 \\ f(X_2) &= 2X_2, \quad 0 < X_2 \leq 1 \end{aligned}$$

When we add the single arrival-time inequality, $X_1 \leq t_1$, as a conditioning event, the region over which the joint density is uniform is reduced to the trapezoidal area defined by $0 < X_1 \leq t_1$, $X_1 < X_2 \leq 1$ (see Figure 3.1). Hence, it is easy to determine that the marginal densities for X_1 and X_2, given $X_1 \leq t_1$, are now as shown in Figure 3.1, and are given by the following expressions:

$$f(X_1|X_1 \leq t_1) = \frac{2 - 2X_1}{t_1(2 - t_1)}, \quad 0 < X_1 \leq t_1$$

$$f(X_2|X_1 \leq t_1) = \begin{cases} \dfrac{2X_2}{t_1(2 - t_1)}, & 0 < X_2 \leq t_1 \\ \dfrac{2}{2 - t_1}, & t_1 < X_2 \leq 1 \end{cases}$$

As expected, the conditional density function for X_1 is still linearly decreasing, but truncated (and scaled accordingly) at the value t_1. The conditional density function

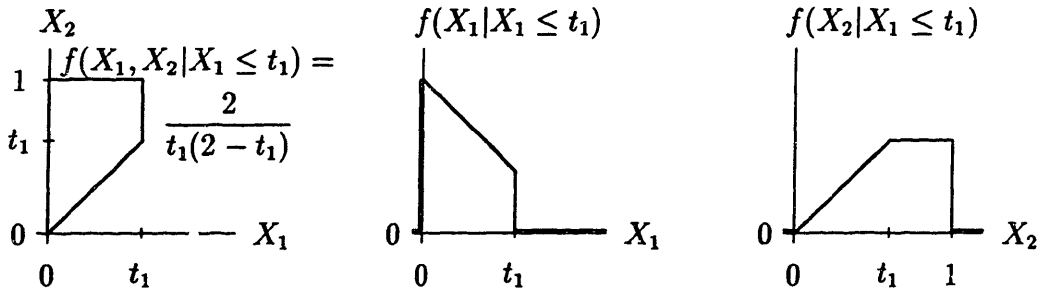


Figure 3.1: Joint and Marginal Densities for X_1 and X_2, Given $X_1 \leq t_1$

for X_2 starts out to be linearly increasing, as before, up to the value of t_1, and then becomes constant for values between t_1 and 1. This makes sense: given that X_2 occurred between t_1 and 1, it was the only arrival during that interval, and so its conditional density function should be uniform. Similarly, given that X_2 occurred between 0 and t_1, we know it was the maximum of two arrivals over that interval and so its conditional density is linearly increasing. Finally, the conditional density for X_1 may be thought of as a weighted average of a uniform function and a linearly decreasing (down to 0) function, where the weighting depends on whether $X_2 > t_1$ or $X_2 \leq t_1$, respectively. (This can be seen clearly by considering the conditional joint density.)

Note that all of the above discussion has depended on knowing the exact value of t_1. But what if that value is not known? It is interesting and instructive to examine the resulting densities in that case. Consider an $M/M/1$ queue with a congestion period of known length and 2 queued customers. Say, however, that the time of service initiation of the first queued customer, t_1, is not known. Before we order the arrivals, then, and before we add the constraint $X_1 \leq t_1$, we know that during the time unit of the congestion period, we have exactly three independent Poisson events from the process which is the combination of the arrival Poisson process and the service Poisson process. Hence, the joint density function for these three events

is uniform over the unit cube. Requiring $X_1 \leq X_2$ and $X_1 \leq t_1$ cuts up the sample space, but the joint density over that new space is still uniform, with value 3, since

$$\Pr[X_1 \leq X_2, X_1 \leq t_1] = \int_0^1 dX_1 \int_{X_1}^1 dX_2 \int_{X_1}^1 dt_1 = 1/3$$

So we now have a partial ordering on the three events: X_1 is the minimum of three Poisson events on the unit time interval; and X_2 and t_1 are the unordered (among themselves) second and third of these events. For simplicity of notation in what follows, we call $\mathcal{C}$ the set of conditioning events given by $X_1 \leq X_2$ and $X_1 \leq t_1$. When we also know the value of one of the variables, say X_1, we denote our conditions as $\mathcal{C}, X_1$.

First note that X_2 and t_1 are indistinguishable in terms of their probability densities: they are both Poisson arrivals over the unit time interval, and both must occur after X_1. Hence, we may use some of the results discussed previously to determine other densities of interest. For instance, consider the two marginal densities depicted in Figure 3.1. The first, $f(X_1|X_1 \leq t_1)$, or, in our new notation, $f(X_1|\mathcal{C}, t_1)$ can also be thought of as representing $f(X_1|\mathcal{C}, X_2)$, for values of X_1 between 0 and X_2. Similarly, $f(X_2|\mathcal{C}, t_1)$ is the same density as $f(t_1|\mathcal{C}, X_2)$, and is linear for values of t_1 between 0 and X_2, and then is uniform between X_2 and 1. It should also be obvious that $f(t_1|\mathcal{C}, X_1) = f(X_2|\mathcal{C}, X_1)$, and that both of these densities are uniform between X_1 and 1. (Given the value of X_1, we simply have two independent Poisson arrivals during $(X_1, 1]$, so they are both uniformly distributed over that interval.) We also know that if we are given both the values of t_1 and X_2, then the density for X_1 will be uniform over the interval between 0 and the minimum of the two given values. Finally, we calculate the marginal densities for X_1 and X_2 (t_1 will have the same marginal density as X_2). The marginal for X_1 is just the density for the minimum of three random variables which are independent and uniform over the unit time interval. This density is given by

$$f(X_1) = 3(1 - X_1)^2, \quad 0 < X_1 \leq 1$$

i.e. it is a convex decreasing function of X_1. We find $f(X_2)$ by the following:

$$\begin{aligned} f(X_1, X_2) &= \int_{X_1}^{1} 3\,dt_1 = 3 - 3X_1, \quad 0 < X_1 \leq X_2 \leq 1 \\ f(X_2) &= \int_{0}^{X_2} (3 - 3X_1)\,dX_1 = 3X_2 - 1.5X_2^2, \quad 0 < X_2 \leq 1 \end{aligned}$$

So we find that the marginals for X_2 and t_1 are concave increasing functions of their random variables.

3.2 Congestion Periods with $N = 3$ Queued Customers

We now proceed to analyze completely a congestion period having three queued customers and return to the case in which we know the exact times of service commencement for all customers during the congestion period. This will lead naturally to some generalizations for N customers, as well as giving us some ideas of ways to approximate the densities of the X's. Again, we first review the densities of the X's without the arrival-time inequality conditions. For three independent random variables which are uniform on the unit time interval, the probability densities for the minimum (X_1), the second smallest (X_2), and the maximum (X_3) of the three random variables are as follows (these are well-known and easily-derivable results from order statistics):

$$\begin{aligned} f(X_1) &= 3(1 - X_1)^2, \quad 0 < X_1 \leq 1 \\ f(X_2) &= 6X_2(1 - X_2), \quad 0 < X_2 \leq 1 \\ f(X_3) &= 3X_3^2, \quad 0 < X_3 \leq 1 \end{aligned}$$

These densities are depicted in Figure 3.2.

We now proceed to the case in which the arrival-time inequality conditions (which we denote by $\mathcal{E}^S(\mathbf{t})$, as defined in Chapter 2) apply and begin by observing that, as in the case of $N = 2$, the joint density of the three ordered arrival times, given the arrival time conditions, is a constant. To find its value, we note that it must be the

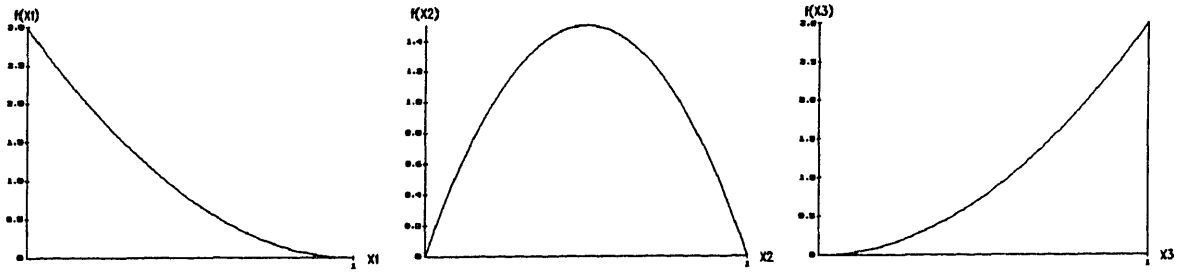


Figure 3.2: Densities for X_1, X_2, and X_3

reciprocal of the volume, V, over which $\mathcal{E}^S(\mathbf{t})$ may occur:

$$\int_0^{t_1} dX_1 \int_{X_1}^{t_2} dX_2 \int_{X_2}^{1} dX_3 = V$$

So we find that

$$V = t_1 t_2 - \frac{t_1 t_2^2}{2} - \frac{t_1^2}{2} + \frac{t_1^3}{6}$$

We may thus find the density for X_1 as follows:

$$\begin{aligned} f[X_1, X_2|\mathcal{E}^S(\mathbf{t})] &= \frac{1}{V}\int_{X_2}^{1} dX_3 = \frac{1-X_2}{V} \\ f[X_1|\mathcal{E}^S(\mathbf{t})] &= \frac{1}{V}\int_{X_1}^{t_2}(1-X_2)dX_2 \\ &= \frac{1}{V}\left[t_2 - \frac{t_2^2}{2} - X_1 + \frac{X_1^2}{2}\right], \quad 0 < X_1 \leq t_1 \end{aligned}$$

Not surprisingly, the density for X_1 is a convex quadratic (just like the density in the unconditioned case), which is truncated at t_1 and scaled appropriately. It may also be thought of as the weighted sum of a convex quadratic, a linearly decreasing function, and a constant, where the weighting is given by the probability of three, two, or one arrival respectively in $(0, t_1]$.

The density for X_2 is found as follows:

$$
\begin{aligned}
f[X_2|\mathcal{E}^S(\mathrm{t})] &= \frac{1}{V}\int_0^{\min(X_2,t_1)}(1-X_2)\,dX_1 \\
&= \begin{cases} \frac{1}{V}\int_0^{X_2}(1-X_2)\,dX_1, & 0 < X_2 \le t_1 \\ \frac{1}{V}\int_0^{t_1}(1-X_2)\,dX_1, & t_1 < X_2 \le t_2 \end{cases} \\
&= \begin{cases} \frac{1}{V}(X_2 - X_2^2), & 0 < X_2 \le t_1 \\ \frac{1}{V}(t_1 - t_1 X_2), & t_1 < X_2 \le t_2 \end{cases}
\end{aligned}
$$

This density is a concave quadratic for values of $X_2 \le t_1$, and it then decreases linearly for values of X_2 between t_1 and t_2. Again, this makes sense: under the conditioning, X_2 is the second earliest of two or three Poisson arrivals between 0 and t_1 (weighted sum of a linearly increasing function and a concave quadratic), and it is the minimum of one or two Poisson arrivals between t_1 and t_2 (weighted sum of a constant and a linearly decreasing function).

Finally, we may find the marginal density for X_3 as follows:

$$
\begin{aligned}
f[X_2, X_3|\mathcal{E}^S(\mathrm{t})] &= \frac{1}{V}\int_0^{\min(X_2,t_1)} dX_1 = \begin{cases} \frac{X_2}{V} & 0 < X_2 \le t_1 \\ \frac{t_1}{V} & t_1 < X_2 \le t_2 \end{cases} \\
f[X_3|\mathcal{E}^S(\mathrm{t})] &= \int_0^{\min(X_3,t_2)} f[X_2, X_3|\mathcal{E}^S(\mathrm{t})]\,dX_2 \\
&= \begin{cases} \int_0^{X_3}\frac{X_2}{V}\,dX_2, & 0 < X_3 \le t_1 \\ \int_0^{t_1}\frac{X_2}{V}\,dX_2 + \int_{t_1}^{X_3}\frac{t_1}{V}\,dX_2, & t_1 < X_3 \le t_2 \\ \int_0^{t_1}\frac{X_2}{V}\,dX_2 + \int_{t_1}^{t_2}\frac{t_1}{V}\,dX_2, & t_2 < X_3 \le 1 \end{cases}
\end{aligned}
$$

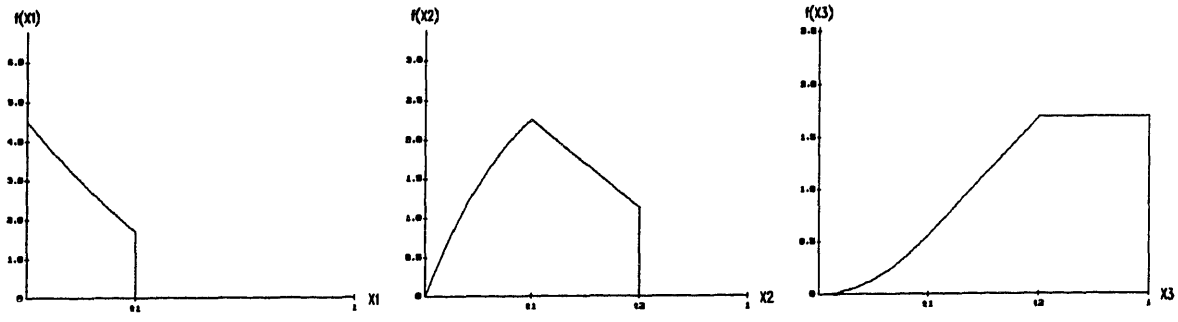


Figure 3.3: Densities for X_1, X_2, and X_3, Given $X_1 \leq t_1, X_2 \leq t_2, X_3 \leq 1$

$$= \begin{cases} \frac{1}{V}\left(\frac{X_3^2}{2}\right), & 0 < X_3 \leq t_1 \\ \frac{1}{V}\left(t_1 X_3 - \frac{t_1^2}{2}\right), & t_1 < X_3 \leq t_2 \\ \frac{1}{V}\left(t_1 t_2 - \frac{t_1^2}{2}\right), & t_2 < X_3 \leq 1 \end{cases}$$

This density is a convex quadratic for values of $X_3 < t_1$, increases linearly for values of X_3 between t_1 and t_2, and then is constant for values of X_3 between t_2 and 1. Again, this makes sense: if $X_3 \leq t_1$, then we know that it is the maximum of three Poisson arrivals over that interval; if $t_1 < X_3 \leq t_2$, then it is either the maximum of two Poisson arrivals (linear increasing) or the only Poisson arrival (constant) over that interval; finally, if $t_2 < X_3 \leq 1$, then we know that it is the only Poisson arrival in that interval and hence its time of arrival is uniformly distributed over the interval.

The densities for X_1, X_2, and X_3 conditioned on the arrival-time inequalities are depicted in Figure 3.3. It is interesting to note that all of these densities are continuous and that X_3 is continuous in slope between 0 and t_2^-. We shall see how this pattern generalizes in the next section.

3.3 Congestion Periods with N Queued Customers

In this section, we prove two theorems about the ordered arrivals, conditioned on the arrival-time inequalities, when we have a general number, N, of queued customers. In particular, we first consider determining something about the density for the arrival time of the k-th customer when the arrival time of the $(k-1)$th customer is known. Then we proceed to make some inferences about the shape of the marginal density for the arrival time of the k-th customer when no other information is available. Specifically, we will prove the following:

Theorem 3.1 *$f_{X_k}(X|\mathcal{E}^S(\mathbf{t}), X_{k-1})$ is a non-increasing, convex, $(N-k)$-th order polynomial, for $X_{k-1} < X \leq t_k$.*

Theorem 3.2 *$f[X_k|\mathcal{E}^S(\mathbf{t})]$ is a continuous function for $0 < X_k < t_k$, which, for $t_{i-1} < X_k \leq t_i$, is an $(N-i)$-th order polynomial. Additionally, $\frac{d^j}{dX_k^j} f[X_k|\mathcal{E}^S(\mathbf{t})]$ is continuous for $0 < X_k < t_{k-j}$.*

Before we prove either of these, however, we need a result from order statistics. This result states that the jth smallest of M uniform arrivals over an interval $(t, t+a]$ has a density that is given by:

$$f_{X_j}(X) = \frac{1}{a^M} \frac{M!}{(j-1)!(M-j)!} (X-t)^{j-1} (t+a-X)^{M-j}, \quad t < X \leq t+a \qquad (3.1)$$

(Note that, when $a = 1$ and $t = 0$, this is a beta density.) This result is easily derived by considering $f_{X_j}(X)dX$ to be the probability that the jth smallest arrival occurs in $(X, X+dX]$. In order for this to happen, we had to have $j-1$ arrivals in $(t, X]$ (which occurs with probability $[(X-t)/a]^{j-1}$); we had to have $M-j$ arrivals in $(X+dX, t+a]$ (probability $[(t+a-X)/a]^{M-j}$); and we had to have a single arrival in $(X, X+dX]$ (probability dX/a). Finally, the M arrivals could have occurred in any of $M!$ orders, but we don't care about the ordering of the arrivals prior to X_j (divide by $(j-1)!$) nor of those after X_j (divide by $(M-j)!$). Combining these results and dividing both sides by dX, we obtain the result given above.

3.3.1 Proof of Theorem 3.1

We now proceed to determine the form for the conditional probability density function for the k-th arrival, given the arrival time of customer $k-1$, $f_{X_k}(X|\mathcal{E}^S(\mathbf{t}), X_{k-1})$. (We have adopted slightly more cumbersome notation for the probability density fuction of X_k in this section only, to indicate that X is the value that random variable X_k takes on, and X_{k-1} is the given time of arrival of customer $k-1$.) Given the time at which X_{k-1} occurs, we know that X_k is the next arrival, and we know that it occurs before t_k. We also know that any number of the remaining $N-k$ customers could also arrive prior to t_k. By conditioning on the total number of arrivals in $(X_{k-1}, t_k]$, we may more easily determine the conditional density for X_k. Specifically, we define B^i to be the following event:

$$\begin{aligned} B^i \equiv\ & \{X_{k+1} \le t_k, \ldots, X_{k+i} \le t_k, t_k < X_{k+i+1} \le t_{k+i+1}, \ldots, t_k < X_N \le t_N\}, \\ & i = 0, 1, \ldots, N-k \end{aligned}$$

Clearly, exactly one of the B^i occurs, so we may write

$$f_{X_k}(X|\mathcal{E}^S(\mathbf{t}), X_{k-1}) = \sum_{i=0}^{N-k} f_{X_k}(X|\mathcal{E}^S(\mathbf{t}), X_{k-1}, B^i) \times \Pr(B^i|\mathcal{E}^S(\mathbf{t}), X_{k-1}) \qquad (3.2)$$

i.e., the conditional density is just a weighted sum of densities which are also conditional on the total number of arrivals, $i+1$ (including arrival k), which occur in the interval $(X_{k-1}, t_k]$. But, given that exactly $i+1$ arrivals occur in that interval, we know that X_k is the minimum of $i+1$ Poisson arrivals on a fixed interval and hence that its density is described by Equation 3.1, with $j = 1, M = i+1, t = X_{k-1}$, and $a = t_k - X_{k-1}$. Hence, we have that

$$\begin{aligned} f_{X_k}(X|\mathcal{E}^S(\mathbf{t}), X_{k-1}, B^i) \ =\ & \frac{i+1}{(t_k - X_{k-1})^{i+1}}(t_k - X)^i, \quad X_{k-1} < X \le t_k \\ & i = 0, 1, \ldots, N-k \end{aligned}$$

So given the time of the $(k-1)$th arrival, the density for the time of the k-th arrival is just the weighted sum of $N-k+1$ polynomials in X, the highest order of which

(when $i = N - k$) is an $(N-k)$-order polynomial; thus, *the weighted sum will just be an $(N-k)$-order polynomial in X.*

Now consider the following derivatives:

$$\frac{d}{dX} f_{X_k}(X|\mathcal{E}^S(\mathbf{t}), X_{k-1}, B^i) = \begin{cases} -\dfrac{i(i+1)}{(t_k - X_{k-1})^{i+1}}(t_k - X)^{i-1}, & i=1,2,\ldots,N-k \\ 0, & i=0 \end{cases}$$

$$\frac{d^2}{dX^2} f_{X_k}(X|\mathcal{E}^S(\mathbf{t}), X_{k-1}, B^i) = \begin{cases} \dfrac{(i-1)i(i+1)}{(t_k - X_{k-1})^{i+1}}(t_k - X)^{i-2}, & i=2,3,\ldots,N-k \\ 0, & i=0,1 \end{cases}$$

Since each of the $N - k + 1$ polynomials in Equation 3.2 has a non-positive first derivative, it is non-increasing in X. Similarly, since each has a non-negative second derivative, it is convex. Hence, since the weighted sum of non-increasing convex functions is also non-increasing and convex, we have that $f_{X_k}(X|\mathcal{E}^S(\mathbf{t}), X_{k-1})$ is a non-increasing, convex, $(N-k)$-th order polynomial, for $X_{k-1} < X \leq t_k$. ∎

3.3.2 Proof of Theorem 3.2

Now we wish to determine the form for the marginal density for the k-th arrival, $f[X_k|\mathcal{E}^S(\mathbf{t})]$. This density is just the integral of the joint density, $f(X_1, X_2, \ldots, X_N)$, over all variables except X_k. From previous arguments, the joint density is constant, with value $1/V$, where V is found from:

$$V = \int_0^{t_1} dX_1 \int_{X_1}^{t_2} dX_2 \cdots \int_{X_{N-1}}^{t_N} dX_N \tag{3.3}$$

Hence, we may write:

$$f[X_k|\mathcal{E}^S(\mathbf{t})] = \frac{1}{V} \times \int_0^{\min(X_k, t_{k-1})} dX_{k-1} \int_0^{\min(X_{k-1}, t_{k-2})} dX_{k-2} \cdots \int_0^{\min(X_2, t_1)} dX_1 \times \int_{X_k}^{t_{k+1}} dX_{k+1} \int_{X_{k+1}}^{t_{k+2}} dX_{k+2} \cdots \int_{X_{N-1}}^{t_N} dX_N, \quad 0 < X_k \leq t_k$$

To investigate further, we define the following quantities:

$$\begin{aligned} H_k(X_j) &\equiv \int_0^{\min(X_j,t_{k-1})} dX_{k-1} \int_0^{\min(X_{k-1},t_{k-2})} dX_{k-2} \cdots \int_0^{\min(X_2,t_1)} dX_1, \\ &\quad k = 2,3,\ldots,N \\ H_1(X_1) &\equiv 1 \\ G_k(X_j) &\equiv \int_{X_j}^{t_{k+1}} dX_{k+1} \int_{X_{k+1}}^{t_{k+2}} dX_{k+2} \cdots \int_{X_{N-1}}^{t_N} dX_N, \\ &\quad k = 1,2,\ldots,N-1 \\ G_N(X_N) &\equiv 1 \end{aligned}$$

and we may then simplify our expression for $f[X_k|\mathcal{E}^S(\mathbf{t})]$ to:

$$f[X_k|\mathcal{E}^S(\mathbf{t})] = \begin{cases} \frac{1}{V} \times H_k(X_k) \times G_k(X_k), & 0 < X_k \leq t_k, \quad k = 1,2,\ldots,N \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

The proof of the theorem proceeds as follows. First we show that $G_k(X_k)$ is an $(N-k)$-th order polynomial in X_k, which is continuous and all of whose derivatives are continuous. Then we show that $H_k(X_k)$ is continuous for all X_k and is a $(k-i)$-th order polynomial in X_k for $t_{i-1} < X_k \leq t_i$. Finally, we show that the jth derivative of $H_k(X_k)$ is just $H_{k-j}(X_k)$ for $0 < X_k < t_{k-j}$, and hence is continuous in that range. Putting all of this information together allows us to prove the theorem easily.

We begin with the following:

Lemma 3.1 *$G_k(X_k)$ is a continuous $(N-k)$-th order polynomial in X_k on the interval $(0,t_k]$, all of whose derivatives are continuous on the interval $(0,t_k)$.*

Proof: If we can prove that $G_k(X_k)$ is an $(N-k)$-th order polynomial in X_k, then the continuities are automatically proved, since any polynomial with finite coefficients is continuous and has continuous derivatives. We proceed by backwards induction, beginning with $k = N-1$, since the $k = N$ case is obvious and depends on a separate definition.

$$k = N-1: \quad G_{N-1}(X_{N-1}) = \int_{X_{N-1}}^{t_N} dX_N = t_N - X_{N-1}, \quad 0 < X_{N-1} \leq t_{N-1}$$

This is an $N-k=N-(N-1)=$ 1st order polynomial in X_{N-1}, as claimed. Now assume the hypothesis for $k+1$:

$$G_k(X_k)=\int_{X_k}^{t_{k+1}} G_{k+1}(X_{k+1})dX_{k+1}, \quad 0<X_k\le t_k$$

By the induction hypothesis, $G_{k+1}(X_{k+1})$ is an $(N-k-1)$th order polynomial on $(0,t_{k+1}]$: hence, its integral is just an $(N-k)$th order polynomial. When evaluated at the given limits, we get the difference between a constant and an $(N-k)$th order polynomial in X_k, which is just an $(N-k)$th order polynomial in X_k. ∎

We continue with the following:

Lemma 3.2 *For $k=1,2,\ldots,N$, $H_k(X_k)$ is continuous for $0<X_k\le t_k$. Furthermore, $H_k(X_k)$ is:*

1. *a $(k-i)$-th order polynomial in X_k for $t_{i-1}<X_k\le t_i$ (where $i=1,2,\ldots,k-1$);*
2. *a zero-th order polynomial (i.e. constant) for $t_{k-1}<X_k\le t_k$.*

Proof: Again the proof proceeds by induction. The lemma is obvious for $k=1$ and depends on a separate definition, so we begin the induction with $k=2$:

$$k=2: \quad H_2(X_2)=\int_0^{\min(X_2,t_1)} dX_1=\begin{cases} X_2, & 0<X_2\le t_1 \\ t_1, & t_1<X_2\le t_2 \end{cases}$$

This function is continuous for all X_2 (notably at the point t_1) and is a first-order polynomial in X_2 for $0<X_2\le t_1$ and is zero-th order for $t_1<X_2\le t_2$. Hence we continue by assuming the hypothesis for $k-1$:

$$\begin{aligned} H_k(X_k) &= \int_0^{\min(X_k,t_{k-1})} H_{k-1}(X_{k-1})dX_{k-1} \\ &= \begin{cases} \displaystyle\int_0^{X_k} H_{k-1}(X_{k-1})dX_{k-1}, & 0<X_k\le t_{k-1} \\ \displaystyle\int_0^{t_{k-1}} H_{k-1}(X_{k-1})dX_{k-1}, & t_{k-1}<X_k\le t_k \end{cases} \end{aligned} \tag{3.5}$$

We now introduce the following notation: we let $\mathcal{P}^j(X)$ represent a jth-order polynomial in X. Then, from the induction hypothesis, we may say

$$H_k(X_k) = \int_0^{t_1} \mathcal{P}^{k-2}(X_{k-1})dX_{k-1}+\int_{t_1}^{t_2} \mathcal{P}^{k-3}(X_{k-1})dX_{k-1}+\cdots$$

$$
\begin{aligned}
&+ \int_{t_{i-1}}^{X_k} \mathcal{P}^{k-i-1}(X_{k-1}) dX_{k-1} \qquad (3.6)\\
&= \text{constants} + \mathcal{P}^{k-i}(X_k) = \mathcal{P}^{k-i}(X_k),\\
&\quad t_{i-1} < X_k \leq t_i, \quad i = 1, 2, \ldots, k-1
\end{aligned}
$$

which proves assertion 1 of the Lemma. We may also say:

$$
\begin{aligned}
H_k(X_k) &= \int_0^{t_1} \mathcal{P}^{k-2}(X_{k-1}) dX_{k-1} + \int_{t_1}^{t_2} \mathcal{P}^{k-3}(X_{k-1}) dX_{k-1} + \cdots\\
&\quad + \int_{t_{k-2}}^{t_{k-1}} \mathcal{P}^{0}(X_{k-1}) dX_{k-1} \qquad (3.7)\\
&= \text{constant},\\
&\quad t_{k-1} < X_k \leq t_k
\end{aligned}
$$

which proves assertion 2 of the Lemma. To see the continuity, consider Equation 3.5. Since the induction hypothesis assures us that $H_{k-1}(X_{k-1})$ is continuous everywhere, then its integral must also be continuous, at least for $X_k \leq t_{k-1}$ and for $X_k > t_{k-1}$. The only remaining question is the point $X_k = t_{k-1}$. But now compare Equation 3.6 for $i = k-1$ and $X_k = t_{k-1}$ to Equation 3.7. The forms are identical and hence $H_k(X_k)$ is also continuous at the point $X_k = t_{k-1}$. ∎

Finally, we prove the following:

Lemma 3.3 $\frac{d^j}{dX_k^j} H_k(X_k) = H_{k-j}(X_k)$ *for* $0 < X_k \leq t_{k-j}$. *Hence,* $\frac{d^j}{dX_k^j} H_k(X_k)$ *is continuous for* $0 < X_k < t_{k-j}$.

Proof: We prove the first part of the Lemma with induction. The second part then follows immediately from Lemma 3.2. We begin the induction with $j = 1$:

$$
\begin{aligned}
\frac{d}{dX_k} H_k(X_k) &= \frac{d}{dX_k} \int_0^{\min(X_k, t_{k-1})} dX_{k-1} \int_0^{\min(X_{k-1}, t_{k-2})} dX_{k-2} \cdots \int_0^{\min(X_2, t_1)} dX_1\\
&= \frac{d}{dX_k} \int_0^{X_k} dX_{k-1} \int_0^{\min(X_{k-1}, t_{k-2})} dX_{k-2} \cdots \int_0^{\min(X_2, t_1)} dX_1, \quad X_k \leq t_{k-1}\\
&= \int_0^{\min(X_k, t_{k-2})} dX_{k-2} \cdots \int_0^{\min(X_2, t_1)} dX_1, \quad X_k \leq t_{k-1}\\
&= H_{k-1}(X_k), \quad 0 < X_k \leq t_{k-1}
\end{aligned}
$$

Now we assume the first part of the Lemma for $j-1$:

$$\begin{aligned}
\frac{d^j}{dX_k^j}H_k(X_k) &= \frac{d}{dX_k}\left[\frac{d^{j-1}}{dX_k^{j-1}}H_k(X_k)\right] = \frac{d}{dX_k}H_{k-j+1}(X_k), \quad X_k \le t_{k-j+1} \\
&= \frac{d}{dX_k}\int_0^{\min(X_k,t_{k-j})} dX_{k-j}\int_0^{\min(X_{k-j},t_{k-j-1})} dX_{k-j-1}\cdots\int_0^{\min(X_2,t_1)} dX_1, \\
&\quad X_k \le t_{k-j+1} \\
&= \frac{d}{dX_k}\int_0^{X_k} dX_{k-j}\int_0^{\min(X_{k-j},t_{k-j-1})} dX_{k-j-1}\cdots\int_0^{\min(X_2,t_1)} dX_1, \\
&\quad X_k \le t_{k-j} \\
&= \int_0^{\min(X_k,t_{k-j-1})} dX_{k-j-1}\cdots\int_0^{\min(X_2,t_1)} dX_1, \quad X_k \le t_{k-j} \\
&= H_{k-j}(X_k), \quad 0 < X_k \le t_{k-j}
\end{aligned}$$

Since the jth derivative of $H_k(X_k)$ is just $H_{k-j}(X_k)$ (a continuous function), when $X_k \le t_{k-j}$, then the jth derivative must also be continuous, for $0 < X_k < t_{k-j}$. ∎

We now have all of the pieces necessary to prove Theorem 3.2. We use the definition of $f[X_k|\mathcal{E}^S(\mathbf{t})]$ given in Equation 3.4. First, since $H_k(X_k)$ and $G_k(X_k)$ are continuous on $(0, t_k]$ from Lemmata 3.1 and 3.2, and $f[X_k|\mathcal{E}^S(\mathbf{t})]$ is a constant times these functions when $0 < X_k \le t_k$, then $f[X_k|\mathcal{E}^S(\mathbf{t})]$ is also continuous, at least for $0 < X_k < t_k$. Second, we know that $G_k(X_k)$ is an $(N-k)$-th order polynomial everywhere (Lemma 3.1), and that $H_k(X_k)$ is a $(k-i)$-th order polynomial for $t_{i-1} < X_k \le t_i$, $i = 1, 2, \ldots, k$ (Lemma 3.2). Hence, a constant times their product yields an $(N-i)$-th order polynomial for $t_{i-1} < X_k \le t_i$, $i = 1, 2, \ldots, k$. Finally, consider the jth derivative of $f[X_k|\mathcal{E}^S(\mathbf{t})]$ (here, we abbreviate $H_k(X_k)$ and $G_k(X_k)$ by H and G respectively):

$$\begin{aligned}
\frac{d^j}{dX_k^j}f[X_k|\mathcal{E}^S(\mathbf{t})] &= \frac{1}{V}\left[G \times \frac{d^jH}{dX_k^j} + \binom{j}{1} \times \frac{dG}{dX_k} \times \frac{d^{j-1}H}{dX_k^{j-1}} + \cdots\right. \\
&\quad \left. + \binom{j}{j-1} \times \frac{d^{j-1}G}{dX_k^{j-1}} \times \frac{dH}{dX_k} + \frac{d^jG}{dX_k^j} \times H\right]
\end{aligned}$$

But we know that $G_k(X_k)$ and all of its derivatives are continuous for $0 < X_k \le t_k$, from Lemma 3.1. We also know that $H_k(X_k)$ and all of its derivatives up to the jth

are continuous for $X_k < t_{k-j}$. Hence, since the product of continuous functions is continuous, we have that $\frac{d^j}{dX_k^j} f[X_k|\mathcal{E}^S(\mathbf{t})]$ is continuous at least for $0 < X_k < t_{k-j}$.

∎

There is one final comment to make about the proof that $f[X_k|\mathcal{E}^S(\mathbf{t})]$ is a polynomial of order $N-i$ for $t_{i-1} < X_k \leq t_i$. We could also have proved this by conditioning on X_k occurring in $(t_{i-1}, t_i]$ and additionally conditioning on l, the total number of arrivals during that interval. That is, if we define B_i^l to be the event that there were a total of exactly l arrivals during $(t_{i-1}, t_i]$, then

$$\begin{aligned} f[X_k|\mathcal{E}^S(\mathbf{t})] &= \sum_{i=1}^{k} \sum_{l=k-i+1}^{N-i+1} f[X_k|B_i^l, t_{i-1} < X_k \leq t_i, \mathcal{E}^S(\mathbf{t})] \\ &\quad \times \Pr[B_i^l, t_{i-1} < X_k \leq t_i|\mathcal{E}^S(\mathbf{t})] \end{aligned}$$

But, given that $t_{i-1} < X_k \leq t_i$ and that there were exactly l arrivals during that interval, then the conditional density for X_k will be the weighted sum of densities of the form given in Equation 3.1, with $M = l$, $t = t_{i-1}$, and $a = t_i - t_{i-1}$, where the weighted sum will be due to the fact that X_k could have been positioned amongst the l arrivals in different ways: i.e., j in Equation 3.1 could take on several different values. However, no matter what the value of j, the result is an $(M-1)$- or, with $M = l$, an $(l-1)$-th order polynomial. Hence, since the maximum value for l is $N-i+1$, then $f[X_k|t_{i-1} < X_k \leq t_i, \mathcal{E}^S(\mathbf{t})]$, which is the only contributor to $f[X_k|\mathcal{E}^S(\mathbf{t})]$ for $t_{i-1} < X_k \leq t_i$ will be an $(N-i)$-th order polynomial, as already shown above.

3.4 Determining the Marginal Density Functions

Theorem 3.2 seems to give us a lot of information about the marginal density functions for the ordered arrivals, conditioned on the arrival-time inequalities. The natural question to ask is whether there is an "easy" technique for determining the exact form of these densities. The answer to that question is yes for the density of X_N: this technique is described in Section 3.4.1. For the other densities, we have found an

$O(N^4)$ algorithm which provides the exact form of all of the other density functions. This algorithm takes a more direct approach to the problem by using integration recursively to find the coefficients for the polynomial arrival time densities. This second algorithm is described in Section 3.4.2.

3.4.1 Determining the Marginal Density Function for X_N Using Continuity Equations

The density function for the last arrival in the congestion period is, as described by Theorem 3.2, a continuous function for $0 < X_N < t_N$, and is a polynomial of order $N - i$ for $t_{i-1} < X_N \leq t_i$. We also know that $\frac{d^j}{dX_N^j} f[X_N|\mathcal{E}^S(\mathbf{t})]$ is continuous for $0 < X_N < t_{N-j}$. Finally, for $0 < X_N \leq t_1$, $f[X_N|\mathcal{E}^S(\mathbf{t})]$ is given by the product $f[X_N|X_N \leq t_1, \mathcal{E}^S(\mathbf{t})] \times \Pr[X_N \leq t_1|\mathcal{E}^S(\mathbf{t})]$. But $f[X_N|X_N \leq t_1, \mathcal{E}^S(\mathbf{t})]$ is just the density function for the maximum of N Poisson arrivals over the interval $(0, t_1]$ and so is given by Equation 3.1 with $j = N, M = N, t = 0$, and $a = t_1$. Hence, $f[X_N|\mathcal{E}^S(\mathbf{t})]$ is just a constant multiplying X_N^{N-1}.

Therefore, we may write $f[X_N|\mathcal{E}^S(\mathbf{t})]$ as follows:

$$f[X_N|\mathcal{E}^S(\mathbf{t})] = \begin{cases} a_{1,(N-1)}X_N^{N-1}, & 0 < X_N \leq t_1 \\ a_{20} + a_{21}X_N + a_{22}X_N^2 + \cdots + a_{2,(N-2)}X_N^{N-2}, & t_1 < X_N \leq t_2 \\ a_{30} + a_{31}X_N + a_{32}X_N^2 + \cdots + a_{3,(N-3)}X_N^{N-3}, & t_2 < X_N \leq t_3 \\ \vdots & \\ a_{(N-1),0} + a_{(N-1),1}X_N, & t_{N-2} < X_N \leq t_{N-1} \\ a_{N0}, & t_{N-1} < X_N \leq t_N \end{cases}$$

So we have a total of $1 + (N-1) + (N-2) + \cdots + 2 + 1 = 1 + \frac{N(N-1)}{2}$ coefficients to determine. We now make the following definition:

$$f^i[X_N|\mathcal{E}^S(\mathbf{t})] \equiv f[X_N|\mathcal{E}^S(\mathbf{t})], \quad t_{i-1} < X_N \leq t_i, \quad i = 1, 2, \ldots, N$$

Our continuity equations tell us the following:

$$\frac{d^j}{dX_N^j} f^{i-1}[X_N|\mathcal{E}^S(\mathbf{t})]\Big]_{X_N=t_{i-1}} = \frac{d^j}{dX_N^j} f^{i}[X_N|\mathcal{E}^S(\mathbf{t})]\Big]_{X_N=t_{i-1}}, \quad \begin{aligned} j &= 0,1,\ldots,N-i, \\ i &= 2,3,\ldots,N \end{aligned}$$

so that we have $\sum_{i=2}^{N} N-i+1 = \frac{N(N-1)}{2}$ equations with which to determine the a_{ij}'s. Therefore, with one additional equation, we would be able, theoretically, to solve for all of these coefficients. But, assuming we have calculated V as defined in Equation 3.3, then we may find $a_{1,(N-1)}$ by equating the following two expressions for $\Pr[X_N \le t_1|\mathcal{E}^S(\mathbf{t})]$:

$$\begin{aligned}
\Pr[X_N \le t_1|\mathcal{E}^S(\mathbf{t})] &= \int_0^{t_1} f^1[X_N|\mathcal{E}^S(\mathbf{t})]\, dX_N \\
&= \int_0^{t_1} a_{1,(N-1)} X_N^{N-1}\, dX_N \\
&= a_{1,(N-1)} \frac{t_1^N}{N} \\
\Pr[X_N \le t_1|\mathcal{E}^S(\mathbf{t})] &= \frac{1}{V}\int_0^{t_1} dX_N \int_0^{X_N} dX_{N-1} \int_0^{X_{N-1}} dX_{N-2} \cdots \int_0^{X_2} dX_1 \\
&= \frac{t_1^N}{N!V} \\
\Longrightarrow \quad a_{1,(N-1)} &= \frac{1}{V(N-1)!}
\end{aligned}$$

Now that we have a sufficient number of equations to determine the values of the coefficients, we must find an efficient method for actually obtaining their values. By expanding our continuity equation above, the method becomes obvious:

$$\begin{aligned}
\frac{d^j}{dX_N^j} f^{i-1}[X_N|\mathcal{E}^S(\mathbf{t})] &= \frac{j!}{0!}a_{(i-1),j} + \frac{(j+1)!}{1!}a_{(i-1),(j+1)}X_N + \frac{(j+2)!}{2!}a_{(i-1),(j+2)}X_N^2 \\
&\quad + \cdots + \frac{(N-i+1)!}{(N-i+1-j)!}a_{(i-1),(N-i+1)}X_N^{N-i+1-j} \\
\frac{d^j}{dX_N^j} f^{i}[X_N|\mathcal{E}^S(\mathbf{t})] &= j!a_{ij} + \frac{(j+1)!}{1!}a_{i,(j+1)}X_N + \frac{(j+2)!}{2!}a_{i,(j+2)}X_N^2 + \cdots \\
&\quad + \frac{(N-i)!}{(N-i-j)!}a_{i,(N-i)}X_N^{N-i-j}
\end{aligned}$$

We can equate the two expressions above at the value $X_N = t_{i-1}$ and rearrange them so that we have an expression for a_{ij} as follows:

$$
\begin{aligned}
a_{1,j} &= \begin{cases} 0, & j = 0, 1, \ldots, N-2 \\ \dfrac{1}{V(N-1)!}, & j = N-1 \end{cases} \\
a_{ij} &= \binom{j}{j} a_{(i-1),j} + \binom{j+1}{j} a_{(i-1),(j+1)} t_{i-1} + \binom{j+2}{j} a_{(i-1),(j+2)} t_{i-1}^2 \\
&\quad + \cdots + \binom{N-i+1}{j} a_{(i-1),(N-i+1)} t_{i-1}^{N-i+1-j} \\
&\quad - \binom{j+1}{j} a_{i,(j+1)} t_{i-1} - \binom{j+2}{j} a_{i,(j+2)} t_{i-1}^2 \\
&\quad - \cdots - \binom{N-i}{j} a_{i,(N-i)} t_{i-1}^{N-i-j} \\
&= \sum_{k=0}^{N-i+1-j} \binom{k+j}{j} a_{(i-1),(k+j)} t_{i-1}^k - \sum_{k=1}^{N-i-j} \binom{k+j}{j} a_{i,(k+j)} t_{i-1}^k, \\
&\quad j = 0, 1, \ldots, N-i, \quad i = 2, 3, \ldots, N
\end{aligned}
$$

Now we can find a_{ij} strictly in terms of other coefficients a_{kl}, with $k < i$, or with $k = i$ and $l > j$. In terms of the expression for $f[X_N|\mathcal{E}^S(\mathbf{t})]$ as defined on page 55, we start at the second equation (the value of $a_{1,(N-1)}$ already having been determined) and first determine the value for $a_{2,(N-2)}$, then we work from right to left (i.e., we decrease the value of j by one at each step), to find the value of a_{2j} in terms of values already found. We then proceed to the third equation and work from right to left and so on.

Since there are $O(N^2)$ coefficients to find, and finding a_{ij} requires calculating $2(N-i-j+1)$ terms, the overall algorithm to determine $f[X_N|\mathcal{E}^S(\mathbf{t})]$ is $O(N^3)$. This seems high, but we are obtaining much more information than we get from simply calculating the beta-matrices. If this method generalized and we were able to find the density function for each of the X's in $O(N^3)$, then we would have an $O(N^4)$ algorithm for determining all possible information about the arrival stream at

every point in time. Unfortunately, for earlier arrivals than X_N, we have many fewer continuity equations, and even though we also have fewer coefficients, the number of equations decreases much more quickly, so that we would have to be inventing multiple additional equations to solve for all of the coefficients. In the next section, we suggest a more straightforward method for calculating the marginal density functions for the ordered arrivals, which also surprisingly results in an $O(N^4)$ algorithm.

3.4.2 Determining the Marginal Density Functions for the X's Using Integration

We now pursue a technique, suggested in the proof of Theorem 3.2 for determining all of the marginal density functions. Specifically, in Equation 3.4, we provide a definition for $f[X_k|\mathcal{E}^S(\mathbf{t})]$ in terms of the quantities V, $H_k(X_k)$, and $G_k(X_k)$. We know that $G_k(X_k)$ is simply an $(N-k)$th order polynomial and so can be defined in terms of $N-k+1$ coefficients of powers of X_k. We also know that $H_k(X_k)$ is a $(k-i)$th order polynomial for $t_{i-1} < X_k \leq t_i$, $i = 1, 2, \ldots, k$. Hence, it can be defined in terms of $1+2+\cdots+(k-1)+k = \dfrac{k(k+1)}{2}$ coefficients of powers of X_k. We now provide an algorithm to find the coefficients for these functions (and, incidentally, to find V), which we may then multiply together to find $f[X_k|\mathcal{E}^S(\mathbf{t})]$.

First, we work with $G_k(X_k)$, which we may expand as follows (here we let g_j^k represent the coefficient of the j-th power of X_k in $G_k(X_k)$, where $0 \leq j \leq N-k$):

$$\begin{aligned} G_k(X_k) &= g_0^k + g_1^k X_k + g_2^k X_k^2 + \cdots + g_{N-k}^k X_k^{N-k}, \\ &\qquad k = 1, 2, \ldots, N-1 \\ G_N(X_N) &= g_0^N = 1 \end{aligned}$$

But note that, because of the definition of $G_k(X_k)$, we also have the following:

$$\begin{aligned} G_k(X_k) &= \int_{X_k}^{t_{k+1}} dX_{k+1} \int_{X_{k+1}}^{t_{k+2}} dX_{k+2} \cdots \int_{X_{N-1}}^{t_N} dX_N \\ &= \int_{X_k}^{t_{k+1}} G_{k+1}(X_{k+1})\, dX_{k+1} \end{aligned}$$

$$
\begin{aligned}
&= \int_{X_k}^{t_{k+1}} \left[g_0^{k+1} + g_1^{k+1} X_{k+1} + g_2^{k+1} X_{k+1}^2 + \cdots + g_{N-k-1}^{k+1} X_{k+1}^{N-k-1} \right] dX_{k+1} \\
&= g_0^{k+1} X + \frac{g_1^{k+1}}{2} X^2 + \frac{g_2^{k+1}}{3} X^3 + \cdots + g_{N-k-1}^{k+1} \frac{X^{N-k}}{N-k} \Bigg]_{X=X_k}^{X=t_{k+1}} \\
&= g_0^{k+1} t_{k+1} + \frac{g_1^{k+1}}{2} t_{k+1}^2 + \frac{g_2^{k+1}}{3} t_{k+1}^3 + \cdots + \frac{g_{N-k-1}^{k+1}}{N-k} t_{k+1}^{N-k} \\
&\quad - g_0^{k+1} X_k - \frac{g_1^{k+1}}{2} X_k^2 - \frac{g_2^{k+1}}{3} X_k^3 - \cdots - \frac{g_{N-k-1}^{k+1}}{N-k} X_k^{N-k}
\end{aligned}
$$

This then gives us a very simple method for determining all of the values for g_j^k in terms of the values of g_m^{k+1}, $m = 0, 1, \ldots, N-k-1$. Specifically, as can be seen from the above, we have:

$$
\begin{aligned}
g_j^k &= -\frac{g_{j-1}^{k+1}}{j}, \quad j = 1, 2, \ldots, N-k, \quad k = 1, 2, \ldots, N-1 \\
g_0^k &= \sum_{m=1}^{N-k} \frac{g_{m-1}^{k+1}}{m} t_{k+1}^m = -\sum_{m=1}^{N-k} g_m^k t_{k+1}^m, \quad k = 1, 2, \ldots, N-1
\end{aligned}
$$

Before we begin discussion of the H functions, note that V may be easily found from the following:

$$
\begin{aligned}
V &= \int_0^{t_1} dX_1 \int_{X_1}^{t_2} dX_2 \cdots \int_{X_{N-1}}^{t_N} dX_N \\
&= \int_0^{t_1} G_1(X_1)\, dX_1 \\
&= \sum_{m=1}^{N} \frac{g_{m-1}^1}{m} t_1^m
\end{aligned}
$$

That is, once $G_1(X_1)$ has been found, V is easily found as a weighted sum of its coefficients.

Finding each of the g_j^k's for $i \neq 0$ requires a single division, and so finding all of the g_j^k's for $j \neq 0$ requires $O(N^2)$ calculations. Finding the value of g_0^k requires that $N-k$ terms be found, and so finding all of the g_0^k's is also an $O(N^2)$ operation. Therefore, determining $G_k(X_k)$ for $k = 1, 2, \ldots, N$, and determining V requires $O(N^2)$ calculations.

We now proceed in a similar manner to determine the exact form for $H_k(X_k)$. Because $H_k(X_k)$ is a different polynomial, depending on the value of X_k, we make

the following definitions (here we let h_j^{ik} represent the coefficient of the j-th power of X_k in $H_k^i(X_k)$, where $0 \le j \le k-i$):

$$\begin{aligned} H_k^i(X_k) &\equiv \begin{cases} H_k(X_k), & t_{i-1} < X_k \le t_i, \quad i = 1,2,\ldots,k \\ 0 & i > k \end{cases} \\ H_1^1(X_1) &= h_0^{11} = 1 \\ H_k^i(X_k) &= h_0^{ik} + h_1^{ik}X_k + h_2^{ik}X_k^2 + \cdots + h_{k-i}^{ik}X_k^{k-i}, \quad i = 1,2,\ldots,k, \quad k = 2,3,\ldots,N \end{aligned}$$

We base what follows on the above definitions and on Equation 3.5:

$$\begin{aligned} H_k^i(X_k) &= \int_0^{t_1} H_{k-1}^1(X_{k-1})dX_{k-1} + \int_{t_1}^{t_2} H_{k-1}^2(X_{k-1})dX_{k-1} + \cdots \\ &\quad + \int_{t_{i-2}}^{t_{i-1}} H_{k-1}^{i-1}(X_{k-1})dX_{k-1} + \int_{t_{i-1}}^{X_k} H_{k-1}^i(X_{k-1})dX_{k-1} \\ &= H_k^{i-1}(X)\Big]_{X=t_{i-1}} + \int_{t_{i-1}}^{X_k} H_{k-1}^i(X_{k-1})dX_{k-1} \end{aligned}$$

Of course, when $i = k$, the last integral above has an integrand which is equal to 0 and hence does not contribute to $H_k^i(X_k)$.

To determine the specific values of the h_j^{ik}'s, we expand the last expression above:

$$\begin{aligned} H_k^i(X_k) &= h_0^{(i-1),k} + h_1^{(i-1),k}t_{i-1} + h_2^{(i-1),k}t_{i-1}^2 + \cdots + h_{k-i+1}^{(i-1),k}t_{i-1}^{k-i+1} \\ &\quad + h_0^{i,(k-1)}X + \frac{h_1^{i,(k-1)}}{2}X^2 + \frac{h_2^{i,(k-1)}}{3}X^3 + \cdots + \frac{h_{k-1-i}^{i,(k-1)}}{k-i}X^{k-i}\Bigg]_{X=t_{i-1}}^{X=X_k} \\ &= h_0^{(i-1),k} + \left[h_1^{(i-1),k} - h_0^{i,(k-1)}\right]t_{i-1} + \left[h_2^{(i-1),k} - \frac{h_1^{i,(k-1)}}{2}\right]t_{i-1}^2 + \cdots \\ &\quad + \left[h_{k-i}^{(i-1),k} - \frac{h_{k-1-i}^{i,(k-1)}}{k-i}\right]t_{i-1}^{k-i} + h_{k-i+1}^{(i-1),k}t_{i-1}^{k-i+1} \\ &\quad + h_0^{i,(k-1)}X_k + \frac{h_1^{i,(k-1)}}{2}X_k^2 + \frac{h_2^{i,(k-1)}}{3}X_k^3 + \cdots + \frac{h_{k-1-i}^{i,(k-1)}}{k-i}X_k^{k-i} \end{aligned}$$

Now we may find the specific values for h_j^{ik} simply by picking off the appropriate coefficients in the expression above:

$$h_j^{ik} = \frac{h_{j-1}^{i,(k-1)}}{j}, \quad j = 1,2,\ldots,k-i$$

$$\begin{aligned} h_j^{ik} &\equiv 0, \quad j > k - i \\ h_0^{ik} &= h_0^{(i-1),k} + \sum_{m=1}^{k-i+1} \left[h_m^{(i-1),k} - h_m^{ik} \right] t_{i-1}^m \end{aligned}$$

We begin by knowing the value of h_0^{11}. Then, as long as we calculate the h_j^{ik}'s in non-decreasing order of both i and k, and as long as we calculate h_0^{ik} after all of the other values of h_j^{ik} have been calculated, then we will always have all of the values we need on the right-hand sides of the expressions above.

Finding each of the h_j^{ik}'s for $j \neq 0$ requires a single division, and so finding all of the h_j^{ik}'s for $j = 1, 2, \ldots, k-i$ and $i = 1, 2, \ldots, k$ and $k = 2, 3, \ldots, N$ requires $O(N^3)$ calculations. Finding the value of h_0^{ik} requires that $k - i + 1$ terms be found, and so finding all of the h_0^{ik}'s is also an $O(N^3)$ operation. Therefore, determining $H_k^i(X_k)$ for $i = 1, 2, \ldots, k$ and $k = 1, 2, \ldots, N$, requires $O(N^3)$ calculations.

Since we have found all of the necessary components to determine $f[X_k|\mathcal{E}^S(\mathbf{t})]$ for $k = 1, 2, \ldots, N$ in $O(N^3)$ calculations, it seems surprising that the overall algorithm should turn out to be $O(N^4)$. Unfortunately, we still must do the appropriate multiplications in order to find each of the $f[X_k|\mathcal{E}^S(\mathbf{t})]$'s explicitly, and this multiplication turns out to be an $O(N^3)$ operation for each function. To see this, consider the following definition:

$$f^i[X_k|\mathcal{E}^S(\mathbf{t})] \equiv f[X_k|\mathcal{E}^S(\mathbf{t})], \quad t_{i-1} < X_k \leq t_i, \quad i = 1, 2, \ldots, k, \quad k = 1, 2, \ldots, N$$

But we may find $f^i[X_k|\mathcal{E}^S(\mathbf{t})]$ by again using Equation 3.4, as follows:

$$f^i[X_k|\mathcal{E}^S(\mathbf{t})] = \frac{1}{V} \times H_k^i(X_k) \times G_k(X_k), \quad i = 1, 2, \ldots, k, \quad k = 1, 2, \ldots, N$$

We know that $H_k^i(X_k)$ has $k-i+1$ terms, and each of these is multiplied by each term of $G_k(X_k)$, of which there are $N - k + 1$. Hence, just to find $f^i[X_k|\mathcal{E}^S(\mathbf{t})]$ requires approximately $(k - i + 1)(N - k + 1)$ operations: thus, to determine the number of calculations required to find $f[X_k|\mathcal{E}^S(\mathbf{t})]$, we must add up this number for $i = 1$ to k: we find that calculation of $f[X_k|\mathcal{E}^S(\mathbf{t})]$ is an $O(Nk^2)$ operation. Finally, to determine all of the $f[X_k|\mathcal{E}^S(\mathbf{t})]$'s, for $k = 1, 2, \ldots, N$, is an $O(N^4)$ operation. It should not

be surprising that it costs more computationally to find the exact density functions for the ordered arrivals than it does to find the beta-matrix, since the beta-matrix only gives information at the specific values of the t_i's, while the density functions give information about all of the arrivals at every instant in the congestion period. Figures 3.4 and 3.5 give the arrival time densities for X_1 through X_{10}, with the t_i's generated randomly on the unit time interval.

3.5 The Density Function for the Unordered Arrivals

We now consider the density function for the unordered times of customer arrivals. Without any conditioning, there are a fixed number (N) of Poisson arrivals over the fixed time interval $(0, t_N]$, so that $f(U_k)$, $k = 1, 2, \ldots, N$ has constant value $1/t_N$, since these arrivals are distributed uniformly over the interval. When the arrival-time inequalities are added as conditions, the density function changes. We claim the following:

Theorem 3.3 *$f[U_k|\mathcal{E}^S(\mathbf{t})]$, $k = 1, 2, \ldots, N$ is constant in the interval $(t_{i-1}, t_i]$, $i = 1, 2, \ldots, N$, with the value of the constant non-increasing as i increases.*

Proof: We may say

$$f[U_k|\mathcal{E}^S(\mathbf{t})] = \sum_{i=1}^{N} f[U_k|t_{i-1} < U_k \leq t_i, \mathcal{E}^S(\mathbf{t})] \times \Pr[t_{i-1} < U_k \leq t_i|\mathcal{E}^S(\mathbf{t})]$$

Each of the conditional densities in the sum above contributes to $f[U_k|\mathcal{E}^S(\mathbf{t})]$ only in the interval $(t_{i-1}, t_i]$. Hence, proving that these conditional densities are uniform proves the first part of the theorem. But we may also say:

$$\begin{aligned} f[U_k|t_{i-1} < U_k \leq t_i, \mathcal{E}^S(\mathbf{t})] \;=\; & \sum_{j=1}^{N-i+1} f[U_k|N(t_{i-1}, t_i) = j, t_{i-1} < U_k \leq t_i, \mathcal{E}^S(\mathbf{t})] \\ & \times \Pr[N(t_{i-1}, t_i) = j|t_{i-1} < U_k \leq t_i, \mathcal{E}^S(\mathbf{t})] \end{aligned}$$

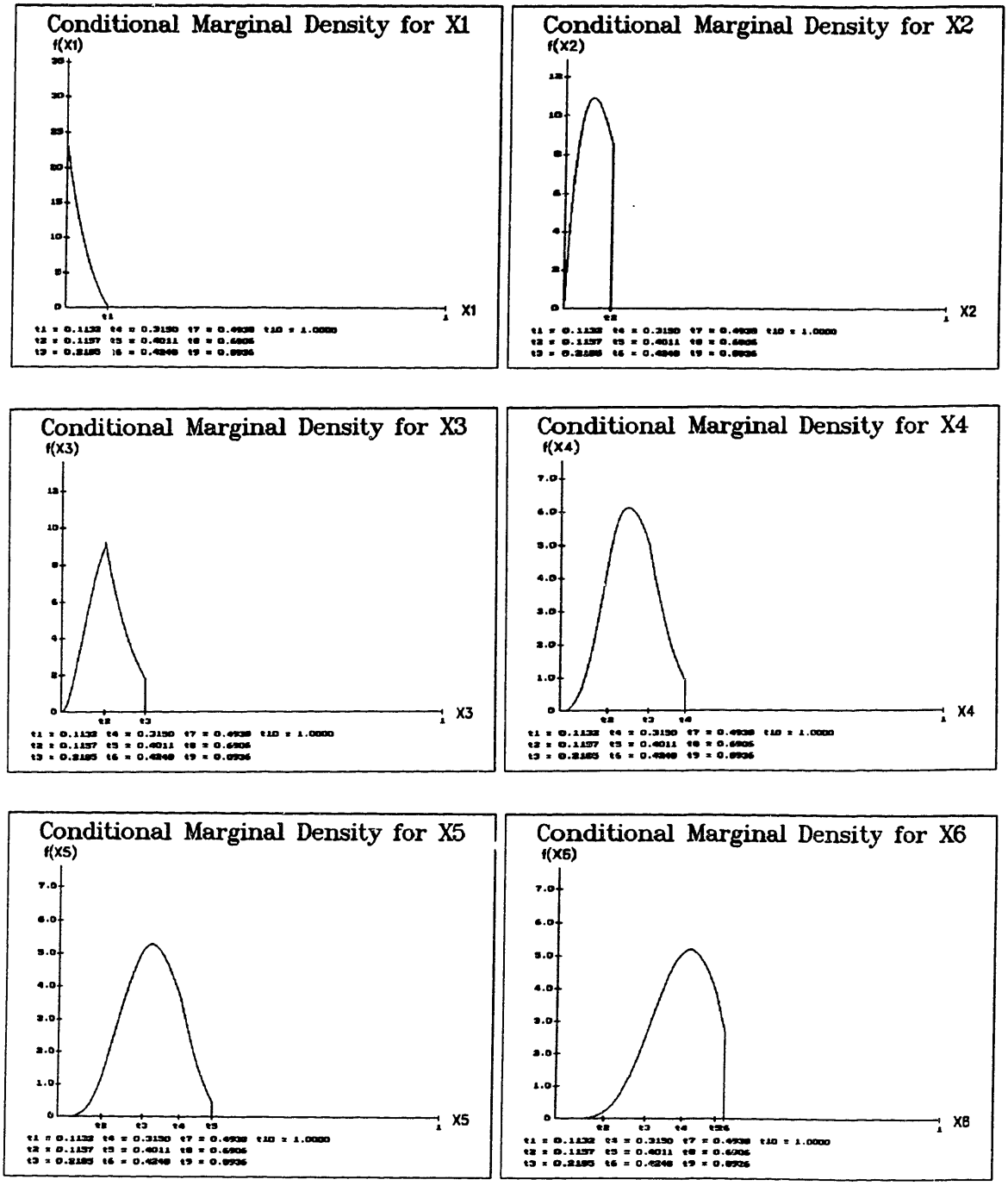


Figure 3.4: Densities for X_1 through X_6, Conditioned on $\mathcal{E}^S(t)$

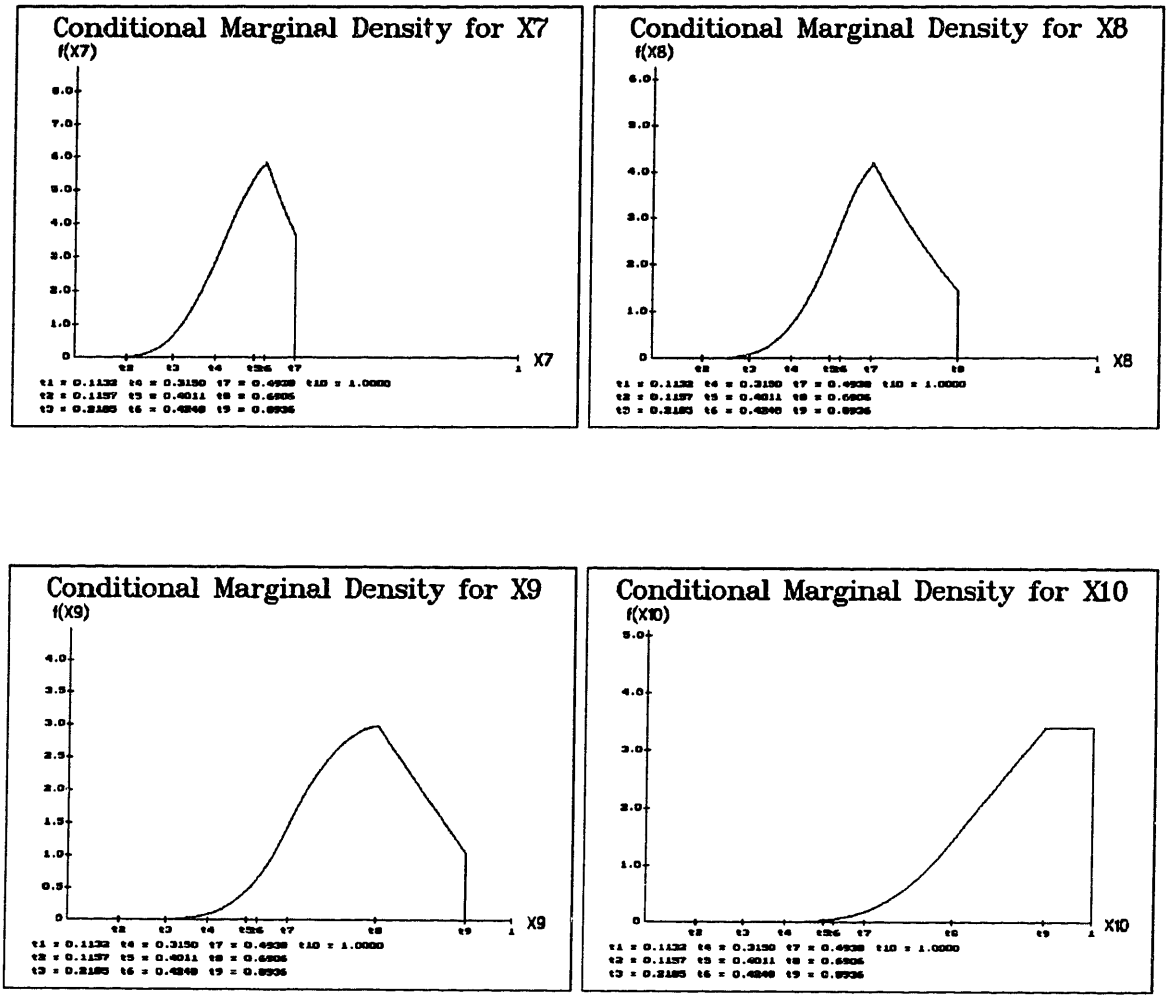


Figure 3.5: Densities for X_7 through X_{10}, Conditioned on $\mathcal{E}^S(\mathrm{t})$

But given that there are exactly j Poisson arrivals, one of which is U_k, over the interval $(t_{i-1}, t_i]$, we know that the density for U_k over that interval is uniform. Since a weighted sum of uniform densities is also uniform, we have established that $f[U_k|\mathcal{E}^S(\mathbf{t})]$ is uniform in the interval $(t_{i-1}, t_i]$, $i = 1, 2, \ldots, N$.

To prove that the value of the constant is non-increasing, we use the concavity of the function $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$. Specifically, say that $f[U_k|\mathcal{E}^S(\mathbf{t})] = u_i$, $t_{i-1} < U_k \leq t_i$. Then, we wish to show that $u_i \geq u_{i+1}$. Consider the expected number of arrivals to the system during the interval $(t_{i-1}, t_i]$. We may represent this as:

$$\mathrm{E}[A(t_{i-1}, t_i)|\mathcal{E}^S(\mathbf{t})] = \mathrm{E}[A(t_i)|\mathcal{E}^S(\mathbf{t})] - \mathrm{E}[A(t_{i-1})|\mathcal{E}^S(\mathbf{t})] \tag{3.8}$$

Because the densities of all of the U_k are symmetric, we may also say:

$$\begin{aligned} \mathrm{E}[A(t_{i-1}, t_i)|\mathcal{E}^S(\mathbf{t})] &= \sum_{k=1}^{N} \mathrm{E}[A(t_{i-1}, t_i)|\mathcal{E}^S(\mathbf{t})] \text{ due to } U_k \\ &= N \times \Pr(t_{i-1} < U_k \leq t_i) \\ &= N \times u_i \times (t_i - t_{i-1}) \end{aligned} \tag{3.9}$$

The concavity of the function $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$ tells us that the slope must be decreasing, i.e. that

$$\frac{\mathrm{E}[A(t_i)|\mathcal{E}^S(\mathbf{t})] - \mathrm{E}[A(t_{i-1})|\mathcal{E}^S(\mathbf{t})]}{t_i - t_{i-1}} \geq \frac{\mathrm{E}[A(t_{i+1})|\mathcal{E}^S(\mathbf{t})] - \mathrm{E}[A(t_i)|\mathcal{E}^S(\mathbf{t})]}{t_{i+1} - t_i}, \quad i = 1, 2, \ldots, N-1 \tag{3.10}$$

Now we substitute Equations 3.8 and 3.9 into 3.10, and we immediately have the following:

$$\frac{N \times u_i \times (t_i - t_{i-1})}{t_i - t_{i-1}} \geq \frac{N \times u_{i+1} \times (t_{i+1} - t_i)}{t_{i+1} - t_i} \implies u_i \geq u_{i+1}$$

Since the above holds for all $i = 1, 2, \ldots, N-1$, the theorem is proved, and the density function for the unordered arrivals, conditioned on the arrival-time inequalities, is just a descending staircase, with the steps located at the t_i's. ∎

Note that the first part of the theorem (the constancy of $f[U_k|\mathcal{E}^S(\mathbf{t})]$ over $(t_{i-1}, t_i]$) may also be proved directly from the linearity of the function $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$ in the interval $(t_{i-1}, t_i]$. We have:

$$\begin{aligned}\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})] &= \sum_{k=1}^{N} \mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})] \text{ due to } U_k \\ &= N \times \Pr[U \leq t|\mathcal{E}^S(\mathbf{t})] \\ &= N \times F_U[t|\mathcal{E}^S(\mathbf{t})]\end{aligned}$$

where $F_U[t|\mathcal{E}^S(\mathbf{t})]$ is just the distribution function for U. Hence, since the left-hand side of the above is linear, so is the right-hand side, i.e. $F_U[t|\mathcal{E}^S(\mathbf{t})]$ is linear, and its derivative, $f_U[t|\mathcal{E}^S(\mathbf{t})]$, is constant.

Another interesting observation may be made from the following (again we use the definition $F_U[t|\mathcal{E}^S(\mathbf{t})] \equiv \Pr[U \leq t|\mathcal{E}^S(\mathbf{t})]$):

$$\begin{aligned}F_U[t|\mathcal{E}^S(\mathbf{t})] &= \sum_{k=1}^{N} \Pr[U \leq t|\text{rand cust is } k\text{-th}, \mathcal{E}^S(\mathbf{t})] \\ &\quad \times \Pr[\text{rand cust is } k\text{-th}|\mathcal{E}^S(\mathbf{t})] \\ &= \sum_{k=1}^{N} F_{X_k}[t|\mathcal{E}^S(\mathbf{t})] \times \frac{1}{N} \\ \Longrightarrow \quad f_U[t|\mathcal{E}^S(\mathbf{t})] &= \frac{1}{N} \sum_{k=1}^{N} f_{X_k}[t|\mathcal{E}^S(\mathbf{t})]\end{aligned}$$

This says that the sum of all of the polynomials, $f[X_k|\mathcal{E}^S(\mathbf{t})]$, in the interval $(t_{i-1}, t_i]$, is a constant, a somewhat counterintuitive result! Figure 3.6 gives the density for U, for the same set of t_i's as those used in Figures 3.4 and 3.5: this density was actually generated by just adding up the polynomials $f[X_1|\mathcal{E}^S(\mathbf{t})]$ through $f[X_{10}|\mathcal{E}^S(\mathbf{t})]$, and then dividing the result by 10, as suggested above.

3.6 The Density Functions for the Waiting Times

Once we have obtained the densities for the arrival times, and assuming a first-come, first-served (FCFS) queue, we can easily find the density functions for the queue

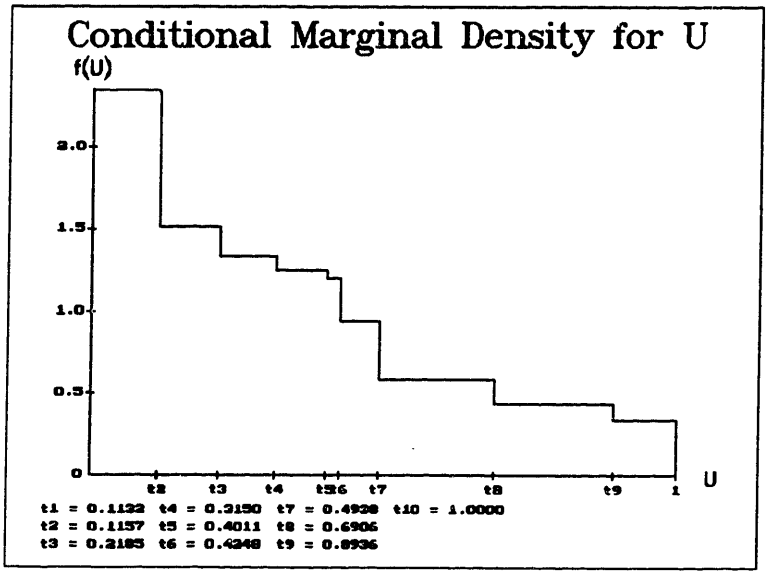


Figure 3.6: Density for U, Conditioned on $\mathcal{E}^S(\mathbf{t})$

waiting times, both of specific ordered customers, and of a random customer. Since we know the density for when customer k arrives, and, under the FCFS assumption, this customer begins service at time t_k, then the density for customer k's waiting time at time t, $f_{W_k}[t|\mathcal{E}^S(\mathbf{t})]$, is just $f_{X_k}[t_k - t|\mathcal{E}^S(\mathbf{t})]$: i.e., the probability that customer k waits t time units is just the probability that that customer arrived at time $t_k - t$. See Figures 3.7 and 3.8 for the densities of the waiting times for customers 1 through 10, in a FCFS queue with the same t_i's as those used in Figures 3.4 and 3.5.

Finally, note that the density for the waiting time of a random customer in a FCFS queue, $f_W[t|\mathcal{E}^S(\mathbf{t})]$, may be found from the following simple observation, similar to that made in finding $f_U[t|\mathcal{E}^S(\mathbf{t})]$:

$$\begin{aligned} f_W[t|\mathcal{E}^S(\mathbf{t})] &= \sum_{k=1}^{N} f_W[t|\text{rand cust is } k\text{-th}, \mathcal{E}^S(\mathbf{t})] \\ &\quad \times \Pr[\text{rand cust is } k\text{-th}|\mathcal{E}^S(\mathbf{t})] \\ &= \sum_{k=1}^{N} f_{W_k}[t|\mathcal{E}^S(\mathbf{t})] \times \frac{1}{N} \end{aligned}$$

Figure 3.9 depicts the density for the waiting time of a random customer in a FCFS queue, for the same set of t_i's as those used for Figures 3.7 and 3.8.

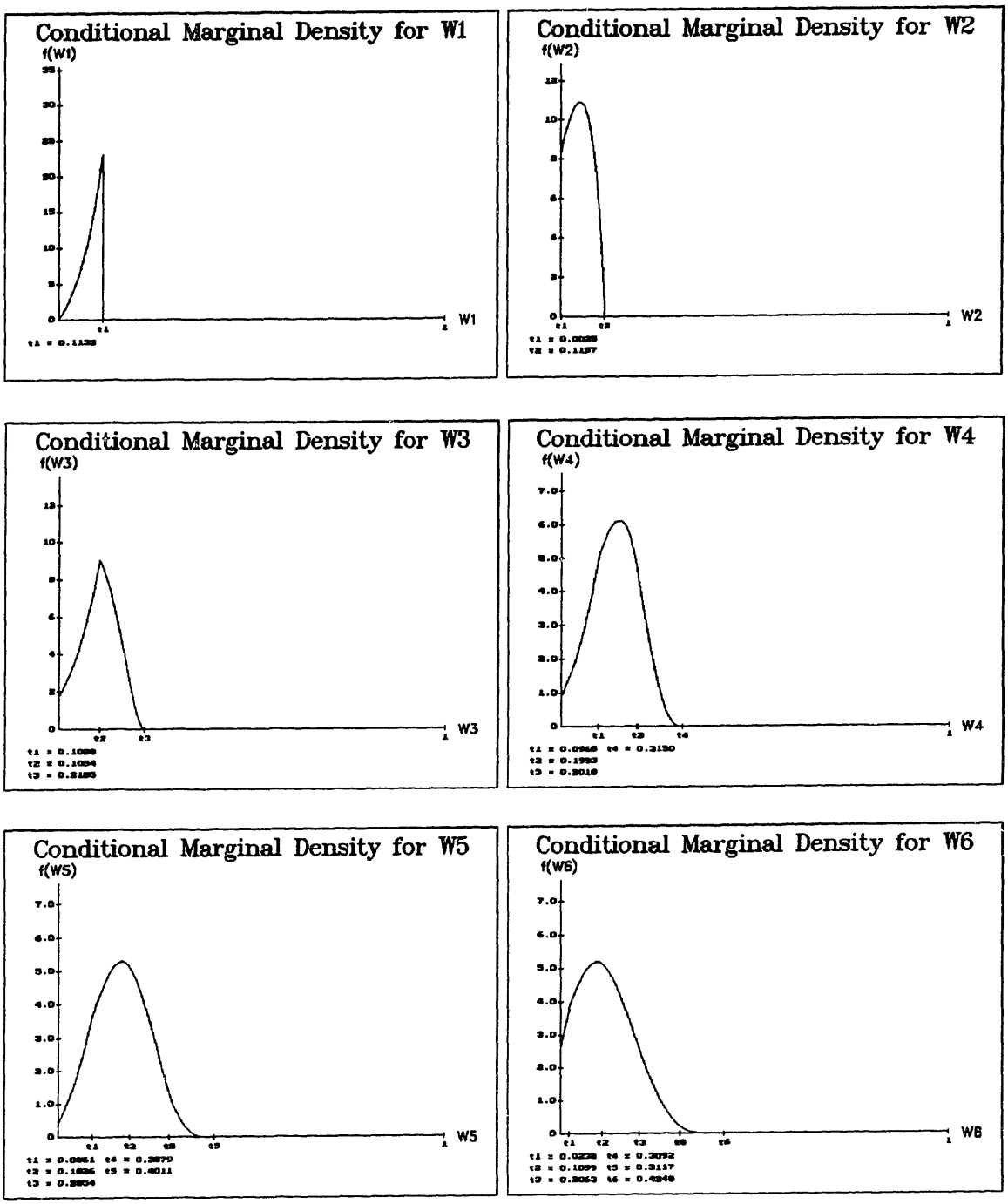


Figure 3.7: Densities for W_1 through W_6, Conditioned on $\mathcal{E}^S(\mathbf{t})$

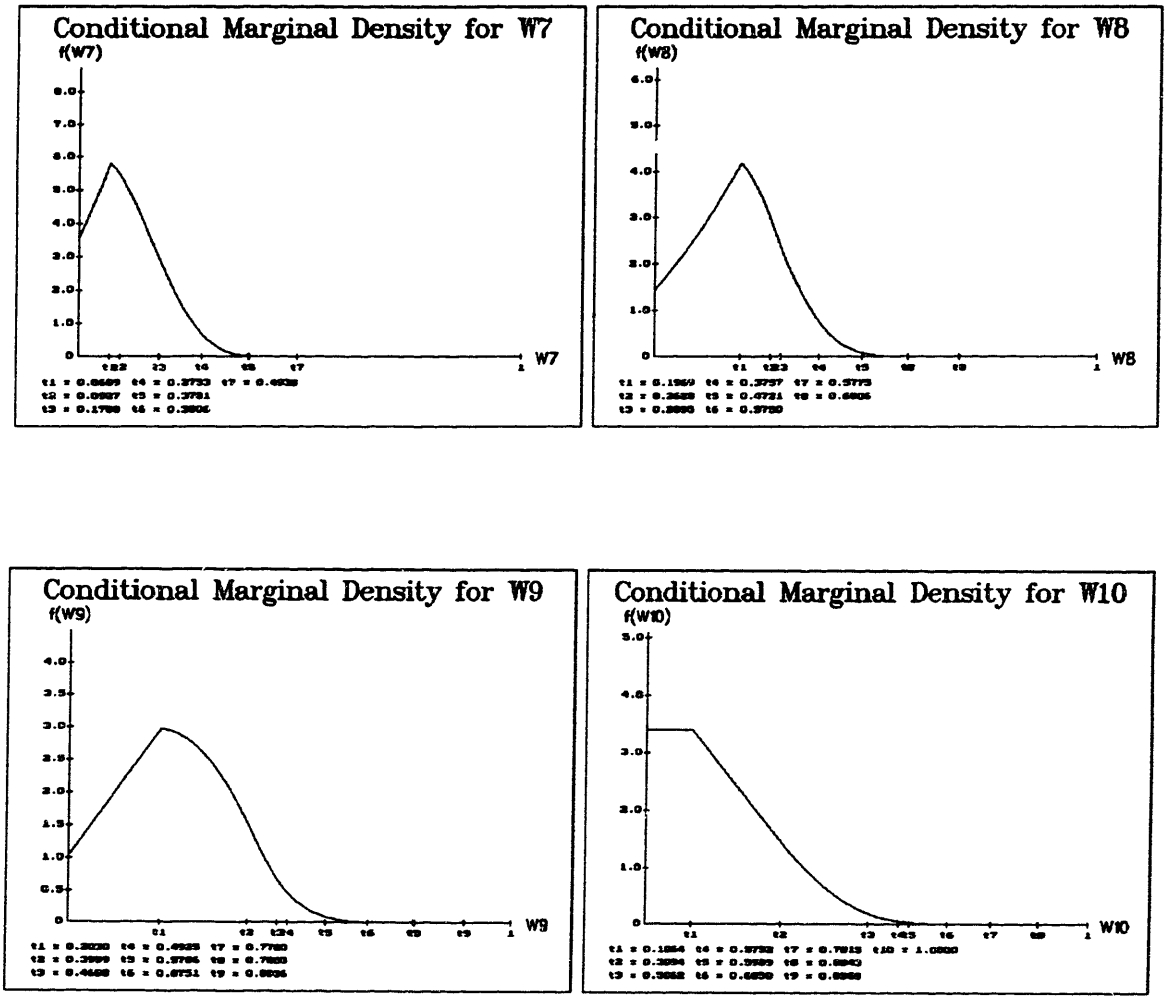


Figure 3.8: Densities for W_7 through W_{10}, Conditioned on $\mathcal{E}^S(\mathbf{t})$

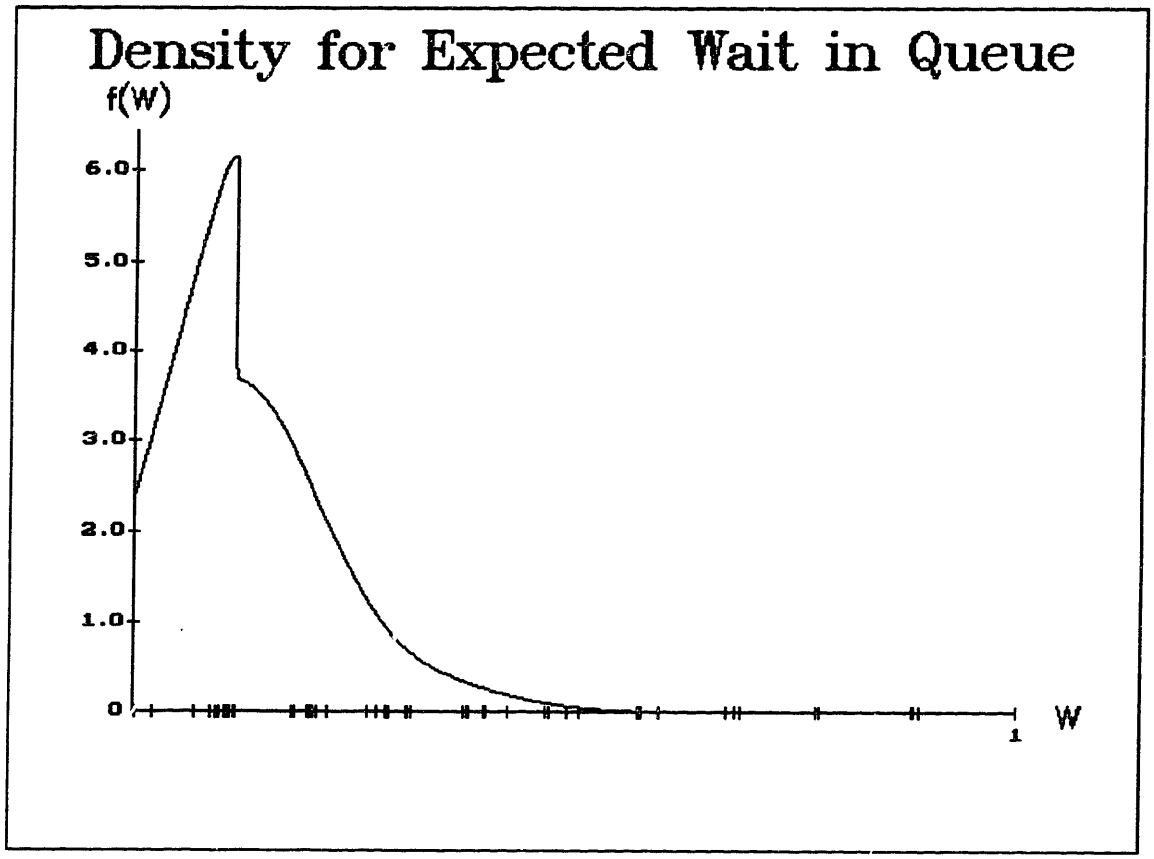


Figure 3.9: Density for W, Conditioned on $\mathcal{E}^S(\mathbf{t})$

3.7 Applications for the Arrival and Wait Time Densities

In this section, we present two applications in which the full densities for the wait times or the arrival times might actually be utilized to improve customer service. The first example is from the banking industry, and the second example is from the airline industry.

The banking application comes from ATM's. Since most ATM installations work on a first-come-first-served (FCFS) basis, it is possible to generate the probability density function (PDF) for the wait in queue for each of the customers during a congestion period. By customer-tagging these PDF's, it is also possible to update these for each customer, each time the customer uses an ATM at that bank. Therefore, theoretically, at least, a customer's monthly statement could include information such as the probability that that customer had to wait more than 5 minutes in queue during the whole month; or the times that the customer visited the ATM and had the shortest expected wait (to encourage the customer to avoid particularly congested times); or the entire PDF for the customer's wait time during the month. This ability to provide customer-specific information could be very valuable in improving an individual customer's service.

The airline application is a little more involved. At least one major airline determines its daily ticket-counter staffing levels by the following procedure:

- For one month during every year, airline agents interview every passenger arriving at the ticket-counter queue and record the time of each passenger's arrival and their flight number.

- Based on these interviews, a cumulative distribution function (CDF) for each flight is created, which represents the customer arrival time pattern for that flight.

- For the next year, at the beginning of each day, for all flights, the flight load factor is multiplied by the corresponding CDF to determine when passengers for each flight will arrive.

- The above are combined for all flights to determine staffing levels at the ticket-counter for all times during the day.

Some of the problems with this type of procedure are as follows:

1. Only one month of data is available, i.e., seasonal changes are not reflected in the CDF.

2. The data are only updated once a year, so that systemic changes in passenger behavior are not recognized until the next time that the polling is done.

3. There is only a single CDF created for each flight, even though the loading of the flight might be reflected in very different customer arrival patterns, i.e., no load-dependent CDF's are available.

4. All of the above sources of inaccuracies could result in over- or under-staffing of the ticket counters. In addition, for one month a year, a substantial number of manhours is consumed by the interview procedure.

Now consider how the ability to generate the arrival-time densities for each customer could simplify this process. As each customer arrives at the ticket counter (enters service), the point-of-sale terminals provide the flight number for that passenger and the number of passengers in his/her party, as well as the time at which the transaction begins. (Here, we consider each transaction to be the set of passengers who are travelling together: the Poisson arrival assumption applies to these clumps of people, rather than to individual passengers.) Note that the loading of the flight for that day could also be included as part of the transactional data. So, the following new procedure is proposed to replace the current procedure:

- For each congestion period, find $f[X_k|\mathcal{E}^S(\mathbf{t})]$ for all k. Tag each of these PDF's with its flight number, the number of passengers the transaction represents, and the flight loading.

- For each flight, create a PDF for passenger arrival times by combining the above PDF's for that flight (and multiplying by number of passengers where necessary). Note that it would be very simple to create several load-dependent PDF's.

- Proceed as before.

This technique eliminates all of the problems listed above! First, the data could be gathered continuously, so that seasonal changes would be detected. Second, the PDF's could be updated in a Bayesian manner as often as desired, so that real changes in passenger arrival trends would be detected immediately. Third, since flight-loading could be included as part of the transactional data, it is very easy to create load-dependent PDF's. Finally, the interview manhours would be eliminated, and the improved accuracy in the data would reduce both under- and over-staffing problems.

Finding the densities for the wait and arrival times of passengers can provide valuable information, but it is a time-consuming process. In some applications, when large congestion periods must be analyzed, or when real-time analysis is desired, even the original QIE algorithm is too slow. In the next chapter, we consider several preliminary approaches to approximating the queue statistics that are found by the QIE algorithm, which provide bounds and approximations to $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$ in much less time than the QIE algorithm.

Chapter 4

Concavity Lower Bound, Uniform Upper Bound, and Trapezoidal Approximation to the QIE

As described in Chapter 2, the computation time required for the QIE to calculate the queue statistics of interest for a given congestion period is proportional to N^3, where N is the number of queued customers in the congestion period. For very large congestion periods, this computational burden may be excessive, particularly if the QIE is being run in a desktop or realtime environment and immediate results are desired. In these cases, it may be useful to have bounds and approximations to apply to the QIE, in order to get an idea of the maximum, minimum, and approximate expected queue lengths and queue waits for customers in such a congestion period.

In this chapter, we present three algorithms which provide, respectively, a lower bound, an upper bound, and an approximation (which is provably not a bound) to the function generated by the QIE, the expected cumulative number of arrivals at any given time during the congestion period, $\mathrm{E}[A(t)|\mathcal{E}^S(\mathrm{t})]$. Although none of these algorithms generates a full β-matrix, it is still possible to determine most of the statistics that the original QIE algorithm generates. Of course, $\mathrm{E}[Q(t)|\mathcal{E}^S(\mathrm{t})]$ comes

directly from $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$, and the time-averaged queue length and expected wait in queue can be found directly by integrating $\mathrm{E}[Q(t)|\mathcal{E}^S(\mathbf{t})]$. We can also find the expected qucue length as encountered by a random arrival, as follows.

$$\begin{aligned}
\mathrm{E}[\ell_Q|\mathcal{E}^S(\mathbf{t})] &= \frac{1}{N}\sum_{i=1}^{N}\mathrm{E}[\text{number in queue seen by arrival } i \\
&= \frac{1}{N}\sum_{i=1}^{N}\mathrm{E}[\text{number in queue seen by departure } i \\
&= \frac{1}{N}\sum_{i=1}^{N}\mathrm{E}[Q(t_i)|\mathcal{E}^S(\mathbf{t})]
\end{aligned}$$

Indeed, if the expressions for $\mathrm{E}[\ell_Q|\mathcal{E}^S(\mathbf{t})]$ and $\Pi[k|\mathcal{E}^S(\mathbf{t})]$ at the end of Chapter 2 are expanded and rearranged, we find that we do indeed get the above result. Unfortunately, we cannot generate the values for $\Pi[k|\mathcal{E}^S(\mathbf{t})]$ without the β-matrix, but for most applications, the other statistics are sufficient.

The lower bound is based on the observation that $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$ is a piecewise-linear, concave function. The upper bound is found by approximating the densities of the arrivals to be uniform over a restricted range. The approximation algorithm uses some of the results of the last chapter and approximates the densities of the arrivals to be trapezoidal over a restricted range. Each of these three algorithms is explored in turn in the first three sections of this chapter. Finally, in the last section, we present computational results to demonstrate how these algorithms perform on a series of simulation runs. We also present computational results for combinations of the algorithms, some of which provide good approximations to the exact QIE.

4.1 Concavity Lower Bound

As shown in [Lars 90], $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$ is a piecewise-linear, concave function, with endpoints $\mathrm{E}[A(t_0)|\mathcal{E}^S(\mathbf{t})] = 0$ and $\mathrm{E}[A(t_N)|\mathcal{E}^S(\mathbf{t})] = N$. We make use of the concavity and the fact that, conditional on $\mathcal{E}^S(\mathbf{t})$, we know $A(t_i) \geq i$, to create our lower bound. We define $A^{LB}(t)$ to be the value of our function, which we claim to be a lower bound

for $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$, for $0 \leq t \leq t_N$; and we generate $A^{LB}(t)$ via the following algorithm:

1. For $i = 0, 1, \ldots, N$, set $A^{LB}(t_i) = i$.
2. Set $i = 0$.
3. For $j = i+1, i+2, \ldots, N$, calculate the slope, m_j, of the line connecting $A^{LB}(t_i)$ to $A^{LB}(t_j)$.
4. Let j^* be the index which maximizes these slopes: i.e., $m_{j^*} = \max_{j=i+1,i+2,\ldots,N} m_j$.
5. For $t_i < t \leq t_{j^*}$, set $A^{LB}(t) = i + m_{j^*}(t - t_i)$, i.e., construct a straight line from $A^{LB}(t_i)$ to $A^{LB}(t_{j^*})$.
6. If $j^* = N$ then we are done; otherwise, set $i = j^*$ and go to step 3.

Claim 4.1 *$A^{LB}(t)$ is a concave, piecewise-linear function, which is a lower bound to $E[A(t)|\mathcal{E}^S(\mathbf{t})]$.*

Proof: The piecewise-linearity is by construction. The concavity is also by construction: if i_n and j_n^* are the values of i and j^* on the nth iteration of step 4 of the algorithm, then we know that $m^n_{j_n^*}$, the slope of the line connecting the points (t_{i_n}, i_n) and $(t_{j_n^*}, j_n^*)$, is greater than the slope of the line connecting (t_{i_n}, i_n) and $(t_{j^*_{n+1}}, j^*_{n+1})$, which implies that $m^n_{j_n^*}$ is greater than $m^{n+1}_{j^*_{n+1}}$, the slope of the line connecting $(t_{j_n^*}, j_n^*) = (t_{i_{n+1}}, i_{n+1})$ and $(t_{j^*_{n+1}}, j^*_{n+1})$, and hence the function is concave. Finally, we know that

$$\begin{aligned} A^{LB}(t_{i_n}) &= i_n \leq \mathrm{E}[A(t_{i_n})|\mathcal{E}^S(\mathbf{t})] \\ A^{LB}(t_{j_n^*}) &= j_n^* \leq \mathrm{E}[A(t_{j_n^*})|\mathcal{E}^S(\mathbf{t})] \end{aligned}$$

Since $A^{LB}(t)$ is linear between t_{i_n} and $t_{j_n^*}$, the concave function $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$ must lie above $A^{LB}(t)$ between these points, and we are done. ∎

Technically, the algorithm for determining $A^{LB}(t)$ is $\mathrm{O}(N^2)$, since in the worst case we might have to repeat steps 3 through 6 of the algorithm N times, and at

iteration n, we must calculate $N - n + 1$ slopes: i.e., in the worst case, we must calculate approximately $\frac{N(N+1)}{2}$ slopes. However, in practice, the algorithm does much better (see Section 4.4).

4.2 Uniform Upper Bound

We now discuss an upper bound to $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$. We expand our notation for $\mathcal{E}^S(\mathbf{t})$ to make explicit the size of the congestion period:

$$\mathcal{E}^S_N(\mathbf{t}) \equiv \mathcal{E}^S(t_0, t_1, t_2, \ldots, t_N)$$

As discussed earlier, our assumption of Poisson arrivals means that, over the fixed time interval $(0, t_N]$, the arrivals are described by N independent uniform random variables, $U_1, U_2, \ldots, U_N$. However, the time conditioning given by $\mathcal{E}^S_N(\mathbf{t})$ places restrictions on these random variables, as follows:

$$\begin{aligned} \min(U_1, U_2, \ldots, U_N) &\leq t_1 \\ \text{2nd smallest}(U_1, U_2, \ldots, U_N) &\leq t_2 \\ &\vdots \\ \max(U_1, U_2, \ldots, U_N) &\leq t_N \end{aligned}$$

These conditions are what cause the analysis to become difficult, in part because, given these conditions, the U_i's are no longer independent. Now consider the following quantity:

$$\mathrm{E}[A(t)|\mathcal{U}_N(\mathbf{t})] \equiv \mathrm{E}[A(t)|U_1 \leq t_1, U_2 \leq t_2, \ldots, U_N \leq t_N, \mathcal{E}^{0,N}(\mathbf{t})]$$

Note that with this new set of conditions, $\mathcal{U}_N(\mathbf{t})$, we guarantee at least i arrivals by t_i, which is the necessary condition for the busy period to continue. Also note that this set of conditions maintains the independence of the U_i's.

We may easily find $\mathrm{E}[A(t)|\mathcal{U}_N(\mathbf{t})]$ as follows:

$$\mathrm{E}[A(t)|\mathcal{U}_N(\mathbf{t})] = \sum_{i=1}^{N} \mathrm{E}[A(t) \text{ due to } U_i|\mathcal{U}_N(\mathbf{t})]$$

$$= \sum_{i=1}^{N} \mathrm{E}[A(t) \text{ due to } U_i | U_i \leq t_i]$$

where the last equation follows because of the independence of the U_i's. But we may easily find the quantity in the summation as follows:

$$A(t) \text{ due to } U_i = \begin{cases} 1, & U_i \leq t \\ 0, & U_i > t \end{cases}$$

Hence, the expectation of the above is found as:

$$\begin{aligned} \mathrm{E}[A(t) \text{ due to } U_i | U_i \leq t_i] &= \Pr(U_i \leq t | U_i \leq t_i) \\ &= \begin{cases} \frac{t}{t_i}, & 0 \leq t \leq t_i \\ 1, & t_i < t \leq t_N \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

Therefore, we have that the value of $\mathrm{E}[A(t)|\mathcal{U}_N(\mathbf{t})]$ is given by:

$$\mathrm{E}[A(t)|\mathcal{U}_N(\mathbf{t})] = \sum_{i=1}^{N} \Pr(U_i \leq t | U_i \leq t_i)$$

and for values of t in $(t_{j-1}, t_j]$ this reduces to:

$$\begin{aligned} \mathrm{E}[A(t)|\mathcal{U}_N(\mathbf{t})] &= \sum_{i=1}^{j-1} 1 + \sum_{i=j}^{N} \frac{t}{t_i} \\ &= j - 1 + \sum_{i=j}^{N} \frac{t}{t_i}, \quad t_{j-1} < t \leq t_j, \quad j = 1, 2, \ldots, N \end{aligned} \tag{4.1}$$

Note that this function is concave and piecewise linear, with breakpoints at the t_i's, just like $\mathrm{E}[A(t)|\mathcal{E}_N^S(\mathbf{t})]$: in Equation 4.1, the summation term is the sum of $N - j + 1$ linear terms and hence is linear itself for $t_{j-1} < t \leq t_j$. As j increases, there are fewer of these positive-slope terms in the summation: hence the slope of the sum decreases, and the function is concave. Also note that when $N = 1$, this function is exactly equal to $\mathrm{E}[A(t)|\mathcal{E}_N^S(\mathbf{t})]$, which starts at the value 0 when $t = 0$, and rises linearly to the value 1 when $t = t_1$.

We now proceed to prove that $\mathrm{E}[A(t)|\mathcal{U}_N(\mathbf{t})]$ is an upper bound to $\mathrm{E}[A(t)|\mathcal{E}_N^S(\mathbf{t})]$, via the following theorem:

Theorem 4.1 $E[A(t)|\mathcal{U}_N(\mathbf{t})] \geq E[A(t)|\mathcal{E}_N^S(\mathbf{t})], \quad 0 \leq t \leq t_N$

Proof: First, because both functions are piecewise linear with the same breakpoints, we only consider the breakpoints. Second, note that

$$\begin{aligned} \mathrm{E}[A(0)|\mathcal{U}_N(\mathbf{t})] &= \mathrm{E}[A(0)|\mathcal{E}_N^S(\mathbf{t})] = 0 \\ \mathrm{E}[A(t_N)|\mathcal{U}_N(\mathbf{t})] &= \mathrm{E}[A(t_N)|\mathcal{E}_N^S(\mathbf{t})] = N \end{aligned}$$

Hence, we need only prove $\mathrm{E}[A(t)|\mathcal{U}_N(\mathbf{t})] \geq \mathrm{E}[A(t)|\mathcal{E}_N^S(\mathbf{t})]$ for $t = t_1, t_2, \ldots, t_{N-1}$. Note that, at the point $t = t_i$, $i = 1, 2, \ldots, N-1$, Equation 4.1 reduces to the following:

$$\mathrm{E}[A(t_i)|\mathcal{U}_N(\mathbf{t})] = i + \sum_{k=i+1}^{N} \frac{t_i}{t_k} \tag{4.2}$$

The proof is by induction. First, we prove it for $N = 2$; then we proceed to the $N = 3$ case, as a demonstration of the induction method. Finally, we generalize for any value of N.

$\boldsymbol{N = 2}$**:** For $N = 2$, we need only prove the theorem at the single value $t = t_1$. We have from Equation 4.2:

$$\mathrm{E}[A(t_1)|\mathcal{U}_2(\mathbf{t})] = 1 + \frac{t_1}{t_2}$$

We may then use standard QIE methods to find that:

$$\mathrm{E}[A(t_1)|\mathcal{E}_2^S(\mathbf{t})] = \frac{2t_2}{2t_2 - t_1}$$

By simple algebraic manipulation, it follows that:

$$1 + \frac{t_1}{t_2} \geq \frac{2t_2}{2t_2 - t_1} \iff t_2 \geq t_1$$

But this is true by definition! Hence, for $N = 2$, $\mathrm{E}[A(t_i)|\mathcal{U}_2(\mathbf{t})] \geq \mathrm{E}[A(t_i)|\mathcal{E}_2^S(\mathbf{t})]$, $i = 0, 1, 2$; and therefore, by the piecewise linearity of both functions, we have that

$$\mathrm{E}[A(t)|\mathcal{U}_2(\mathbf{t})] \geq \mathrm{E}[A(t)|\mathcal{E}_2^S(\mathbf{t})], \quad 0 \leq t \leq t_2$$

$N = 3$: Now we define a set of three events, $\mathcal{B}_3^1(\mathbf{t}), \mathcal{B}_3^2(\mathbf{t})$, and $\mathcal{B}_3^3(\mathbf{t})$, which we claim are three mutually exclusive, collectively exhaustive subsets of $\mathcal{E}_3^S(\mathbf{t})$.

$$\begin{aligned}
\mathcal{B}_3^1(\mathbf{t}) &\equiv \{\min(U_1, U_2) \le t_2\} \bigcap \{\max(U_1, U_2) \le t_3\} \bigcap \{0 < U_3 \le t_1\} \bigcap \mathcal{E}^{0,3}(\mathbf{t}) \\
\mathcal{B}_3^2(\mathbf{t}) &\equiv \{\min(U_1, U_2) \le t_1\} \bigcap \{\max(U_1, U_2) \le t_3\} \bigcap \{t_1 < U_3 \le t_2\} \bigcap \mathcal{E}^{0,3}(\mathbf{t}) \\
\mathcal{B}_3^3(\mathbf{t}) &\equiv \{\min(U_1, U_2) \le t_1\} \bigcap \{\max(U_1, U_2) \le t_2\} \bigcap \{t_2 < U_3 \le t_3\} \bigcap \mathcal{E}^{0,3}(\mathbf{t})
\end{aligned}$$

Clearly, these are mutually exclusive, since U_3 is defined over a different interval for each event. Also, these are certainly subsets of $\mathcal{E}_3^S(\mathbf{t})$, since, in each case,

$$\begin{aligned}
\min(U_1, U_2, U_3) &\le t_1 \\
\text{2nd smallest}(U_1, U_2, U_3) &\le t_2 \\
\max(U_1, U_2, U_3) &\le t_3
\end{aligned} \tag{4.3}$$

is satisfied. Finally, they are collectively exhaustive, because $\mathcal{E}_3^S(\mathbf{t})$ is exactly satisfied, without slack, by each of these events. This can be easily seen by contradiction: say I have a point (U_1, U_2, U_3) in $\mathcal{E}_3^S(\mathbf{t})$ which I claim is not an element of any of the events above, and say that $t_{j-1} < U_3 \le t_j$ for this point $(j = 1, 2, 3)$. So if the point is in any event, it will be in event $\mathcal{B}_3^j(\mathbf{t})$. If I claim that this point violates the first constraint of $\mathcal{B}_3^j(\mathbf{t})$, then the point is not in $\mathcal{E}_3^S(\mathbf{t})$ because it violates the first $(j = 2, 3)$ or second $(j = 1)$ constraint of set 4.3 above. If I claim that the point violates the second constraint of $\mathcal{B}_3^j(\mathbf{t})$, then the point is not in $\mathcal{E}_3^S(\mathbf{t})$ because it violates the second $(j = 3)$ or third $(j = 1, 2)$ constraint of set 4.3 above. Hence, if the point is not in one of the $\mathcal{B}_3^j(\mathbf{t})$'s, then it is not in $\mathcal{E}_3^S(\mathbf{t})$, or, conversely, if the point is in $\mathcal{E}_3^S(\mathbf{t})$, then it must be in one of the $\mathcal{B}_3^j(\mathbf{t})$'s, so the $\mathcal{B}_3^j(\mathbf{t})$'s are mutually exclusive, collectively exhaustive subsets of $\mathcal{E}_3^S(\mathbf{t})$.

Now we show that

$$\mathrm{E}[A(t_i)|\mathcal{B}_3^j(\mathbf{t})] \le \mathrm{E}[A(t_i)|\mathcal{U}_3(\mathbf{t})], \quad j = 1, 2, 3; \quad i = 1, 2$$

Since $\mathcal{B}_3^1(\mathbf{t}), \mathcal{B}_3^2(\mathbf{t})$, and $\mathcal{B}_3^3(\mathbf{t})$ are mutually exclusive, collectively exhaustive subsets

of $\mathcal{E}_3^S(\mathbf{t})$, this is equivalent to showing that

$$\mathrm{E}[A(t_i)|\mathcal{E}_3^S(\mathbf{t})] \leq \mathrm{E}[A(t_i)|\mathcal{U}_3(\mathbf{t})], \quad i = 1, 2$$

First, we show in detail that $\mathrm{E}[A(t_1)|\mathcal{B}_3^1(\mathbf{t})] \leq \mathrm{E}[A(t_1)|\mathcal{U}_3(\mathbf{t})]$.

***i* =1, *j* =1:**

$$\begin{aligned}
\mathrm{E}[A(t_1)|\mathcal{B}_3^1(\mathbf{t})] &= \mathrm{E}[A(t_1) \text{ due to } U_1, U_2|\mathcal{B}_3^1(\mathbf{t})] + \mathrm{E}[A(t_1) \text{ due to } U_3|\mathcal{B}_3^1(\mathbf{t})] \\
&= \mathrm{E}[A(t_1) \text{ due to } U_1, U_2|\mathcal{E}_2^S(t_2, t_3)] + \mathrm{E}[A(t_1) \text{ due to } U_3|U_3 \leq t_1] \\
&= \mathrm{E}[A(t_1)|\mathcal{E}_2^S(t_2, t_3)] + 1
\end{aligned}$$

The second equation above follows because, conditioned on $\mathcal{B}_3^1(\mathbf{t})$, U_1 and U_2 are independent of U_3 (although U_1 and U_2 are themselves conditionally dependent). The second term of the last equation above follows because we know $U_3 \leq t_1$. The first term of the last equation above follows because, given $\mathcal{B}_3^1(\mathbf{t})$, and considering only U_1 and U_2, it is as if we have a congestion period with $N = 2$ and with departures at t_2 and t_3; and we are interested in the expected number of arrivals by the intermediate time, t_1. From our $N = 2$ proof, we know:

$$\begin{aligned}
\mathrm{E}[A(t_1)|\mathcal{E}_2^S(t_2, t_3)] + 1 &\leq \mathrm{E}[A(t_1)|\mathcal{U}_2(t_2, t_3)] + 1 \\
&= \frac{t_1}{t_2} + \frac{t_1}{t_3} + 1 \\
&= \mathrm{E}[A(t_1)|\mathcal{U}_3(t_1, t_2, t_3)] \\
\Longrightarrow \qquad \mathrm{E}[A(t_1)|\mathcal{B}_3^1(\mathbf{t})] &\leq \mathrm{E}[A(t_1)|\mathcal{U}_3(\mathbf{t})]
\end{aligned}$$

We briefly enumerate the other five cases, which follow by exactly the same logic:

***i* =1, *j* =2:**

$$\begin{aligned}
\mathrm{E}[A(t_1)|\mathcal{B}_3^2(\mathbf{t})] &= \mathrm{E}[A(t_1) \text{ due to } U_1, U_2|\mathcal{B}_3^2(\mathbf{t})] + \mathrm{E}[A(t_1) \text{ due to } U_3|\mathcal{B}_3^2(\mathbf{t})] \\
&= \mathrm{E}[A(t_1)|\mathcal{E}_2^S(t_1, t_3)] + 0 \\
&\leq \mathrm{E}[A(t_1)|\mathcal{U}_2(t_1, t_3)] \\
&= 1 + \frac{t_1}{t_3} \quad < \quad 1 + \frac{t_1}{t_2} + \frac{t_1}{t_3} \\
&= \mathrm{E}[A(t_1)|\mathcal{U}_3(\mathbf{t})]
\end{aligned}$$

$i = 1, j = 3$:

$$\begin{aligned}
\mathrm{E}[A(t_1)|\mathcal{B}_3^3(\mathbf{t})] &= \mathrm{E}[A(t_1) \text{ due to } U_1, U_2|\mathcal{B}_3^3(\mathbf{t})] + \mathrm{E}[A(t_1) \text{ due to } U_3|\mathcal{B}_3^3(\mathbf{t})] \\
&= \mathrm{E}[A(t_1)|\mathcal{E}_2^S(t_1, t_2)] + 0 \\
&\leq \mathrm{E}[A(t_1)|\mathcal{U}_2(t_1, t_2)] \\
&= 1 + \frac{t_1}{t_2} \quad < \quad 1 + \frac{t_1}{t_2} + \frac{t_1}{t_3} \\
&= \mathrm{E}[A(t_1)|\mathcal{U}_3(\mathbf{t})]
\end{aligned}$$

$i = 2, j = 1$:

$$\begin{aligned}
\mathrm{E}[A(t_2)|\mathcal{B}_3^1(\mathbf{t})] &= \mathrm{E}[A(t_2) \text{ due to } U_1, U_2|\mathcal{B}_3^1(\mathbf{t})] + \mathrm{E}[A(t_2) \text{ due to } U_3|\mathcal{B}_3^1(\mathbf{t})] \\
&= \mathrm{E}[A(t_2)|\mathcal{E}_2^S(t_2, t_3)] + 1 \\
&\leq \mathrm{E}[A(t_2)|\mathcal{U}_2(t_2, t_3)] + 1 \\
&= 1 + \frac{t_2}{t_3} + 1 \\
&= \mathrm{E}[A(t_2)|\mathcal{U}_3(\mathbf{t})]
\end{aligned}$$

$i = 2, j = 2$:

$$\begin{aligned}
\mathrm{E}[A(t_2)|\mathcal{B}_3^2(\mathbf{t})] &= \mathrm{E}[A(t_2) \text{ due to } U_1, U_2|\mathcal{B}_3^2(\mathbf{t})] + \mathrm{E}[A(t_2) \text{ due to } U_3|\mathcal{B}_3^2(\mathbf{t})] \\
&= \mathrm{E}[A(t_2)|\mathcal{E}_2^S(t_1, t_3)] + 1 \\
&\leq \mathrm{E}[A(t_2)|\mathcal{U}_2(t_1, t_3)] + 1 \\
&= 1 + \frac{t_2}{t_3} + 1 \\
&= \mathrm{E}[A(t_2)|\mathcal{U}_3(\mathbf{t})]
\end{aligned}$$

$i = 2, j = 3$:

$$\begin{aligned}
\mathrm{E}[A(t_2)|\mathcal{B}_3^3(\mathbf{t})] &= \mathrm{E}[A(t_2) \text{ due to } U_1, U_2|\mathcal{B}_3^3(\mathbf{t})] + \mathrm{E}[A(t_2) \text{ due to } U_3|\mathcal{B}_3^3(\mathbf{t})] \\
&= \mathrm{E}[A(t_2)|\mathcal{E}_2^S(t_1, t_2)] + 0 \\
&\leq \mathrm{E}[A(t_2)|\mathcal{U}_2(t_1, t_2)] \\
&= 2 \quad < \quad 2 + \frac{t_2}{t_3} \\
&= \mathrm{E}[A(t_2)|\mathcal{U}_3(\mathbf{t})]
\end{aligned}$$

Hence, we have shown that

$$\mathrm{E}[A(t_i)|\mathcal{B}_3^j(\mathbf{t})] \leq \mathrm{E}[A(t_i)|\mathcal{U}_3(\mathbf{t})], \quad i = 1,2; \;\; j = 1,2,3$$

Therefore, we may say:

$$\begin{aligned}
\mathrm{E}[A(t_i)|\mathcal{E}_3^S(\mathbf{t})] &= \sum_{j=1}^{3} \mathrm{E}[A(t_i)|\mathcal{B}_3^j(\mathbf{t})] \times \Pr[\mathcal{B}_3^j(\mathbf{t})|\mathcal{E}_3^S(\mathbf{t})] \\
&\leq \sum_{j=1}^{3} \mathrm{E}[A(t_i)|\mathcal{U}_3(\mathbf{t})] \times \Pr[\mathcal{B}_3^j(\mathbf{t})|\mathcal{E}_3^S(\mathbf{t})] \\
&= \mathrm{E}[A(t_i)|\mathcal{U}_3(\mathbf{t})], \quad i = 1,2
\end{aligned}$$

Again, because of the linearity of both functions between the breakpoints, this is equivalent to having proved, for $N = 3$:

$$\mathrm{E}[A(t)|\mathcal{E}_3^S(\mathbf{t})] \leq \mathrm{E}[A(t)|\mathcal{U}_3(\mathbf{t})], \quad 0 \leq t \leq t_3$$

General N : We proceed in a manner parallel to the $N = 3$ case. First, we define N events $\mathcal{B}_N^1(\mathbf{t}), \mathcal{B}_N^2(\mathbf{t}), \ldots, \mathcal{B}_N^N(\mathbf{t})$, where:

$$\mathcal{B}_N^j(\mathbf{t}) \equiv \left\{ \begin{array}{rcl}
\{\min(U_1, U_2, \ldots, U_{N-1}) & \leq & t_1\} \\
\bigcap \{\text{2nd smallest}(U_1, U_2, \ldots, U_{N-1}) & \leq & t_2\} \\
\vdots & & \\
\bigcap \{(j-1)\text{th smallest}(U_1, U_2, \ldots, U_{N-1}) & \leq & t_{j-1}\} \\
\bigcap \{j\text{th smallest}(U_1, U_2, \ldots, U_{N-1}) & \leq & t_{j+1}\} \\
\bigcap \{(j+1)\text{th smallest}(U_1, U_2, \ldots, U_{N-1}) & \leq & t_{j+2}\} \\
\vdots & & \\
\bigcap \{\max(U_1, U_2, \ldots, U_{N-1}) & \leq & t_N\} \\
\bigcap \{t_{j-1} < U_N \leq t_j\} & \bigcap & \mathcal{E}^{0,N}(\mathbf{t})
\end{array} \right.$$

where $j = 1, 2, \ldots, N$. We claim that these are mutually exclusive, collectively exhaustive subsets of $\mathcal{E}_N^S(\mathbf{t})$, just as in the $N = 3$ case. They are certainly mutually exclusive, since U_N is defined over a different interval for each $\mathcal{B}_N^j(\mathbf{t})$. They are also certainly subsets of $\mathcal{E}_N^S(\mathbf{t})$. The ith condition of $\mathcal{E}_N^S(\mathbf{t})$, ith smallest$(U_1, U_2, \ldots, U_N) \leq t_i$,

is satisfied by the ith condition of $\mathcal{B}_N^j(\mathbf{t})$, for $i = 1, 2, \ldots, j-1$; it is satisfied by $t_{j-1} < U_N \leq t_j$ for $i = j$; and it is satisfied by the $(i-1)$th condition of $\mathcal{B}_N^j(\mathbf{t})$ for $i = j+1, j+2, \ldots, N$.

Finally, the argument for the $\mathcal{B}_N^j(\mathbf{t})$'s being collectively exhaustive may be made by contradiction, as before: say I have a set of $U_1, U_2, \ldots, U_N = \mathbf{U}$, which I claim is not contained in any of the $\mathcal{B}_N^j(\mathbf{t})$'s, and say that, for $\mathbf{U}$, $t_{i-1} < U_N \leq t_i$. So, if $\mathbf{U}$ is in any of the $\mathcal{B}_N^j(\mathbf{t})$'s, it will be in $\mathcal{B}_N^i(\mathbf{t})$. Now I claim that $\mathbf{U}$ violates constraint k of $\mathcal{B}_N^i(\mathbf{t})$, where $k = 1, 2, \ldots, N-1$. But then $\mathbf{U}$ cannot be contained in $\mathcal{E}_N^S(\mathbf{t})$. For $k \leq i-1$, constraint k being violated means that the kth smallest of $U_1, U_2, \ldots, U_{N-1}$ is greater than t_k, and since we know $U_N > t_k$, this is equivalent to the kth smallest of $U_1, U_2, \ldots, U_N$ being greater then t_k, which violates constraint k of $\mathcal{E}_N^S(\mathbf{t})$. Similarly, for $k \geq i$, constraint k being violated means that the kth smallest of $U_1, U_2, \ldots, U_{N-1}$ is greater than t_{k+1}, and since we know $U_N \leq t_{k+1}$, this is equivalent to the $(k+1)$th smallest of $U_1, U_2, \ldots, U_N$ being greater then t_{k+1}, which violates constraint $k+1$ of $\mathcal{E}_N^S(\mathbf{t})$. Hence, if $\mathbf{U}$ is not in any of the $\mathcal{B}_N^j(\mathbf{t})$'s, then it is also not in $\mathcal{E}_N^S(\mathbf{t})$, so that the $\mathcal{B}_N^j(\mathbf{t})$'s are indeed a set of mutually exclusive, collectively exhaustive subsets of $\mathcal{E}_N^S(\mathbf{t})$. In a manner parallel to that used in the $N = 3$ proof, we will show that:

$$\begin{aligned} & \mathrm{E}[A(t_i)|\mathcal{B}_N^j(\mathbf{t})] \leq \mathrm{E}[A(t_i)|\mathcal{U}_N(\mathbf{t})], \quad j = 1, 2, \ldots, N; \ i = 1, 2, \ldots, N-1 \\ \Longrightarrow \quad & \mathrm{E}[A(t_i)|\mathcal{E}_N^S(\mathbf{t})] \leq \mathrm{E}[A(t_i)|\mathcal{U}_N(\mathbf{t})], \quad i = 1, 2, \ldots, N-1 \end{aligned}$$

We first make the induction assumption that:

$$\mathrm{E}[A(t)|\mathcal{E}_{N-1}^S(\mathbf{t})] \leq \mathrm{E}[A(t)|\mathcal{U}_{N-1}(\mathbf{t})], \quad 0 \leq t \leq t_{N-1}$$

and that this is true for any set of t_i's satisfying $0 < t_1 < t_2 < \cdots < t_{N-1}$. We now show that $\mathrm{E}[A(t_i)|\mathcal{B}_N^j(\mathbf{t})] \leq \mathrm{E}[A(t_i)|\mathcal{U}_N(\mathbf{t})]$ by considering the three cases $i < j$, $i = j$, and $i > j$.

i < j

$$\mathrm{E}[A(t_i)|\mathcal{B}_N^j(\mathbf{t})] = \mathrm{E}[A(t_i) \text{ due to } U_1, U_2, \ldots, U_{N-1}|\mathcal{B}_N^j(\mathbf{t})]$$

$$
\begin{aligned}
&+ \mathrm{E}[A(t_i) \text{ due to } U_N | \mathcal{B}_N^j(\mathbf{t})] \\
&= \mathrm{E}[A(t_i) \text{ due to } U_1, U_2, \ldots, U_{N-1} \\
&\qquad | \mathcal{E}_{N-1}^S(t_1, t_2, \ldots, t_{j-1}, t_{j+1}, t_{j+2}, \ldots, t_N)] \\
&\quad + \mathrm{E}[A(t_i) \text{ due to } U_N | t_{j-1} < U_N \leq t_j] \\
&= \mathrm{E}[A(t_i) | \mathcal{E}_{N-1}^S(t_1, t_2, \ldots, t_{j-1}, t_{j+1}, t_{j+2}, \ldots, t_N)] + 0 \\
&\leq \mathrm{E}[A(t_i) | \mathcal{U}_{N-1}(t_1, t_2, \ldots, t_{j-1}, t_{j+1}, t_{j+2}, \ldots, t_N)] \\
&= i + \frac{t_i}{t_{i+1}} + \frac{t_i}{t_{i+2}} + \cdots + \frac{t_i}{t_{j-1}} + \frac{t_i}{t_{j+1}} + \cdots + \frac{t_i}{t_N} \\
&< i + \sum_{k=i+1}^{N} \frac{t_i}{t_k} \\
&= \mathrm{E}[A(t_i) | \mathcal{U}_N(\mathbf{t})]
\end{aligned}
$$

$i = j$

$$
\begin{aligned}
\mathrm{E}[A(t_i) | \mathcal{B}_N^i(\mathbf{t})] &= \mathrm{E}[A(t_i) \text{ due to } U_1, U_2, \ldots, U_{N-1} | \mathcal{B}_N^i(\mathbf{t})] \\
&\quad + \mathrm{E}[A(t_i) \text{ due to } U_N | \mathcal{B}_N^i(\mathbf{t})] \\
&= \mathrm{E}[A(t_i) \text{ due to } U_1, U_2, \ldots, U_{N-1} \\
&\qquad | \mathcal{E}_{N-1}^S(t_1, t_2, \ldots, t_{i-1}, t_{i+1}, t_{i+2}, \ldots, t_N)] \\
&\quad + \mathrm{E}[A(t_i) \text{ due to } U_N | t_{i-1} < U_N \leq t_i] \\
&= \mathrm{E}[A(t_i) | \mathcal{E}_{N-1}^S(t_1, t_2, \ldots, t_{i-1}, t_{i+1}, t_{i+2}, \ldots, t_N)] + 1 \\
&\leq \mathrm{E}[A(t_i) | \mathcal{U}_{N-1}(t_1, t_2, \ldots, t_{i-1}, t_{i+1}, t_{i+2}, \ldots, t_N)] + 1 \\
&= i - 1 + \frac{t_i}{t_{i+1}} + \frac{t_i}{t_{i+2}} + \cdots + \frac{t_i}{t_N} + 1 \\
&= i + \sum_{k=i+1}^{N} \frac{t_i}{t_k} \\
&= \mathrm{E}[A(t_i) | \mathcal{U}_N(\mathbf{t})]
\end{aligned}
$$

$i > j$

$$
\begin{aligned}
\mathrm{E}[A(t_i) | \mathcal{B}_N^j(\mathbf{t})] &= \mathrm{E}[A(t_i) \text{ due to } U_1, U_2, \ldots, U_{N-1} | \mathcal{B}_N^j(\mathbf{t})] \\
&\quad + \mathrm{E}[A(t_i) \text{ due to } U_N | \mathcal{B}_N^j(\mathbf{t})]
\end{aligned}
$$

$$
\begin{aligned}
&= \mathrm{E}[A(t_i) \text{ due to } U_1, U_2, \ldots, U_{N-1} \\
&\qquad |\mathcal{E}^S_{N-1}(t_1, t_2, \ldots, t_{j-1}, t_{j+1}, t_{j+2}, \ldots, t_N)] \\
&\quad + \mathrm{E}[A(t_i) \text{ due to } U_N | t_{j-1} < U_N \leq t_j] \\
&= \mathrm{E}[A(t_i)|\mathcal{E}^S_{N-1}(t_1, t_2, \ldots, t_{j-1}, t_{j+1}, t_{j+2}, \ldots, t_N)] + 1 \\
&\leq \mathrm{E}[A(t_i)|\mathcal{U}_{N-1}(t_1, t_2, \ldots, t_{j-1}, t_{j+1}, t_{j+2}, \ldots, t_N)] + 1 \\
&= \sum_{k=1}^{j-1} 1 + \sum_{k=j+1}^{i} 1 + \sum_{k=i+1}^{N} \frac{t_i}{t_k} + 1 \\
&= i + \sum_{k=i+1}^{N} \frac{t_i}{t_k} \\
&= \mathrm{E}[A(t_i)|\mathcal{U}_N(\mathbf{t})]
\end{aligned}
$$

Therefore, we may say

$$
\begin{aligned}
\mathrm{E}[A(t_i)|\mathcal{E}^S_N(\mathbf{t})] &= \sum_{j=1}^{N} \mathrm{E}[A(t_i)|\mathcal{B}^j_N(\mathbf{t})] \times \Pr[\mathcal{B}^j_N(\mathbf{t})|\mathcal{E}^S_N(\mathbf{t})] \\
&\leq \sum_{j=1}^{N} \mathrm{E}[A(t_i)|\mathcal{U}_N(\mathbf{t})] \times \Pr[\mathcal{B}^j_N(\mathbf{t})|\mathcal{E}^S_N(\mathbf{t})] \\
&= \mathrm{E}[A(t_i)|\mathcal{U}_N(\mathbf{t})], \quad i = 1, 2, \ldots, N-1
\end{aligned}
$$

Again, because of the linearity of both functions between the breakpoints, this is equivalent to

$$\mathrm{E}[A(t)|\mathcal{E}^S_N(\mathbf{t})] \leq \mathrm{E}[A(t)|\mathcal{U}_N(\mathbf{t})], \quad 0 \leq t \leq t_N$$

and thus the theorem is proved. ∎

The algorithm for determining $\mathrm{E}[A(t)|\mathcal{U}_N(\mathbf{t})]$ is $\mathrm{O}(N^2)$, since in order to determine $\mathrm{E}[A(t_i)|\mathcal{U}_N(\mathbf{t})], i = 1, 2, \ldots, N-1$, we must perform $N - i$ divisions and additions, and

$$\sum_{i=1}^{N-1} (N - i) = N(N-1) - \frac{N(N-1)}{2} = \frac{N(N-1)}{2}$$

The calculations that are performed are so simple, though, compared to those that must be performed to determine $\mathrm{E}[A(t)|\mathcal{E}^S_N(\mathbf{t})]$ for the exact QIE, that the time savings is quite dramatic. In Section 4.4, we present typical runs and computation times for

the exact QIE and for the upper bound. We also consider weighted combinations of the upper bound and the lower bound presented in the previous section, which, in some cases, do quite well in *approximating* the value of $\mathrm{E}[A(t)|\mathcal{E}_N^S(\mathbf{t})]$ for the exact QIE.

4.3 Trapezoidal Approximation

In this section, we explore an approach to approximating the value of $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$, an approximation which is based loosely on the observations of Chapter 3. First note that, in the previous section, we could obtain the identical upper bound to $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$ by creating a set of N approximations to the densities of the ordered arrival times, where the approximation to $f[X_k|\mathcal{E}^S(\mathbf{t})]$ would be a function which is uniform between 0 and t_k. (This is just a different interpretation of what we did in the last section.) With the trapezoidal approximation, we create a different set of approximations to the densities of the ordered arrivals, conditioned on the arrival-time inequalities. Specifically, we create a set of density functions $f[Y_k], k = 1, 2, \ldots, N$, which are approximations to the functions $f[X_k|\mathcal{E}^S(\mathbf{t})]$.

Recall that in the case of the upper bound with $N = 1$, the density function $f(U_1|U_1 \leq t_1)$ is identical to the density function $f(X_1|X_1 \leq t_1)$: both are uniform with value $1/t_1$, for values between 0 and t_1, and hence the expected number of arrivals by time t is identical for both the exact QIE algorithm and for the upper bound. The trapezoidal approximation matches the exact density functions both when $N = 1$ ($f[Y_1]$ is just uniform over $(0, t_1]$) and $N = 2$. Specifically, in the case of $N = 2$, we begin with density functions which are identical to the density functions $f[X_1|\mathcal{E}^S(\mathbf{t})]$ and $f[X_2|\mathcal{E}^S(\mathbf{t})]$ (see Figure 3.1). Note that $f[X_2|\mathcal{E}^S(\mathbf{t})]$ is trapezoidal in shape. Since we are only really interested in the expected number of arrivals by t_1 and will linearly interpolate to find the expected number of arrivals for times less than t_1, we may also approximate $f[X_1|\mathcal{E}^S(\mathbf{t})]$ by a uniform function over $(0, t_1]$, to

simplify the analysis (particularly as we extrapolate this algorithm to larger values of N). With this approximation, then, we have:

$$f[Y_1] = \frac{1}{t_1}, \quad 0 < Y_1 \leq t_1$$

$$f[Y_2] = \begin{cases} \dfrac{2Y_2}{t_1(2-t_1)}, & 0 < Y_2 \leq t_1 \\ \dfrac{2}{2-t_1}, & t_1 < Y_2 \leq t_2 \end{cases}$$

When we calculate the expected number of arrivals by t_1, we get the same value as that calculated by the exact QIE algorithm (not surprisingly), a lower value than that calculated by the upper bound algorithm. But $f[Y_1]$ is identical to $f[U_1|U_1 \leq t_1, U_2 \leq t_2]$. The difference is that $f[U_2|U_1 \leq t_1, U_2 \leq t_2]$ is uniform, while $f[Y_2]$ is trapezoidal: $f[Y_2]$ essentially causes the other arrival to occur later and thus reduces the expected number of arrivals by time t_1. This then suggests a way to improve on the upper bound to the expected number of arrivals. Namely, rather than using uniform density functions for all of the arrivals, which causes the arrivals to occur too soon, we use trapezoidal density functions for all of the arrivals (except the first, which we still approximate by a uniform), which causes these arrivals to occur later and thus reduces the expected number of arrivals by the t_i's.

Specifically, we have the following. We define $f[Y_k]$ to be an approximation to $f[X_k|\mathcal{E}^S(\mathbf{t})]$. We would like $f[Y_k]$ to be linearly increasing up to t_{k-1} and then to be uniform for $t_{k-1} < Y_k \leq t_k$, i.e., we require that $Y_i \leq t_i$ for all i so that we know that we have enough arrivals by t_i for the busy period to continue. After working out what the values must be in order to ensure that the area of $f[Y_k]$ is unity, we have the following:

$$f[Y_k] = \begin{cases} \dfrac{2Y_k}{t_{k-1}(2t_k - t_{k-1})}, & 0 < Y_k \leq t_{k-1} \\ \dfrac{2}{2t_k - t_{k-1}}, & t_{k-1} < Y_k \leq t_k \end{cases} \qquad k = 1, 2, \ldots, N$$

Now we define the trapezoidal approximation to the expected number of arrivals by time t_i, which we denote by $A^{TR}(t_i)$, to be the expected number of arrivals by time

t_i due to the Y_k's. Specifically:

$$\begin{aligned} A^{TR}(t_i) &\equiv \mathrm{E}[A(t_i) \text{ due to the } Y_k\text{'s}] \\ &= \sum_{k=1}^{N} \mathrm{E}[A(t_i) \text{ due to } Y_k] \end{aligned}$$

But the quantity in the summation is easily found as follows:

$$\begin{aligned} A(t_i) \text{ due to } Y_k &= \begin{cases} 1, & Y_k \le t_i \\ 0, & Y_k > t_i \end{cases} \\ \Longrightarrow \mathrm{E}[A(t_i) \text{ due to } Y_k] &= \Pr[Y_k \le t_i] \\ &= \begin{cases} 1, & k = 1, 2, \ldots, i \\ \dfrac{t_i^2}{t_{k-1}(2t_k - t_{k-1})}, & i < k \le N \end{cases} \end{aligned}$$

Therefore, we find that $A^{TR}(t_i)$ is given by the following:

$$A^{TR}(t_i) = i + \sum_{k=i+1}^{N} \frac{t_i^2}{t_{k-1}(2t_k - t_{k-1})} \tag{4.4}$$

If we actually used the Y_k's to evaluate $A^{TR}(t)$ for values of t not equal to a t_i, then $A^{TR}(t)$ would not be a piecewise linear function but would be quadratic. Hence, we *define* $A^{TR}(t)$ for values of t in the range $t_{i-1} < t < t_i$ to be the linear interpolation between $A^{TR}(t_{i-1})$ and $A^{TR}(t_i)$. Specifically, we have:

$$A^{TR}(t) = A^{TR}(t_{i-1}) + \frac{t - t_{i-1}}{t_i - t_{i-1}} \left\{ A^{TR}(t_i) - A^{TR}(t_{i-1}) \right\}, \quad t_{i-1} < t < t_i$$

Of course, only the values of the function at the t_i's are used in calculating the queue statistics of interest; the above is provided for completeness.

We now proceed to demonstrate that the trapezoidal approximation is neither an upper bound nor a lower bound (proof by counterexample); that it is neither concave nor convex (also proof by counterexample); and that it does give values for the expected number of arrivals by time t_i which are lower than those provided by the upper bound, i.e., we have

$$A^{TR}(t_i) \le \mathrm{E}[A(t_i) | \mathcal{U}_N(\mathbf{t})], \quad i = 1, 2, \ldots, N \tag{4.5}$$

Consider a congestion period with 3 customers and with service commencement times given by:

$$\begin{aligned} t_1 &= 15 \\ t_2 &= 19 \\ t_3 &= 20 \end{aligned}$$

We may use Equation 4.4 to calculate the expected number of arrivals by time t_i using the trapezoidal approximation. To find the exact expected number of arrivals by time t_i, either we may use the beta-matrix technique described in Chapter 2; or we may integrate and sum the densities given in Section 3.2 for X_2 and X_3. The values we get are the following:

Time	$\mathrm{E}[A(t)\|\mathcal{E}^S(\mathbf{t})]$	$A^{TR}(t)$
$t_1 = 15$	2.2931	2.2161
$t_2 = 19$	2.8678	2.9048
$t_3 = 20$	3.0000	3.0000

Note that $\mathrm{E}[A(t_1)|\mathcal{E}^S(\mathbf{t})] > A^{TR}(t_1)$. However, we also have that $\mathrm{E}[A(t_2)|\mathcal{E}^S(\mathbf{t})] < A^{TR}(t_2)$. This is sufficient to demonstrate that $A^{TR}(t)$ is neither an upper nor a lower bound to $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$.

Next we wish to consider the concavity of the function $A^{TR}(t)$ in this particular $N = 3$ example. We define m_i^{QIE} to be the slope of the line connecting $(t_{i-1}, \mathrm{E}[A(t_{i-1})|\mathcal{E}^S(\mathbf{t})])$ with $(t_i, \mathrm{E}[A(t_i)|\mathcal{E}^S(\mathbf{t})])$. Similarly, we define m_i^{TR} to be the slope of the line connecting $(t_{i-1}, A^{TR}(t_{i-1}))$ with $(t_i, A^{TR}(t_i))$. Then we have:

$$\begin{aligned} m_i^{QIE} &= \frac{\mathrm{E}[A(t_i)|\mathcal{E}^S(\mathbf{t})] - \mathrm{E}[A(t_{i-1})|\mathcal{E}^S(\mathbf{t})]}{t_i - t_{i-1}} \\ m_i^{TR} &= \frac{A^{TR}(t_i) - A^{TR}(t_{i-1})}{t_i - t_{i-1}} \end{aligned}$$

which gives us the following for the slopes in the $t_1 = 15, t_2 = 19, t_3 = 20$ example:

Interval	m_i^{QIE}	m_i^{TR}
(t_0, t_1)	0.1529	0.1477
(t_1, t_2)	0.1437	0.1722
(t_2, t_3)	0.1322	0.0952

As can be seen from the above table and as we already knew, the exact QIE estimate is a piecewise-linear, concave function whose slope decreases as t increases. As can also be seen, the trapezoidal approximation of the cumulative number of arrivals by time t is convex in the interval (t_0, t_2), and is concave in the interval (t_1, t_3). This is sufficient to demonstrate that $A^{TR}(t)$ is neither a convex nor a concave function of t.

Finally, we wish to demonstrate that Equation 4.5 holds. We know the following:

$$\begin{aligned} A^{TR}(t_i) &= \sum_{k=1}^{N} \mathrm{E}[A(t_i) \text{ due to } Y_k] \\ \mathrm{E}[A(t_i)|\mathcal{U}_N(\mathbf{t})] &= \sum_{k=1}^{N} \mathrm{E}[A(t_i) \text{ due to } U_k | U_k \leq t_k] \end{aligned}$$

We also know that:

$$\begin{aligned} \mathrm{E}[A(t_i) \text{ due to } Y_k] &= \Pr[Y_k \leq t_i] \\ \mathrm{E}[A(t_i) \text{ due to } U_k | U_k \leq t_k] &= \Pr[U_k \leq t_i | U_k \leq t_k] \end{aligned}$$

We can easily show dominance of the right-hand side of the bottom expression over the right-hand side of the top expression. When $k \leq i$, both right-hand sides are unity. When $k > i$, we have:

$$\begin{aligned} \Pr[Y_k \leq t_i] &= \frac{t_i^2}{t_{k-1}(2t_k - t_{k-1})}, \quad k > i \\ &= \frac{t_{k-1}}{2t_k - t_{k-1}}, \quad i = k-1 \\ \Pr[U_k \leq t_i | U_k \leq t_k] &= \frac{t_i}{t_k}, \quad k > i \\ &= \frac{t_{k-1}}{t_k}, \quad i = k-1 \end{aligned}$$

Both functions begin at the origin, and $\Pr[Y_k \leq t_i]$ is convex in t_i while $\Pr[U_k \leq t_i | U_k \leq t_k]$ is linear. Hence, if the linear function still dominates the convex function

at the greatest possible value of t_i (since we require $k > i$, this would be at the point $t_i = t_{k-1}$), then it also dominates everywhere in between the origin and this point. But considering the above expressions when $i = k - 1$, we see that the dominance exists if and only if $2t_k - t_{k-1} \geq t_k \iff t_k \geq t_{k-1}$. This is true by definition, and hence we have the stochastic dominance (of $U_k|U_k \leq t_k$ over Y_k) to give us the result of Equation 4.5:

$$\begin{aligned} \Pr[Y_k \leq t_i] &\leq \Pr[U_k \leq t_i | U_k \leq t_k], \quad i = 1, 2, \ldots, N, \quad k = 1, 2, \ldots, N \\ \Longrightarrow A^{TR}(t_i) &\leq \mathrm{E}[A(t_i)|\mathcal{U}_N(\mathbf{t})], \quad i = 1, 2, \ldots, N \end{aligned}$$

Because both $A^{TR}(t)$ and $\mathrm{E}[A(t)|\mathcal{U}_N(\mathbf{t})]$ are or have been defined to be linear for values of t between the t_i's, then the inequality above actually holds for all values of t in $[0, t_N]$.

4.4 Computational Results

In this section, we present figures and tables which document the results of runs of the concavity lower bound algorithm, the uniform upper bound algorithm, and the trapezoidal approximation algorithm. We then consider combining pairs of these algorithms to get better estimates of the exact QIE expected queue length function. We include results from simulation of an M/M/1 queue. These data were generated by three simulation runs of 100 hours with Poisson arrivals at rate 10 per hour, a single server, and exponential service time with expected value of 3 minutes for the first run (giving a value of $\rho = 0.5$) and 4 minutes for the second and third runs (giving a value of $\rho = 0.67$). There were 6 congestion periods in the first run which had greater than 11 customers. They had 14, 13, 14, 18, 21, and 12 customers, respectively. In the following results, we consider either all 6 of these runs, or just the two longest, with $N = 18$ and $N = 21$. We also examine the two longest runs in the other two congestion periods, both of which, coincidentally, had $N = 58$ customers. These runs are presented as examples of very long congestion periods. The statistics that are

used for comparison of the various algorithms include: $E[L_Q|\ldots]$, the time-averaged number of customers in queue; $E[W_Q|\ldots]$, the average wait in queue; and δ, the approximation error, which we define to be the absolute area between $E[Q(t)|\mathcal{E}^S(\mathbf{t})]$ as generated by the exact QIE algorithm, and the approximation algorithm's estimate of the same function, divided by the duration of the congestion period, t_N. The run times to generate the expected cumulative number of arrivals by the t_i's for the different algorithms are also compared. Runs were on a 386/387-based Northgate Computer Systems PC. Each run time presented below is actually the average of 1000 (for long runs, presented to hundredths of a second), 3000 (for intermediate runs, presented to thousandths of a second significance) or 10,000 (for short runs, presented to ten-thousandths of a second significance) run times from different runs of the program on the same data. This averaging was necessary, because the system clock is only updated every 0.0549254 seconds [Scan 83], so to get accuracies better than 0.1 seconds, many runs must be averaged.

The first results are for the concavity lower bound. In Figures 4.1 and 4.2 we show the expected queue lengths as generated by the original QIE algorithm versus the expected queue lengths as generated by the concavity lower bound algorithm. The queue statistics and run times for these congestion periods, under the original QIE algorithm and the concavity lower bound, are presented in Table 4.1. Note that the run times for the concavity lower bound algorithm are very small. Also note that they are not monotonic with the size of the congestion period. This is because the number of slopes that must be calculated to generate the concave hull depends on the t-vector as well as the number of customers in the congestion period. For example, it is much faster to compute the concave hull of a set of (t_i, i) that describe a convex function than it is a set that describe a concave function. In general, this bound is not very good. At the end of this section, we will see how to combine it with either the uniform upper bound or the trapezoidal approximation to get a better approximation to the original QIE function.

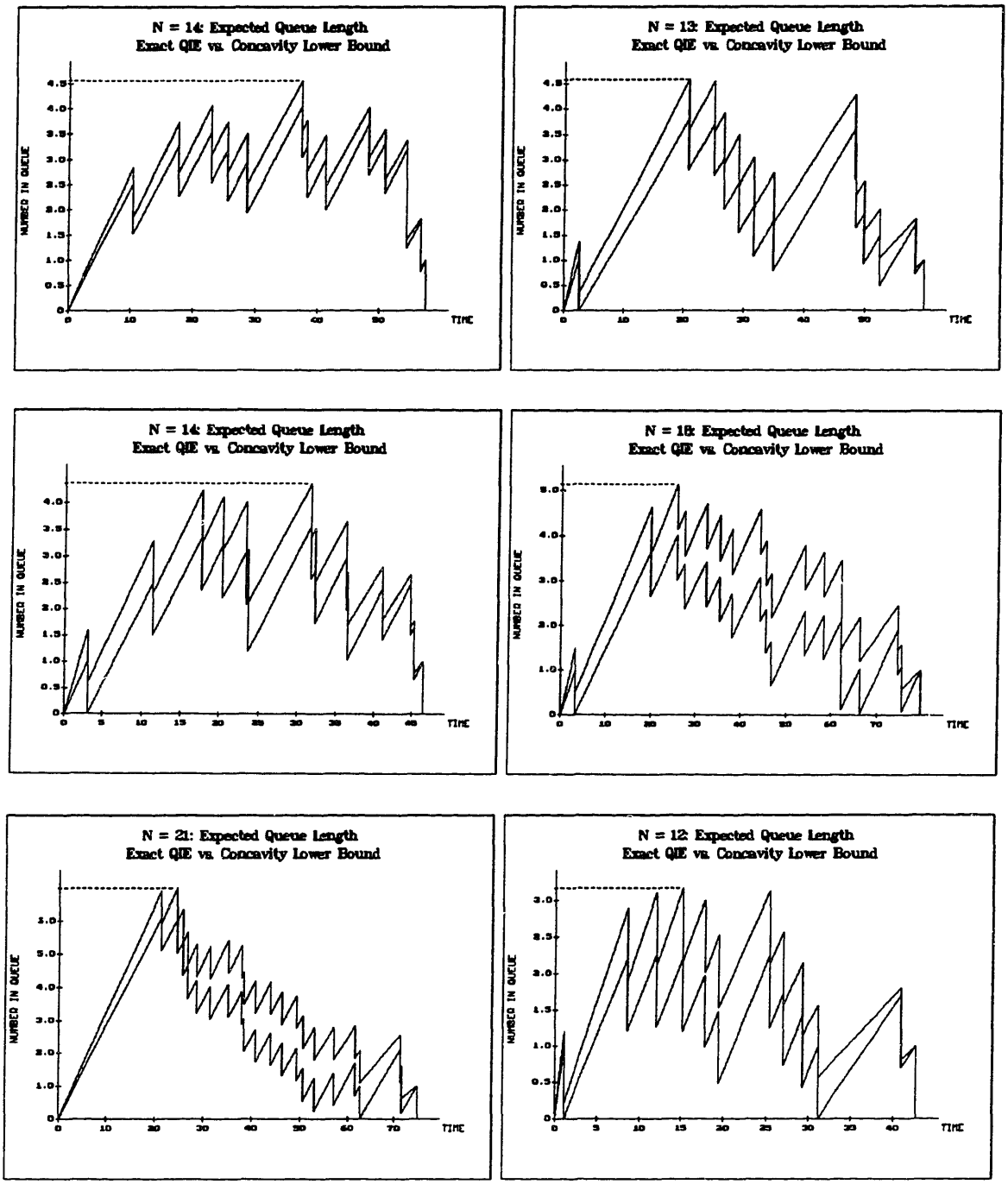
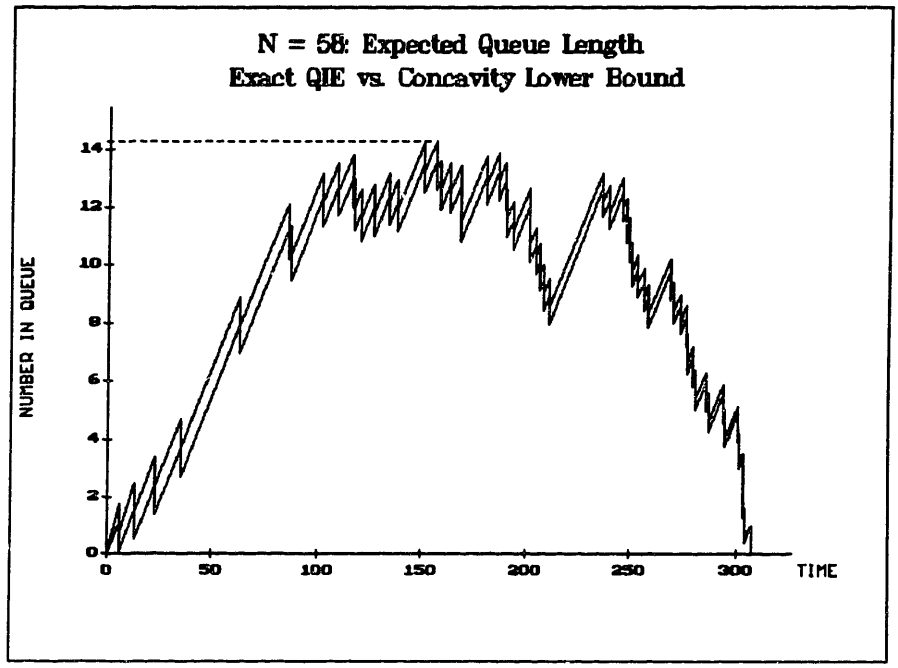


Figure 4.1: Expected Queue Length for Congestion Periods of 14, 13, 14, 18, 21, and 12 Customers: Exact QIE vs. Concavity Lower Bound



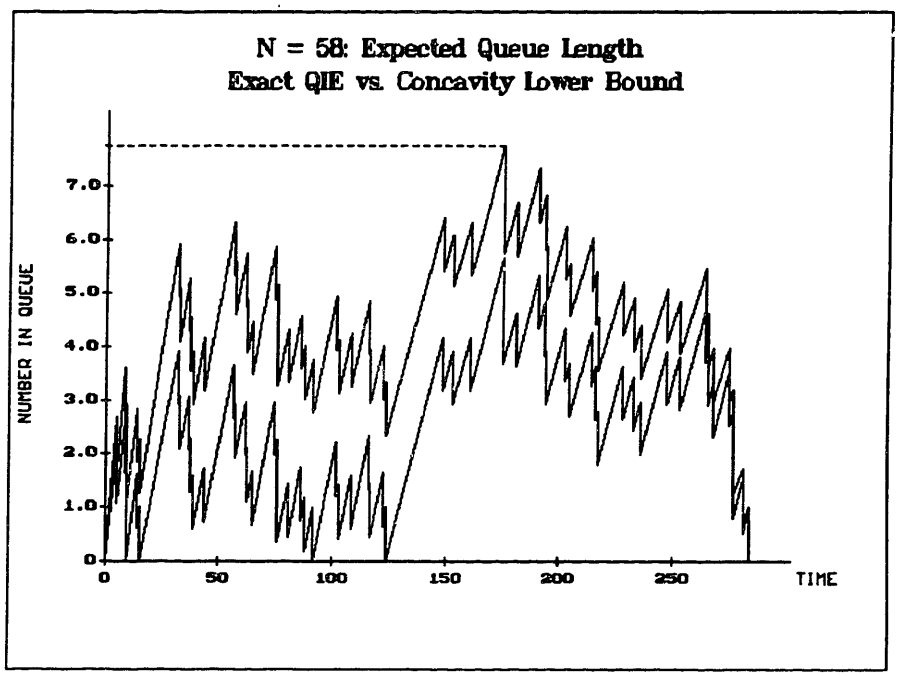


Figure 4.2: Expected Queue Length for Two Congestion Periods of 58 Customers: Exact QIE vs. Concavity Lower Bound

Size of Cong. Period	Data Used	E[L_Q|...]	E[W_Q|...] (minutes)	δ	Run Time (secs)
$N = 14$	Orig. QIE	2.8067	11.5727	0	0.172
	Concavity LB	2.4254	10.0005	0.3813	0.0011
$N = 13$	Orig. QIE	2.5231	11.5515	0	0.133
	Concavity LB	1.8718	8.5697	0.6513	0.0014
$N = 14$	Orig. QIE	2.6239	8.6997	0	0.173
	Concavity LB	1.9440	6.4456	0.6798	0.0014
$N = 18$	Orig. QIE	2.8649	12.6644	0	0.404
	Concavity LB	1.8121	8.0103	1.0528	0.0017
$N = 21$	Orig. QIE	3.3871	12.0615	0	0.694
	Concavity LB	2.4346	8.6698	0.9524	0.0016
$N = 12$	Orig. QIE	1.8193	6.4488	0	0.103
	Concavity LB	1.1878	4.2104	0.6315	0.0013
$N = 58$ (1)	Orig. QIE	9.3605	49.5175	0	31.15
	Concavity LB	8.6635	45.8302	0.6970	0.0031
$N = 58$ (2)	Orig. QIE	4.4114	21.5511	0	31.22
	Concavity LB	2.4802	12.1168	1.9312	0.0074

Table 4.1: Comparison of QIE and Concavity Lower Bound Algorithms for Eight Congestion Periods

The next set of results are for the uniform upper bound. In Figures 4.3 and 4.4, we show the expected queue lengths as generated by the original QIE algorithm versus the expected queue lengths as generated by the uniform upper bound algorithm. The queue statistics for these congestion periods, under the original QIE algorithm and the uniform upper bound, are presented in Table 4.2. Note that the run times for the uniform upper bound algorithm are also very short. This time, however, they are monotonic with the length of the congestion period, since the number of calculations that must be executed for a single congestion period is deterministic with the length of the congestion period and in no way depends on the t-vector. Note that there is a slight discrepancy between the runtimes of the two $N = 58$ congestion periods. This discrepancy is also evident in the runtimes for the trapezoidal approximation. The source of the discrepancy is unknown, although it is conjectured that the processor may handle different t-vectors slightly differently, which could result in slightly different runtimes. Again, we will see better performance of this algorithm at the end of this section when it is combined with the concavity lower bound.

Next, we present results for the trapezoidal approximation algorithm. In Figures 4.5 and 4.6, we show the expected queue lengths as generated by the original QIE algorithm versus the expected queue lengths as generated by the trapezoidal approximation algorithm. The queue statistics for these congestion periods, under the original QIE algorithm and the uniform upper bound, are presented in Table 4.3. The run times for the trapezoidal algorithm are again monotonic with the length of the congestion period. These runs take just slightly less than twice as long as the uniform upper bound runs on the same data. Note that this approximation does much better than the uniform upper bound. The tendency of the trapezoidal approximation algorithm is to underestimate the queue lengths at the beginning of the congestion period and then to overestimate them towards the end of the congestion period. The reason for this is that at the beginning of the congestion period, most of

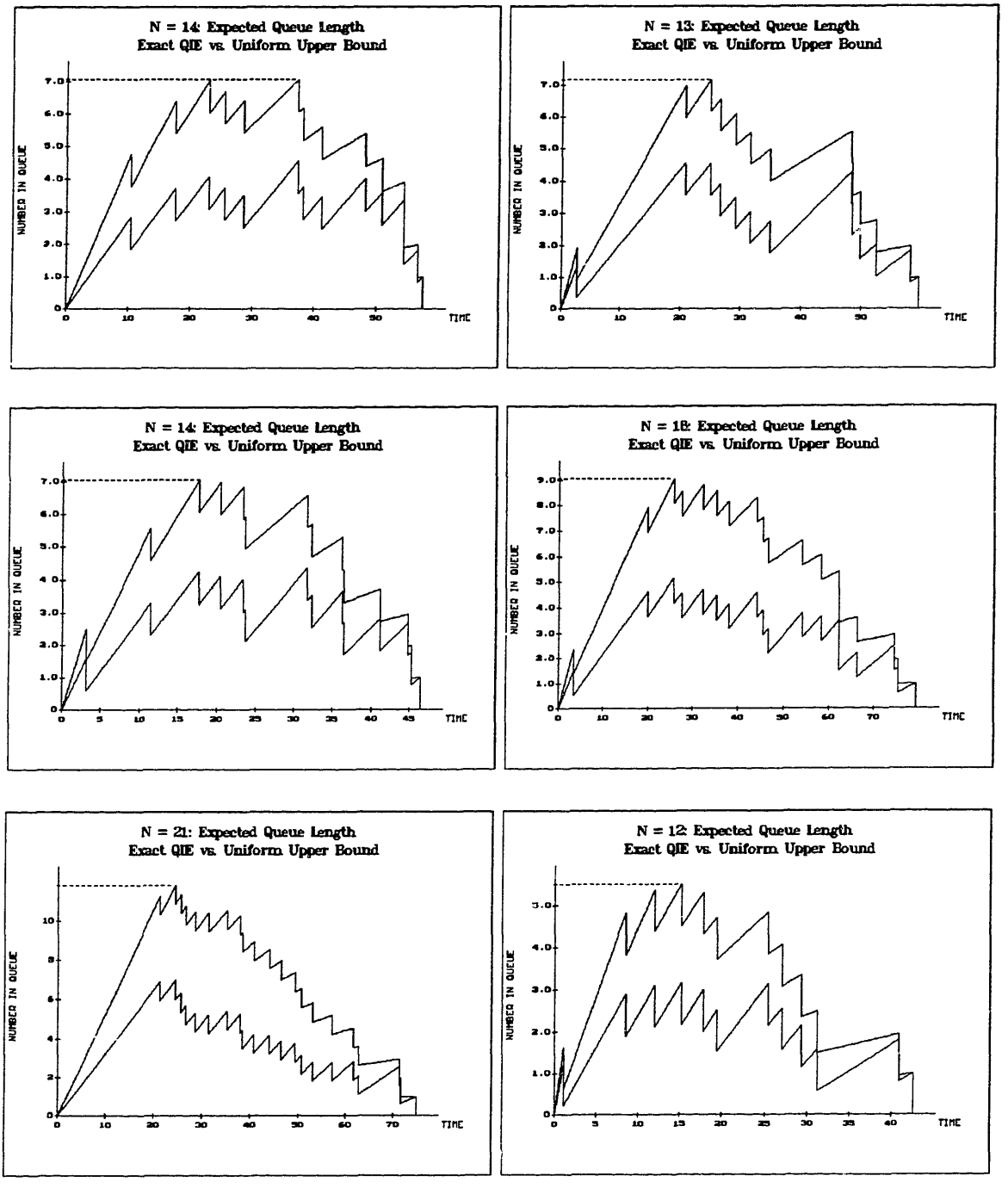
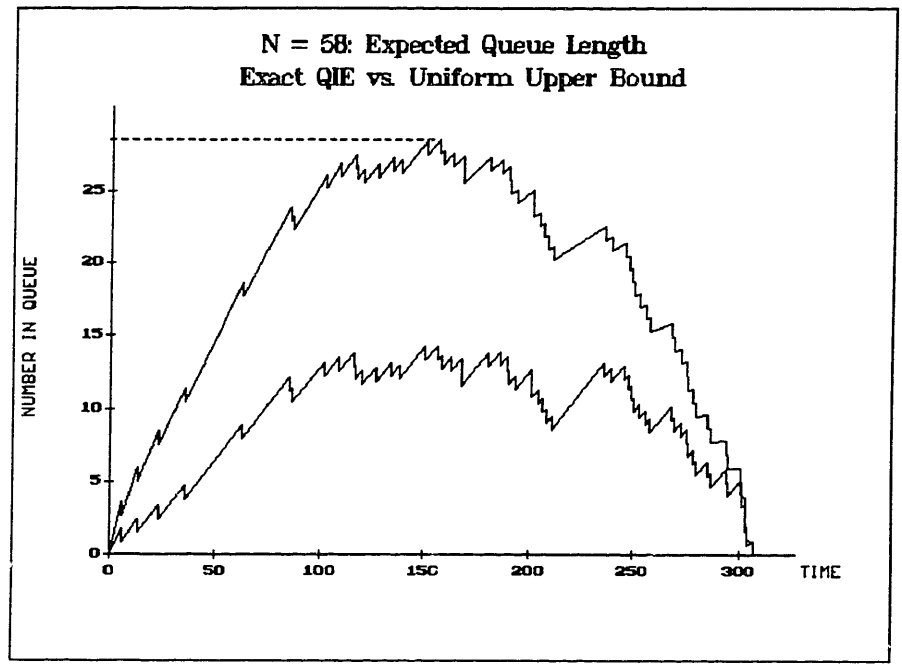


Figure 4.3: Expected Queue Length for Congestion Periods of 14, 13, 14, 18, 21, and 12 Customers: Exact QIE vs. Uniform Upper Bound



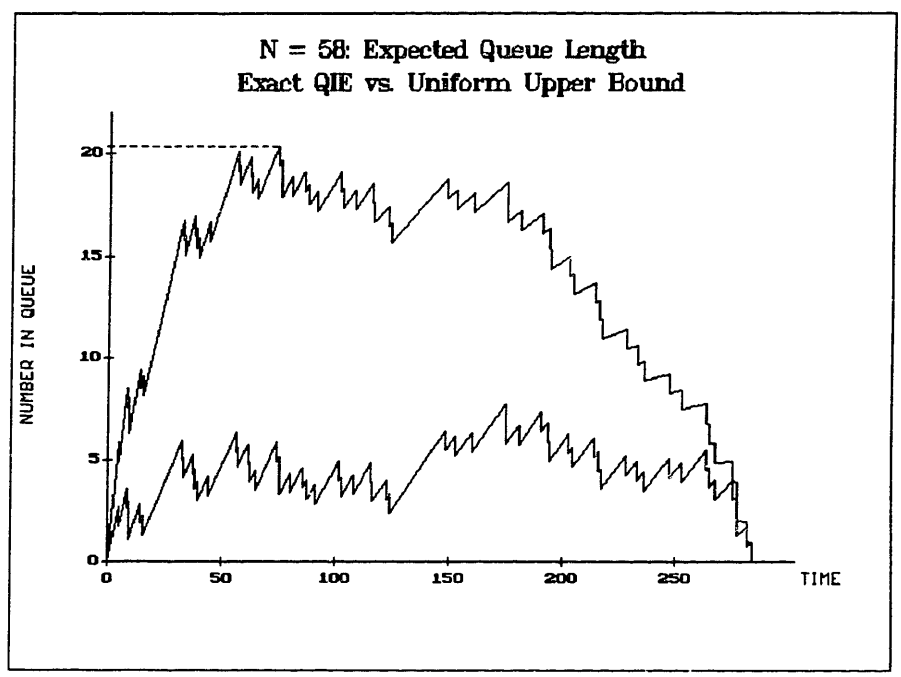


Figure 4.4: Expected Queue Length for Two Congestion Periods of 58 Customers: Exact QIE vs. Uniform Upper Bound

Size of Cong. Period	Data Used	E[L_Q|...]	E[W_Q|...] (minutes)	δ	Run Time (secs)
$N = 14$	Orig. QIE	2.8067	11.5727	0	0.172
	Uniform UB	4.7127	19.4318	1.9060	0.0021
$N = 13$	Orig. QIE	2.5231	11.5515	0	0.133
	Uniform UB	4.0867	18.7099	1.5636	0.0018
$N = 14$	Orig. QIE	2.6239	8.6997	0	0.173
	Uniform UB	4.4573	14.7788	1.8335	0.0021
$N = 18$	Orig. QIE	2.8649	12.6644	0	0.404
	Uniform UB	5.3502	23.6505	2.4853	0.0033
$N = 21$	Orig. QIE	3.3871	12.0615	0	0.694
	Uniform UB	6.3740	22.6982	2.9870	0.0043
$N = 12$	Orig. QIE	1.8193	6.4488	0	0.103
	Uniform UB	3.1765	11.2596	1.3572	0.0016
$N = 58$ (1)	Orig. QIE	9.3605	49.5175	0	31.15
	Uniform UB	18.8317	99.6206	9.4712	0.0310
$N = 58$ (2)	Orig. QIE	4.4114	21.5511	0	31.22
	Uniform UB	14.0497	68.6376	9.6383	0.0299

Table 4.2: Comparison of QIE and Uniform Upper Bound Algorithms for Eight Congestion Periods

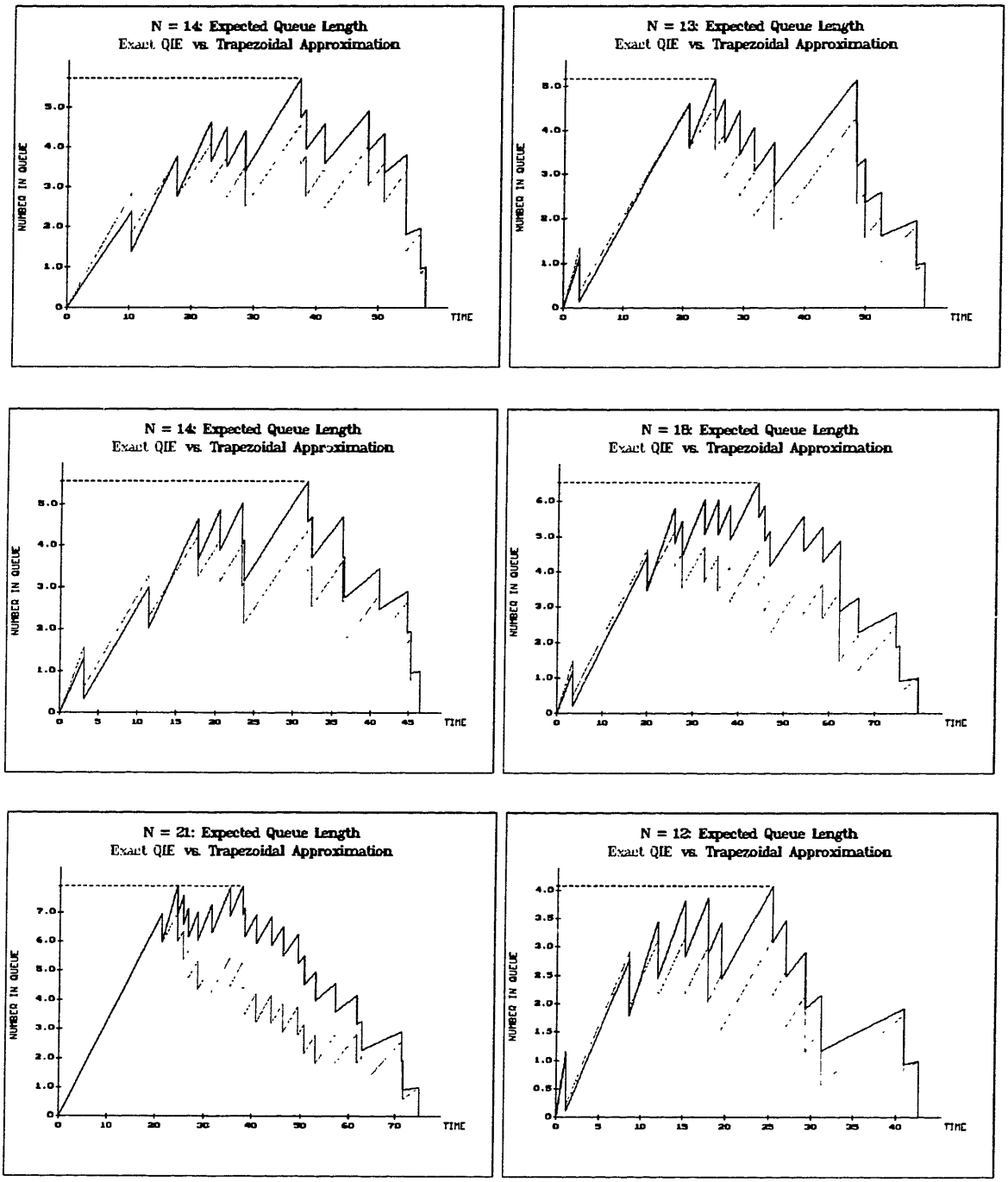


Figure 4.5: Expected Queue Length for Congestion Periods of 14, 13, 14, 18, 21, and 12 Customers: Exact QIE vs. Trapezoidal Approximation

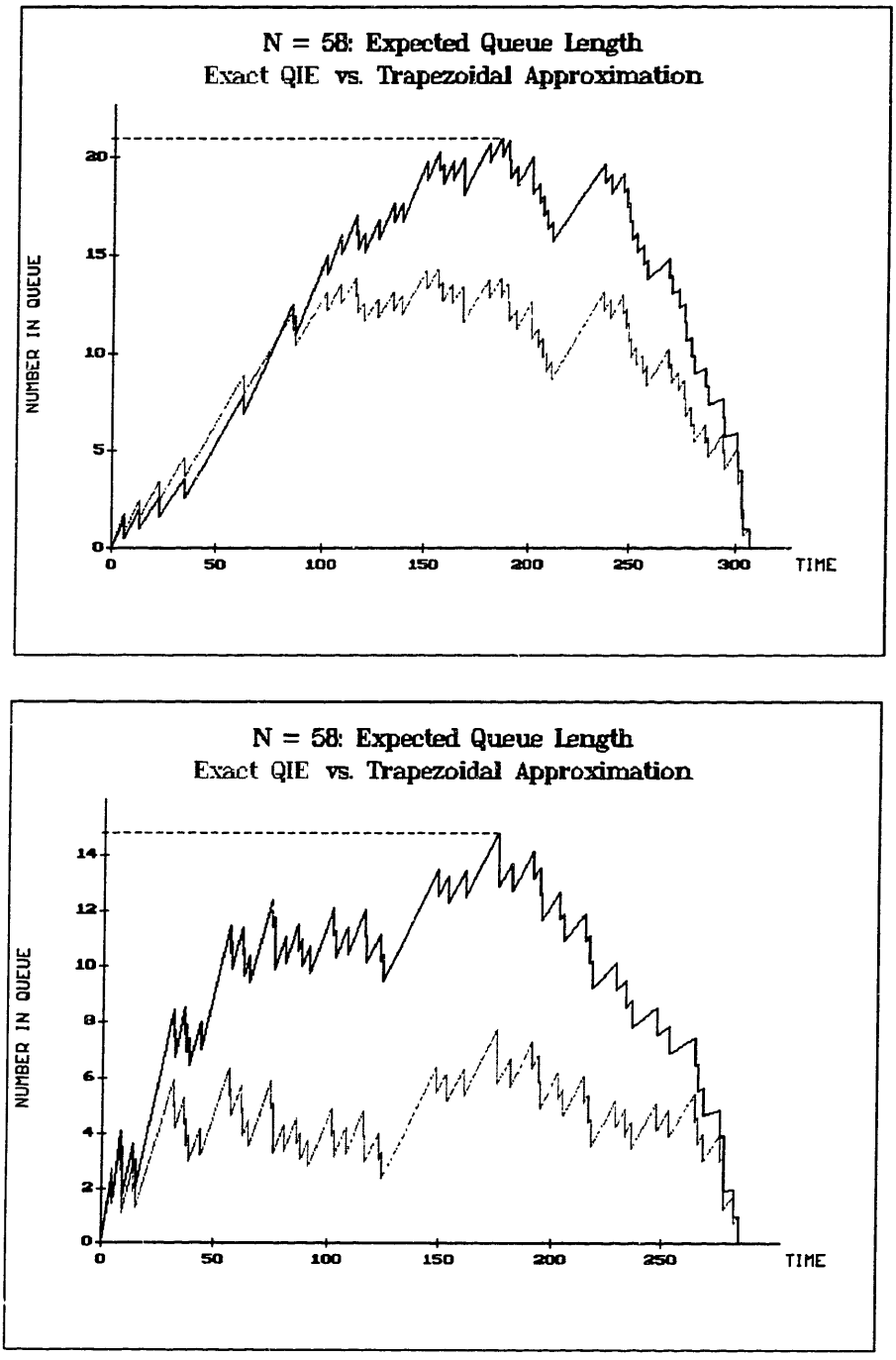


Figure 4.6: Expected Queue Length for Two Congestion Periods of 58 Customers: Exact QIE vs. Trapezoidal Approximation

Size of Cong. Period	Data Used	E[L_Q|...]	E[W_Q|...] (minutes)	δ	Run Time (secs)
$N = 14$	Orig. QIE	2.8067	11.5727	0	0.172
	Trap. Approx.	3.2745	13.5020	0.6072	0.0040
$N = 13$	Orig. QIE	2.5231	11.5515	0	0.133
	Trap. Approx.	2.9582	13.5434	0.5026	0.0035
$N = 14$	Orig. QIE	2.6239	8.6997	0	0.173
	Trap. Approx.	3.1007	10.2807	0.6086	0.0040
$N = 18$	Orig. QIE	2.8649	12.6644	0	0.404
	Trap. Approx.	3.7207	16.4474	0.9669	0.0065
$N = 21$	Orig. QIE	3.3871	12.0615	0	0.694
	Trap. Approx.	4.6003	16.3817	1.2132	0.0086
$N = 12$	Orig. QIE	1.8193	6.4488	0	0.103
	Trap. Approx.	2.2399	7.9396	0.4691	0.0030
$N = 58$ (1)	Orig. QIE	9.3605	49.5175	0	31.15
	Trap. Approx.	12.6367	66.8488	3.6746	0.0650
$N = 58$ (2)	Orig. QIE	4.4114	21.5511	0	31.22
	Trap. Approx.	9.4367	46.1014	5.0322	0.0641

Table 4.3: Comparison of QIE and Trapezoidal Approximation Algorithms for Eight Congestion Periods

the contribution to the expected queue length is from the early arrivals. The probability densities for the very early arrivals tend to be skewed much earlier than a trapezoidal density would suggest, leading to an underestimation of queue length. On the other hand, at the middle and end of the congestion period, the later arrivals are the only ones contributing to the expected queue length, and these tend to be skewed even more heavily towards the later times than a trapezoidal density, leading to an overestimation of queue length.

Finally, we present results for mixtures of the uniform upper bound with the concavity lower bound, and of the trapezoidal approximation with the concavity lower bound. In Figures 4.7 and 4.8, we show the expected queue lengths as generated by the original QIE algorithm versus the expected queue lengths as generated by a mixture of the uniform upper bound and the concavity lower bound. Then, in Figures 4.9 and 4.10, we show the expected queue lengths as generated by the original QIE algorithm versus the expected queue lengths as generated by a mixture of the trapezoidal approximation and the concavity lower bound. In both cases, the $N = 18$, $N = 21$, and the two $N = 58$ congestion periods are analyzed, with various fractions of the contributing algorithms. Also in both cases, we started by trying a 50/50 combination of each algorithm. We then tried two other combinations for each mixture. For the uniform upper bound mixture, it was clear that the 50/50 combination still had too much contribution from the upper bound, so both 40/60 and 30/70 combinations were tried for the smaller congestion periods, and 30/70 and 20/80 combinations were tried for the $N = 58$ congestion periods. For the trapezoidal approximation mixture, the 50/50 mixture seemed to perform well for the smaller congestion periods, so we tried combinations on either side of 50/50, namely 40/60 and 60/40. For the $N = 58$ congestion periods, we started with a 50/50 combination and the tried 40/60 and 30/70 combinations. It should be stressed that these choices were strictly ad hoc and depended on the results of these specific congestion periods. It appears that the amount of the concavity lower bound to be included in either

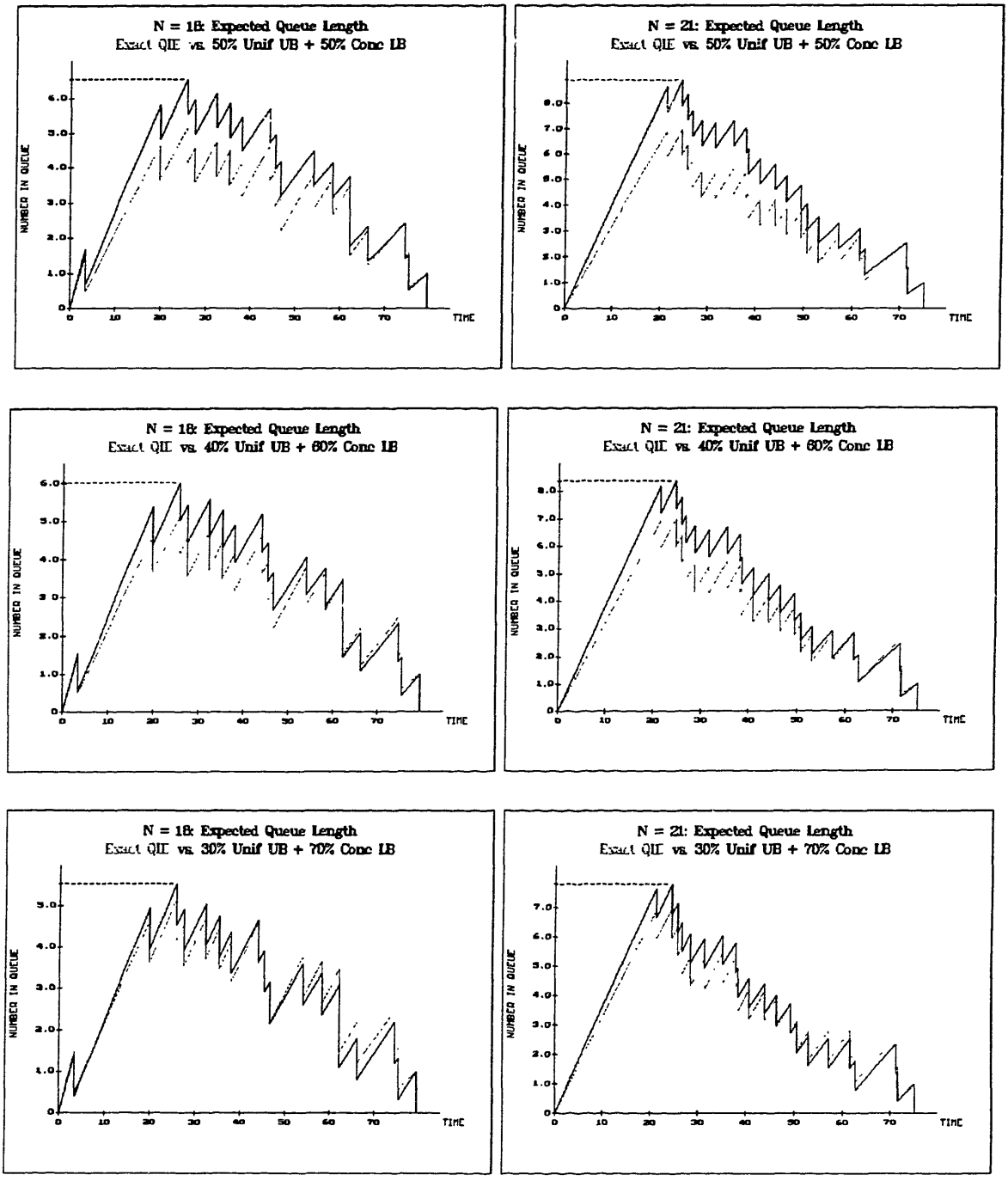


Figure 4.7: Expected Queue Length for Congestion Periods of 18 and 21 Customers: Exact QIE vs. Uniform UB/Concavity LB Combinations

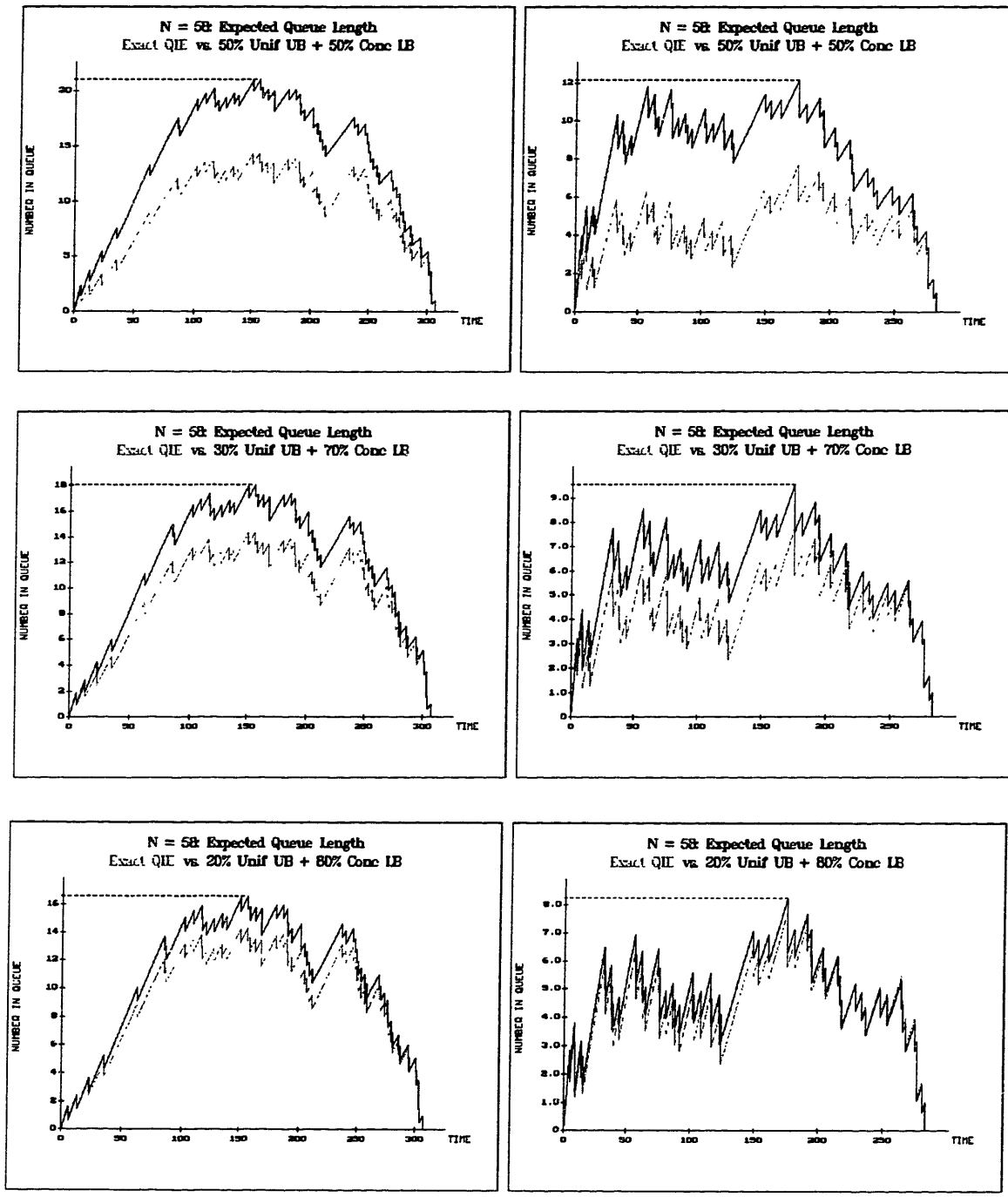


Figure 4.8: Expected Queue Length for Two Congestion Periods of 58 Customers: Exact QIE vs. Uniform UB/Concavity LB Combinations

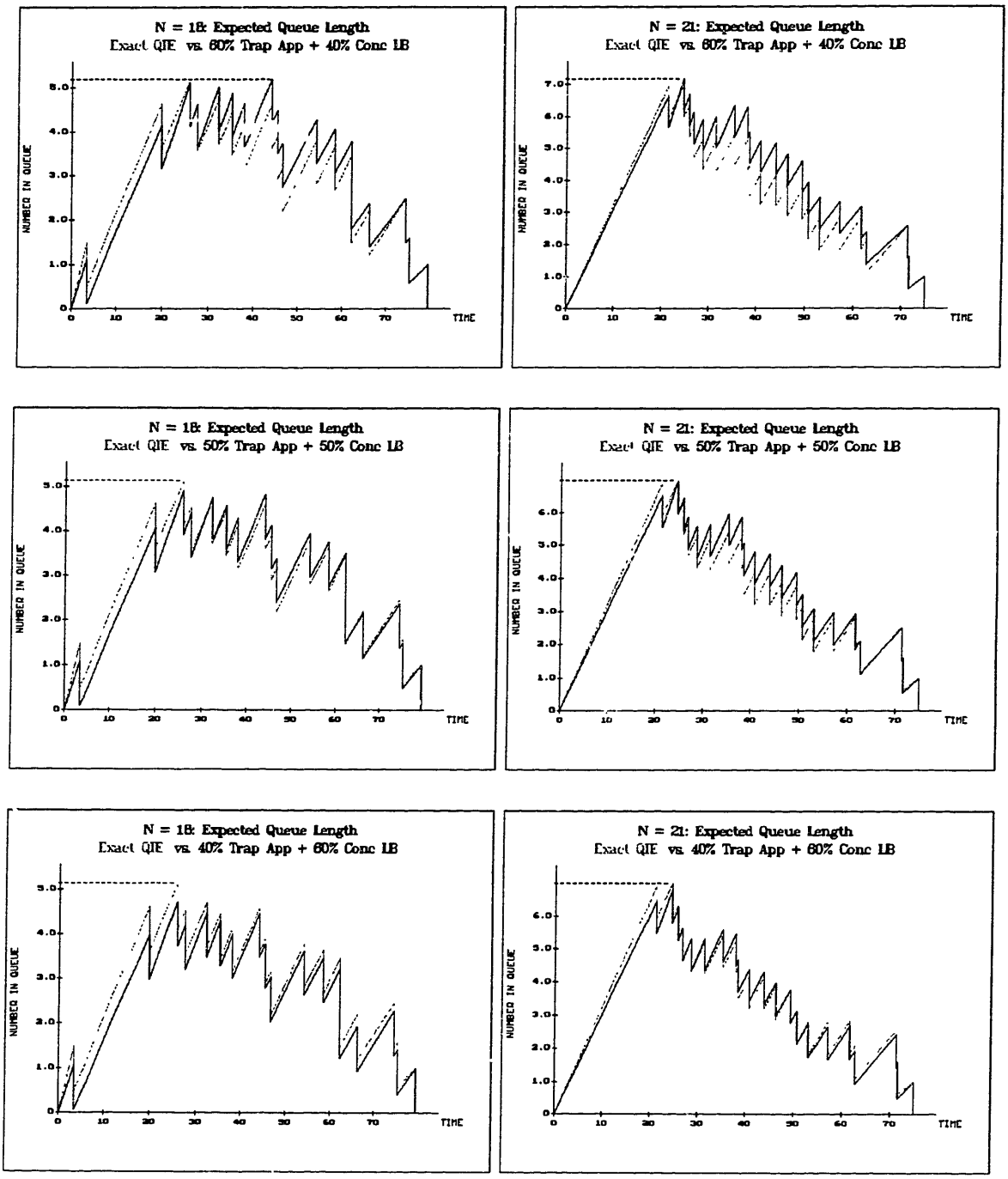


Figure 4.9: Expected Queue Length for Congestion Periods of 18 and 21 Customers: Exact QIE vs. Trapezoidal App/Concavity LB Combinations

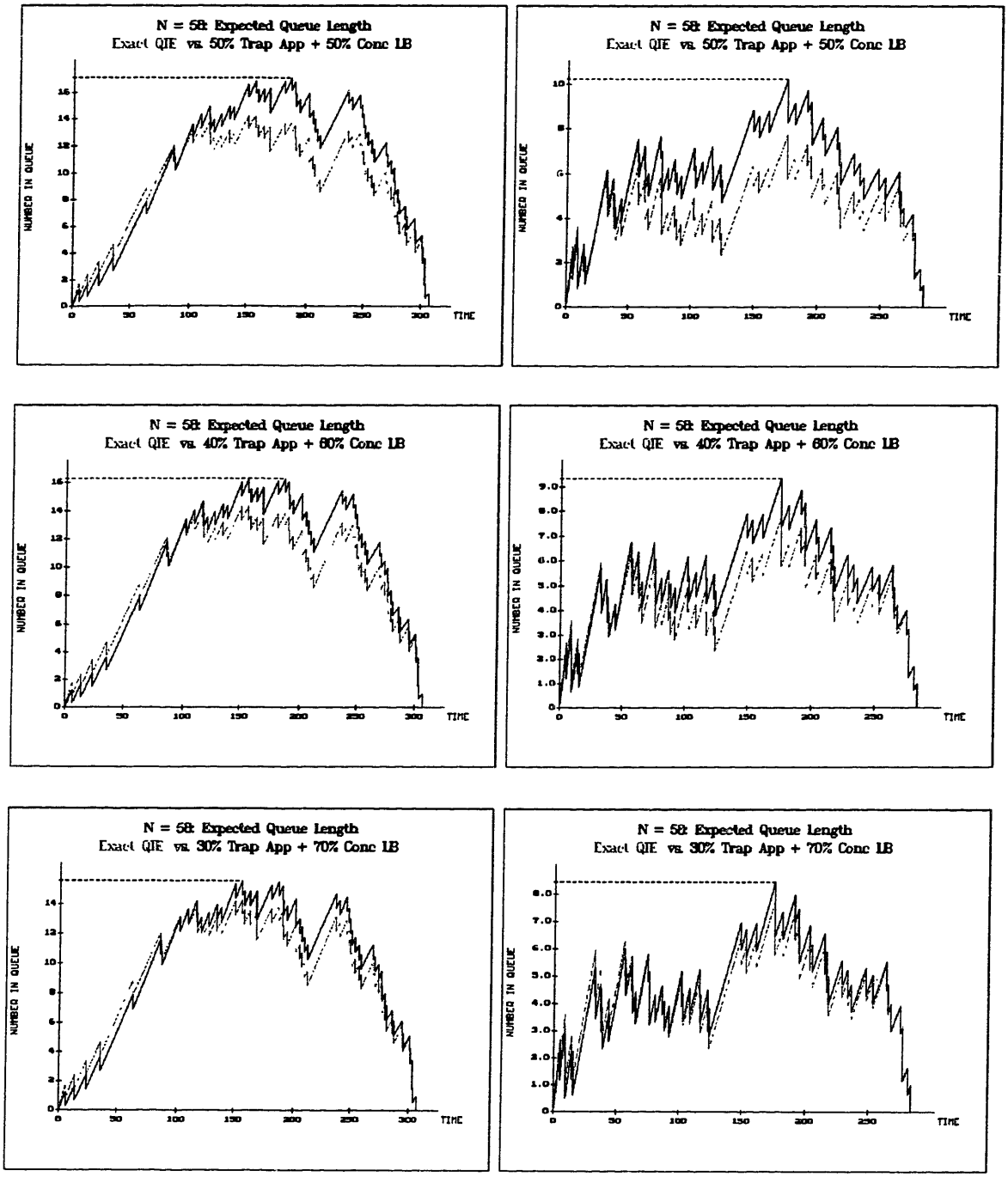


Figure 4.10: Expected Queue Length for Two Congestion Periods of 58 Customers: Exact QIE vs. Trapezoidal App/Concavity LB Combinations

combination increases with the size of the congestion period. The specific manner by which this occurs is one area which needs to be investigated in a methodical manner.

The queue statistics for these four congestion periods, under the original QIE algorithm and the mixture algorithms, are presented in Tables 4.4 and 4.5. Note that we do not present run times here, since the run times will simply be the sum of the times for the individual algorithms plus a nominal amount of time to add the two values together with the appropriate weighting: i.e., the run times are still very short. The improvement in approximating the QIE is quite dramatic over any of the algorithms alone, but work needs to be done to determine a reliable technique for choosing the best ratio as a function of the size of the congestion period.

Size of Cong. Period	Data Used	E[L_Q|...]	E[W_Q|...] (minutes)	δ
$N = 18$	Orig. QIE	2.8649	12.6644	0
	50% UUB + 50% CLB	3.5812	15.8304	1.0669
	40% UUB + 60% CLB	3.2274	14.2664	0.4167
	30% UUB + 70% CLB	2.8735	12.7024	0.2172
$N = 21$	Orig. QIE	3.3871	12.0615	0
	50% UUB + 50% CLB	4.4043	15.6840	1.0197
	40% UUB + 60% CLB	4.0104	14.2812	0.6467
	30% UUB + 70% CLB	3.6165	12.8783	0.3714
$N = 58$ (1)	Orig. QIE	9.3605	49.5175	0
	50% UUB + 50% CLB	13.7476	72.7254	4.3871
	30% UUB + 70% CLB	11.7140	61.9673	2.3543
	20% UUB + 80% CLB	10.6971	56.5882	1.3475
$N = 58$ (2)	Orig. QIE	4.4114	21.5511	0
	50% UUB + 50% CLB	8.2650	40.3772	3.8536
	30% UUB + 70% CLB	5.9511	29.0730	1.5435
	20% UUB + 80% CLB	4.7941	23.4209	0.4278

Table 4.4: Comparison of QIE and Various Mixtures of the Uniform Upper Bound (UUB) and the Concavity Lower Bound (CLB) Algorithms, for Congestion Periods with $N = 18$, 21, and 58

Size of Cong. Period	Data Used	E[L_Q|...]	E[W_Q|...] (minutes)	δ
$N = 18$	Orig. QIE	2.8649	12.6644	0
	60% TRA + 40% CLB	2.9573	13.0725	0.3276
	50% TRA + 50% CLB	2.7664	12.2288	0.8212
	40% TRA + 60% CLB	2.5756	11.3851	0.2894
$N = 21$	Orig. QIE	3.3871	12.0615	0
	60% TRA + 40% CLB	3.7340	13.2970	0.4400
	50% TRA + 50% CLB	3.5175	12.5258	0.2631
	40% TRA + 60% CLB	3.3009	11.7546	0.1617
$N = 58$ (1)	Orig. QIE	9.3605	49.5175	0
	50% TRA + 50% CLB	10.6501	56.3395	1.7393
	40% TRA + 60% CLB	10.2528	54.2376	1.3616
	30% TRA + 70% CLB	9.8555	52.1357	0.9942
$N = 58$ (2)	Orig. QIE	4.4114	21.5511	0
	50% TRA + 50% CLB	5.9585	29.1091	1.5826
	40% TRA + 60% CLB	5.2628	25.7106	0.9350
	30% TRA + 70% CLB	4.5672	22.3122	0.4043

Table 4.5: Comparison of QIE and Various Mixtures of the Trapezoidal Approximation (TRA) and the Concavity Lower Bound (CLB) Algorithms, for Congestion Periods with $N = 18$, 21, and 58

Chapter 5

Generalizing the Set of Conditioning Inequalities

The basic ideas in this chapter were motivated by the consideration of large congestion periods. The ideas presented explore the consequences of changing the set of conditioning events on which the $\beta_{ki}(\mathbf{t})$'s are calculated. A general theorem is presented and proved, which leads to bounds on the quantity $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$. Finally, a general algorithm, to find the $\beta_{ki}(\mathbf{t})$'s under any set of conditioning events, is presented. In the next chapter, this algorithm is specialized in a couple of ways to take advantage of the special structure of the modified set of conditioning events. These specialized algorithms result in some significant computational savings over the original QIE algorithm, although they are not as fast as the algorithms presented in the last chapter.

5.1 Motivation for Changing $\mathcal{E}^S(\mathbf{t})$

Note that in the calculation of the $\beta_{ki}(\mathbf{t})$ values, we condition on all of the arrival-time inequalities given by $\mathcal{E}^S(\mathbf{t})$. But if $N = 100$, and we are calculating $\beta_{75,60}(\mathbf{t})$ (the probability that X_{75} occurs before t_{60}), it is unlikely that $\beta_{75,60}(\mathbf{t})$ will be much affected

by the part of $\mathcal{E}^S(\mathbf{t})$ given by $X_1 \leq t_1$ (otherwise denoted by $1 \leq N(t_1) \leq 100$). Hence, the idea arises that perhaps a good approximation to the $\beta_{ki}(\mathbf{t})$ values may be found by considering only the parts of $\mathcal{E}^S(\mathbf{t})$ that are likely to have a significant impact on $\beta_{ki}(\mathbf{t})$. One question that arises is, what does omission of some of the arrival-time inequalities do to the expected cumulative number of arrivals by time t? To answer this, consider the example given above. If we omit the single arrival-time inequality, $X_1 \leq t_1$, then we are allowing X_1 to occur at a later time, which would seem to increase the probabilities that all of the X's occur later, in turn decreasing $\beta_{ki}(\mathbf{t})$ for all k and i values of interest. The decrease in all of these probabilities would mean that there would be a concomitant decrease in the expected cumulative number of arrivals by time t, and hence we speculate that the expected cumulative number of arrivals in this case would be a lower bound to $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$.

Now consider what would happen if we also eliminated the arrival-time inequality, $X_2 \leq t_2$. This allows X_2 to occur later, which again increases the probabilities that all of the other X's occur later, which would decrease $\beta_{ki}(\mathbf{t})$ and the expected cumulative number of arrivals even further than the original reduction due to omitting $X_1 \leq t_1$. Hence, by starting with a minimal number of the original arrival-time inequalities and then adding more and more of them, we get a series of stochastically dominant lower bounds to $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$, where the bound gets better and better as we add more conditioning inequalities. Our general hypothesis here is that omitting any (and any number) of the conditions contained in $\mathcal{E}^S(\mathbf{t})$ (except the two boundary conditions) and then calculating $\mathrm{E}[A(t)|\mathcal{E}^R(\mathbf{t})]$, the expected cumulative number of arrivals conditioned on $\mathcal{E}^R(\mathbf{t})$ (the reduced set of inequalities), gives a lower bound to $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$.

To see what impact reducing the set of inequalities has on our formulation of the set of bounds on the $A(t_i)$'s, we look at a simple example. Say $N = 4$, and we decide to eliminate the inequality $X_2 \leq t_2$ from $\mathcal{E}^S(\mathbf{t})$. Then, our reduced set of inequalities,

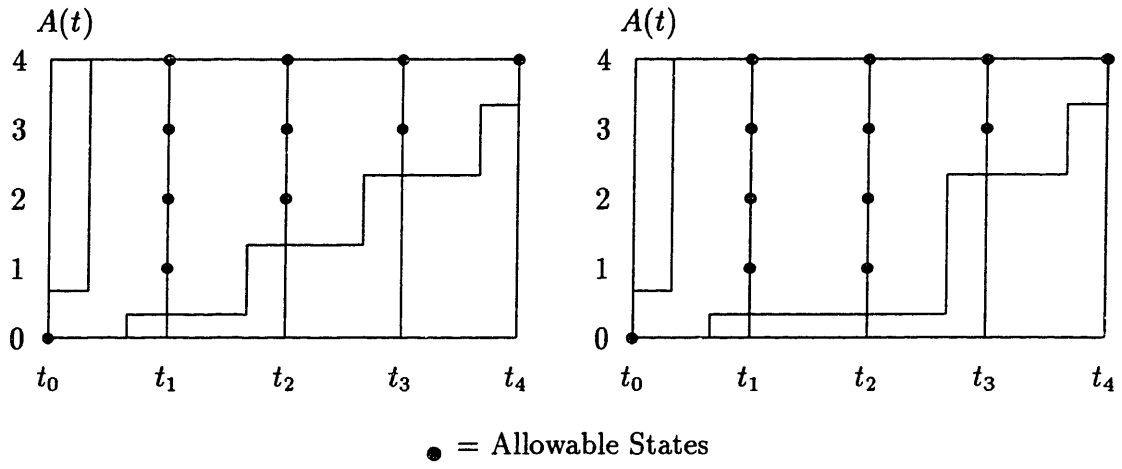


Figure 5.1: Bounds for $\mathcal{E}^S(\mathbf{t})$ (Left) and $\mathcal{E}^R(\mathbf{t})$ (Right: Eliminate $X_2 \leq t_2$)

$\mathcal{E}^R(\mathbf{t})$, would be given by:

$$\begin{aligned} 0 &\leq A(t_0) \leq 0 \\ 1 &\leq A(t_1) \leq 4 \\ 3 &\leq A(t_3) \leq 4 \\ 4 &\leq A(t_4) \leq 4 \end{aligned}$$

But because $A(t)$ is defined as a counting process, we know that $A(t_1) \leq A(t_2) \leq A(t_3)$, which says that there are still implicit bounds on $A(t_2)$, which are given by $1 \leq A(t_2) \leq 4$. In the original $\mathcal{E}^S(\mathbf{t})$, the bounds on $A(t_2)$ were given by $2 \leq A(t_2) \leq 4$, so the elimination of the inequality $X_2 \leq t_2$ has the effect of lowering the lower bound on $A(t_2)$. See Figure 5.1 for a graphical interpretation of the two sets of bounds. Note that by eliminating the inequality $X_2 \leq t_2$, the state $A(t_2) = 1$ is admitted to the set of allowable states.

Now let us consider another way to change the bounds on the $A(t_i)$'s. Say that in our $N = 100$ example, we decide that it is very unlikely that all 100 customers arrived before t_1 (giving a queue length of 100), so we would like not to have to calculate the probability of this (presumably rare) event in our algorithm. What does this assumption do to the bounds on the $A(t_i)$'s and to the expected cumulative number of arrivals by time t? Clearly, we now require $1 \leq A(t_1) \leq 99$, i.e., the upper bound

on $A(t_1)$ has been lowered from 100 to 99. By making this change, we are not allowing X_{100} to occur before t_1, i.e., we are forcing it to occur later, which again increases the probabilities that all of the X's occur later, again decreasing $\beta_{ki}(\mathbf{t})$ for all k and all i. The decrease in all of these probabilities would mean that there would be a concomitant decrease in the expected cumulative number of arrivals by time t: i.e., we hypothesize that eliminating large-queue events from $\mathcal{E}^S(\mathbf{t})$ and then calculating $\mathrm{E}[A(t)|\mathcal{E}^Q(\mathbf{t})]$, the expected cumulative number of arrivals conditioned on the new set of bounds, would give a lower bound to $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$. We look at a simple $N = 4$ example, this time adding the constraint that our maximum queue length may never exceed the value 3 (this is equivalent to lowering the upper bound on $A(t_1)$). So our new set of conditions, $\mathcal{E}^Q(\mathbf{t})$, is given by:

$$\begin{aligned} 0 \leq\ & A(t_0) \leq 0 \\ 1 \leq\ & A(t_1) \leq 3 \\ 2 \leq\ & A(t_2) \leq 4 \\ 3 \leq\ & A(t_3) \leq 4 \\ 4 \leq\ & A(t_4) \leq 4 \end{aligned}$$

See Figure 5.2 for a graphical interpretation of this new set of bounds.

Consider now how these ideas might generalize. We begin with $\mathcal{E}^S(\mathbf{t})$, and then we lower either a lower bound or an upper bound to one of the $A(t_i)$'s. In both cases, we hypothesize the values of $\beta_{ki}(\mathbf{t})$ for all k and i are reduced, and hence the expected cumulative number of arrivals by time t is also reduced, i.e., in both cases a lower bound to $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$ is generated. There was nothing special about starting with the set of bounds given by $\mathcal{E}^S(\mathbf{t})$, however. And we might also speculate that raising either the upper or lower bound on some $A(t_i)$ should give an upper bound to $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$. We now present a general theorem which combines all of these ideas and does indeed give the hypothesized results.

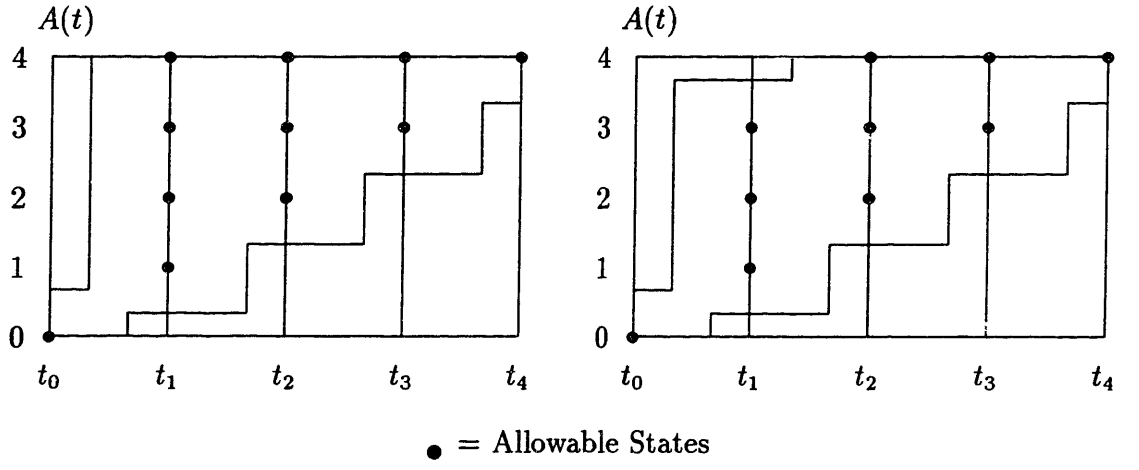


Figure 5.2: Bounds for $\mathcal{E}^S$(t) (Left) and $\mathcal{E}^Q$(t) (Right: Max Queue Length ≤ 3)

5.2 Proof of Stochastic Dominance Theorem

In this section, we present and prove a theorem which validates the general conjectures made in the last section. We begin with some definitions and notation. First, we generalize on the Poisson assumption. Namely, we assume that, given N arrivals in $(0, t_N]$, the times of the unordered arrivals are independent and identically distributed (under Poisson arrivals, the additional assumption of uniformity is made). The distribution function for these arrivals is given by F. Now consider B, any reasonable set of bounds on the $A(t_i)$'s, i.e., a set which does not violate the counting process nature of $A(t)$. Specifically, in terms of Equation 2.4, we require the following:

$$\begin{aligned} l_0 &= u_0 = 0 \\ l_N &= u_N = N \\ l_{i-1} &\leq l_i \qquad i = 1, 2, \ldots, N \\ u_{i-1} &\leq u_i \qquad i = 1, 2, \ldots, N \\ l_i &\leq u_i \qquad i = 0, 1, \ldots, N \end{aligned} \tag{5.1}$$

The first two of these requirements are just the boundary conditions of the process; the next two are direct consequences of the fact that $A(t_{i-1}) \leq A(t_i)$; and the last is a consequence of requiring that there be at least one allowable state for each of

the $A(t_i)$'s. We also adapt another convenience in notation, which will be used in this section only. Since we are only looking at $A(t)$ and $F(t)$ at the values $t = t_i$, we abbreviate $A(t_i)$ by A_i and $F(t_i)$ by F_i, i.e. we have, for the explication and proof of the theorem only:

$$A_i \equiv A(t_i), \quad F_i \equiv F(t_i), \qquad i = 0, 1, \ldots, N$$

Finally, since we are only considering a single t-vector throughout the proof, we omit it in our notation for conditioning events, e.g., instead of $\mathcal{E}^B(\mathbf{t})$, we use $\mathcal{E}^B$.

The general formulation of the theorem is the following:

Theorem 5.1 $\Pr[A_i \geq k | A_j \geq m, \mathcal{E}^B] \;\geq\; \Pr[A_i \geq k | \mathcal{E}^B], \qquad m \leq u_j$

i.e., increasing the minimum number of arrivals that have occurred by time t_j also increases the probability that there have been at least k arrivals by time t_i.

Proof: Note that this theorem is symmetric in i and j: i.e., it may be expressed as:

$$\Pr[A_i \geq k, A_j \geq m | \mathcal{E}^B] \;\geq\; \Pr[A_i \geq k | \mathcal{E}^B] \times \Pr[A_j \geq m | \mathcal{E}^B]$$

In this formulation it is easy to see that it is trivially true when $i = j$, for then the theorem reduces to proving (when $k \geq m$)

$$\Pr[A_i \geq k | \mathcal{E}^B] \;\geq\; \Pr[A_i \geq k | \mathcal{E}^B] \times \Pr[A_i \geq m | \mathcal{E}^B]$$

Because of the symmetry, we may assume, without loss of generality, that $i > j$. We now eliminate some other trivial cases. First, when $k \leq m$, the left-hand side of the theorem is unity, making it trivially true, so we assume $k > m$. Second, when $k \leq l_i$, then both sides of the theorem are unity; and when $k > u_i$, then both sides are zero: hence, we assume $l_i < k \leq u_i$. Finally, both sides of the theorem are equal when $m \leq l_j$ (no new information added), so we assume $l_j < m$. Summarizing all of the assumptions, then, we have:

$$\begin{array}{ccccc} i & > & j & \qquad & k > m \\ l_i & < & k \leq u_i & & l_j < m \leq u_j \end{array}$$

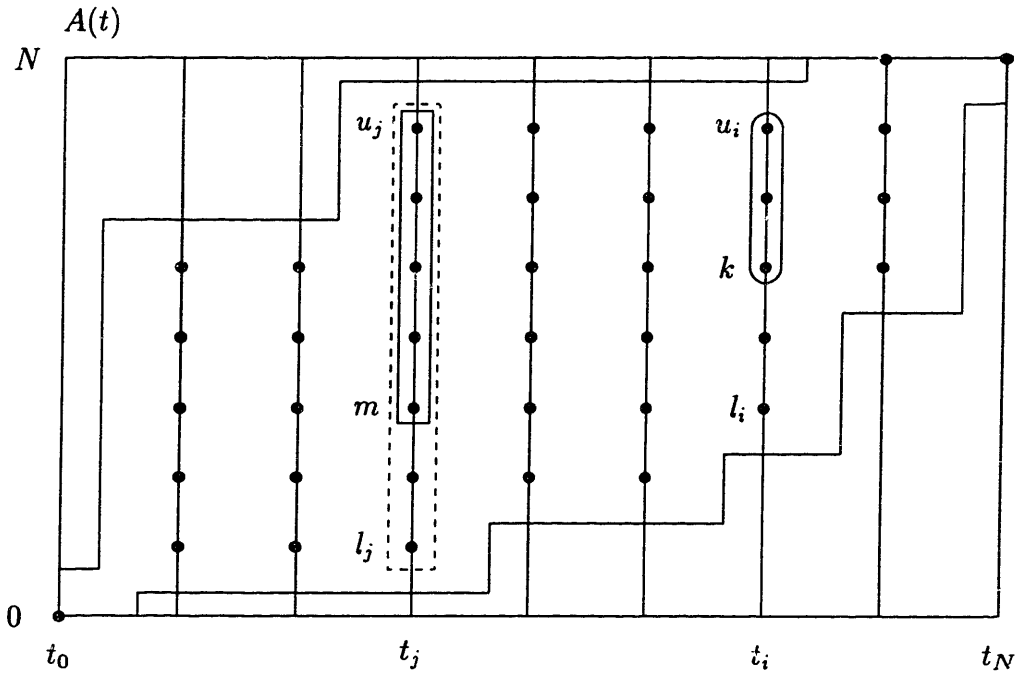


Figure 5.3: Depiction of Theorem 5.1

See Figure 5.3 for a graphical interpretation of the theorem: the probability of entering the circled states is higher, given we start from the boxed states, than it is, given we start from the dashed-boxed states.

We begin the proof with the following Lemma:

Lemma 5.1

$$\Pr[A_a \geq b | A_{a-1} = c, \mathcal{E}^{B \geq a}, \mathcal{E}^{0,N}] \geq \Pr[A_a \geq b | A_{a-1} = c-1, \mathcal{E}^{B \geq a}, \mathcal{E}^{0,N}], \quad c \leq u_a$$

The definition of $\mathcal{E}^{B \geq a}$ is equivalent to that given in Equation 2.7 (although here we omit explicit reference to the t-vector). Figure 5.4 gives a graphical interpretation of the Lemma: we wish to prove that the probability of entering the circled states at time t_a has a higher probability when $A(t_{a-1}) = c$ than when $A(t_{a-1}) = c-1$. Again, we make the following assumptions to avoid trivialities:

$$\begin{aligned} b &> c \\ l_a < b &\leq u_a \\ 0 < c &< u_a \end{aligned}$$

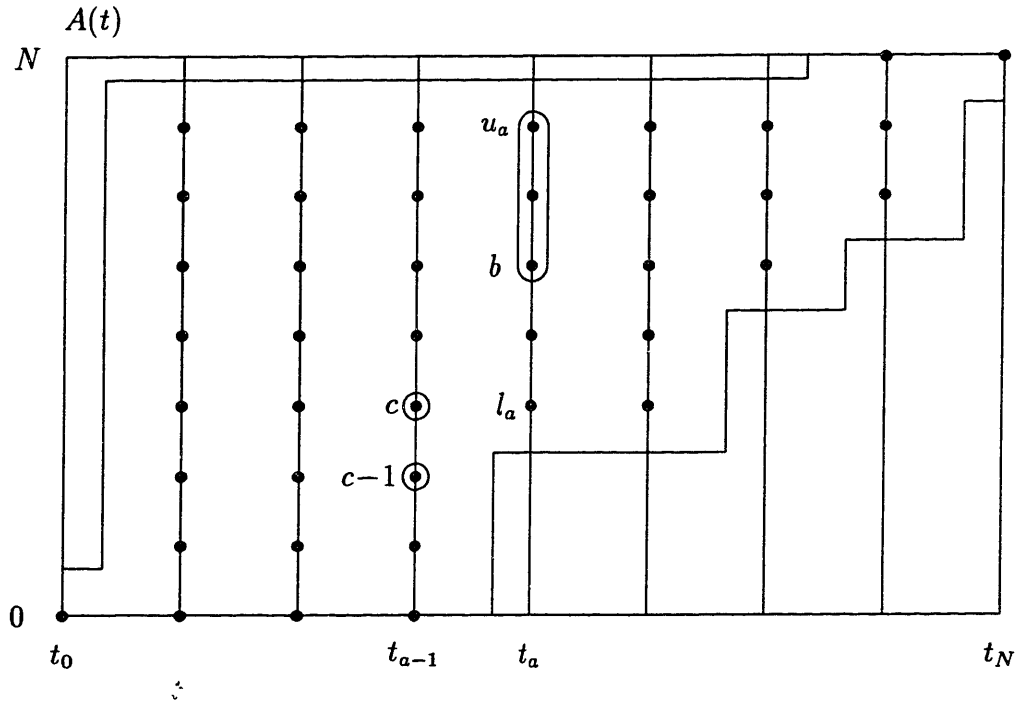


Figure 5.4: Sample Bounds for Lemma 5.1

We may expand the left-hand side of the lemma as follows, just using the definition of conditional probability:

$$\frac{\Pr[A_{a-1}=c, A_a \geq b, \mathcal{E}^{B\geq a}|\mathcal{E}^{0,N}]}{\Pr[A_{a-1}=c, \mathcal{E}^{B\geq a}|\mathcal{E}^{0,N}]} =$$

$$\frac{\Pr[A_{a-1}=c, A_a \geq b, \mathcal{E}^{B\geq a}|\mathcal{E}^{0,N}]}{\Pr[A_{a-1}=c, A_a \geq b, \mathcal{E}^{B\geq a}|\mathcal{E}^{0,N}] + \Pr[A_{a-1}=c, A_a < b, \mathcal{E}^{B\geq a}|\mathcal{E}^{0,N}]}$$

In a similar manner, we expand the right-hand side of the lemma to get:

$$\frac{\Pr[A_{a-1}=c-1, A_a \geq b, \mathcal{E}^{B\geq a}|\mathcal{E}^{0,N}]}{\Pr[A_{a-1}=c-1, A_a \geq b, \mathcal{E}^{B\geq a}|\mathcal{E}^{0,N}] + \Pr[A_{a-1}=c-1, A_a < b, \mathcal{E}^{B\geq a}|\mathcal{E}^{0,N}]}$$

We now make the following definitions:

$$W \equiv \Pr[A_{a-1}=c, A_a \geq b, \mathcal{E}^{B\geq a}|\mathcal{E}^{0,N}]$$

$$X \equiv \Pr[A_{a-1}=c-1, A_a < b, \mathcal{E}^{B\geq a}|\mathcal{E}^{0,N}]$$

$$Y \equiv \Pr[A_{a-1}=c, A_a < b, \mathcal{E}^{B\geq a}|\mathcal{E}^{0,N}]$$

$$Z \equiv \Pr[A_{a-1}=c-1, A_a \geq b, \mathcal{E}^{B\geq a}|\mathcal{E}^{0,N}]$$

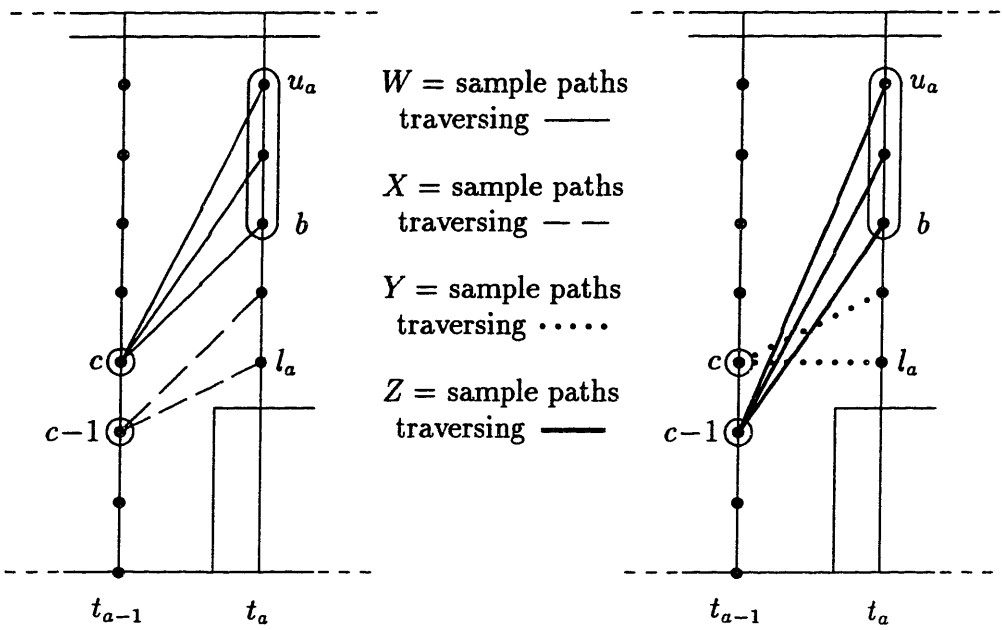


Figure 5.5: Depiction of Sample Paths Comprising W, X, Y, and Z

See Figure 5.5 for a depiction of the set of sample paths that each represents. W is the probability of all sample paths which enter state c at time t_{a-1}, then enter a state with index at least equal to b at time t_a and then conform to the barriers given by $\mathcal{E}^{B\geq a}$ for times greater than t_a and less than t_N. X, Y, and Z are defined similarly. Substituting these in, we see that the Lemma may be restated:

$$\begin{aligned} & \frac{W}{W+Y} \geq \frac{Z}{Z+X} \\ \iff \quad & W \times X \geq Y \times Z \end{aligned}$$

Each of these four probabilities may be expressed as a sum of multinomial probabilities, as follows:

$$\begin{aligned} W &= \Pr[A_{a-1} = c,\ A_a \geq b,\ \mathcal{E}^{B\geq a} | \mathcal{E}^{0,N}] \\ &= \sum_{\Gamma_W} \Pr[A_{a-1} = c, A_a = \gamma_a, A_{a+1} = \gamma_{a+1} \ldots, A_{N-1} = \gamma_{N-1}, A_N = N | \mathcal{E}^{0,N}] \\ &= \sum_{\Gamma_W} \frac{N!}{c!(\gamma_a - c)!(\gamma_{a+1} - \gamma_a)! \ldots (N - \gamma_{N-1})!} \\ &\quad \times F_{a-1}^c (F_a - F_{a-1})^{(\gamma_a - c)} (F_{a+1} - F_a)^{(\gamma_{a+1} - \gamma_a)} \ldots (F_N - F_{N-1})^{(N - \gamma_{N-1})} \end{aligned}$$

where Γ_W is the set of all γ's $(\gamma_a, \gamma_{a+1} \ldots, \gamma_{N-1})$ which satisfy the basic definition of a counting process, i.e., we have $c \leq \gamma_a \leq \gamma_{a+1} \leq \cdots \leq \gamma_{N-1} \leq N$, and which also satisfy:

$$\begin{array}{rcccl} b & \leq & \gamma_a & \leq & u_a \\ l_{a+1} & \leq & \gamma_{a+1} & \leq & u_{a+1} \\ & & \vdots & & \\ l_{N-1} & \leq & \gamma_{N-1} & \leq & u_{N-1} \end{array}$$

Note that X, Y, and Z may be expressed similarly.

When we multiply W by X, or Y by Z, we get a sum of terms, each of which is the product of two multinomial terms similar to the above. We claim that, for any term in $Y \times Z$, consisting of the product of one term from Y and one from Z, we may find a *unique* term in $W \times X$, consisting of the product of one term from W and one term from X, such that

$$\text{term from } W \times X \;\geq\; \text{term from } Y \times Z$$

Specifically, let $(A_{a-1}, A_a, \ldots, A_{N-1}, A_N)$ represent any point in the sample space (where we always have $A_N = N$). Now say that our term from $Y \times Z$ is given by the product of

$$\Pr[c, y_a, y_{a+1}, \ldots, y_{N-1}, N | \mathcal{E}^{0,N}] \;\times\; \Pr[c-1, z_a, z_{a+1}, \ldots, z_{N-1}, N | \mathcal{E}^{0,N}]$$

where we know $c \leq y_a < b$, $z_a \geq b$, and, of course, the y's and z's are non-decreasing with their indices. Then we choose our term from $W \times X$ to be the following product:

$$\begin{array}{rl} & \Pr[c, w_a = z_a, w_{a+1} = z_{a+1}, \ldots, w_{N-1} = z_{N-1}, N | \mathcal{E}^{0,N}] \\ \times & \Pr[c-1, x_a = y_a, x_{a+1} = y_{a+1}, \ldots, x_{N-1} = y_{N-1}, N | \mathcal{E}^{0,N}] \end{array}$$

We know the first term exists in W, since $w_a = z_a \geq b > c$ and the z's are non-decreasing. Similarly, the second term exists in X, since $c - 1 < c \leq y_a = x_a < b$ and the y's are non-decreasing. We would like to show the following:

Claim 5.1

$$\begin{aligned} & \Pr[c, z_a, z_{a+1}, \ldots, z_{N-1}, N|\mathcal{E}^{0,N}] \times \Pr[c-1, y_a, y_{a+1}, \ldots, y_{N-1}, N|\mathcal{E}^{0,N}] \\ \geq\ & \Pr[c, y_a, y_{a+1}, \ldots, y_{N-1}, N|\mathcal{E}^{0,N}] \times \Pr[c-1, z_a, z_{a+1}, \ldots, z_{N-1}, N|\mathcal{E}^{0,N}] \end{aligned}$$

We expand each of the above into its multinomial form to find that the claim is true if and only if

$$\begin{aligned} & \frac{N!}{c!(z_a-c)!(z_{a+1}-z_a)!\ldots(N-z_{N-1})!} \\ & \times F_{a-1}^{c}(F_a-F_{a-1})^{(z_a-c)}\ldots(F_N-F_{N-1})^{(N-z_{N-1})} \\ \times\ & \frac{N!}{(c-1)!(y_a-c+1)!(y_{a+1}-y_a)!\ldots(N-y_{N-1})!} \\ & \times F_{a-1}^{(c-1)}(F_a-F_{a-1})^{(y_a-c+1)}\ldots(F_N-F_{N-1})^{(N-y_{N-1})} \\ \stackrel{?}{\geq}\ & \frac{N!}{c!(y_a-c)!(y_{a+1}-y_a)!\ldots(N-y_{N-1})!} \\ & \times F_{a-1}^{c}(F_a-F_{a-1})^{(y_a-c)}\ldots(F_N-F_{N-1})^{(N-y_{N-1})} \\ \times\ & \frac{N!}{(c-1)!(z_a-c+1)!(z_{a+1}-z_a)!\ldots(N-z_{N-1})!} \\ & \times F_{a-1}^{(c-1)}(F_a-F_{a-1})^{(z_a-c+1)}\ldots(F_N-F_{N-1})^{(N-z_{N-1})} \end{aligned}$$

Fortunately, all of the F terms and most of the factorial terms cancel. After cancellation, we find that the claim is true if and only if:

$$\begin{aligned} \frac{1}{(z_a-c)!(y_a-c+1)!} & \stackrel{?}{\geq} \frac{1}{(y_a-c)!(z_a-c+1)!} \\ \iff \frac{(z_a-c+1)!}{(z_a-c)!} & \stackrel{?}{\geq} \frac{(y_a-c+1)!}{(y_a-c)!} \\ \iff z_a-c+1 & \stackrel{?}{\geq} y_a-c+1 \\ \iff z_a & \stackrel{?}{\geq} y_a \end{aligned}$$

But since $z_a \geq b$ and $y_a < b$, the above is true, and hence the claim is true. Since for each term in $Y \times Z$ we can find a unique corresponding term in $W \times X$, then

$$Y \times Z = \sum \text{all } Y \times Z \text{ terms} \leq \sum \text{all corresponding } W \times X \text{ terms} \leq W \times X$$

where the last inequality follows since there may be additional terms in $W \times X$ that have not been accounted for. ∎ (Lemma 5.1)

Now we proceed to prove the following:

Lemma 5.2

$$\Pr[A_a \geq b | A_{a-n} = c, \mathcal{E}^{B \geq a-n+1}, \mathcal{E}^{0,N}] \geq \Pr[A_a \geq b | A_{a-n} = c-1, \mathcal{E}^{B \geq a-n+1}, \mathcal{E}^{0,N}],$$
$$c \leq u_{a-n+1}$$

Again, to avoid trivialities, we assume the following:

$$\begin{aligned} b &> c \\ l_a < b &\leq u_a \\ 0 < c &\leq \min(u_a - 1, u_{a-n+1}) \end{aligned}$$

Figure 5.6 gives a graphical interpretation of the Lemma: we wish to prove that the probability of entering the circled states at time t_a has a higher probability when $A(t_{a-n}) = c$ than when $A(t_{a-n}) = c-1$. The proof is by induction on n, where the $n = 1$ case was proved in Lemma 5.1. Hence, we assume the Lemma to be true for n and prove that this implies its truth for $n+1$.

First we expand the left-hand side of the Lemma for $n+1$:

$$\begin{aligned} & \Pr[A_a \geq b | A_{a-n-1} = c, \mathcal{E}^{B \geq a-n}, \mathcal{E}^{0,N}] \\ = & \sum_{d=l_{a-n}}^{u_{a-n}} \Pr[A_a \geq b, A_{a-n} = d | A_{a-n-1} = c, \mathcal{E}^{B \geq a-n}, \mathcal{E}^{0,N}] \\ = & \sum_{d=l_{a-n}}^{u_{a-n}} \Big\{ \Pr[A_a \geq b | A_{a-n} = d, \mathcal{E}^{B \geq a-n+1}, \mathcal{E}^{0,N}] \\ & \times \Pr[A_{a-n} = d | A_{a-n-1} = c, \mathcal{E}^{B \geq a-n}, \mathcal{E}^{0,N}] \Big\} \end{aligned} \tag{5.2}$$

Notice that the first term in the last sum above does not need to be conditioned on the value of A_{a-n-1}, since the process is Markovian. Also note that:

$$\{A_{a-n} = d\} \bigcap \mathcal{E}^{B \geq a-n} = \{A_{a-n} = d\} \bigcap \mathcal{E}^{B \geq a-n+1}$$

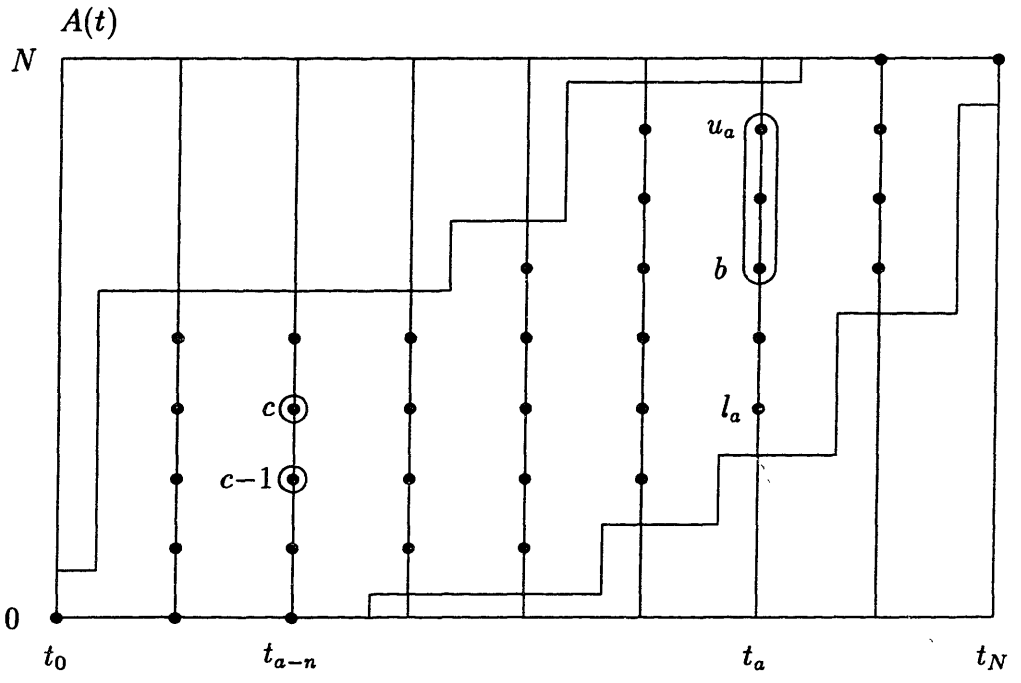


Figure 5.6: Sample Bounds for Lemma 5.2

Now we rewrite expression 5.2 as follows:

$$\begin{aligned}
&\sum_{d=l_{a-n}}^{u_{a-n}} \Big\{ \Pr[A_a \geq b | A_{a-n} = d, \mathcal{E}^{B \geq a-n+1}, \mathcal{E}^{0,N}] \\
&- \Pr[A_a \geq b | A_{a-n} = d-1, \mathcal{E}^{B \geq a-n+1}, \mathcal{E}^{0,N}] \Big\} \\
&\quad \times \Pr[A_{a-n} \geq d | A_{a-n-1} = c, \mathcal{E}^{B \geq a-n}, \mathcal{E}^{0,N}] \qquad (5.3)
\end{aligned}$$

Note the inequality in the last term. Also note that we have defined

$$\Pr[A_a \geq b | A_{a-n} = l_{a-n} - 1, \mathcal{E}^{B \geq a-n+1}, \mathcal{E}^{0,N}] \equiv 0$$

We know from Lemma 5.1 that

$$\begin{aligned}
&\Pr[A_{a-n} \geq d | A_{a-n-1} = c, \mathcal{E}^{B \geq a-n}, \mathcal{E}^{0,N}] \\
\geq\ &\Pr[A_{a-n} \geq d | A_{a-n-1} = c-1, \mathcal{E}^{B \geq a-n}, \mathcal{E}^{0,N}] \qquad (5.4)
\end{aligned}$$

We also know from the induction hypothesis that the difference term in braces in

expression 5.3 must be nonnegative. Combining these results we have that

$$\begin{aligned}
\text{Expression 5.3} \;\geq\; & \sum_{d=l_{a-n}}^{u_{a-n}} \Big\{ \Pr[A_a \geq b | A_{a-n} = d, \mathcal{E}^{B \geq a-n+1}, \mathcal{E}^{0,N}] \\
& - \Pr[A_a \geq b | A_{a-n} = d-1, \mathcal{E}^{B \geq a-n+1}, \mathcal{E}^{0,N}] \Big\} \\
& \times \Pr[A_{a-n} \geq d | A_{a-n-1} = c-1, \mathcal{E}^{B \geq a-n}, \mathcal{E}^{0,N}] \\
= \; & \sum_{d=l_{a-n}}^{u_{a-n}} \Big\{ \Pr[A_a \geq b | A_{a-n} = d, \mathcal{E}^{B \geq a-n+1}, \mathcal{E}^{0,N}] \\
& \times \Pr[A_{a-n} = d | A_{a-n-1} = c-1, \mathcal{E}^{B \geq a-n}, \mathcal{E}^{0,N}] \Big\} \\
= \; & \Pr[A_a \geq b | A_{a-n-1} = c-1, \mathcal{E}^{B \geq a-n}, \mathcal{E}^{0,N}]
\end{aligned}$$

which proves Lemma 5.2. ∎ (Lemma 5.2)

Now we can prove Theorem 5.1. From Lemma 5.2 we know that

$$\Pr[A_i \geq k | A_j = c, \mathcal{E}^{B \geq j+1}, \mathcal{E}^{0,N}] \geq \Pr[A_i \geq k | A_j = c-1, \mathcal{E}^{B \geq j+1}, \mathcal{E}^{0,N}]$$

Since the value of A_j is exactly specified, and the process is Markovian, we also have that:

$$\begin{aligned}
\Pr[A_i \geq k | A_j = c-1, \mathcal{E}^{B \geq j+1}, \mathcal{E}^{0,N}] &= \Pr[A_i \geq k | A_j = c-1, \mathcal{E}^{B}] \\
\Pr[A_i \geq k | A_j = c, \mathcal{E}^{B \geq j+1}, \mathcal{E}^{0,N}] &= \Pr[A_i \geq k | A_j = c, \mathcal{E}^{B}]
\end{aligned}$$

Combining the above two results, we have that:

$$\Pr[A_i \geq k | A_j = c, \mathcal{E}^B] \;\geq\; \Pr[A_i \geq k | A_j = c-1, \mathcal{E}^B] \tag{5.5}$$

$$\Longrightarrow \Pr[A_i \geq k | A_j = c, \mathcal{E}^B] \;\geq\; \Pr[A_i \geq k | A_j = b, \mathcal{E}^B], \quad c \geq b \tag{5.6}$$

since we can just apply Equation 5.5 recursively to get Equation 5.6. We would like to prove that:

$$\begin{aligned}
& \Pr[A_i \geq k | A_j \geq m, \mathcal{E}^B] \;\overset{?}{\geq}\; \Pr[A_i \geq k | \mathcal{E}^B] \\
\Longleftrightarrow & \sum_{b=m}^{u_j} \Pr[A_i \geq k, A_j = b | A_j \geq m, \mathcal{E}^B] \;\overset{?}{\geq}\; \sum_{b=l_j}^{u_j} \Pr[A_i \geq k, A_j = b | \mathcal{E}^B] \\
\Longleftrightarrow & \sum_{b=m}^{u_j} \frac{\Pr[A_i \geq k | A_j = b, \mathcal{E}^B] \Pr[A_j = b | \mathcal{E}^B]}{\Pr[A_j \geq m | \mathcal{E}^B]} \;\overset{?}{\geq}\; \sum_{b=l_j}^{u_j} \Pr[A_i \geq k | A_j = b, \mathcal{E}^B] \Pr[A_j = b | \mathcal{E}^B]
\end{aligned}$$

Since the denominator on the left-hand side above is not a function of b, we may bring it over to the right. We also multiply the left-hand side by unity, in the form of $\Pr[A_j \geq m|\mathcal{E}^B] + \Pr[A_j < m|\mathcal{E}^B]$, so we now want to show that:

$$\begin{aligned}
& \sum_{b=m}^{u_j} \Pr[A_i \geq k|A_j = b, \mathcal{E}^B] \Pr[A_j = b|\mathcal{E}^B] \left\{\Pr[A_j \geq m|\mathcal{E}^B] + \Pr[A_j < m|\mathcal{E}^B]\right\} \\
\stackrel{?}{\geq} & \sum_{b=l_j}^{u_j} \Pr[A_i \geq k|A_j = b, \mathcal{E}^B] \Pr[A_j = b|\mathcal{E}^B] \Pr[A_j \geq m|\mathcal{E}^B] \\
\iff & \sum_{b=m}^{u_j} \Pr[A_i \geq k|A_j = b, \mathcal{E}^B] \Pr[A_j = b|\mathcal{E}^B] \Pr[A_j < m|\mathcal{E}^B] \\
\stackrel{?}{\geq} & \sum_{b=l_j}^{m-1} \Pr[A_i \geq k|A_j = b, \mathcal{E}^B] \Pr[A_j = b|\mathcal{E}^B] \Pr[A_j \geq m|\mathcal{E}^B] \\
\iff & \sum_{b=m}^{u_j} \sum_{c=l_j}^{m-1} \Pr[A_i \geq k|A_j = b, \mathcal{E}^B] \Pr[A_j = b|\mathcal{E}^B] \Pr[A_j = c|\mathcal{E}^B] \\
\stackrel{?}{\geq} & \sum_{b=l_j}^{m-1} \sum_{c=m}^{u_j} \Pr[A_i \geq k|A_j = b, \mathcal{E}^B] \Pr[A_j = b|\mathcal{E}^B] \Pr[A_j = c|\mathcal{E}^B]
\end{aligned}$$

If we compare the two sides of the above inequality, we see that the limits on the sums are the same, but reversed with respect to b and c. Hence, we now simply change indices on the left-hand side so that what was b becomes c and vice versa. We also reverse the order of summation, so that we now want to show that:

$$\begin{aligned}
& \sum_{b=l_j}^{m-1} \sum_{c=m}^{u_j} \Pr[A_i \geq k|A_j = c, \mathcal{E}^B] \Pr[A_j = c|\mathcal{E}^B] \Pr[A_j = b|\mathcal{E}^B] \\
\stackrel{?}{\geq} & \sum_{b=l_j}^{m-1} \sum_{c=m}^{u_j} \Pr[A_i \geq k|A_j = b, \mathcal{E}^B] \Pr[A_j = b|\mathcal{E}^B] \Pr[A_j = c|\mathcal{E}^B]
\end{aligned}$$

But now it is easy to show term-by-term dominance: i.e., to show that term (b, c) on the left-hand side is greater than or equal to term (b, c) on the right-hand side:

$$\begin{aligned}
& \Pr[A_i \geq k|A_j = c, \mathcal{E}^B] \Pr[A_j = c|\mathcal{E}^B] \Pr[A_j = b|\mathcal{E}^B] \\
\stackrel{?}{\geq} & \Pr[A_i \geq k|A_j = b, \mathcal{E}^B] \Pr[A_j = b|\mathcal{E}^B] \Pr[A_j = c|\mathcal{E}^B] \\
\iff & \Pr[A_i \geq k|A_j = c, \mathcal{E}^B] \stackrel{?}{\geq} \Pr[A_i \geq k|A_j = b, \mathcal{E}^B]
\end{aligned}$$

But we know $c \geq m$ and $b \leq m - 1$ so that we have $c > b$. Hence, by Equation 5.6, we know that the above is true. But since we have term-by-term dominance, the sum

also dominates (identical ranges for b and c on both sides), and hence the theorem is true! ∎ (Theorem 5.1)

Note the implications of this theorem. First, as the theorem is stated, decreasing the lower bound on some $A(t_j)$ decreases the values of $\beta_{ki}(\mathbf{t})$, while increasing the lower bound increases the $\beta_{ki}(\mathbf{t})$'s. Similarly, we may rearrange the statement of the theorem as follows:

$$\begin{aligned}
& \Pr[A_i \geq k | A_j \geq m, \mathcal{E}^B] \geq \Pr[A_i \geq k | \mathcal{E}^B] \\
\iff & 1 - \Pr[A_i \geq k | A_j \geq m, \mathcal{E}^B] \leq 1 - \Pr[A_i \geq k | \mathcal{E}^B] \\
\iff & \Pr[A_i \leq k | A_j \geq m, \mathcal{E}^B] \leq \Pr[A_i \leq k | \mathcal{E}^B] \\
\iff & \Pr[A_i \leq k, A_j \geq m | \mathcal{E}^B] \leq \Pr[A_i \leq k | \mathcal{E}^B] \times \Pr[A_j \geq m | \mathcal{E}^B] \\
\iff & \Pr[A_j \geq m | A_i \leq k, \mathcal{E}^B] \leq \Pr[A_j \geq m | \mathcal{E}^B]
\end{aligned}$$

This tells us that decreasing the upper bound on some $A(t_j)$ will also decrease the values of $\beta_{ki}(\mathbf{t})$, while increasing the upper bound increases the $\beta_{ki}(\mathbf{t})$'s. In all of these cases, whenever all of the $\beta_{ki}(\mathbf{t})$ values are decreased (increased), we are also decreasing (increasing) the expected cumulative number of arrivals function, and hence we are finding a lower (upper) bound to $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$. In the next section we find an algorithm to determine the arrival-time probabilities under any general set of bounds. These probabilities may then be used to find upper and lower bounds to $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$, as just described.

5.3 An Algorithm for Finding Arrival-Time Probabilities Under General Bounds

We have introduced the notion of a set of general bounds, B, on the $A(t_i)$'s. The question arises, is there an easy algorithm to find quantities comparable to the $\beta_{ki}(\mathbf{t})$'s, i.e. is there an easy way to find the quantity

$$\beta_{ki}^B(\mathbf{t}) \equiv \Pr[A(t_i) \geq k | \mathcal{E}^B(\mathbf{t})], \quad k = 1, 2, \ldots, N, \quad i = 1, 2, \ldots, N$$

(where we have made explicit the set of bounds that the probability is conditioned on)? The answer is yes, and the derivation runs parallel to that given in Chapter 2 for the recursion algorithm to find the $\beta_{ki}(\mathbf{t})$'s. Here, we present the general form and again generalize to I.I.D. arrivals on the interval $(0, t_N]$, rather than restricting the derivation to the Poisson case. We also maintain our notation that F represents the distribution function for the unordered arrivals and that F_i represents the probability that an arrival occurred prior to t_i.

Note that

$$\beta^B_{ki}(\mathbf{t}) = \begin{cases} 1, & k = 1, 2, \ldots, l_i, \\ 0, & k = u_i+1, u_i+2, \ldots, N, \end{cases} \qquad i = 1, 2, \ldots, N$$

We now describe the method for calculating the other values for $\beta^B_{ki}(\mathbf{t})$. First, we define $\mathcal{E}^{B\leq i}(\mathbf{t})$ and $\mathcal{E}^{B\geq i}(\mathbf{t})$ similarly to the definitions of $\mathcal{E}^{S\leq i}(\mathbf{t})$ and $\mathcal{E}^{S\geq i}(\mathbf{t})$ in Equations 2.6 and 2.7. Now we may begin:

$$\begin{aligned} \beta^B_{ki}(\mathbf{t}) &= \Pr[A(t_i) \geq k | \mathcal{E}^B(\mathbf{t})] \\ &= \Pr[A(t_i) \geq k+1 | \mathcal{E}^B(\mathbf{t})] + \Pr[A(t_i) = k | \mathcal{E}^B(\mathbf{t})] \end{aligned}$$

Recognizing the first term above as $\beta^B_{(k+1),i}(\mathbf{t})$ when $k < u_i$ and zero when $k = u_i$, we get that:

$$\beta^B_{ki}(\mathbf{t}) = \begin{cases} \beta^B_{(k+1),i}(\mathbf{t}) + \Pr[A(t_i) = k | \mathcal{E}^B(\mathbf{t})], & k = l_i+1, l_i+2, \ldots, u_i-1, \\ \Pr[A(t_i) = k | \mathcal{E}^B(\mathbf{t})], & k = u_i, \end{cases}$$
$$i = 1, 2, \ldots, N-1$$

So clearly the term of interest to calculate is $\Pr[A(t_i) = k | \mathcal{E}^B(\mathbf{t})]$, which we do as follows:

$$\begin{aligned} \Pr[A(t_i) = k | \mathcal{E}^B(\mathbf{t})] &= \frac{\Pr[\mathcal{E}^B(\mathbf{t}) | A(t_i) = k, \mathcal{E}^{0,N}(\mathbf{t})] \times \Pr[A(t_i) = k | \mathcal{E}^{0,N}(\mathbf{t})]}{\Pr[\mathcal{E}^B(\mathbf{t}) | \mathcal{E}^{0,N}(\mathbf{t})]} \\ &= \frac{1}{\Pr[\mathcal{E}^B(\mathbf{t}) | \mathcal{E}^{0,N}(\mathbf{t})]} \Big\{ \Pr[\mathcal{E}^{B\leq i}(\mathbf{t}) | A(t_i) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\ &\quad \times \Pr[\mathcal{E}^{B\geq i}(\mathbf{t}) | A(t_i) = k, \mathcal{E}^{0,N}(\mathbf{t})] \times \Pr[A(t_i) = k | \mathcal{E}^{0,N}(\mathbf{t})] \Big\} \\ &= \frac{1}{\tilde{\alpha}^B_{NN}(\mathbf{t})} \left\{ \tilde{\alpha}^B_{ki}(\mathbf{t}) \times \tilde{\eta}^B_{ki}(\mathbf{t}) \times \binom{N}{k} (F_i)^k (F_N - F_i)^{N-k} \right\} \end{aligned}$$

Note that we may break up $\Pr[\mathcal{E}^B(\mathbf{t})|A(t_i)=k,\mathcal{E}^{0,N}(\mathbf{t})]$ into $\tilde{\alpha}^B_{ki}(\mathbf{t})\times\tilde{\eta}^B_{ki}(\mathbf{t})$, because, given the value of $A(t_i)$, events prior to t_i are conditionally independent of events subsequent to t_i. The last term in the braces above is again due to the fact that the number of arrivals by time t_i, given N arrivals by time t_N, is a binomial random variable with "p" equal to F_i, and "$1-p$" equal to $F_N - F_i$ (where $F_N = 1$). Note that in the case of Poisson arrivals, $F_i = \frac{t_i}{t_N}$ and $F_N - F_i = \frac{t_N - t_i}{t_N}$. We have defined $\tilde{\alpha}^B_{ki}(\mathbf{t})$ and $\tilde{\eta}^B_{ki}(\mathbf{t})$ similarly to the definitions of $\tilde{\alpha}_{ki}(\mathbf{t})$ and $\tilde{\eta}_{ki}(\mathbf{t})$ in Equations 2.8 and 2.9, i.e., we have that:

$$\tilde{\alpha}^B_{ki}(\mathbf{t}) \equiv \Pr[\mathcal{E}^{B\leq i}(\mathbf{t})|A(t_i)=k,\mathcal{E}^{0,N}(\mathbf{t})]$$
$$\tilde{\eta}^B_{ki}(\mathbf{t}) \equiv \Pr[\mathcal{E}^{B\geq i}(\mathbf{t})|A(t_i)=k,\mathcal{E}^{0,N}(\mathbf{t})]$$

We also have that:

$$\begin{aligned}\Pr[\mathcal{E}^B(\mathbf{t})|\mathcal{E}^{0,N}(\mathbf{t})] &= \Pr[\mathcal{E}^{B\leq N}(\mathbf{t})|\mathcal{E}^{0,N}(\mathbf{t})] = \tilde{\alpha}^B_{NN}(\mathbf{t})\\ &= \Pr[\mathcal{E}^{B\geq 0}(\mathbf{t})|\mathcal{E}^{0,N}(\mathbf{t})] = \tilde{\eta}^B_{00}(\mathbf{t})\end{aligned}$$

The next task is to determine the values in the $\tilde{\alpha}^B$-matrix for $k=0,1,\ldots,N$ and for $i=1,2,\ldots,N$. First, it should be obvious that:

$$\begin{aligned}\tilde{\alpha}^B_{0i}(\mathbf{t}) &= \Pr[\mathcal{E}^{B\leq 1}(\mathbf{t})|A(t_i)=0,\mathcal{E}^{0,N}(\mathbf{t})] = 1, \quad \text{if } l_i=0,\ i=1,2,\ldots,N-1\\ \tilde{\alpha}^B_{k1}(\mathbf{t}) &= \Pr[\mathcal{E}^{B\leq 1}(\mathbf{t})|A(t_1)=k,\mathcal{E}^{0,N}(\mathbf{t})] = 1, \quad k=l_1,l_1+1,\ldots,u_1\\ \tilde{\alpha}^B_{ki}(\mathbf{t}) &= 0, \quad k=0,1,\ldots,l_i-1 \text{ or } k=u_i+1,u_i+2,\ldots,N,\\ &\quad i=1,2,\ldots,N-1\end{aligned} \tag{5.7}$$

We also define $\tilde{\alpha}^B_{kN}(\mathbf{t}) \equiv 0$ for $k=0,1,\ldots,N-1$. Now consider the following:

$$\begin{aligned}\tilde{\alpha}^B_{ki}(\mathbf{t}) &= \Pr[\mathcal{E}^{B\leq i}(\mathbf{t})|A(t_i)=k,\mathcal{E}^{0,N}(\mathbf{t})], \quad k=l_i,l_i+1,\ldots,u_i \text{ and } k>0\\ &= \sum_{j=\max(0,k-u_{i-1})}^{k-l_{i-1}} \Pr[\mathcal{E}^{B\leq i-1}(\mathbf{t}), l_i\leq A(t_i)\leq u_i, A(t_{i-1},t_i)=j|A(t_i)=k,\mathcal{E}^{0,N}(\mathbf{t})]\end{aligned}$$

Here, as in Chapter 2, j represents the number of arrivals which may occur between t_{i-1} and t_i, while still conforming to the bounds given by $\mathcal{E}^B(\mathbf{t})$. Of course, we would

get the greatest possible number of arrivals if $A(t_{i-1})$ were as small as possible, i.e., if $A(t_{i-1}) = l_{i-1}$. In this case, the number of arrivals in $(t_{i-1}, t_i]$ would be $k - l_{i-1}$: hence, the upper limit on the sum above. To understand the lower limit, we know that we must have a non-negative number of arrivals in $(t_{i-1}, t_i]$; however, when $u_{i-1} < k$, then $A(t_{i-1})$ can be no larger than u_{i-1}, and therefore, the number of arrivals in $(t_{i-1}, t_i]$ can be no smaller than $k - u_{i-1}$: hence, the lower limit on the sum. We continue:

$$\begin{aligned}
\tilde{\alpha}^B_{ki}(\mathbf{t}) &= \sum_{j=\max(0,k-u_{i-1})}^{k-l_{i-1}} \Pr[\mathcal{E}^{B\leq i-1}(\mathbf{t}) | A(t_{i-1}, t_i) = j, A(t_i) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\
&\quad \times \Pr[A(t_{i-1}, t_i) = j | A(t_i) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\
&= \sum_{j=\max(0,k-u_{i-1})}^{k-l_{i-1}} \Pr[\mathcal{E}^{B\leq i-1}(\mathbf{t}) | A(t_{i-1}) = k - j, \mathcal{E}^{0,N}(\mathbf{t})] \\
&\quad \times \Pr[A(t_{i-1}, t_i) = j | A(t_i) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\
&= \sum_{j=\max(0,k-u_{i-1})}^{k-l_{i-1}} \tilde{\alpha}^B_{(k-j),(i-1)}(\mathbf{t}) \times \binom{k}{j} \left(\frac{F_{i-1}}{F_i}\right)^{k-j} \left(\frac{F_i - F_{i-1}}{F_i}\right)^j, \\
&\quad k = l_i, l_i+1, \ldots, u_i \text{ and } k > 0, \quad i = 2, 3, \ldots, N
\end{aligned} \tag{5.8}$$

Here, $\frac{F_{i-1}}{F_i}$ is the probability that one of the arrivals occurs prior to t_{i-1}, given that it occurs prior to t_i. So we first fill in the first column and zero-th row of the matrix with ones and zeroes; and we fill in other zeroes as indicated by Equation 5.7. Then we proceed to the second column, and the third column, etc., each time calculating the unknown values using the values from the previous column.

Now let us compare Equation 5.8 to Equation 2.10, where our set of bounds, B, is given by S (see Equation 2.5). First consider the limits on the sum: since $k \leq N$ and $u_i = N, \quad i = 1, 2, \ldots, N$, then the bottom limit of zero makes sense. Similarly, we have $l_i = i, \quad i = 1, 2, \ldots, N$, so $l_{i-1} = i - 1$, and the upper limit on the sum should be $k - (i-1) = k - i + 1$, which it is. Also, F_i here corresponds to $\frac{t_i}{t_N}$ in the Poisson case. Finally, the values of k and i for which the recursion equation is valid also correspond, since $l_i = i$ and $u_i = N$, in this special case.

Next, we must determine the values in the $\tilde{\eta}^B$-matrix for $i = 0, 1, \ldots, N-1$ and

for $k = 0, 1, \ldots, N$. Again, it should be obvious that:

$$\begin{aligned}
\tilde{\eta}^B_{Ni}(\mathbf{t}) &= \Pr[\mathcal{E}^{B\geq i}(\mathbf{t})|A(t_i) = N, \mathcal{E}^{0,N}(\mathbf{t})] = 1, \text{ if } u_i = N, \ i = 1, 2, \ldots, N-1 \\
\tilde{\eta}^B_{k,(N-1)}(\mathbf{t}) &= \Pr[\mathcal{E}^{B\geq N-1}(\mathbf{t})|A(t_{N-1}) = k, \mathcal{E}^{0,N}(\mathbf{t})] = 1, \ k = l_{N-1}, l_{N-1}+1, \ldots, u_{N-1} \\
\tilde{\eta}^B_{ki}(\mathbf{t}) &= 0, \quad k = 0, 1, \ldots, l_i - 1 \text{ or } k = u_i+1, u_i+2, \ldots, N, \\
&\quad i = 1, 2, \ldots, N-1
\end{aligned} \tag{5.9}$$

We also define $\tilde{\eta}^B_{k0}(\mathbf{t}) \equiv 0$ for $k = 1, 2, \ldots, N$. Now consider the following:

$$\begin{aligned}
\tilde{\eta}^B_{ki}(\mathbf{t}) &= \Pr[\mathcal{E}^{B\geq i}(\mathbf{t})|A(t_i) = k, \mathcal{E}^{0,N}(\mathbf{t})], \quad k = l_i, l_i+1, \ldots, u_i \text{ and } k < N \\
&= \sum_{j=\max(0,l_{i+1}-k)}^{u_{i+1}-k} \Pr[\mathcal{E}^{B\geq i+1}(\mathbf{t}), l_i \leq A(t_i) \leq u_i, A(t_i, t_{i+1}) = j | A(t_i) = k, \mathcal{E}^{0,N}(\mathbf{t})]
\end{aligned}$$

Again, as in Chapter 2, j represents the number of arrivals which may occur between t_i and t_{i+1}, while still conforming to the bounds given by $\mathcal{E}^B(\mathbf{t})$. Of course, we would get the greatest possible number of arrivals if $A(t_{i+1})$ were as large as possible, i.e., if $A(t_{i+1}) = u_{i+1}$. In this case, the number of arrivals in $(t_i, t_{i+1}]$ would be $u_{i+1} - k$: hence, the upper limit on the sum above. To understand the lower limit, we know that we must have a non-negative number of arrivals in $(t_i, t_{i+1}]$; however, when $l_{i+1} > k$, then $A(t_{i+1})$ can be no smaller than l_{i+1}, and hence, the number of arrivals in $(t_i, t_{i+1}]$ can be no smaller than $l_{i+1} - k$: hence, the lower limit on the sum. We continue:

$$\begin{aligned}
\tilde{\eta}^B_{ki}(\mathbf{t}) &= \sum_{j=\max(0,l_{i+1}-k)}^{u_{i+1}-k} \Pr[\mathcal{E}^{B\geq i+1}(\mathbf{t})|A(t_i, t_{i+1}) = j, A(t_i) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\
&\quad \times \Pr[A(t_i, t_{i+1}) = j | A(t_i) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\
&= \sum_{j=\max(0,l_{i+1}-k)}^{u_{i+1}-k} \Pr[\mathcal{E}^{B\geq i+1}(\mathbf{t})|A(t_{i+1}) = k + j, \mathcal{E}^{0,N}(\mathbf{t})] \\
&\quad \times \Pr[A(t_i, t_{i+1}) = j | A(t_i) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\
&= \sum_{j=\max(0,l_{i+1}-k)}^{u_{i+1}-k} \tilde{\eta}^B_{(k+j),(i+1)}(\mathbf{t}) \times \binom{N-k}{j} \left(\frac{F_{i+1}-F_i}{F_N-F_i}\right)^j \left(\frac{F_N-F_{i+1}}{F_N-F_i}\right)^{N-k-j}, \\
&\quad k = l_i, l_i+1, \ldots, u_i \text{ and } k < N, \ i = 0, 1, \ldots, N-2
\end{aligned} \tag{5.10}$$

Again, we have used fractional forms of the distribution function to represent conditional probabilities of arrivals. So we first fill in the last column and bottom row of

the matrix with ones and zeroes; and we fill in other zeroes as indicated by Equation 5.9. Then we proceed to the second-to-last column, and the third-to-last column, etc., each time calculating the unknown values using the values from the column to the right.

Now let us compare Equation 5.10 to Equation 2.11, where our set of bounds, B, is again given by S. First consider the limits on the sum: since $u_i = N,\ \ i = 1, 2, \ldots, N$, then the upper limit of $N - k$ makes sense. Similarly, we have $l_i = i,\ \ i = 1, 2, \ldots, N$, so $l_{i+1} = i + 1$. We also know that $k \geq i$, so as long as $k > i$, then we will have $l_{i+1} - k \leq 0$, so the lower limit of zero makes sense here. When $k = i$, then the lower limit of the sum in Equation 2.11 should be 1, since it is not possible to have zero arrivals in $(t_i, t_{i+1}]$ in this case. However, since $\tilde{\eta}^B_{i,i+1}(\mathbf{t})$ has already been defined to be zero, we may start the sum at $j = 0$: hence the lower limit. Again, the values of k and i for which the recursion equation is valid also correspond, since $l_i = i$ and $u_i = N$, in this special case.

We now present the following definitions, which make all of the above equations simpler and are comparable to the equations given in [Lars 90]:

$$\begin{aligned} \alpha^B_{ki}(\mathbf{t}) &\equiv \tilde{\alpha}^B_{ki}(\mathbf{t}) \times [F(t_i)]^k \\ \eta^B_{ki}(\mathbf{t}) &\equiv \tilde{\eta}^B_{ki}(\mathbf{t}) \times [F(t_N) - F(t_i)]^{N-k} \\ &= \tilde{\eta}^B_{ki}(\mathbf{t}) \times [1 - F(t_i)]^{N-k} \end{aligned}$$

where, for clarity, we have expanded our notation for the distribution function for the unordered arrivals. With these definitions, we have the following for the definitions of the α^B-matrix:

$$\alpha^B_{0i}(\mathbf{t}) = 1, \quad \text{if } l_i = 0,\ \ i = 1, 2, \ldots, N-1 \tag{5.11}$$

$$\alpha^B_{k1}(\mathbf{t}) = [F(t_1)]^k, \quad k = l_1, l_1+1, \ldots, u_1 \tag{5.12}$$

$$\begin{aligned} \alpha^B_{ki}(\mathbf{t}) &= 0, \quad k = 0, 1, \ldots, l_i - 1 \text{ or } k = u_i+1, u_i+2, \ldots, N, \\ &\qquad i = 1, 2, \ldots, N \end{aligned} \tag{5.13}$$

$$\alpha^B_{ki}(\mathbf{t}) = \sum_{j=\max(0,k-u_{i-1})}^{k-l_{i-1}} \alpha^B_{(k-j),(i-1)}(\mathbf{t}) \times \binom{k}{j} [F(t_i) - F(t_{i-1})]^j ,$$
$$k = l_i, l_i+1, \ldots, u_i \text{ and } k > 0, \quad i = 2, 3, \ldots, N \tag{5.14}$$

The η^B-matrix is defined by the following:

$$\eta^B_{Ni}(\mathbf{t}) = 1, \quad \text{if } u_i = N, \ i = 1, 2, \ldots, N-1 \tag{5.15}$$

$$\eta^B_{k,(N-1)}(\mathbf{t}) = [1 - F(t_{N-1})]^{N-k}, \quad k = l_{N-1}, l_{N-1}+1, \ldots, u_{N-1} \tag{5.16}$$

$$\eta^B_{ki}(\mathbf{t}) = 0, \quad k = 0, 1, \ldots, l_i - 1 \text{ or } k = u_i+1, u_i+2, \ldots, N,$$
$$i = 0, 1, \ldots, N-1 \tag{5.17}$$

$$\eta^B_{ki}(\mathbf{t}) = \sum_{j=\max(0,l_{i+1}-k)}^{u_{i+1}-k} \eta^B_{(k+j),(i+1)}(\mathbf{t}) \times \binom{N-k}{j} [F(t_{i+1}) - F(t_i)]^j ,$$
$$k = l_i, l_i+1, \ldots, u_i \text{ and } k < N, \quad i = 0, 1, \ldots, N-2 \tag{5.18}$$

Finally, we have for the β^B-matrix:

$$\beta^B_{ki}(\mathbf{t}) = 1, \quad k = 1, 2, \ldots, l_i, \ i = 1, 2, \ldots, N \tag{5.19}$$

$$\beta^B_{ki}(\mathbf{t}) = 0, \quad k = u_i+1, u_i+2, \ldots, N, \ i = 1, 2, \ldots, N-1 \tag{5.20}$$

$$\beta^B_{Ni}(\mathbf{t}) = \Pr[A(t_i) = N | \mathcal{E}^B(\mathbf{t})]$$
$$= \begin{cases} \dfrac{\tilde{\alpha}^B_{Ni}(\mathbf{t})\tilde{\eta}^B_{Ni}(\mathbf{t})[F(t_i)]^N}{\tilde{\alpha}^B_{NN}(\mathbf{t})} = \dfrac{\alpha^B_{Ni}(\mathbf{t})}{\alpha^B_{NN}(\mathbf{t})}, & \text{if } u_i = N \\ 0, & \text{if } u_i < N \end{cases}$$
$$i = 1, 2, \ldots, N-1 \tag{5.21}$$

$$\beta^B_{ki}(\mathbf{t}) = \beta^B_{(k+1),i}(\mathbf{t}) + \frac{1}{\alpha^B_{NN}(\mathbf{t})} \left\{ \binom{N}{k} \alpha^B_{ki}(\mathbf{t})\eta^B_{ki}(\mathbf{t}) \right\},$$
$$k = l_i+1, l_i+2, \ldots, u_i \text{ and } k < N, \ i = 1, 2, \ldots, N-1 \tag{5.22}$$

Hence, we begin by filling in the ones and zeroes as indicated; then we calculate the bottom row of the matrix; finally we calculate each column by a multiplication of elements from the α^B-matrix and the η^B-matrix, which we then add to the element of the β^B-matrix just below the one being calculated.

Of course, in order to calculate $\mathrm{E}[A(t_i)|\mathcal{E}^B(\mathbf{t})]$, we merely add up all of the values in column i of the β^B-matrix, just as in the case of the standard QIE. Similarly, all

of the statistics corresponding to those generated by the standard QIE algorithm are calculated in exactly the same manner as described at the end of Chapter 2. However, in order to calculate the expected cumulative number of arrivals to the system under this general set of bounds at intermediate values of t, $\mathrm{E}[A(t)|\mathcal{E}^B(\mathbf{t})]$, we cannot simply linearly interpolate between the values of the function at the t_i's, as in the Poisson case. Instead, we must use the following:

$$\begin{aligned}\mathrm{E}[A(t)|\mathcal{E}^B(\mathbf{t})] &= \mathrm{E}[A(t_{i-1})|\mathcal{E}^B(\mathbf{t})] \\ &\quad + \left\{\mathrm{E}[A(t_i)|\mathcal{E}^B(\mathbf{t})] - \mathrm{E}[A(t_{i-1})|\mathcal{E}^B(\mathbf{t})]\right\} \frac{F(t) - F(t_{i-1})}{F(t_i) - F(t_{i-1})} \\ &\quad t_{i-1} < t \leq t_i, \quad i = 1, 2, \ldots, N\end{aligned}$$

The validity of this expression may be seen by conditioning on the number of arrivals at t_{i-1} and t_i, using the independence of the arrivals, and then just weighting and adding.

This general algorithm would appear to be very useful, but its efficacy in reducing runtimes relative to the original QIE algorithm, when used in the manners suggested earlier in this chapter, is not evident. In the next chapter, we specialize this algorithm in the two cases of using a reduced set of conditioning inequalities and of imposing a maximum queue length. These algorithms have special structures which allow a simpler calculation of their beta-matrices and hence of the expected cumulative number of arrivals by time t.

Chapter 6

Two Lower Bound Algorithms Based on Changing the Conditioning Information

In this chapter, we explore two specific algorithms, based on the general algorithm in Chapter 5. Both algorithms assume a return to the assumption of Poisson arrivals that was abandoned in the last chapter. In the first section, we consider reducing the set of conditioning inequalities. When this is done, it is possible to construct matrices similar to the α- and η-matrices, but which only consider the times at which the conditioning inequalities apply. Hence, the matrices so constructed have fewer columns and require less computation. In the second section, we provide some results of sample runs of this algorithm, considering its accuracy and its runtimes. In the third section, we consider adding maximum queue length information to the set of conditioning inequalities. This has the effect of allowing calculation of all of the matrices of interest, without calculation of the large-queue events: thus, the lower left-hand corner of these matrices are zeroed out, again reducing the amount of required computation. Finally, we provide results of sample runs of this second algorithm that again examine both its accuracy and the reduction in computation that is achieved.

6.1 Algorithm Based on Reducing the Set of Conditioning Inequalities

The notion of reducing the set of conditioning inequalities upon which the β-matrices are based was first introduced in Chapter 5. The basic idea is that perhaps not all of these inequalities are necessary in order to get a good estimate of what the queue length is actually doing. We showed in Chapter 5 that reducing the set of inequalities leads to a lower bound on the expected cumulative number of arrivals by any given time. We now proceed to show the specifics of implementing this lower-bound algorithm and investigate its computational complexity. The first step is to demonstrate how to find $\mathrm{E}[A(t)|\mathcal{E}^R(\mathrm{t})]$, the expected cumulative number of arrivals by time t, conditioned on the reduced set of inequalities, R.

First, we need a few definitions. Say that, in terms of the ordered arrival times, $X_1, X_2, \ldots, X_N$, we decide that the following conditions will comprise the set that we consider:

$$\begin{aligned} X_{I_1} &\leq I_1 \\ X_{I_2} &\leq I_2 \\ &\vdots \\ X_{I_C} = X_N &\leq I_C = N \end{aligned}$$

So, I_m is the index of the m-th arrival time inequality in our set of C conditions. Note that we will always include $X_N \leq N$ in our set, since this is one of the boundary conditions of the process. Now it is easy to translate this set of conditions into a comparable set in terms of the $A(t_i)$'s (here, we also define $I_0 \equiv 0$):

$$\begin{aligned} I_0 = 0 \quad &\leq \quad A(t_0) \quad \leq \quad 0 \\ I_0 = 0 \quad &\leq \quad A(t_1) \quad \leq \quad N \\ &\vdots \end{aligned}$$

$$
\begin{array}{rcccl}
I_0 = 0 & \leq & A(t_{I_1-1}) & \leq & N \\
I_1 & \leq & A(t_{I_1}) & \leq & N \\
I_1 & \leq & A(t_{I_1+1}) & \leq & N \\
 & & \vdots & & \\
I_1 & \leq & A(t_{I_2-1}) & \leq & N \\
I_2 & \leq & A(t_{I_2}) & \leq & N \\
I_2 & \leq & A(t_{I_2+1}) & \leq & N \\
 & & \vdots & & \\
I_{C-2} & \leq & A(t_{I_{C-1}-1}) & \leq & N \\
I_{C-1} & \leq & A(t_{I_{C-1}}) & \leq & N \\
I_{C-1} & \leq & A(t_{I_{C-1}+1}) & \leq & N \\
 & & \vdots & & \\
I_{C-1} & \leq & A(t_{N-1}) & \leq & N \\
I_C = N & \leq & A(t_N) & \leq & N
\end{array}
$$

We let R represent the set of bounds for any such reduced set of inequalities, so that we have for R:

$$
\begin{aligned}
l_{I_m} = l_{I_m+1} = \cdots = l_{I_{m+1}-1} &= I_m, \\
&\quad m = 0, 1, \ldots, C-1 \\
l_N = l_{I_C} &= I_C = N \\
u_0 &= 0 \\
u_1 = u_2 = \cdots = u_N &= N
\end{aligned}
$$

Now that we have defined all of these quantities, we could go ahead and compute full α^R-, η^R-, and β^R-matrices, as described in Section 5.3. But this would not provide us with any computational savings. Instead, we calculate these quantities only at the t_{I_m}'s, $m = 1, 2, \ldots, C$, and claim that, from these values, we may calculate the entire

function, $\mathrm{E}[A(t)|\mathcal{E}^R(\mathbf{t})]$. First, it is certainly possible to calculate the I_m-th column of the β^R-matrix from the comparable columns of the α^R- and η^R-matrices, simply by using Equations 5.19 through 5.22. Now we must show that it is possible to calculate the α^R- and the η^R-matrices recursively, where we only calculate the values in the columns representing $t_{I_m}, m = 1, 2, \ldots, C$. We proceed to show this by way of the following claims.

Claim 6.1

$$\begin{aligned}\alpha^R_{k,I_1}(\mathbf{t}) &\equiv \Pr[\mathcal{E}^{R\leq I_1}(\mathbf{t})|A(t_{I_1}) = k, \mathcal{E}^{0,N}(\mathbf{t})] \times \left(\frac{t_{I_1}}{t_N}\right)^k \\ &= \begin{cases} 0, & k = 0, 1, \ldots, I_1 - 1 \\ \left(\frac{t_{I_1}}{t_N}\right)^k, & k = I_1, I_1 + 1, \ldots, N \end{cases}\end{aligned}$$

Proof: The first part of the claim comes directly from Equation 5.13 and the fact that $l_{I_1} = I_1$. The second part of the claim results because, when $k \geq I_1$, the probability that all of the bounds on the $A(t_i)$'s, for $i \leq I_1$, are met, is just unity. ∎

Claim 6.2

$$\begin{aligned}\alpha^R_{k,I_m}(\mathbf{t}) &\equiv \Pr[\mathcal{E}^{R\leq I_m}(\mathbf{t})|A(t_{I_m}) = k, \mathcal{E}^{0,N}(\mathbf{t})] \times \left(\frac{t_{I_m}}{t_N}\right)^k \\ &= \begin{cases} 0, & k = 0, 1, \ldots, I_m - 1 \\ \sum_{j=0}^{k-I_{m-1}} \alpha^R_{(k-j),I_{m-1}}(\mathbf{t}) \binom{k}{j} \left(\frac{t_{I_m} - t_{I_{m-1}}}{t_N}\right)^j, & k = I_m, I_m + 1, \ldots, N \end{cases} \\ & m = 2, 3, \ldots, C\end{aligned}$$

Proof: Again, the first part of the claim comes directly from Equation 5.13 and the fact that $l_{I_m} = I_m$. The second part of the claim may be derived as follows:

$$\begin{aligned}\tilde{\alpha}^R_{k,I_m}(\mathbf{t}) &\equiv \Pr[\mathcal{E}^{R\leq I_m}(\mathbf{t})|A(t_{I_m}) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\ &= \sum_{j=0}^{k-I_{m-1}} \Pr[\mathcal{E}^{R\leq I_m}(\mathbf{t}), A(t_{I_{m-1}}, t_{I_m}) = j|A(t_{I_m}) = k, \mathcal{E}^{0,N}(\mathbf{t})], \\ & k = I_m, I_m + 1, \ldots, N, \quad m = 2, 3, \ldots, C\end{aligned}$$

Here, j is the number of arrivals that occur between $t_{I_{m-1}}$ and t_{I_m}. The lower limit on the sum is due to the fact that if $A(t_{I_{m-1}}) = k$, then there are zero arrivals in the interval. The upper limit results because, in order for the probability being summed to be non-zero, it must be the case that $A(t_{I_{m-1}}) \geq l_{I_{m-1}} = I_{m-1}$. This means that the greatest number of arrivals we can have over the interval is $k - I_{m-1}$. Continuing:

$$\begin{aligned}
\tilde{\alpha}^R_{k,I_m}(\mathbf{t}) &= \sum_{j=0}^{k-I_{m-1}} \Pr[\mathcal{E}^{R\leq I_{m-1}}(\mathbf{t}), I_{m-1} \leq A(t_{I_{m-1}+1}) \leq N, \ldots, I_{m-1} \leq A(t_{I_m-1}) \leq N, \\
&\quad I_m \leq A(t_{I_m}) \leq N, A(t_{I_{m-1}}, t_{I_m}) = j | A(t_{I_m}) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\
&= \sum_{j=0}^{k-I_{m-1}} \Pr[I_{m-1} \leq A(t_{I_{m-1}+1}) \leq N, \ldots, I_{m-1} \leq A(t_{I_m-1}) \leq N, \\
&\quad I_m \leq A(t_{I_m}) \leq N | \mathcal{E}^{R\leq I_{m-1}}(\mathbf{t}), A(t_{I_{m-1}}, t_{I_m}) = j, A(t_{I_m}) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\
&\quad \times \Pr[\mathcal{E}^{R\leq I_{m-1}}(\mathbf{t}) | A(t_{I_{m-1}}, t_{I_m}) = j, A(t_{I_m}) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\
&\quad \times \Pr[A(t_{I_{m-1}}, t_{I_m}) = j | A(t_{I_m}) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\
&= \sum_{j=0}^{k-I_{m-1}} 1 \times \tilde{\alpha}^R_{(k-j),I_{m-1}}(\mathbf{t}) \times \binom{k}{j} \left(\frac{t_{I_{m-1}}}{t_{I_m}}\right)^{k-j} \left(\frac{t_{I_m} - t_{I_{m-1}}}{t_{I_m}}\right)^j
\end{aligned}$$

The key here is that there are no new conditions imposed between $t_{I_{m-1}}$ and t_{I_m}, so that the probability that the numbers of arrivals at the intermediate times stay within their bounds, given the number of arrivals at the two endpoints of the interval, is just unity. By using the standard definition that $\alpha^R_{ki}(\mathbf{t}) \equiv \tilde{\alpha}^R_{ki}(\mathbf{t})\left(\frac{t_i}{t_N}\right)^k$, we immediately get the second result of Claim 6.2:

$$\begin{aligned}
\alpha^R_{k,I_m}(\mathbf{t}) &= \sum_{j=0}^{k-I_{m-1}} \alpha^R_{(k-j),I_{m-1}}(\mathbf{t}) \binom{k}{j} \left(\frac{t_{I_m} - t_{I_{m-1}}}{t_N}\right)^j, \\
&\quad k = I_m, I_m + 1, \ldots, N, \quad m = 2, 3, \ldots, N
\end{aligned}$$

and so the claim has been proved. ∎

We continue now with two similar claims regarding the η^R-matrices.

Claim 6.3

$$\eta^R_{k,I_{C-1}}(\mathbf{t}) \equiv \Pr[\mathcal{E}^{R\geq I_{C-1}}(\mathbf{t}) | A(t_{I_{C-1}}) = k, \mathcal{E}^{0,N}(\mathbf{t})] \times \left(\frac{t_N - t_{I_{C-1}}}{t_N}\right)^{N-k}$$

$$= \begin{cases} 0, & k = 0, 1, \ldots, I_{C-1} - 1 \\ \left(\frac{t_N - t_{I_{C-1}}}{t_N}\right)^{N-k}, & k = I_{C-1}, I_{C-1} + 1, \ldots, N \end{cases}$$

Proof: The first part of the claim comes directly from Equation 5.17 and the fact that $l_{I_{C-1}} = I_{C-1}$. The second part of the claim results because, when $k \geq I_{C-1}$, the probability that all of the bounds on the $A(t_i)$'s, for $i \geq I_{C-1}$, are met, is just unity. ∎

Claim 6.4

$$\begin{aligned} \eta^R_{k,I_m}(\mathbf{t}) &\equiv \Pr[\mathcal{E}^{R \geq I_m}(\mathbf{t}) | A(t_{I_m}) = k, \mathcal{E}^{0,N}(\mathbf{t})] \times \left(\frac{t_N - t_{I_m}}{t_N}\right)^{N-k} \\ &= \begin{cases} 0, & k = 0, 1, \ldots, I_m - 1 \\ \sum\limits_{j=\max(0, I_{m+1}-k)}^{N-k} \eta^R_{(k+j),I_{m+1}}(\mathbf{t}) \binom{N-k}{j} \left(\frac{t_{I_{m+1}} - t_{I_m}}{t_N}\right)^j, & \\ & k = I_m, I_m + 1, \ldots, N \end{cases} \\ & m = 1, 2, \ldots, C-2 \end{aligned}$$

Proof: Again, the first part of the claim comes directly from Equation 5.17 and the fact that $l_{I_m} = I_m$. The second part of the claim may be derived as follows:

$$\begin{aligned} \tilde{\eta}^R_{k,I_m}(\mathbf{t}) &\equiv \Pr[\mathcal{E}^{R \geq I_m}(\mathbf{t}) | A(t_{I_m}) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\ &= \sum_{j=\max(0, I_{m+1}-k)}^{N-k} \Pr[\mathcal{E}^{R \geq I_m}(\mathbf{t}), A(t_{I_m}, t_{I_{m+1}}) = j | A(t_{I_m}) = k, \mathcal{E}^{0,N}(\mathbf{t})], \\ & k = I_m, I_m + 1, \ldots, N, \quad m = 1, 2, \ldots, C-2 \end{aligned}$$

Here, j is the number of arrivals that occur between t_{I_m} and $t_{I_{m+1}}$. The upper limit on the sum is due to the fact that if $A(t_{I_{m+1}}) = N$, then there are $N - k$ arrivals in the interval. The lower limit results because, first, there must have been at least zero arrivals in the interval. But in the case that $k < I_{m+1}$, we know that $A(t_{I_{m+1}}) \geq I_{m+1}$, and hence, the number of arrivals in the interval must be at least $I_{m+1} - k$. Continuing:

$$\tilde{\eta}^R_{k,I_m}(\mathbf{t}) = \sum_{j=\max(0, I_{m+1}-k)}^{N-k} \Pr[\mathcal{E}^{R \geq I_{m+1}}(\mathbf{t}), I_m \leq A(t_{I_m}) \leq N, I_m \leq A(t_{I_{m+1}}) \leq N,$$

$$
\begin{aligned}
&\ldots, I_m \le A(t_{I_{m+1}-1}) \le N, A(t_{I_m}, t_{I_{m+1}}) = j | A(t_{I_m}) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\
= &\sum_{j=\max(0,I_{m+1}-k)}^{N-k} \Pr[I_m \le A(t_{I_m}) \le N, I_m \le A(t_{I_{m+1}}) \le N, \ldots, \\
&I_m \le A(t_{I_{m+1}-1}) \le N | \mathcal{E}^{R \ge I_{m+1}}(\mathbf{t}), A(t_{I_m}, t_{I_{m+1}}) = j, A(t_{I_m}) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\
&\times \Pr[\mathcal{E}^{R \ge I_{m+1}}(\mathbf{t}) | A(t_{I_m}, t_{I_{m+1}}) = j, A(t_{I_m}) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\
&\times \Pr[A(t_{I_m}, t_{I_{m+1}}) = j | A(t_{I_m}) = k, \mathcal{E}^{0,N}(\mathbf{t})] \\
= &\sum_{j=\max(0,I_{m+1}-k)}^{N-k} 1 \times \tilde{\eta}^R_{(k+j),I_{m+1}}(\mathbf{t}) \times \binom{N-k}{j} \left(\frac{t_{I_{m+1}} - t_{I_m}}{t_N - t_{I_m}}\right)^j \left(\frac{t_N - t_{I_{m+1}}}{t_N - t_{I_m}}\right)^{N-k-j}
\end{aligned}
$$

The key here is that there are no new conditions imposed between t_{I_m} and $t_{I_{m+1}}$, so that the probability that the numbers of arrivals at the intermediate times stay within their bounds, given the number of arrivals at the two endpoints of the interval, is just unity. By using the standard definition that $\eta^R_{ki}(\mathbf{t}) \equiv \tilde{\eta}^R_{ki}(\mathbf{t})\left(\frac{t_N - t_i}{t_N}\right)^{N-k}$, we immediately get the second result of Claim 6.4:

$$
\begin{aligned}
\eta^R_{k,I_m}(\mathbf{t}) &= \sum_{j=\max(0,I_{m+1}-k)}^{N-k} \eta^R_{(k+j),I_{m+1}}(\mathbf{t}) \binom{N-k}{j} \left(\frac{t_{I_{m+1}} - t_{I_m}}{t_N}\right)^j, \\
&\quad k = I_m, I_m + 1, \ldots, N, \quad m = 1, 2, \ldots, C-2
\end{aligned}
$$

and so the claim has been proved. ∎

We have now seen how to construct limited α^R- and η^R-matrices, calculating only the columns corresponding to $t_{I_m}, m = 1, 2, \ldots, C$. We also pointed out that the corresponding columns of the β^R-matrix may be calculated from these limited matrices, using Equations 5.19, 5.21, and 5.22. That is, we have:

$$
\begin{aligned}
\beta^R_{k,I_m}(\mathbf{t}) &= 1, \quad k = 1, 2, \ldots, I_m, \quad m = 1, 2, \ldots, C \\
\beta^R_{N,I_m}(\mathbf{t}) &= \Pr[A(t_i) = N | \mathcal{E}^R(\mathbf{t})] \\
&= \frac{\tilde{\alpha}^R_{N,I_m}(\mathbf{t}) \tilde{\eta}^R_{N,I_m}(\mathbf{t}) \left(\frac{t_{I_m}}{t_N}\right)^N}{\tilde{\alpha}^R_{NN}(\mathbf{t})} = \frac{\alpha^R_{N,I_m}(\mathbf{t})}{\alpha^R_{NN}(\mathbf{t})}, \\
&\quad m = 1, 2, \ldots, C-1 \\
\beta^R_{k,I_m}(\mathbf{t}) &= \beta^R_{(k+1),I_m}(\mathbf{t}) + \frac{1}{\alpha^R_{NN}(\mathbf{t})} \left\{ \binom{N}{k} \alpha^R_{k,I_m}(\mathbf{t}) \eta^R_{k,I_m}(\mathbf{t}) \right\}, \\
&\quad k = I_m + 1, I_m + 2, \ldots, N-1, \quad m = 1, 2, \ldots, C-1
\end{aligned}
$$

Now it is easy to calculate the values of $\mathrm{E}[A(t_{I_m})|\mathcal{E}^R(\mathbf{t})]$ by using the familiar

$$\begin{aligned}\mathrm{E}[A(t_{I_m})|\mathcal{E}^R(\mathbf{t})] &= \sum_{k=1}^{N} \Pr[A(t_{I_m}) \geq k|\mathcal{E}^R(\mathbf{t})] \\ &= \sum_{k=1}^{N} \beta^R_{k,I_m}(\mathbf{t})\end{aligned}$$

We now make the following claim:

Claim 6.5

$$\begin{aligned}\mathrm{E}[A(t)|\mathcal{E}^R(\mathbf{t})] &= \mathrm{E}[A(t_{I_{m-1}})|\mathcal{E}^R(\mathbf{t})]\frac{t_{I_m}-t}{t_{I_m}-t_{I_{m-1}}} + \mathrm{E}[A(t_{I_m})|\mathcal{E}^R(\mathbf{t})]\frac{t-t_{I_{m-1}}}{t_{I_m}-t_{I_{m-1}}}, \\ & t_{I_{m-1}} < t \leq t_{I_m}, \quad m = 1,2,\ldots,C\end{aligned}$$

i.e. $\mathrm{E}[A(t)|\mathcal{E}^R(\mathbf{t})]$ *is linear when* $t_{I_{m-1}} < t \leq t_{I_m}$ *and* $m = 1,2,\ldots,C$.

Proof: The proof exactly parallels the proof in [Lars 90] of Lemma 3. First, we condition both on the value of $A(t_{I_{m-1}})$ and on the number of arrivals that occurred during the interval $(t_{I_{m-1}}, t_{I_m}]$. That is, say we are given that $A(t_{I_{m-1}}) = n$ and that $A(t_{I_{m-1}}, t_{I_m}) = j$. Then, since there are no conditions imposed on when the j arrivals may occur within the interval, those arrivals are conditionally independent and uniform over the interval, so that the cumulative expected number of arrivals by time t grows linearly with t, i.e., we have:

$$\begin{aligned}\mathrm{E}[A(t)|A(t_{I_{m-1}}) = n, A(t_{I_{m-1}}, t_{I_m}) = j, \mathcal{E}^R(\mathbf{t})] &= n + j\frac{t-t_{I_{m-1}}}{t_{I_m}-t_{I_{m-1}}}, \\ n = I_{m-1}, I_{m-1}+1, \ldots, N-j, \quad j &= 0,1,\ldots,N-I_{m-1}\end{aligned}$$

where the limits on n and j are given to correspond to the bounds on $A(t_{I_{m-1}})$ and on $A(t_{I_m})$. Now we remove the conditioning, first on $A(t_{I_{m-1}})$:

$$\begin{aligned}&\mathrm{E}[A(t)|A(t_{I_{m-1}}, t_{I_m}) = j, \mathcal{E}^R(\mathbf{t})] \\ = &\sum_{n=I_{m-1}}^{N-j} \mathrm{E}[A(t)|A(t_{I_{m-1}}) = n, A(t_{I_{m-1}}, t_{I_m}) = j, \mathcal{E}^R(\mathbf{t})] \\ &\times \Pr[A(t_{I_{m-1}}) = n|A(t_{I_{m-1}}, t_{I_m}) = j, \mathcal{E}^R(\mathbf{t})]\end{aligned}$$

$$= \sum_{n=I_{m-1}}^{N-j} \left\{ n + j\frac{t - t_{I_{m-1}}}{t_{I_m} - t_{I_{m-1}}} \right\} \times \Pr[A(t_{I_{m-1}}) = n | A(t_{I_{m-1}}, t_{I_m}) = j, \mathcal{E}^R(\mathbf{t})]$$

$$= \mathrm{E}[A(t_{I_{m-1}}) | A(t_{I_{m-1}}, t_{I_m}) = j, \mathcal{E}^R(\mathbf{t})] + j\frac{t - t_{I_{m-1}}}{t_{I_m} - t_{I_{m-1}}}$$

Finally we remove the conditioning on $A(t_{I_{m-1}}, t_{I_m})$:

$$\begin{aligned}
\mathrm{E}[A(t)|\mathcal{E}^R(\mathbf{t})] &= \sum_{j=0}^{N-I_{m-1}} \mathrm{E}[A(t)|A(t_{I_{m-1}}, t_{I_m}) = j, \mathcal{E}^R(\mathbf{t})] \times \Pr[A(t_{I_{m-1}}, t_{I_m}) = j|\mathcal{E}^R(\mathbf{t})] \\
&= \sum_{j=0}^{N-I_{m-1}} \left\{ \mathrm{E}[A(t_{I_{m-1}})|A(t_{I_{m-1}}, t_{I_m}) = j, \mathcal{E}^R(\mathbf{t})] + j\frac{t - t_{I_{m-1}}}{t_{I_m} - t_{I_{m-1}}} \right\} \\
&\quad \times \Pr[A(t_{I_{m-1}}, t_{I_m}) = j|\mathcal{E}^R(\mathbf{t})] \\
&= \mathrm{E}[A(t_{I_{m-1}})|\mathcal{E}^R(\mathbf{t})] + \mathrm{E}[A(t_{I_{m-1}}), A(t_{I_m})|\mathcal{E}^R(\mathbf{t})]\frac{t - t_{I_{m-1}}}{t_{I_m} - t_{I_{m-1}}}
\end{aligned}$$

After substituting $\mathrm{E}[A(t_{I_m})|\mathcal{E}^R(\mathbf{t})] - \mathrm{E}[A(t_{I_{m-1}})|\mathcal{E}^R(\mathbf{t})]$ for $\mathrm{E}[A(t_{I_{m-1}}), A(t_{I_m})|\mathcal{E}^R(\mathbf{t})]$ and doing some rearranging, we see that the claim is indeed true. ∎

So we have now shown how to find the entire function, $\mathrm{E}[A(t)|\mathcal{E}^R(\mathbf{t})]$, while only calculating C columns of the α^R-, η^R-, and β^R-matrices. We call this the QIE^R algorithm. Since we have not found the entire β^R-matrix, it is not possible to find the $\Pi[k|\mathcal{E}^R(\mathbf{t})]$'s, the probabilities that a random arrival finds k customers in queue. However, as described at the beginning of Chapter 4, it is possible to generate all of the other queue statistics that are generated by the standard QIE algorithm. Of course, if we needed the $\Pi[k|\mathcal{E}^R(\mathbf{t})]$'s, it would be possible to generate them just by calculating the full β^R-matrix, but then no computational savings would be realized, so it is hard to imagine why one would choose to do this when, for the same computational effort, one could implement the full QIE algorithm.

We now consider two specific ways of implementing the QIE^R algorithm. The first and most obvious way would be to select a subset of the total set of conditions and use that subset to calculate the entire function $\mathrm{E}[A(t)|\mathcal{E}^R(\mathbf{t})]$. So, we might choose $C = 10$, and, in the case of a congestion period with $N = 50$ customers, we might choose to include conditions 5, 10, 15, 20, 25, 30, 35, 40, 45, and 50 (recall that we must include condition N). Note that, rather than spacing these conditions evenly

with regard to the number of the condition, it might also be of interest to space them evenly with regard to time. For example, say that the congestion period in question had $N = 4$ and $C = 2$. By spacing evenly with regard to condition, we would be tempted to use $A(t_2) \geq 2$ and $A(t_4) \geq 4$ as our two conditions. However, if our **t**-vector were $t_1 = 1$, $t_2 = 2$, $t_3 = 5$, $t_4 = t_N = 10$, it might make more sense to choose the conditions corresponding to t_3 and t_4. In either case, we have some method of choosing a set of conditions which is then used to calculate the entire function, $\mathrm{E}[A(t)|\mathcal{E}^R(\mathbf{t})]$. In this case, the total computational complexity of the QIE^R algorithm would be $O(N^2C)$, because we calculate C columns of the α^R- and η^R-matrices, each of which has potentially as many as N values, and each of these values is produced by up to N computations from the previously calculated column.

We claim that, in this implementation, $\mathrm{E}[A(t)|\mathcal{E}^R(\mathbf{t})]$ is a concave function. The proof of this claim follows very closely the proof that $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$ is a concave function in [Lars 90], so we present only a sketch here. We have the following:

Claim 6.6 *$E[A(t)|\mathcal{E}^R(\mathbf{t})]$ is a concave function of t, for $0 < t \leq t_N$.*

Proof: We will show that the function is concave on $(t_{I_{k-1}}, t_{I_{k+1}}]$ for $k = 1, 2, \ldots, C-1$ by fixing the value of $A(t_{I_{k-1}})$ and then adding up weighted concave functions. We consider two cases.

Case 1. $A(t_{I_{k-1}}) = m$, $l_{I_k} \leq m \leq N$

Then of the remaining $N - m$ arrivals, we may have any number n of them, with $n \geq \max(0, l_{I_{k+1}} - m)$, uniformly and independently distributed over $(t_{I_{k-1}}, t_{I_{k+1}}]$. Of course, the contribution to $\mathrm{E}[A(t)|\mathcal{E}^R(\mathbf{t}), A(t_{I_{k-1}}) = m]$, $t_{I_{k-1}} < t \leq t_{I_{k+1}}$ of the arrivals prior to $t_{I_{k-1}}$ is just a constant (of value m), and arrivals after $t_{I_{k+1}}$ contribute nothing. Finally, with n fixed, the contribution of the n uniform arrivals will be a straight line. Hence, after unconditioning on n and adding in the constant, we find that, in this case, $\mathrm{E}[A(t)|\mathcal{E}^R(\mathbf{t}), A(t_{I_{k-1}}) = m]$, $t_{I_{k-1}} < t \leq t_{I_{k+1}}$ is a straight line, which is a concave function.

Case 2. $A(t_{I_{k-1}}) = m$, $l_{I_{k-1}} \leq m < l_{I_k}$

Again, we temporarily fix the number of arrivals in $(t_{I_{k-1}}, t_{I_{k+1}}]$ to be n, where $n \geq l_{I_{k+1}} - m$, and we define B_{mn} as follows:

$$B_{mn} \equiv \{A(t_{I_{k-1}}) = m, A(t_{I_{k+1}}) = m + n\}$$

Without the time constraint at t_{I_k}, we would still have that $\mathrm{E}[A(t)|B_{mn}], t_{I_{k-1}} < t \leq t_{I_{k+1}}$ is a straight line, starting at the value m at $t_{I_{k-1}}$ and ending at the value $m+n$ at $t_{I_{k+1}}$. However, we no longer have uniform independent arrivals, since we have the intermediate requirement that $A(t_{I_k}) \geq l_{I_k}$ (in Case 1, this is automatically satisfied, because of the value of m). It should be obvious that

$$\mathrm{E}[A(t_{I_k})|B_{mn}, A(t_{I_k}) \geq l_{I_k}] \geq \mathrm{E}[A(t_{I_k})|B_{mn}], \quad t_{I_{k-1}} < t \leq t_{I_{k+1}}$$

since the expectation of a random variable, after it is restricted to the upper values of its range, can only increase. It should also be clear that

$$\mathrm{E}[A(t)|B_{mn}, A(t_{I_k}) \geq l_{I_k}] = \mathrm{E}[A(t)|B_{mn}, \mathcal{E}^R(\mathbf{t})], \quad t_{I_{k-1}} < t \leq t_{I_{k+1}}$$

since, in the given range for t, with the endpoints fixed, the only part of $\mathcal{E}^R(\mathbf{t})$ which affects the expected cumulative number of arrivals is the intermediate condition at t_{I_k}. It is certainly still true that for values of t between $t_{I_{k-1}}$ and t_{I_k} and between t_{I_k} and $t_{I_{k+1}}$, $\mathrm{E}[A(t)|B_{mn}, A(t_{I_k}) \geq l_{I_k}]$ will still be linear (consider conditioning on the value of $A(t_{I_k})$, weighting, and adding). Hence, $\mathrm{E}[A(t)|B_{mn}, A(t_{I_k}) \geq l_{I_k}]$ must be a concave function in the given range for t, since it is piecewise-linear; it has values equal to those of the straight line given by $\mathrm{E}[A(t)|B_{mn}]$ at its endpoints; and it has a value greater than or equal to that of the straight line given by $\mathrm{E}[A(t)|B_{mn}]$ at its single breakpoint, t_{I_k}. Therefore, we can say that $\mathrm{E}[A(t)|B_{mn}, \mathcal{E}^R(\mathbf{t})]$ is concave in the given range, and, after unconditioning on n, since the sum of concave functions is concave, we have that $\mathrm{E}[A(t)|\mathcal{E}^R(\mathbf{t}), A(t_{I_{k-1}}) = m], \ t_{I_{k-1}} < t \leq t_{I_{k+1}}$ is a concave function.

To complete the proof, we merely uncondition on m by weighting by the appropriate probabilities and adding. Again, since the sum of concave functions is concave,

and we do not alter concavity by weighting, we have that $\mathrm{E}[A(t)|\mathcal{E}^R(\mathbf{t})]$ is concave on $(t_{I_{k-1}}, t_{I_{k+1}}]$. Since there was nothing special about the selected value of k, we conclude that $\mathrm{E}[A(t)|\mathcal{E}^R(\mathbf{t})]$ is a concave, increasing, piecewise-linear function, with breakpoints at the t_{I_k}'s, with $k = 1, 2, \ldots, C-1$. ∎

We now continue with specifics of this first type of implementation. One pitfall to be aware of in this method is that, since we no longer have the requirement $A(t_i) \geq i$ at all values of i, it is possible that the expected queue length at some of these unconditioned t_i's could be negative. An obvious remedy to this problem would be first to calculate $\mathrm{E}[A(t)|\mathcal{E}^R(\mathbf{t})]$ for all t_i; then to require that $\mathrm{E}[A(t_i)|\mathcal{E}^R(\mathbf{t})] \geq i$ for all i; and next to linearize the function between the new values at the t_i's. We know that the original $\mathrm{E}[A(t)|\mathcal{E}^R(\mathbf{t})]$ is a concave function, but adding the requirement that no queue lengths be negative can cause the new function to violate concavity. Hence, to get an even better bound, as a final step, we may take the concave hull of the modified function. This technique ensures that we generate no negative queue lengths and that the resulting function is concave and piecewise linear. Note that this function will still be a lower bound to the actual $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$.

We may find a series of stochastically dominant lower bounds to the exact QIE algorithm by using this first implementation (with or without the combined improvements just suggested). In our $N = 50$ example, if we begin with the two conditions at t_{25} and t_{50}, we get a fairly weak lower bound. As we add conditions one at a time, each $\mathrm{E}[A(t)|\mathcal{E}^R(\mathbf{t})]$ is an upper bound to the previously generated function, while still being a lower bound to the exact $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$ (from Theorem 5.1). Since the algorithm is essentially $\mathrm{O}(N^2)$ to find $\mathrm{E}[A(t)|\mathcal{E}^R(\mathbf{t})]$ for the two-condition case (see the discussion in the next two paragraphs), and is $\mathrm{O}(N^3)$ for the exact QIE, then we essentially have a continuum of algorithms available, with range in computational complexity between $\mathrm{O}(N^2)$ and $\mathrm{O}(N^3)$, and with accuracy increasing as complexity. At this point, it is not clear where in this range the algorithm becomes "good enough," but it is speculated that it is closer to the N^2 end of the range than the N^3.

The computational results in the next section provide some illustration of the types of output that result for various values of C.

A second way to implement the QIE^R algorithm is found by recalling the original intent for devising the algorithm, which was that perhaps the expected cumulative number of arrivals by time t_i would only depend heavily on conditions in the vicinity of t_i. So, as a first approach, say that we only wish to consider the condition $X_i \leq t_i$ when calculating the expected cumulative number of arrivals by t_i. So, if we let R_i represent the set of bounds corresponding to the two conditions, $X_i \leq t_i$ and $X_N \leq t_N$, then we would like to calculate $\mathrm{E}[A(t_i)|\mathcal{E}^{R_i}(\mathbf{t})]$. This can surely be done: for each value of i, we calculate column i of the α^{R_i}-matrix and of the η^{R_i}-matrix, plus the value $\alpha_{NN}^{R_i}(\mathbf{t})$. From this, we can find all elements of the i-th column of the β^{R_i}-matrix. Finally, we add up all of the elements in that column to get our value for $\mathrm{E}[A(t_i)|\mathcal{E}^{R_i}(\mathbf{t})]$.

Note how different this is from the previous implementation: here we choose a different subset of conditions for each t_i and calculate the expected cumulative number of arrivals at that time based on these different conditions. This would appear to increase the computational complexity substantially, since, for each i, we would have an $\mathrm{O}(N^2C)$ operation. There are N values of i, so the entire algorithm in this implementation would appear to be $\mathrm{O}(N^3C)$, greater than the original QIE. This would be the case if we chose a large and different set of conditions for each i, but the current example being considered requires closer scrutiny. In this case, we have $C = 2$, $I_1 = i$, and $I_2 = N$. So, in order to calculate column i of the α^{R_i}-matrix, we need only use Claim 6.1: i.e., calculation of the entire column requires only $N - i$ calculations. We also have to calculate $\alpha_{NN}^{R_i}(\mathbf{t})$. Using Claim 6.2, we see that this also requires $N - i$ calculations. Next, we have to calculate the i-th column of the η^{R_i}-matrix. Using Claim 6.3, we see that this also requires $N - i$ calculations. Finally, there are $N - i - 1$ non-trivial entries in the i-th column of the β^{R_i}-matrix, each of which requires $\mathrm{O}(1)$ operation. Hence, we see that calculation of $\mathrm{E}[A(t_i)|\mathcal{E}^{R_i}(\mathbf{t})]$

requires $N - i$ operations, and hence, calculation of the entire function is an $O(N^2)$ operation.

One advantage of this implementation is that we do require $A(t_i) \geq i$ for every i, so we don't have the problem of generating negative queue lengths. This function is not necessarily concave, so we can also improve performance by taking its concave hull. This implementation actually performs quite well (see Section 6.2), especially considering that we are only looking at a single arrival-time inequality at each time. But, as already mentioned, if we try to look at a different set of $C > 2$ inequalities at each t_i, the computational complexity becomes $O(N^3C)$, greater than that of the original QIE algorithm.

In Section 6.2, we present the results of several runs of this algorithm. We consider both implementations, and for the first implementation (using a single set of conditions to generate $E[A(t)|\mathcal{E}^R(t)]$), we consider both the method of choosing conditions by spacing the conditions evenly and by spacing them evenly in time. We compare all of the implementations to each other, and to the original QIE algorithm, to demonstrate their accuracy and their improved runtimes. We also consider the improvements found by adding the requirements that no queue length be negative and that the functions be concave. These additions improve our estimates while adding very little to the runtimes.

6.2 Computational Results of the QIE^R Algorithm

We include here results from simulation of an M/M/1 queue. These data were generated by three simulation runs with Poisson arrivals at rate 10 per hour, a single server, and exponential service times with expected values of 3 minutes for the first run (giving a value of $\rho = 0.5$) and 4 minutes for the last two runs (giving a value of $\rho = 0.67$). Runs were on a 386/387-based Northgate Computer Systems PC. Each run time given below is an average of 3000 run times (for shorter runs, presented to

thousandths of a second) or 1000 run times (for longer runs, presented to hundredths of a second) from different runs of the program on the same data. This averaging was necessary because the system clock is only updated every 0.0549254 seconds [Scan 83], so to get accuracy greater than 0.1 seconds, many runs must be averaged.

We compare the QIE^R algorithm to the standard QIE algorithm. First, we consider the case in which a fixed number of C conditions is used to generate our entire function, $\mathrm{E}[A(t)|\mathcal{E}^R(\mathrm{t})]$. Then, we examine the case in which a single local condition is used at each t_i to generate the expected cumulative number of arrivals by that time. The statistics that are used for comparison of the algorithms include: $\mathrm{E}[L_Q|\ldots]$, the time-averaged number of customers in queue; $\mathrm{E}[W_Q|\ldots]$, the average wait in queue; and δ, the approximation error, which we define to be the absolute area between $\mathrm{E}[Q(t)|\mathcal{E}^S(\mathrm{t})]$ as generated by the exact QIE algorithm, and the lower bound algorithm's estimate of the same function, divided by the duration of the congestion period, t_N. The run times to generate the beta-matrix for the different algorithms are also compared.

First, consider choosing a fixed number of C conditions to generate the entire expected queue length function. We will examine the two longest congestion periods in the first simulation run, one with $N = 18$, and the other with $N = 21$, as well as the longest congestion period from each of the other two runs, each with $N = 58$. We consider different values of C. For the two smaller congestion periods, we begin with a total of $C = 5$ conditions, then increase that number to $C = 8$, and finally finish with $C = 10$ conditions. For the $N = 58$ congestion periods, we consider $C = 5$, $C = 10$, and $C = 20$. In each case, we choose which k conditions to include (actually, we only choose $k - 1$, since the N-th condition must always be included) by trying to space them evenly by condition number. So, for example, in the $N = 18$ case, we begin with conditions 4, 7, 11, 14, and 18. In the $N = 21$ case, we begin with conditions 4, 8, 12, 16, and 21. In the $N = 58$ case, we begin with conditions 11, 23, 35, 47, and 58. Because we would like to show a series of stochastically dominant

bounds, we then proceed to generate the next set of conditions by adding conditions to the original 5 in all cases. For the $N = 18$ case, we add conditions 2, 9, and 16; for the $N = 21$ case, we add conditions 2, 6, and 18; and for the $N = 58$ case, we add conditions 5, 17, 29, 41, and 53. Although the conditions in the $N = 21$ case are no longer very evenly spaced, adding the last two to generate the $C = 10$ set of conditions again restores the even spacing. In the $N = 18$ case, we add conditions 5 and 12; in the $N = 21$ case, we add conditions 10 and 14; and in the $N = 58$ case, we add conditions 2, 8, 14, 20, 26, 32, 38, 44, 50, and 56. The selected conditions are summarized below.

Size of Cong. Pd.	Number of Conditions	Conditions Selected
$N = 18$	$C = 5$	4,7,11,14,18
	$C = 8$	2,4,7,9,11,14,16,18
	$C = 10$	2,4,5,7,9,11,12,14,16,18
$N = 21$	$C = 5$	4,8,12,16,21
	$C = 8$	2,4,6,8,12,16,18,21
	$C = 10$	2,4,6,8,10,12,14,16,18,21
$N = 58$	$C = 5$	11,23,35,47,58
	$C = 10$	5,11,17,23,29,35,41,47,53,58
	$C = 20$	2,5,8,11,14,17,20,23,26,29, 32,35,38,41,44,47,50,53,56,58

Figures 6.1 and 6.2 present the expected queue length for the four congestion periods and the sets of conditions above, both for the exact QIE algorithm and for the QIE^R algorithm. There are a couple of things to point out with regard to these figures. First, we have incorporated the "no negative queue-lengths" improvement suggested in the last section, but have not yet incorporated the concavity filter. Second, although the total number of conditions is 5, 8, 10, or 20, the number of

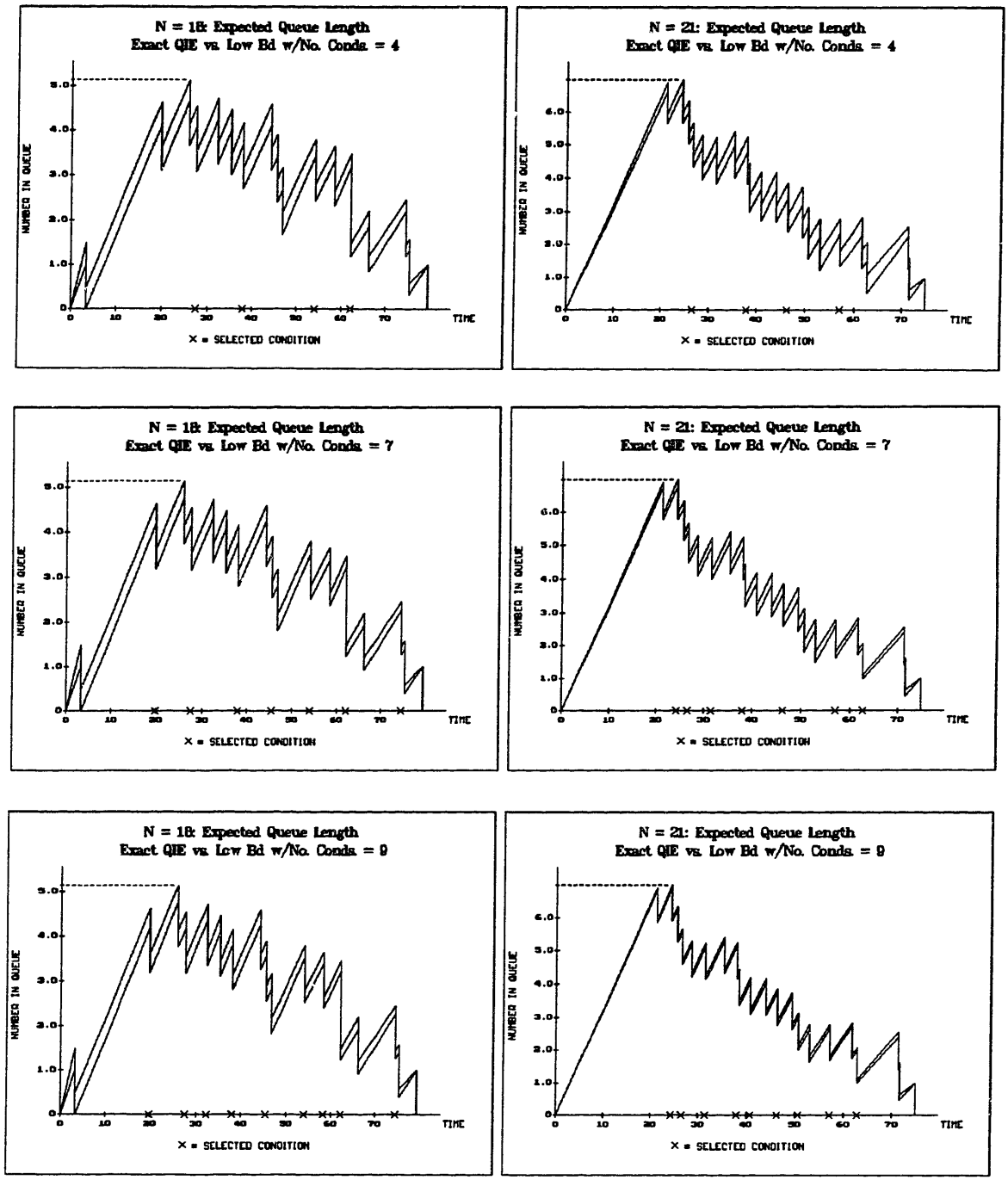


Figure 6.1: Expected Queue Length for Congestion Periods with $N = 18$ and $N = 21$: Exact QIE vs. QIE^R Algorithm, for $C = 5, 8$, and 10, No Concavity Filter

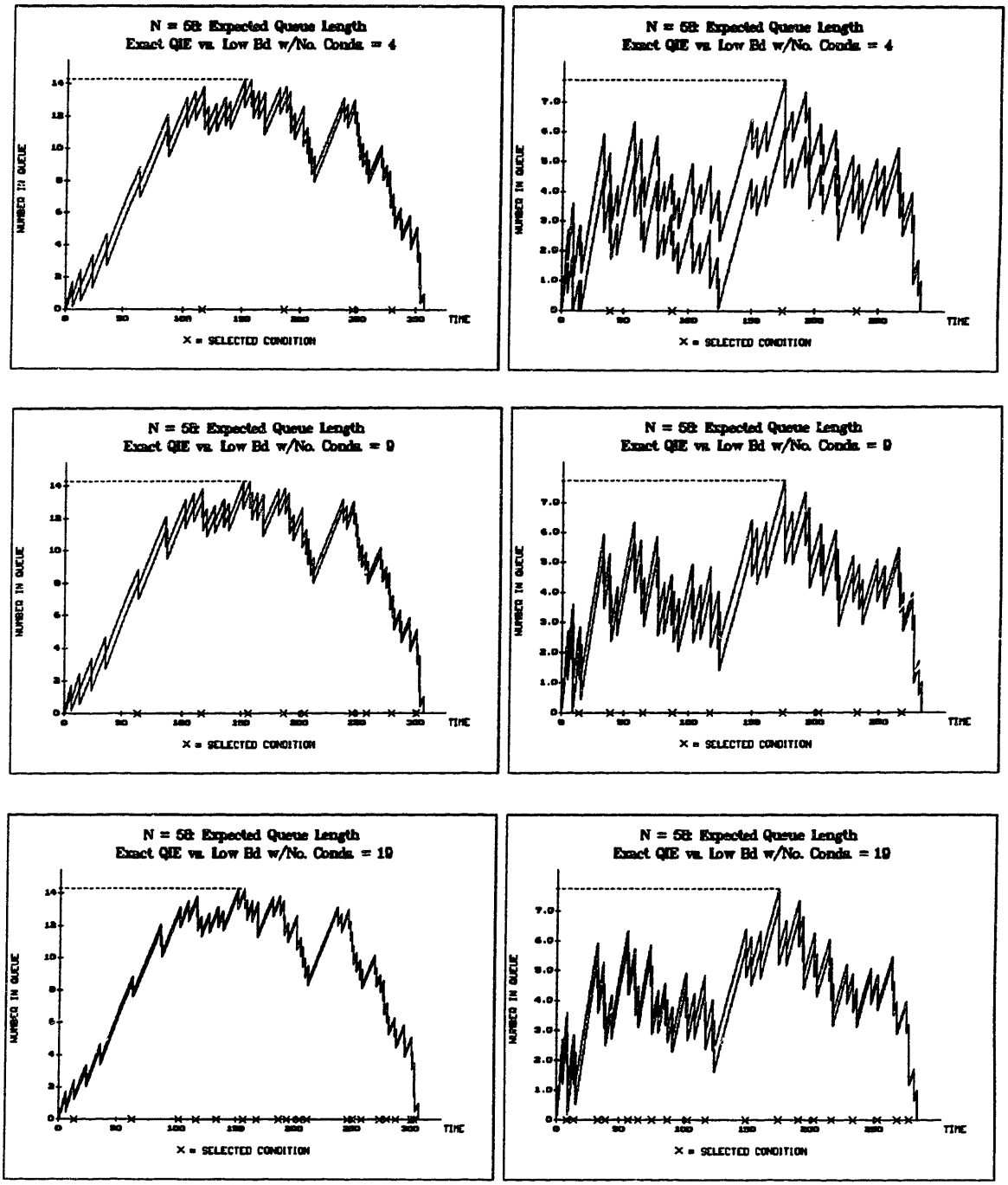


Figure 6.2: Expected Queue Length for Two Congestion Periods with $N = 58$: Exact QIE vs. QIE^R Algorithm, for $C = 5, 10$, and 20, No Concavity Filter

internal conditions selected by the user is actually one less than this, since the N-th condition must always be selected: this is the number of conditions indicated on the graphs themselves. The queue statistics for this set of graphs are presented in Table 6.1. The algorithm does very well on these congestion periods, and the monotonic increase in the values of the lower bounding queue statistics, as well as the monotonic decrease in δ, may be seen. Note that the reported run times will vary as the set of chosen conditions changes. This is due to the way the matrices are computed, the left-most columns having many more elements to compute than the right-most columns. To test the variance that would result, we ran both the $N = 18$ and the $N = 21$ congestion periods with the two extreme sets of conditions for the case $C = 5$, and the results given in Table 6.2 were achieved. These results are depicted in Figure 6.3: note that we have not implemented the concavity filter for these runs. Note that there is quite a range of run times. As one would expect, none of the extreme cases does as well at estimating the queue length and queue statistics as the cases where the conditions are spaced. Surprisingly, the extreme case which takes the least amount of time does much better than that which takes the most, and approaches the accuracy of the mixed conditions cases.

Figures 6.4 and 6.5 present the expected queue length for the four congestion periods and the sets of conditions above, both for the exact QIE algorithm and for the QIE^R algorithm, when concavity filtering is added. The queue statistics for this set of graphs is presented in Table 6.3. Notice that, in the $N = 21$ case and in the first $N = 58$ case, the queue statistics do not change when concavity filtering is added. That is because the original queue length calculation did not go negative, so that the original concave function was not modified when checked for negative queue lengths. Hence, taking the concave hull of a concave function gives back the same function, i.e., nothing is changed. Since we did check for concavity, though, the run times are slightly increased. This small increase can be considered to be the maximum time that the concavity filter could add to an $N = 21$ ($N = 58$) congestion period, since

Size of Cong. Period	Algorithm Used	E[L_Q|...]	E[W_Q|...] (minutes)	δ	Run Time (seconds)
$N = 18$	QIE	2.8649	12.6644	0	0.404
	QIE^R, $C = 5$	2.4537	10.8465	0.4112	0.088
	QIE^R, $C = 8$	2.5237	11.1560	0.3412	0.175
	QIE^R, $C = 10$	2.5373	11.2160	0.3276	0.222
$N = 21$	QIE	3.3870	12.0614	0	0.694
	QIE^R, $C = 5$	3.0348	10.8069	0.3523	0.138
	QIE^R, $C = 8$	3.1969	11.3841	0.1902	0.297
	QIE^R, $C = 10$	3.2905	11.7174	0.0966	0.338
$N = 58$ (1)	QIE	9.3605	49.5175	0	31.15
	QIE^R, $C = 5$	8.6864	45.9513	0.6741	1.79
	QIE^R, $C = 10$	8.7223	46.1412	0.6382	4.62
	QIE^R, $C = 20$	9.1154	48.2206	0.2452	10.31
$N = 58$ (2)	QIE	4.4114	21.5511	0	31.22
	QIE^R, $C = 5$	3.0226	14.7664	1.3888	1.79
	QIE^R, $C = 10$	3.7435	18.2884	0.6679	4.64
	QIE^R, $C = 20$	3.9568	19.3302	0.4546	10.40

Table 6.1: Comparison of QIE and QIE^R Algorithms (No Concavity Filter) for Congestion Periods with $N = 18$, 21, and 58

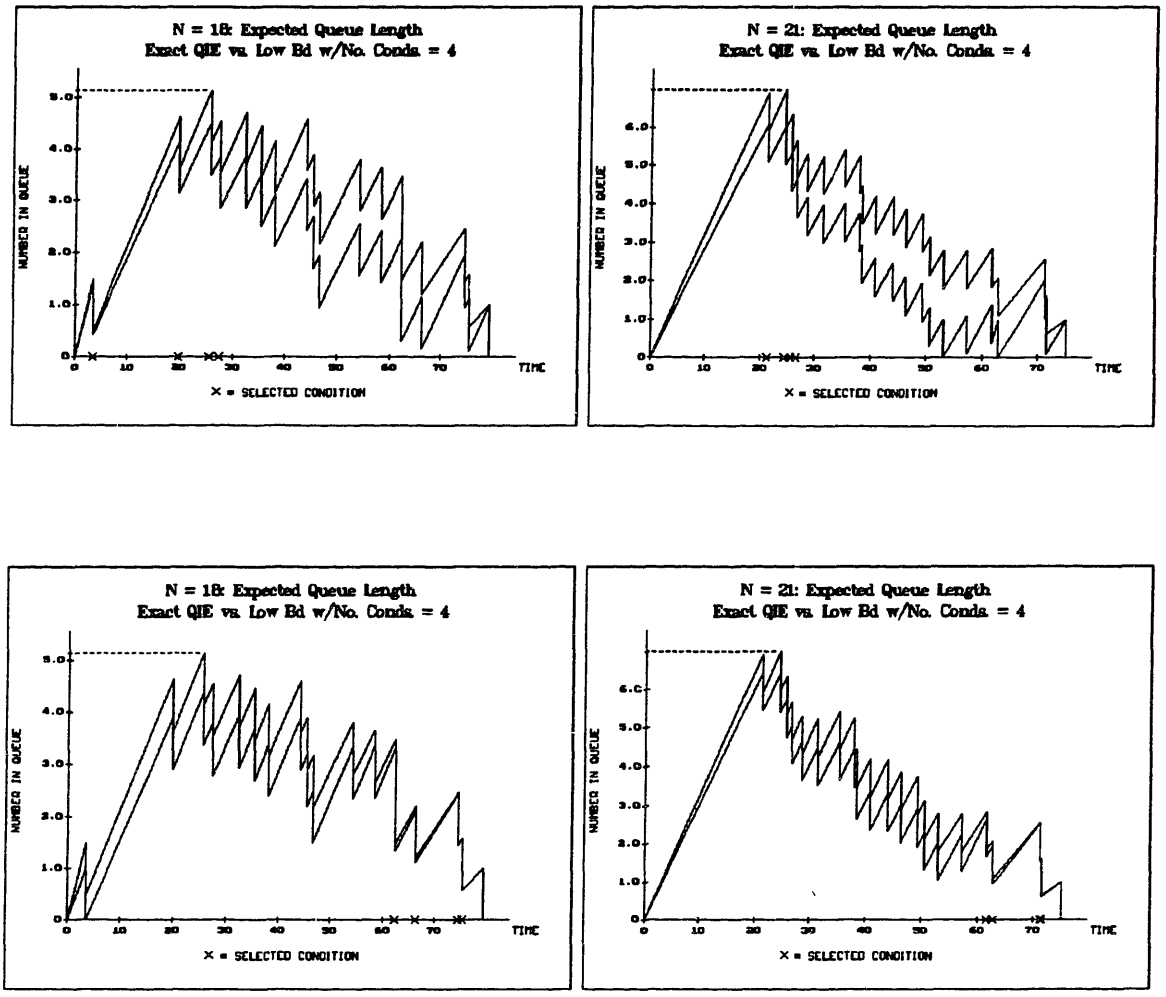


Figure 6.3: Expected Queue Length for Congestion Periods with $N = 18$ and $N = 21$: Exact QIE vs. QIE^R Algorithm, for Extreme Conditions, No Concavity Filter

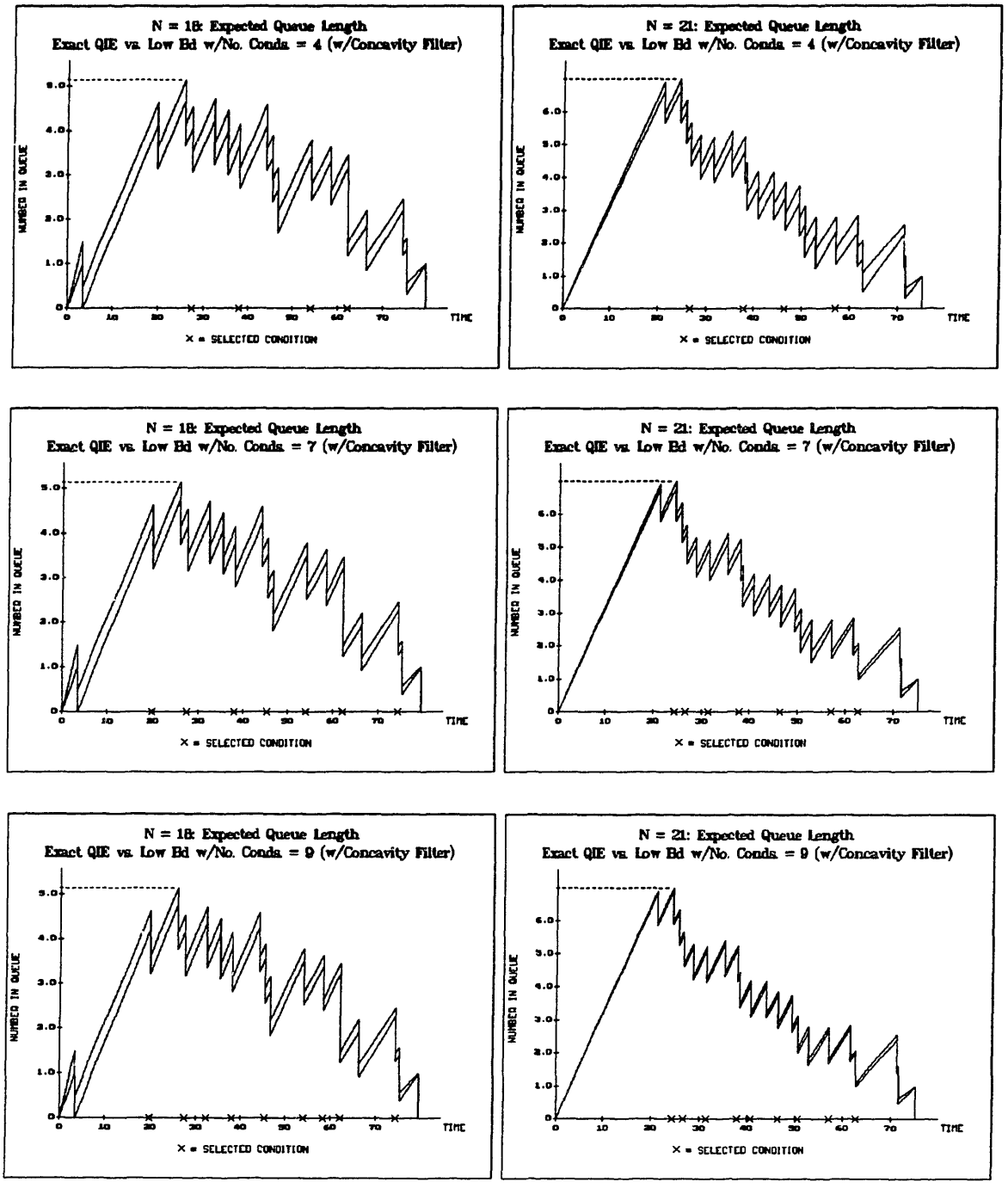


Figure 6.4: Expected Queue Length for Congestion Periods with $N = 18$ and $N = 21$: Exact QIE vs. QIE^R Algorithm, for $C = 5, 8$, and 10, with Concavity Filter

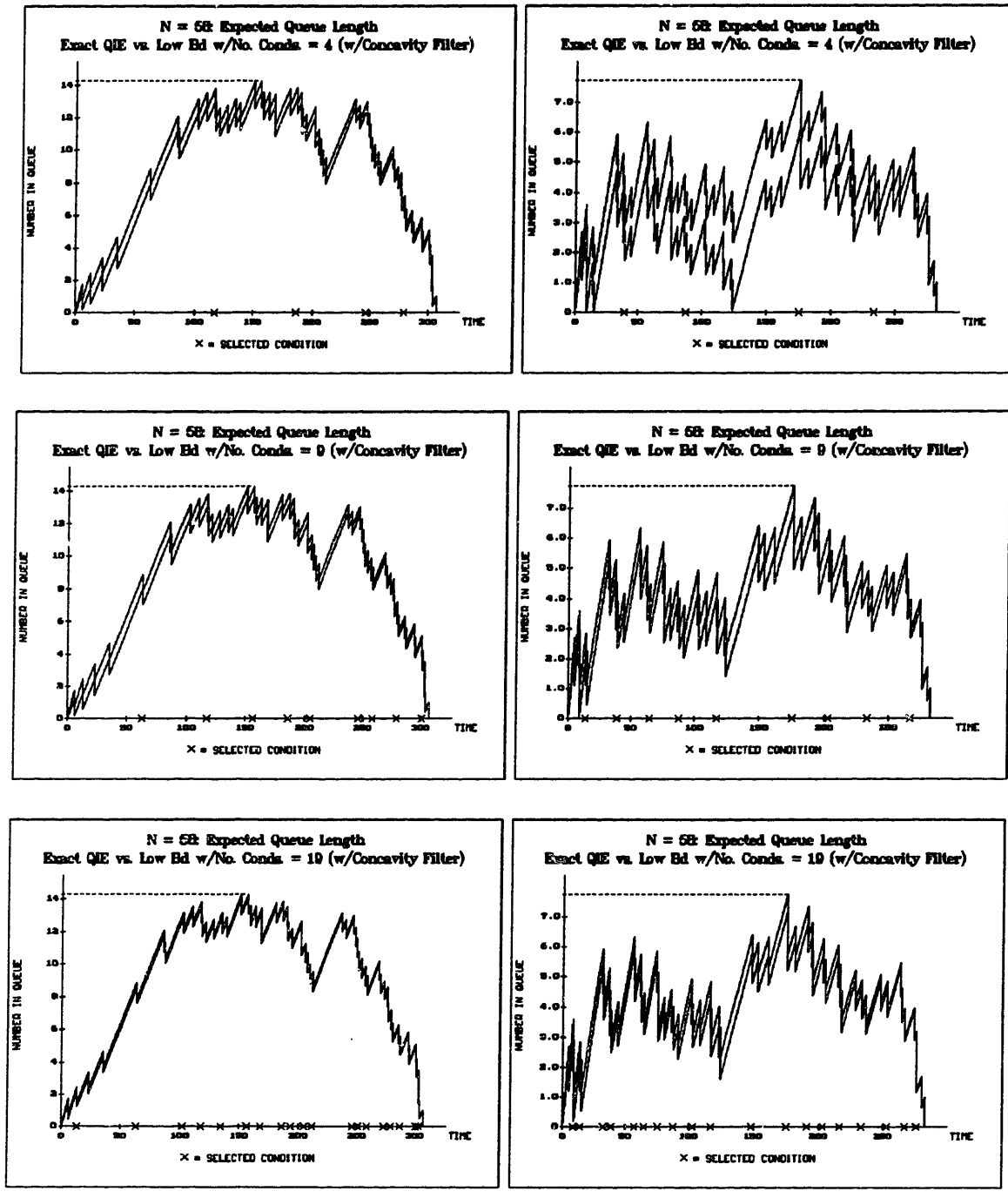


Figure 6.5: Expected Queue Length for Two Congestion Periods with $N = 58$: Exact QIE vs. QIE^R Algorithm, for $C = 5, 10$, and 20, with Concavity Filter

Size of Cong. Pd.	Alg. Used	Conds.	E[L_Q|...]	E[W_Q|...] (minutes)	δ	Run Time (seconds)
$N = 18$	QIE	all	2.8649	12.6644	0	0.404
	QIE^R	1,2,3,4,18	2.1184	9.3645	0.7465	0.209
	QIE^R	14,15,16,17,18	2.3800	10.5205	0.4850	0.014
$N = 21$	QIE	all	3.3870	12.0614	0	0.694
	QIE^R	1,2,3,4,21	2.3429	8.3431	1.0442	0.319
	QIE^R	17,18,19,20,21	2.9262	10.4201	0.4609	0.015

Table 6.2: Comparison of QIE and QIE^R Algorithms for Congestion Periods of $N = 18$ and $N = 21$, with $C = 5$ and Extreme Conditions

an already concave function must be checked at every breakpoint for concavity. In the $N = 18$ case, small improvements are noted, but only because the original QIE^R algorithm produced a negative queue length at time t_1, which was adjusted, resulting in a non-concave function. Similarly, in the second $N = 58$ case, small improvements are noted for the $C = 5$ case only.

Now, we consider trying to space the conditions evenly in time, rather than by condition number, and see if that results in any change in the queue statistics. We consider the case $C = 5$ for both large congestion periods ($N = 18$ and $N = 21$) in the first run and for the $N = 58$ congestion period from each of the last two runs. We look at the results with and without concavity filtering. Figures 6.6 and 6.7 present the expected queue length for the two congestion periods with $N = 18$ and $N = 21$, and the two with $N = 58$, respectively, both for the exact QIE algorithm and for the QIE^R algorithm, with $C = 5$, and with the conditions chosen to be evenly spaced in time. The set of conditions that are chosen are 2, 6, 10, and 14 for the $N = 18$ congestion period; 1, 7, 13, and 17 for the $N = 21$ congestion period; 5, 13, 23, and 35 for the first $N = 58$ congestion period; and 13, 28, 35, and 46 for the second $N = 58$ congestion period. The results are shown both with and without

Size of Cong. Period	Algorithm Used	E[L_Q|...]	E[W_Q|...] (minutes)	δ	Run Time (seconds)
$N = 18$	QIE	2.8649	12.6644	0	0.404
	QIE^R, $C = 5$	2.4594	10.8715	0.4056	0.091
	QIE^R, $C = 8$	2.5252	11.1628	0.3397	0.178
	QIE^R, $C = 10$	2.5388	11.2227	0.3261	0.226
$N = 21$	QIE	3.3870	12.0614	0	0.694
	QIE^R, $C = 5$	3.0348	10.8069	0.3523	0.143
	QIE^R, $C = 8$	3.1969	11.3841	0.1902	0.302
	QIE^R, $C = 10$	3.2905	11.7174	0.0966	0.342
$N = 58$ (1)	QIE	9.3605	49.5175	0	31.15
	QIE^R, $C = 5$	8.6864	45.9513	0.6741	1.80
	QIE^R, $C = 10$	8.7223	46.1412	0.6382	4.64
	QIE^R, $C = 20$	9.1154	48.2206	0.2452	10.34
$N = 58$ (2)	QIE	4.4114	21.5511	0	31.22
	QIE^R, $C = 5$	3.0558	14.9285	1.3556	1.81
	QIE^R, $C = 10$	3.7435	18.2884	0.6679	4.66
	QIE^R, $C = 20$	3.9568	19.3302	0.4546	10.43

Table 6.3: Comparison of QIE and QIE^R Algorithms, with Concavity Filter, for Congestion Periods with $N = 18$, 21, and 58

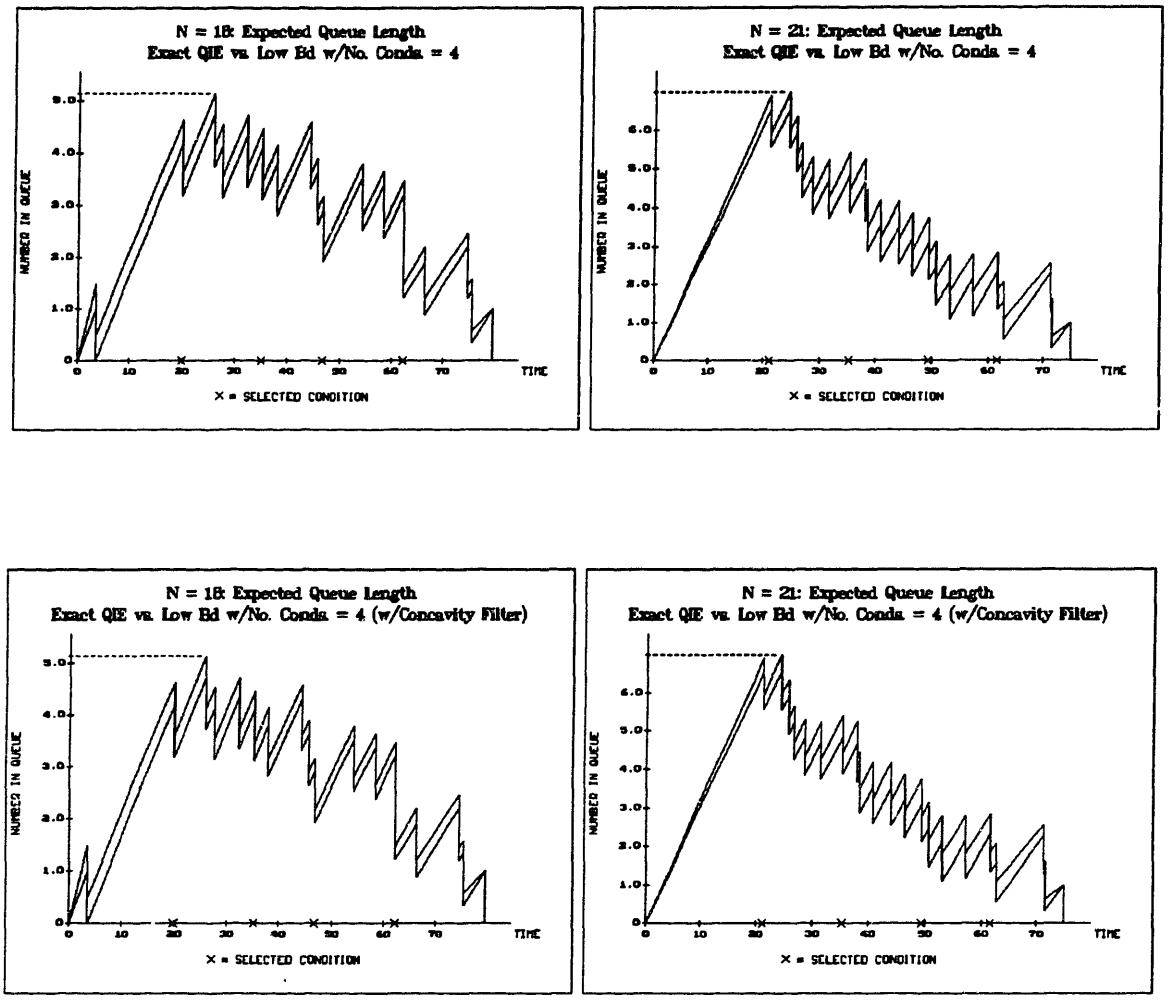


Figure 6.6: Expected Queue Length for Congestion Periods with $N = 18$ and $N = 21$: Exact QIE vs. QIE^R Algorithm, for 5 Time-Spaced Conditions, with and without Concavity Filter

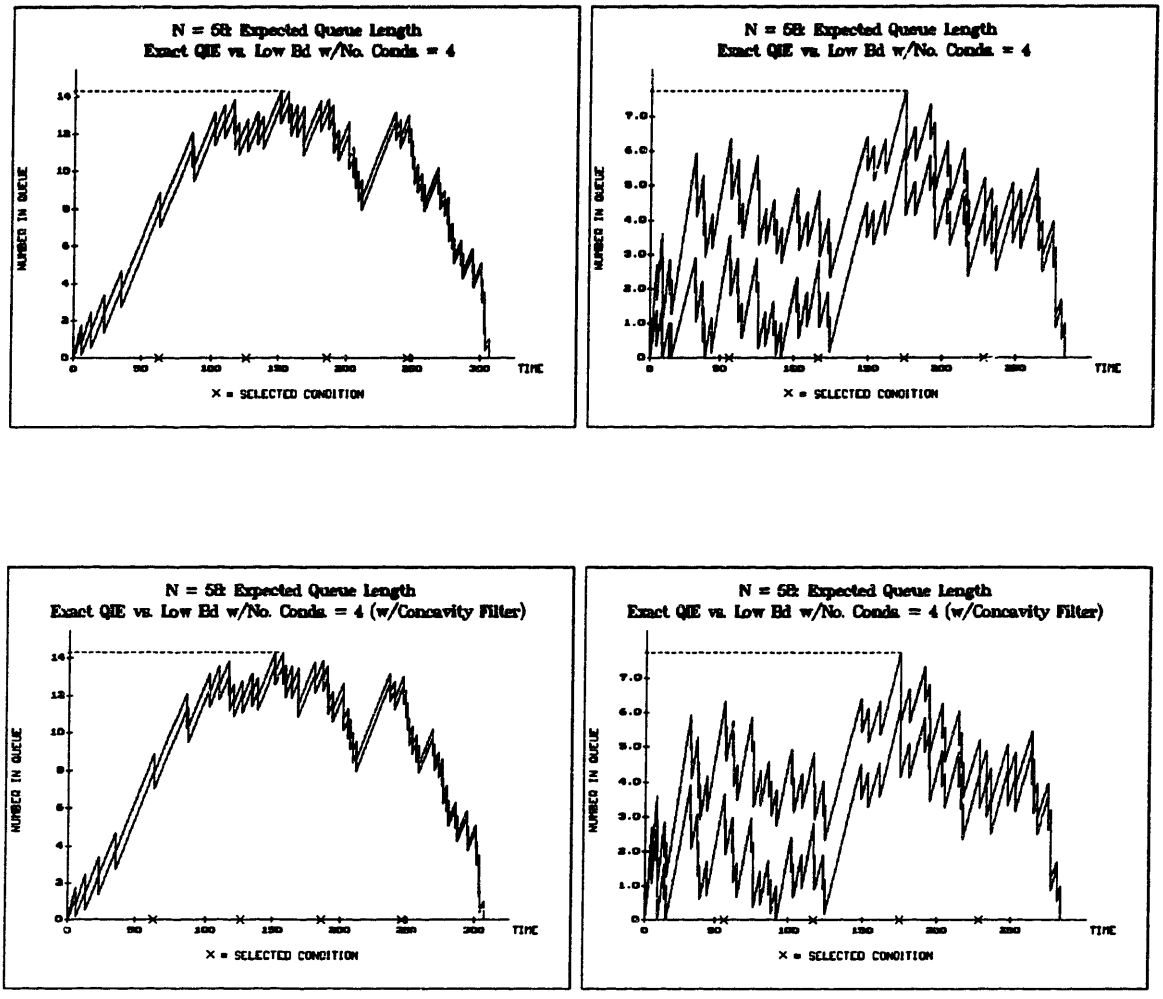


Figure 6.7: Expected Queue Length for Two Congestion Periods with $N = 58$: Exact QIE vs. QIE^R Algorithm, for 5 Time-Spaced Conditions, with and without Concavity Filter

concavity filtering. The queue statistics for this set of runs are presented in Table 6.4. Again, concavity filtering makes a difference only in the $N = 18$ case and the second $N = 58$ case. Although spacing evenly in time gives slightly better estimates in the $N = 18$ example, it gives worse estimates in the other three examples, so we can draw no definite conclusions as to a preferred method for selecting the best set of conditions to use, although we would be inclined to use condition-spacing as a first try. Presumably, there is some optimal set of conditions that gives the best estimate for any given congestion period and C. This is an area which requires exhaustive study.

Size of Cong. Pd.	Algorithm Used	Concavity Filter?	E[L_Q|...]	E[W_Q|...] (minutes)	δ
$N = 18$	QIE	—	2.8649	12.6644	0
	QIE^R, CS	No	2.4537	10.8465	0.4112
	QIE^R, CS	Yes	2.4594	10.8715	0.4056
	QIE^R, TS	No	2.5281	11.1755	0.3368
	QIE^R, TS	Yes	2.5304	11.1855	0.3345
$N = 21$	QIE	—	3.3870	12.0614	0
	QIE^R, CS	No	3.0348	10.8069	0.3523
	QIE^R, CS	Yes	3.0348	10.8069	0.3523
	QIE^R, TS	No	2.9594	10.5385	0.4277
	QIE^R, TS	Yes	2.9594	10.5385	0.4277
$N = 58$ (1)	QIE	—	9.3605	49.5175	0
	QIE^R, CS	No	8.6864	45.9513	0.6741
	QIE^R, CS	Yes	8.6864	45.9513	0.6741
	QIE^R, TS	No	8.6862	45.9506	0.6743
	QIE^R, TS	Yes	8.6862	45.9506	0.6743
$N = 58$ (2)	QIE	—	4.4114	21.5511	0
	QIE^R, CS	No	3.0226	14.7664	1.3888
	QIE^R, CS	Yes	3.0558	14.9285	1.3556
	QIE^R, TS	No	2.6105	12.7530	1.8009
	QIE^R, TS	Yes	2.7288	13.3309	1.6826

Table 6.4: Comparison of QIE and QIE^R Algorithms, with and without Concavity Filter, Condition-Spaced (CS) and Time-Spaced (TS), for Congestion Periods with $N = 18$, 21, and 58

Next, consider the second implementation suggested in the last section, namely, using a single local condition at each t_i: i.e., only using $A(t_i) \geq i$ to calculate $\mathrm{E}[A(t_i)|\ldots]$. We examine the six longest congestion periods in our first simulation run, with numbers of customers 14, 13, 14, 18, 21, and 12, as well as the two $N = 58$ congestion periods, one from each of the other two runs. Figures 6.8 and 6.9 present the expected queue length for the eight congestion periods, both for the exact QIE algorithm and for the QIE^R algorithm. These figures do not incorporate the concavity filter. The same set of data, but with incorporation of the concavity filter for the QIE^R algorithm, is presented in Figures 6.10 and 6.11. The queue statistics for both sets of graphs (with and without concavity filtering) are presented in Table 6.5. Note that, even though we are considering only a single condition at each t_i, the algorithm does fairly well, even without concavity filtering. When the concave hull of the function is taken, marked improvements may be seen. In fact, for $N = 18$, after concavity filtering, this algorithm is comparable in accuracy to, yet runs much faster than, the multiple global conditions implementation for $C = 10$. In the $N = 21$ example, after concavity filtering, this algorithm is close to the accuracy of the other implementation, with $C = 5$, although its run time is slightly greater. For the first $N = 58$ congestion period, after concavity filtering, the algorithm is close to the accuracy of the multiple global conditions, $C = 20$, case, yet runs much faster. And for the second $N = 58$ congestion period, after concavity filtering, the algorithm is between the $C = 5$ and $C = 10$ multiple global conditions implementation in both accuracy and runtime. In general, however, the run times are quite fast and adding concavity filtering adds very little to them, even less than was added in the previous implementation of the QIE^R algorithm. As already discussed, this is because the multiple global conditions implementation produces a concave function, while that produced by the single local condition implementation may be far from concave. It is paradoxical that the more the lower bound is improved by concavity filtering, the less time the concavity filter takes to run!

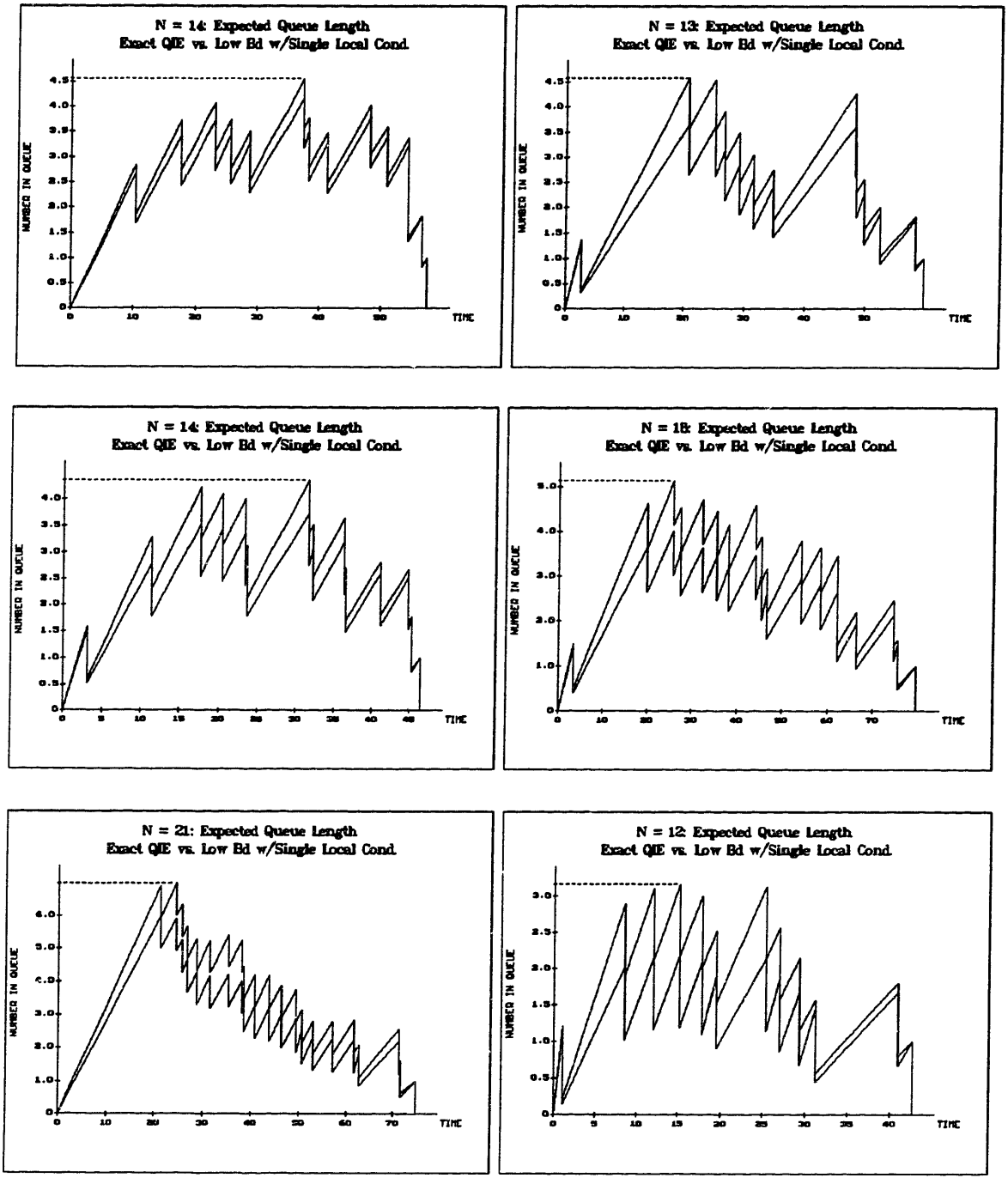


Figure 6.8: Expected Queue Length for Congestion Periods of $N = 14$, 13, 14, 18, 21, and 12: Exact QIE vs. QIE^R, Single Local Condition, No Concavity Filter

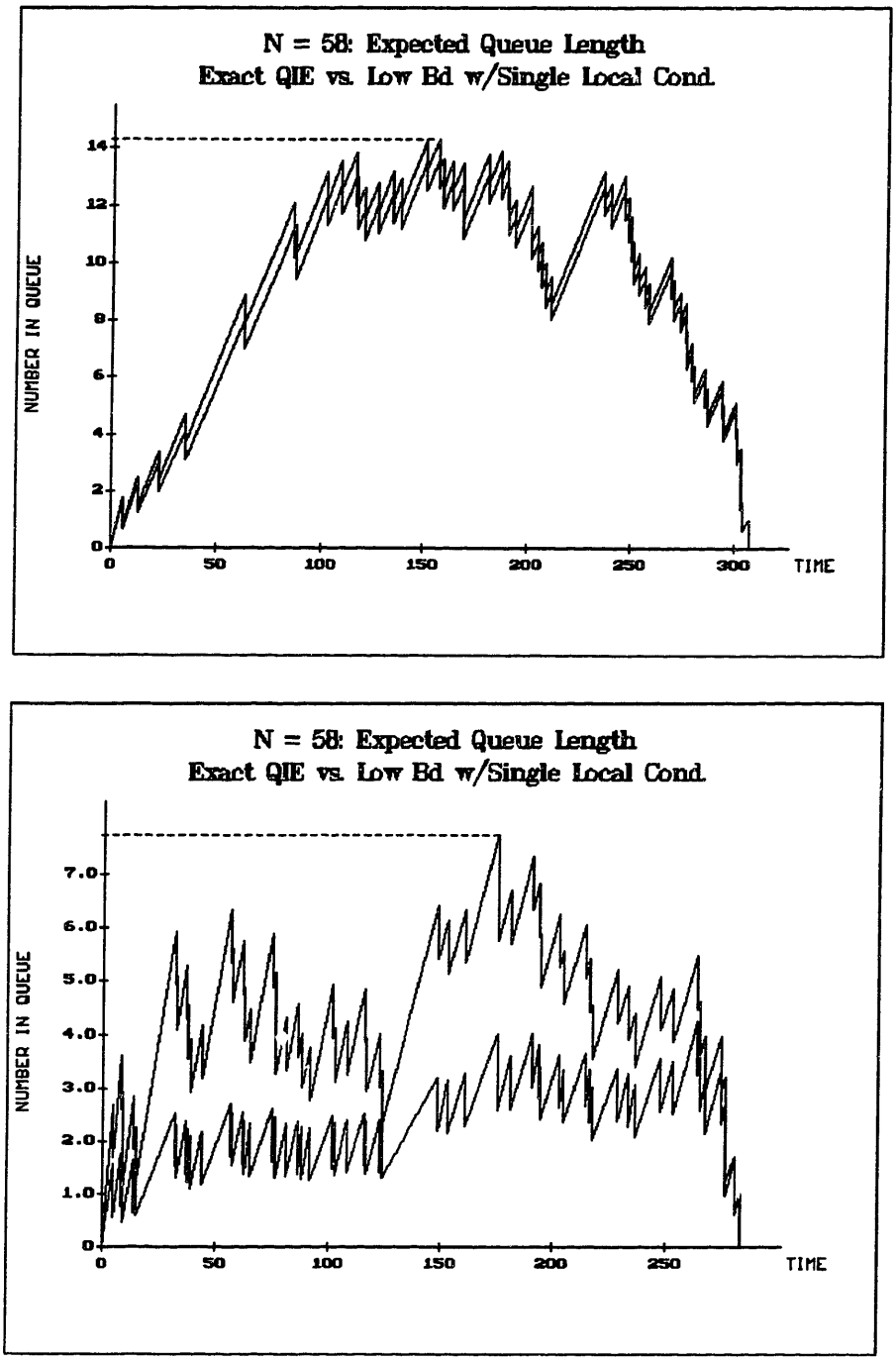


Figure 6.9: Expected Queue Length for Two Congestion Periods of $N = 58$: Exact QIE vs. QIE^R, Single Local Condition, No Concavity Filter

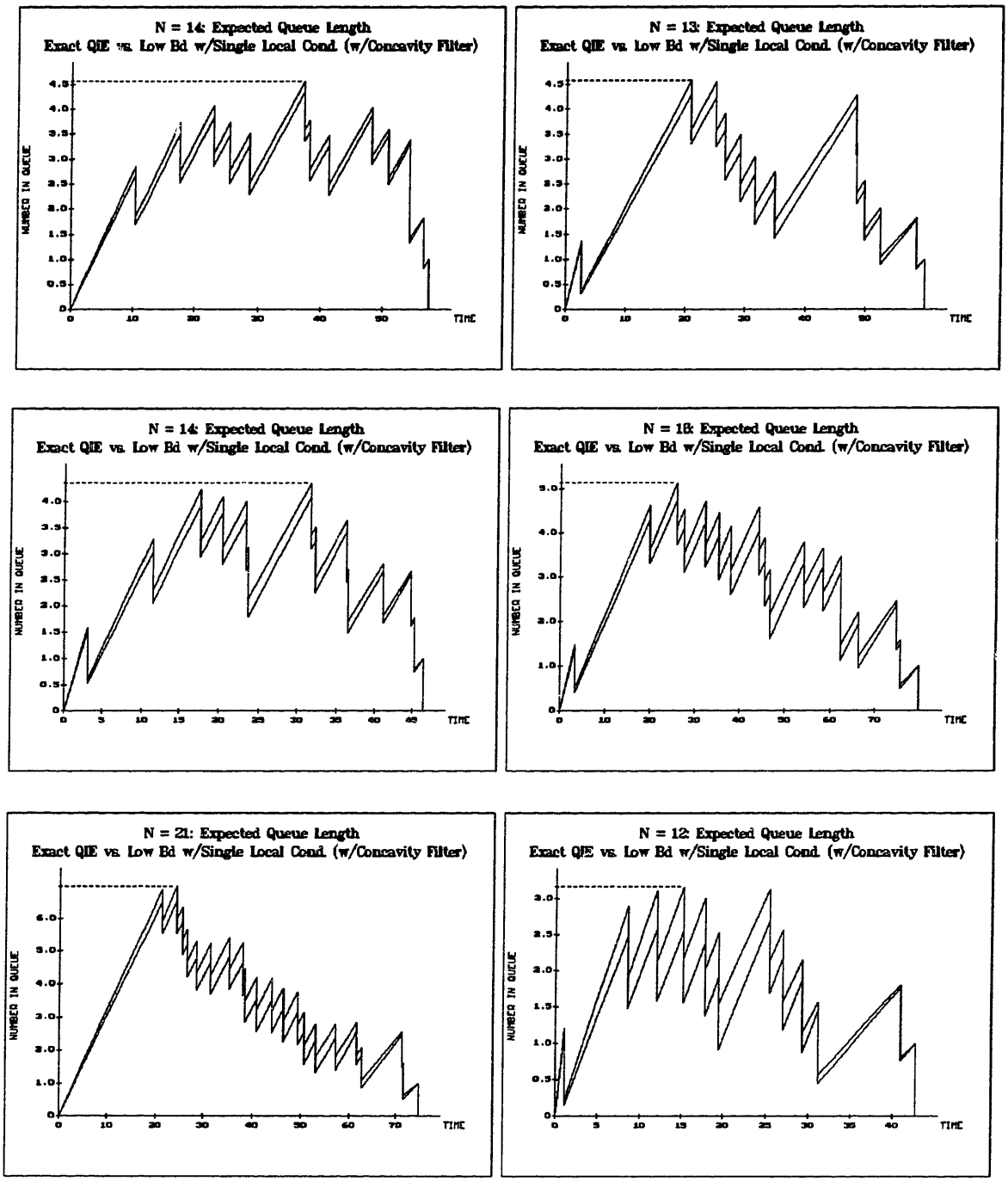


Figure 6.10: Expected Queue Length for Congestion Periods of $N = 14$, 13, 14, 18, 21, and 12: Exact QIE vs. QIE^R, Single Local Condition, with Concavity Filter

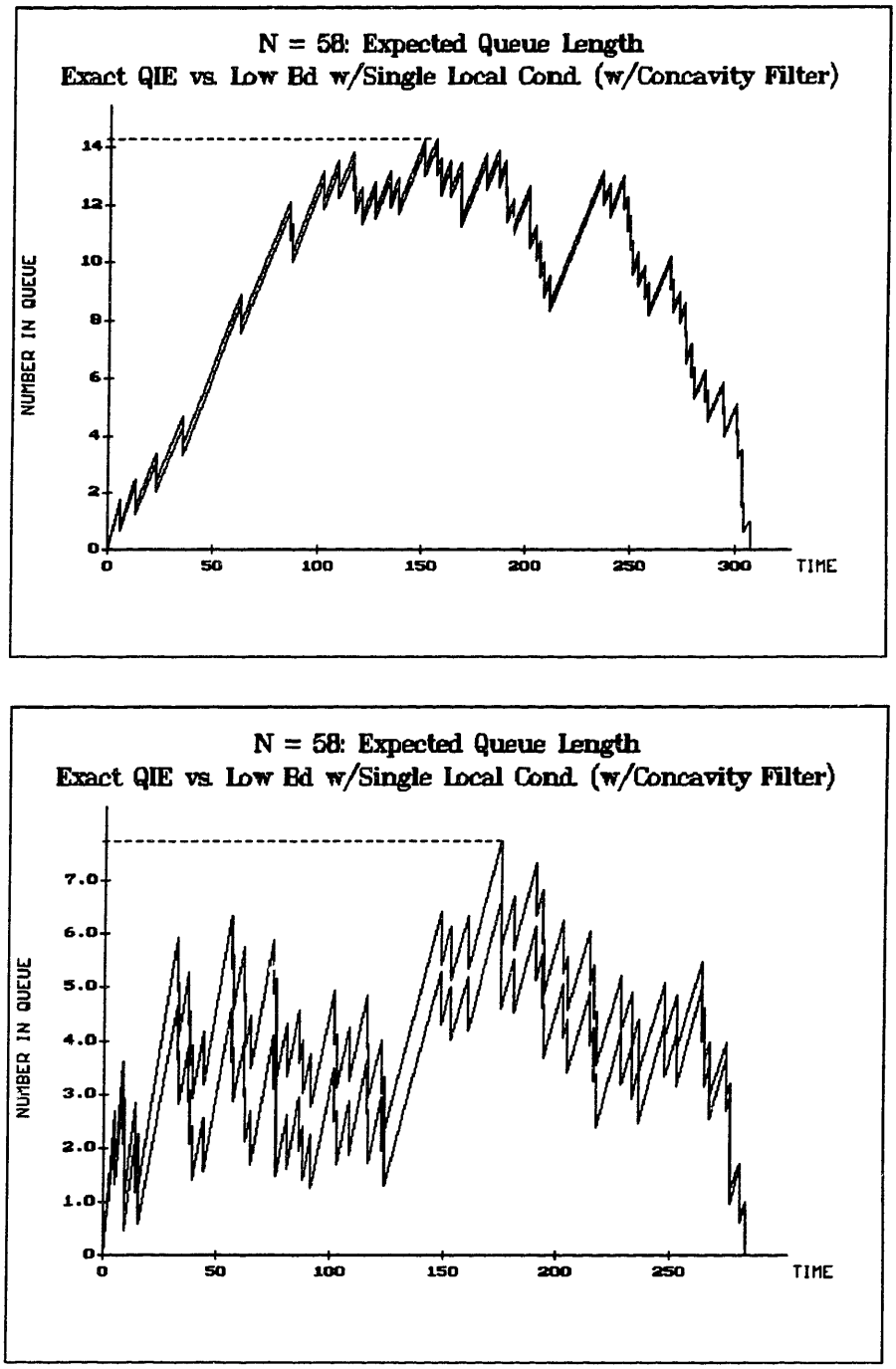


Figure 6.11: Expected Queue Length for Two Congestion Periods of $N = 58$: Exact QIE vs. QIE^R, Single Local Condition, with Concavity Filter

Size of Cong. Pd.	Algorithm Used	E[L_Q|...]	E[W_Q|...] (minutes)	δ	Run Time (seconds)
	QIE	2.8066	11.5726	0	0.172
$N = 14$	QIE^R, SLC, NCF	2.5848	10.6581	0.2218	0.050
	QIE^R, SLC, CF	2.6379	10.8770	0.1687	0.051
	QIE	2.5231	11.5514	0	0.133
$N = 13$	QIE^R, SLC, NCF	2.0712	9.4823	0.4519	0.041
	QIE^R, SLC, CF	2.3126	10.5878	0.2105	0.042
	QIE	2.6238	8.6997	0	0.173
$N = 14$	QIE^R, SLC, NCF	2.2293	7.3917	0.3945	0.050
	QIE^R, SLC, CF	2.4109	7.9936	0.2129	0.051
	QIE	2.8649	12.6644	0	0.404
$N = 18$	QIE^R, SLC, NCF	2.2082	9.7612	0.6568	0.099
	QIE^R, SLC, CF	2.5288	11.1783	0.3362	0.101
	QIE	3.3870	12.0614	0	0.694
$N = 21$	QIE^R, SLC, NCF	2.7255	9.7058	0.6615	0.153
	QIE^R, SLC, CF	3.0194	10.7523	0.3676	0.154
	QIE	1.8193	6.4488	0	0.103
$N = 12$	QIE^R, SLC, NCF	1.2786	4.5323	0.5407	0.033
	QIE^R, SLC, CF	1.4959	5.3025	0.3234	0.034
	QIE	9.3605	49.5175	0	31.15
$N = 58$ (1)	QIE^R, SLC, NCF	8.7584	46.3322	0.6021	2.86
	QIE^R, SLC, CF	9.1113	48.1992	0.2492	2.86
	QIE	4.4114	21.5511	0	31.22
$N = 58$ (2)	QIE^R, SLC, NCF	2.3265	11.3658	2.0849	2.87
	QIE^R, SLC, CF	3.2942	16.0933	1.1172	2.87

Table 6.5: Comparison of QIE and QIE^R, Single Local Condition (SLC), No Concavity Filter (NCF) and Concavity Filter (CF), for Eight Congestion Periods

We have presented two implementations of the QIE^R algorithm, both of which provide quite good lower bounds to the actual QIE calculations, yet cut down significantly on the running time required. Substantial work remains to be done to determine how to choose the set of conditions to use in the multiple global conditions implementation, both in terms of improving the bound and reducing the run times.

6.3 Algorithm Based on Restricting the Maximum Queue Length

The second idea that was introduced in Chapter 5 as a way to change the conditioning inequalities was the notion of assuming a maximum queue length. That is, we conjectured that an approximation to the expected cumulative number of arrivals function could be found by disregarding large-queue events, thereby avoiding calculation of their (presumably small) probabilities. We showed in Chapter 5 that introducing a maximum queue length constraint, i.e., conditioning on the maximum queue length being less than or equal to some value Q, also leads to a lower bound on the values in the β-matrix and hence on the expected cumulative number of arrivals by time t. We now show the specifics of implementing this algorithm, and show how the computational complexity is reduced by the zeroing out of many elements of the three matrices.

As already stated, the condition that we will add to $\mathcal{E}^S(\mathbf{t})$ is that the queue length never exceed the length Q during the congestion period in question. Because we have defined $Q(t)$ as a right-continuous function, this means that, between t_{i-1} and t_i, the queue length may only increase, until the instant t_i, at which time $Q(t_i)$ is decremented by one from its value at $Q(t_i^-)$. So in order to ensure that the queue length never exceed Q during the congestion period, we must require $Q(t_i) \leq Q-1, i = 1, 2, \ldots, N$. By using Equation 2.12, it is easy to see what this constraint does to the

bounds on the $A(t_i)$'s:

$$\begin{aligned}
\text{Condition from } \mathcal{E}^S(\mathbf{t}): \quad & i \leq A(t_i) \leq N \\
\text{Additional Condition:} \quad & Q(t_i) \leq Q-1 \\
\Longleftrightarrow \quad & A(t_i) \leq i+Q-1 \\
\text{Total Conditions:} \quad & i \leq A(t_i) \leq \min(N, i+Q-1) \\
\Longleftrightarrow \quad & i \leq A(t_i) \leq i+Q-1, \quad i=1,2,\ldots,N-Q \\
& i \leq A(t_i) \leq N, \quad i=N-Q+1, N-Q+2,\ldots,N
\end{aligned}$$

So this means that we have for our set Q of bounds on the $A(t_i)$'s:

$$\begin{aligned}
l_i &= i, \quad i=0,1,\ldots,N \\
u_0 &= 0 \\
u_i &= i+Q-1, \quad i=1,2,\ldots,N-Q \\
u_i &= N, \quad i=N-Q+1, N-Q+2,\ldots,N
\end{aligned}$$

(Here, we are using Q both to represent the maximum queue length that might have been achieved during the congestion period, and also to represent the set of bounds under $\mathcal{E}^S(\mathbf{t})$ and the additional condition that the maximum queue length did not exceed Q: the context should make clear which definition is being used, and any confusion is outweighed by the confusion inherent in introducing yet another symbol.) Note that we also have some implicit constraints on the value of Q, namely, we require $1 \leq Q \leq N$, since we have to have at least one person in queue prior to every t_i in order for the congestion period to continue, and since N is the maximum queue length achievable for a congestion period of N customers.

We now proceed to find the α^Q-, η^Q-, and β^Q-matrices, using the above specific definitions for the l_i's and the u_i's, and using the results of Chapter 5. These results actually follow quite readily from Equations 5.11 through 5.22 and so will not be derived in detail here, but are presented via the following claims. We begin with the α^Q-matrix.

Claim 6.7

$$\begin{aligned}
\alpha^Q_{k1}(\mathbf{t}) &= \left(\frac{t_1}{t_N}\right)^k, \quad k = 1,2,\ldots,Q \\
\alpha^Q_{ki}(\mathbf{t}) &= 0, \quad k = 0,1,\ldots,i-1 \text{ or } k = i+Q, i+Q+1,\ldots,N, \\
&\quad i = 1,2,\ldots,N \\
\alpha^Q_{ki}(\mathbf{t}) &= \sum_{j=0}^{k-i+1} \alpha^Q_{(k-j),(i-1)}(\mathbf{t}) \times \binom{k}{j} \left(\frac{t_i - t_{i-1}}{t_N}\right)^j, \\
&\quad k = i, i+1,\ldots,\min(i+Q-1,N), \quad i = 2,3,\ldots,N
\end{aligned}$$

Proof: Most of the above follows directly from Equations 5.11 through 5.14. The only explanation required is for the lower limit on the sum in the last part of the claim above. Consider the situation in which $k = i+Q-1$ and $i = 2,3,\ldots,N-Q+1$. Then u_{i-1} is actually equal to $i+Q-2$, so that $k-u_{i-1} = 1$, rather than the zero indicated as the lower limit. But we claim that, in this special case, the $j = 0$ term is zero, so starting the sum at zero is actually valid. This can be seen by considering the first term of the sum, which, for the values of k and i above, and $j = 0$, is $\alpha^Q_{(i+Q-1),(i-1)}(\mathbf{t})$. But this expression has the value of its first subscript greater than the value of its second subscript by the amount Q, and so, by the second part of the claim above, this α^Q term, and hence the whole $j = 0$ term, is zero. ∎

We now continue with the η^Q-matrix.

Claim 6.8

$$\begin{aligned}
\eta^Q_{Ni}(\mathbf{t}) &= 1, \quad i = N-Q+1, N-Q+2,\ldots,N-1 \\
\eta^Q_{(N-1),(N-1)}(\mathbf{t}) &= \frac{t_N - t_{N-1}}{t_N} \\
\eta^Q_{ki}(\mathbf{t}) &= 0, \quad k = 0,1,\ldots,i-1 \text{ or } k = i+Q, i+Q+1,\ldots,N, \\
&\quad i = 0,1,\ldots,N-1 \\
\eta^Q_{ki}(\mathbf{t}) &= \sum_{j=0}^{\min(N-k,i+Q-k)} \eta^Q_{(k+j),(i+1)}(\mathbf{t}) \times \binom{N-k}{j} \left(\frac{t_{i+1} - t_i}{t_N}\right)^j, \\
&\quad k = i, i+1,\ldots,\min(N-1,i+Q-1), \quad i = 0,1,\ldots,N-2
\end{aligned}$$

Proof: Most of the above follows directly from Equations 5.15 through 5.18. The only explanations required are for the limits on the sum in the last part of the claim above. First consider the upper limit, which, from Equation 5.18, should be $u_{i+1} - k$. We know we can express $u_i = \min(N, i+Q-1)$ so that we also have $u_{i+1} = \min(N, i+Q)$ and finally, $u_{i+1} - k = \min(N-k, i+Q-k)$. Now consider the lower limit, which, from Equation 5.18, should be $\max(0, l_{i+1} - k)$. We know $l_{i+1} = i+1$, so we will have the maximum equal to zero except in the single case when $k = i$. But we claim that, in this special case, the $j = 0$ term is zero, so starting the sum at zero is actually valid. This can be seen by considering the first term of the sum, which, for the values $k = i$ and $j = 0$, is $\eta^Q_{i,(i+1)}(\mathbf{t})$. But this expression has a first subscript whose value is one less than that of its second subscript, and so, by the third part of the claim above, this η^Q term, and hence the whole $j = 0$ term, is zero. ∎

Finally, we have the following for the β^Q-matrix.

Claim 6.9

$$\begin{aligned}
\beta^Q_{ki}(\mathbf{t}) &= 1, \quad k = 1, 2, \ldots, i, \quad i = 1, 2, \ldots, N \\
\beta^Q_{ki}(\mathbf{t}) &= 0, \quad k = i+Q, i+Q+1, \ldots, N, \quad i = 1, 2, \ldots, N-Q \\
\beta^Q_{Ni}(\mathbf{t}) &= \frac{\alpha^Q_{Ni}(\mathbf{t})}{\alpha^Q_{NN}(\mathbf{t})}, \quad i = N-Q+1, N-Q+2, \ldots, N-1 \\
\beta^Q_{ki}(\mathbf{t}) &= \beta^Q_{(k+1),i}(\mathbf{t}) + \frac{1}{\alpha^Q_{NN}(\mathbf{t})} \left\{ \binom{N}{k} \alpha^Q_{ki}(\mathbf{t}) \eta^Q_{ki}(\mathbf{t}) \right\}, \\
&\qquad k = i+1, i+2, \ldots, \min(N-1, i+Q-1), \quad i = 1, 2, \ldots, N-2
\end{aligned}$$

Proof: These expressions follow directly from Equations 5.19 through 5.22 and require no elaboration. ∎

We now have a complete method for determining the full β^Q-matrix, which in turn may be used in the standard way to generate all of the queue statistics that are generated by the original QIE algorithm. We call this algorithm the QIE^Q algorithm. Now let us examine its computational complexity. We first generate the α^Q- and η^Q-matrices. Consider the last parts of both Claims 6.7 and 6.8. In both cases, for

column i, we calculate at most Q elements in that column. To find the complexity of calculating each of these elements for the α^Q-matrix, we see that we have the most terms to add up when $k = i + Q - 1$: in this case, we must add up $Q + 1$ terms to find the single element, $\alpha^Q_{(i+Q-1),i}(\mathbf{t})$. Similarly, for the η^Q-matrix, we have the most terms to add up when $k = i$: in this case, we must also add up $Q + 1$ terms to find the single element, $\eta^Q_{ii}(\mathbf{t})$. Hence, calculation of each column of these matrices is $O(Q^2)$, and since each matrix has N columns, calculation of the entire α^Q- and η^Q-matrices is an $O(NQ^2)$ operation. Finally, we multiply elements of these matrices to generate elements of the β^Q-matrix. There are N columns of the β^Q-matrix to be filled in; each of these columns has at most $Q-1$ elements to be calculated (see Claim 6.9); and each of these elements requires an $O(1)$ computation: hence, computation of the β^Q-matrix, after computation of the other two matrices, is $O(NQ)$, and the computational complexity of the entire QIE^Q algorithm is $O(NQ^2)$.

This is quite a savings over the standard QIE algorithm, and, for large congestion periods, even a modest value for Q (on the order of 10 or 15) can result in a fairly tight bound (see Section 6.4). The bound can be improved even further by again forcing concavity on the function, $\mathrm{E}[A(t)|\mathcal{E}^Q(\mathbf{t})]$. That is, one would find the $\mathrm{E}[A(t)|\mathcal{E}^Q(\mathbf{t})]$ function, and then take the concave hull of that function in order to get a tighter lower bound on $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$.

An important point to note with regard to the QIE^Q algorithm is that, in some environments, we may actually know that, during some congestion period, the queue length never exceeded the value, Q. For example, we may have a finite-capacity waiting room, whose capacity we know to be Q, and by some means of observation, we may be able to determine that the capacity was never exceeded during a given congestion period. Incorporating this information into the QIE model is exactly equivalent to assuming bounds, Q, on the $A(t_i)$'s, as described above. So we could run the QIE^Q algorithm on the given data set, and, not only would we realize computational savings, but we should also actually do a better job in estimating the expected cu-

mulative number of arrivals by time t than we would do, running the original QIE algorithm. This is due to the fact that the original algorithm takes into account all possible arrival patterns, including those that generate large queues; but, given that the queue length was limited, we know that those particular arrival patterns did not occur, and by omitting them from consideration, we are actually more likely to be close to the actual arrival pattern.

And what if, during a congestion period, the capacity of our waiting room were exceeded and we were given the times of transition between the states "capacity exceeded" and "capacity not exceeded?" As in the case above, we are given some partial queue length information which should make our estimates of the queue statistics better than if we did not have the information. Also, since we are given perfect information as to the queue length at several points during the congestion period, it should be possible to break up the congestion period and analyze each section separately, thereby also reducing the computational burden. These are the ideas that are explored in the next chapter.

In the next section, we present the results of several sample runs of the QIE^Q algorithm. We consider two cases. First, we consider using the algorithm as a lower bound to $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$, and we examine how good the bound is and how fast the runtimes are, relative to the original QIE algorithm. Second, we look at actual simulations with queue lengths less than or equal to Q, and we calculate $\mathrm{E}[A(t)|\mathcal{E}^Q(\mathbf{t})]$ and compare its performance to that of $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$: in this case, we expect the QIE^Q algorithm to be better and to have faster runtimes. These expectations are indeed borne out.

6.4 Computational Results of the QIE^Q Algorithm

We include here results from simulation of an M/M/1 queue. These data were generated by three simulation runs with Poisson arrivals at rate 10 per hour, a single

server, and exponential service times with expected values of 3 minutes for the first run (giving a value of $\rho = 0.5$) and 4 minutes for the last two runs (giving a value of $\rho = 0.67$). Runs were on a 386/387-based Northgate Computer Systems PC. Each run time given below is an average of 3000 run times (for shorter runs, presented to thousandths of a second) or 1000 run times (for longer runs, presented to hundredths of a second) from different runs of the program on the same data. This averaging was necessary because the system clock is only updated every 0.0549254 seconds [Scan 83], so to get accuracy greater than 0.1 seconds, many runs must be averaged.

We compare the QIE^Q algorithm to the standard QIE algorithm. First, we consider the case in which the QIE^Q algorithm is used as a lower bound to the exact QIE algorithm. Then, we examine the case in which maximum queue length data are available, so that the QIE^Q algorithm may be used to improve our estimates. The statistics that are used for comparison of the algorithms include: $\mathrm{E}[L_Q|\ldots]$, the time-averaged number of customers in queue; $\mathrm{E}[W_Q|\ldots]$, the average wait in queue; δ, the approximation error, which we define to be the absolute area between $\mathrm{E}[Q(t)|\mathcal{E}^S(\mathbf{t})]$ as generated by the exact QIE algorithm, and the lower bound algorithm's estimate of the same function, divided by the duration of the congestion period, t_N; and ϵ, the time-averaged error, defined to be the absolute area between the actual queue length graph and the QIE (or QIE^Q) expected queue length graph, divided by the total time of the congestion period. The run times to generate the beta-matrix for the different algorithms are also compared.

When we are not given data regarding the maximum queue length but have a long congestion period to analyze, we may wish to lower bound the QIE output by using the QIE^Q algorithm. Examples of this are presented in Figures 6.12 and 6.13. These graphs illustrate the mean queue length as estimated by the QIE algorithm, compared with the same quantity as estimated by the QIE^Q algorithm, for congestion periods of 18, 21, and 58 customers. We present these comparisons for values of Q of 5, 8, and 10 for the two shorter congestion periods, and $Q = 5$, 10, and 15 for the two $N = 58$.

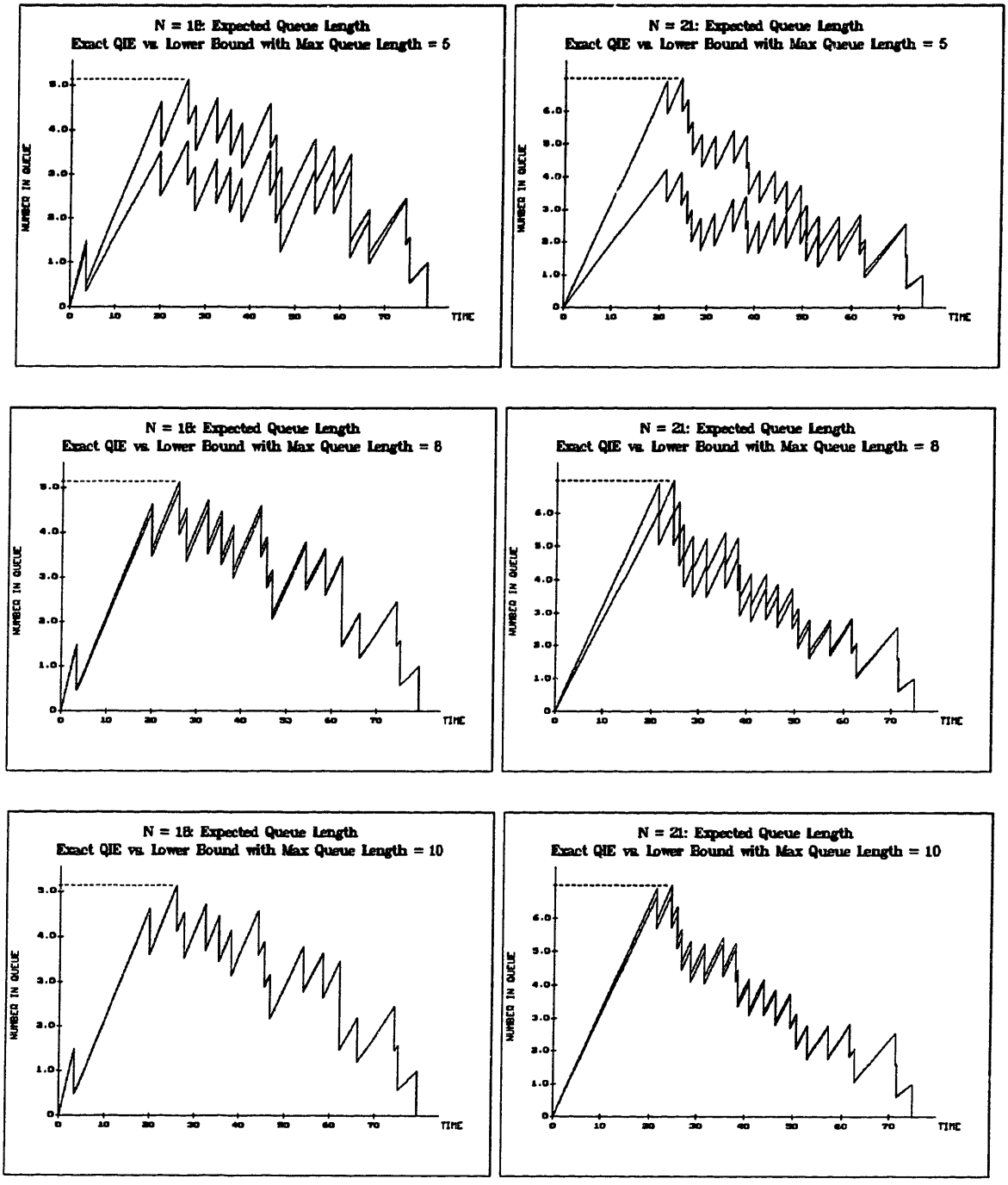


Figure 6.12: Expected Queue Length for Congestion Periods of $N = 18$ and $N = 21$: QIE vs. QIE^Q, with $Q = 5, 8$, and 10, No Concavity Filter

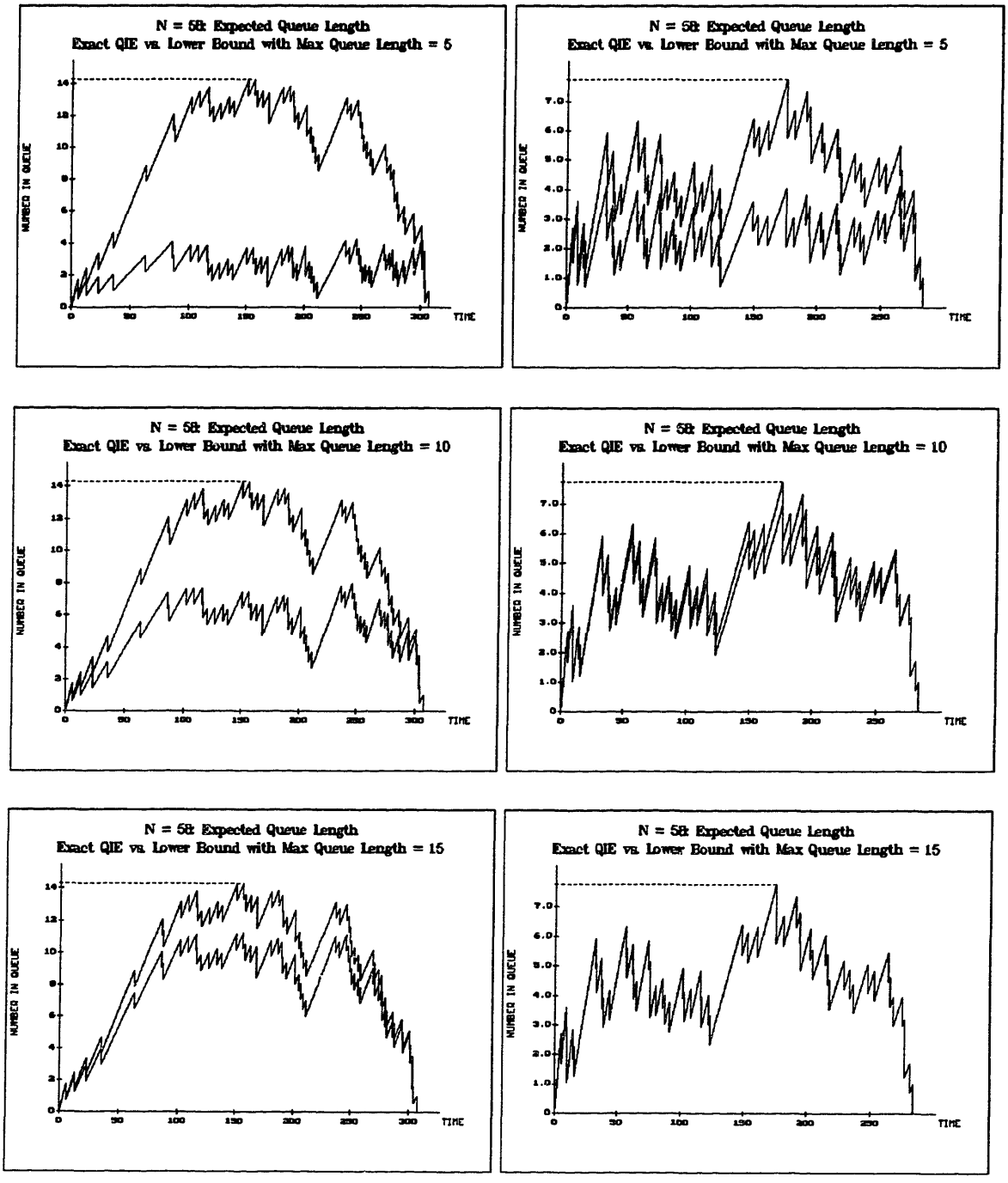


Figure 6.13: Expected Queue Length for Two Congestion Periods of $N = 58$: QIE vs. QIE^Q, with $Q = 5, 10$, and 15, No Concavity Filter

congestion periods. Note that as Q increases, the bound gets tighter, and the values of $\mathrm{E}[L_Q|\mathcal{E}^Q(\mathbf{t})]$ also increase, up to the maximum given by the standard QIE algorithm, while the value of δ decreases to 0. Also note that, for the two shorter congestion periods and the second $N = 58$ congestion period with $Q = 10$, the QIE^Q algorithm is very close to the exact QIE algorithm yet requires considerably less running time. For the first $N = 58$ congestion period, we must have at least $Q = 15$ to obtain a bound that is reasonably close, but again, the running time is still far less than that of the exact QIE algorithm. Comparisons of the values of the time-average queue length, the expected wait in queue, the approximation error, and the running times for these data are provided in Table 6.6.

Recall that we suggested that the QIE^Q algorithm could be improved by taking the concave hull of the expected cumulative number of arrivals by time t. This concave hull is still a lower bound, but is tighter than that generated by the QIE^Q algorithm alone. In Figures 6.14 and 6.15, we present the output of the same congestion periods and values of Q we considered above, but this time we add concavity filtering to the QIE^Q algorithm. Comparisons of the values of the time-average queue length, the expected wait in queue, the approximation error, and the running times for these data are provided in Table 6.7. Note that the run times are not significantly increased by the additional task of the concavity filtering, but that in many instances, the bound is tightened quite significantly after the concave hull has been taken. This is particularly evident in the first $N = 58$ congestion period, where, even with $Q = 5$, we get quite a good bound after concavity filtering.

Next, we compare the QIE^Q algorithm to the standard QIE algorithm, when we actually have maximum queue length data available. Consider a congestion period with $N = 11$, as shown in Figure 6.16, and suppose we have a pressure-sensitive mat at position $Q + 1$ in the queue, and we determine that, over the course of the given congestion period, the mat was never depressed, so the queue length could not have exceeded Q. The figure depicts the exact queue length for the period (which,

Size of Cong. Pd.	Algorithm Used	E[L_Q|...]	E[W_Q|...] (minutes)	δ	Run Time (seconds)
$N = 18$	QIE	2.8649	12.6644	0	0.404
	QIE^Q, $Q = 5$	2.1541	9.5221	0.7108	0.084
	QIE^Q, $Q = 8$	2.7703	12.2461	0.0946	0.167
	QIE^Q, $Q = 10$	2.8568	12.6282	0.0082	0.235
$N = 21$	QIE	3.3871	12.0615	0	0.694
	QIE^Q, $Q = 5$	2.1681	7.7205	1.2190	0.102
	QIE^Q, $Q = 8$	2.9874	10.6383	0.3996	0.208
	QIE^Q, $Q = 10$	3.2773	11.6706	0.1098	0.301
$N = 58$ (1)	QIE	9.3605	49.5175	0	31.15
	QIE^Q, $Q = 5$	2.4845	13.1428	6.8761	0.423
	QIE^Q, $Q = 10$	5.0879	26.9151	4.2726	1.20
	QIE^Q, $Q = 15$	7.5336	39.8530	1.8269	2.69
$N = 58$ (2)	QIE	4.4114	21.5511	0	31.22
	QIE^Q, $Q = 5$	2.4109	11.7779	2.0005	0.423
	QIE^Q, $Q = 10$	4.0442	19.7574	0.3672	1.20
	QIE^Q, $Q = 15$	4.4041	21.5154	0.0073	2.69

Table 6.6: Comparison of QIE and QIE^Q Algorithms, for Four Congestion Periods, No Concavity Filter

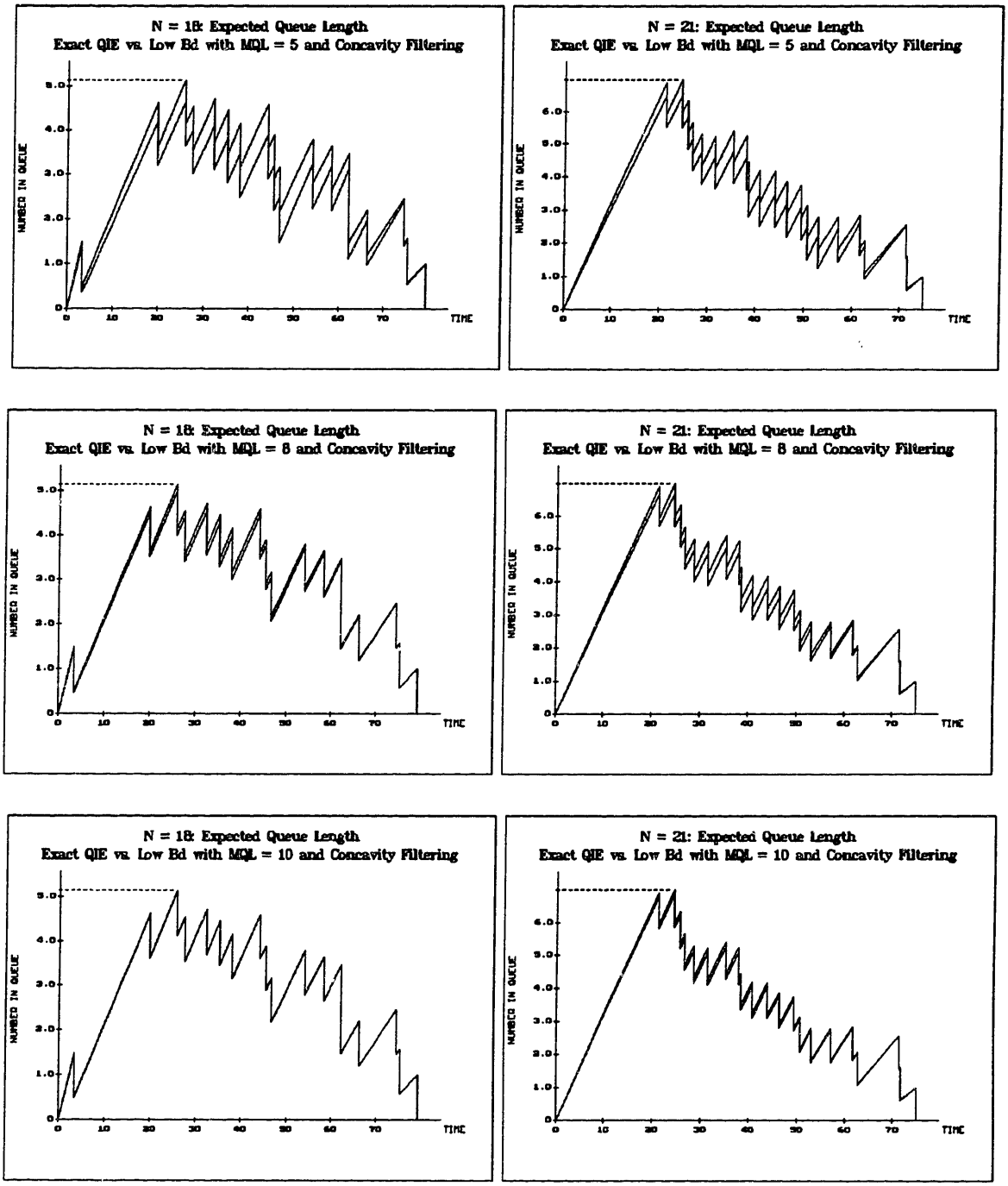


Figure 6.14: Expected Queue Length for Congestion Periods of $N = 18$ and $N = 21$: QIE vs. QIE^Q, with $Q = 5, 8$, and 10, with Concavity Filter

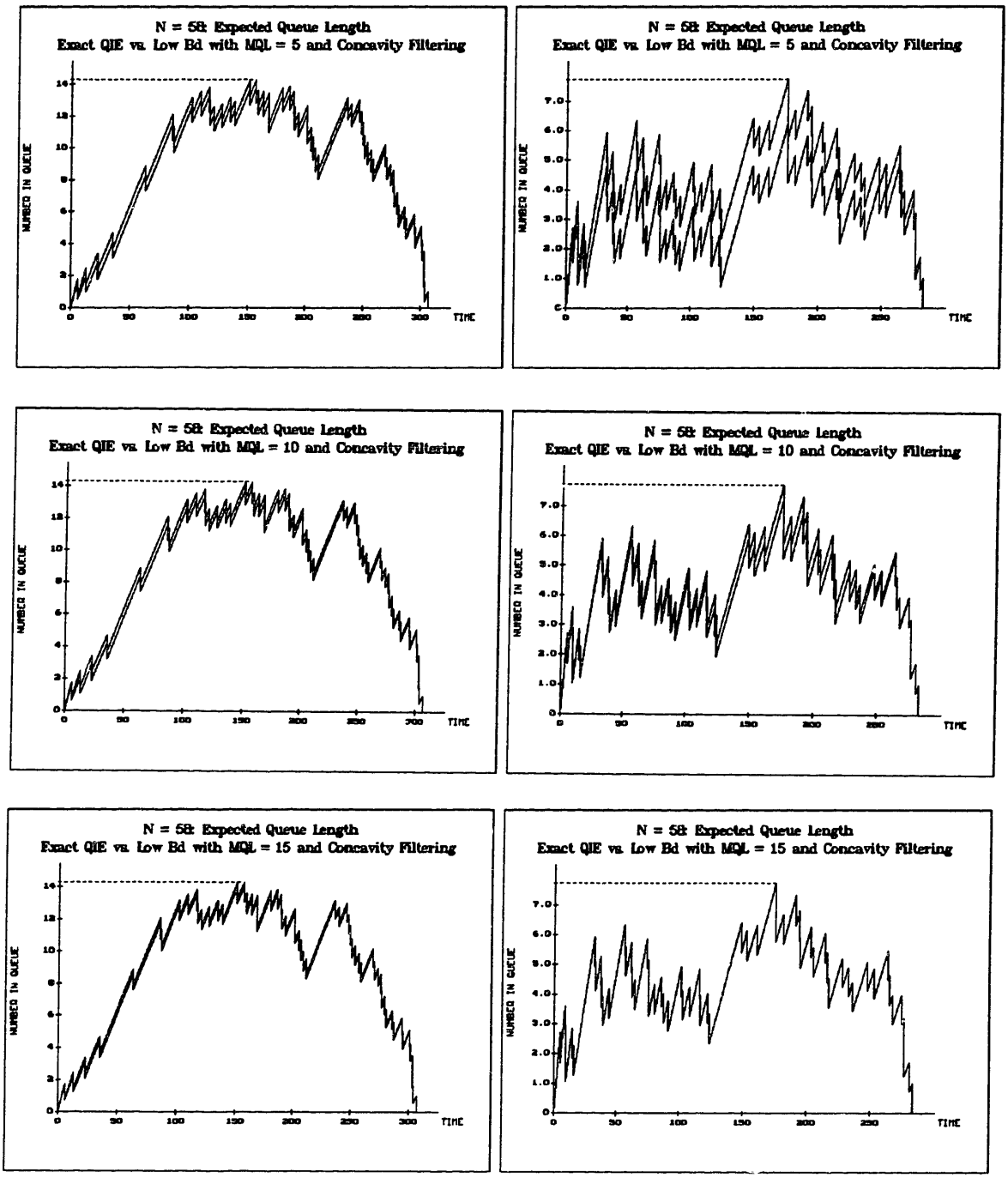


Figure 6.15: Expected Queue Length for Two Congestion Periods of $N = 58$: QIE vs. QIE^Q, with $Q = 5, 10$, and 15, with Concavity Filter

Size of Cong. Pd.	Algorithm Used	E[L_Q|...]	E[W_Q|...] (minutes)	δ	Run Time (seconds)
$N = 18$	QIE	2.8649	12.6644	0	0.404
	QIE^Q, $Q = 5$	2.4668	10.9045	0.3981	0.088
	QIE^Q, $Q = 8$	2.7791	12.2849	0.0858	0.173
	QIE^Q, $Q = 10$	2.8568	12.6282	0.0082	0.242
$N = 21$	QIE	3.3871	12.0615	0	0.694
	QIE^Q, $Q = 5$	3.0083	10.7125	0.3788	0.108
	QIE^Q, $Q = 8$	3.1955	11.3793	0.1915	0.215
	QIE^Q, $Q = 10$	3.3098	11.7863	0.0773	0.308
$N = 58$ (1)	QIE	9.3605	49.5175	0	31.15
	QIE^Q, $Q = 5$	8.8658	46.9004	0.4947	0.427
	QIE^Q, $Q = 10$	8.9758	47.4823	0.3847	1.20
	QIE^Q, $Q = 15$	9.1104	48.1942	0.2501	2.70
$N = 58$ (2)	QIE	4.4114	21.5511	0	31.22
	QIE^Q, $Q = 5$	3.1403	15.3413	1.2711	0.435
	QIE^Q, $Q = 10$	4.0915	19.9884	0.3199	1.23
	QIE^Q, $Q = 15$	4.4041	21.5154	0.0073	2.73

Table 6.7: Comparison of QIE and QIE^Q Algorithms, for Four Congestion Periods, with Concavity Filter

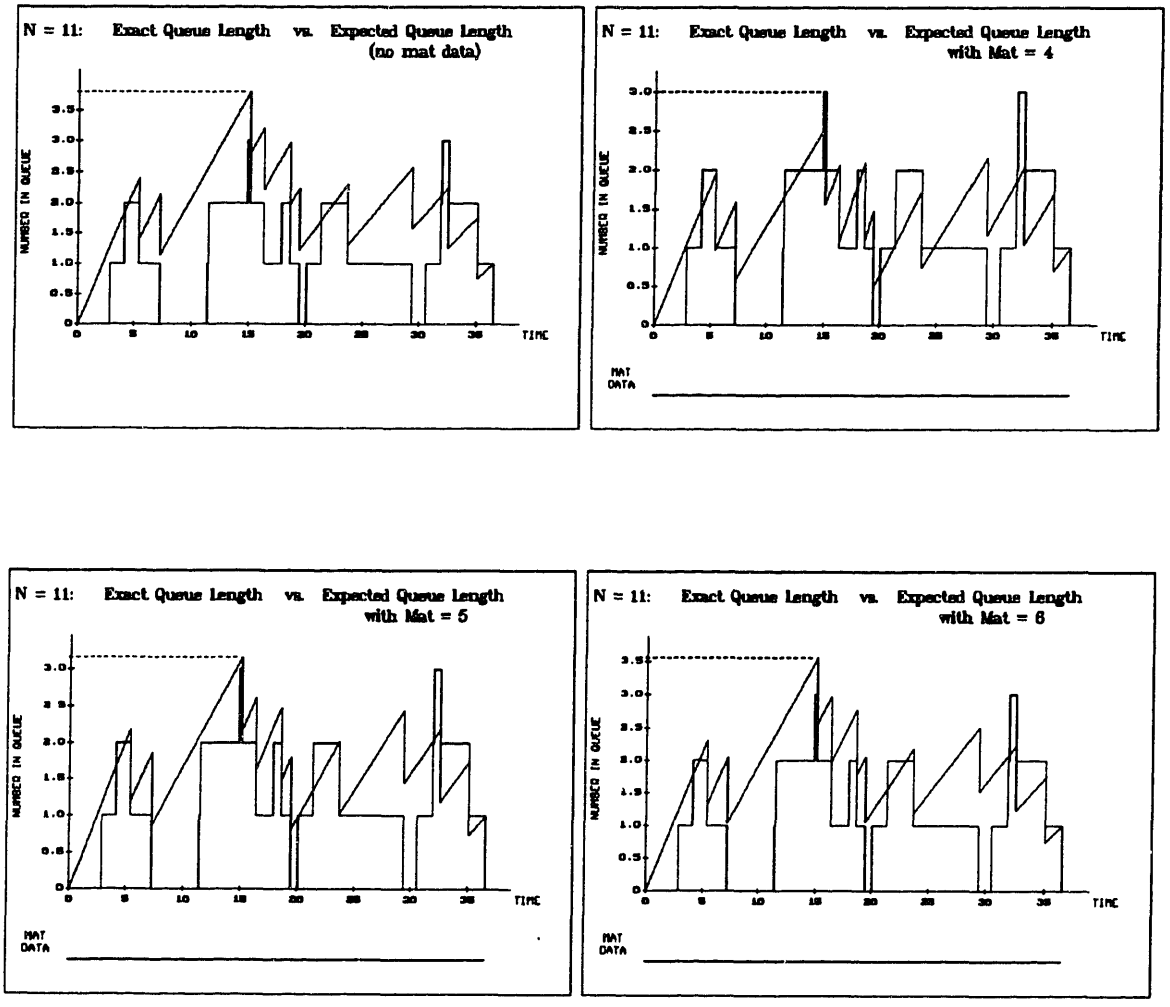


Figure 6.16: Exact Queue Length vs. Expected Queue Length for a Congestion Period with $N = 11$: Standard QIE and QIE^Q with $Q = 3, 4$, and 5

Algorithm Used	E[L_Q|...] (actual = 1.1161)	E[W_Q|...] (minutes) (actual = 3.7160)	ϵ	Run Time (seconds)
QIE	1.9230	6.4025	0.9249	0.080
QIE^Q, $Q = 3$	1.3654	4.5460	0.5309	0.027
QIE^Q, $Q = 4$	1.6467	5.4826	0.6947	0.034
QIE^Q, $Q = 5$	1.8107	6.0286	0.8293	0.043

Table 6.8: Comparison of QIE and QIE^Q Algorithms (with Max Queue Length Data Given) for a Congestion Period with $N = 11$

in fact, never exceeds 3) and, superimposed, depicts the QIE (or QIE^Q) expected queue length. As can be seen, the standard QIE overestimates the expected queue length, while the QIE^Q estimate with $Q = 3$ is quite close to the actual data. With $Q = 5$, the QIE^Q is actually quite close to the standard QIE output: even though the standard QIE algorithm considers many more possible events (all those with queue length greater than 5 and less than 12), those events are of relatively low probability and so do not have much impact on the final expected queue length. Comparative statistics for these congestion periods are provided in Table 6.8. Notice that the algorithm with the shortest run time gives the best estimate of the data!

We continue with the idea of having partial queue-length data available in the next chapter. We again consider having a mat at position $Q + 1$ in the queue, but this time we allow mat transitions to occur during the congestion period. This allows us to partition the congestion period and analyze the partitions separately, thereby giving us more accurate estimates in shorter runtimes.

Chapter 7

Adding Partial Queue Length Information to Transactional Data

The QIE^Q algorithm discussed in the last chapter raises the following interesting issue. Suppose that the queue actually had some sort of sensing mechanism, for example a pressure-sensitive mat placed at position M in the queue, such that we would be able to detect all queue transitions from $M-1$ to M, as well as all transitions from M to $M-1$. Here, we ignore the transients of customers stepping over the mat just to achieve a position in queue which is less than M. We also assume that there is no queue transit delay: that is when a customer leaves the queue to enter service, the entire queue immediately shifts forward one position. Then, for a congestion period during which *no* transitions are observed, we have exactly the situation in the previous chapter, with $M = Q + 1$: the mat information allows us to discard the large-queue events which we know did not occur.

For a congestion period during which mat transitions *are* observed, we clearly have new information at the points of transition and so should be able to use this information to improve the QIE performance. In addition, because the state of the queue is known exactly at the points of transition, we may break the congestion period down into "congestion period partitions" and analyze each of these separately, thereby

significantly reducing the complexity of the computation. Specifically, assume, as before, that $t_1, t_2, \ldots, t_N$ are the times of service commencement for customers 1 through N. Now define a *mat cycle* as any period of time during which the mat is continuously depressed. Similarly, define a *non-mat cycle* as any period of time within a single congestion period between two mat cycles. Say there are Γ mat cycles during the congestion period, with $\Gamma \geq 0$ (hence, there are $\Gamma - 1$ non-mat cycles whenever $\Gamma \geq 1$). Then define $d_1, d_2, \ldots, d_\Gamma$ as the times at which the mat is depressed; and define $r_1, r_2, \ldots, r_\Gamma$ as the times at which the mat is released. Note that any given mat release time must coincide with one of the t_i's. Figure 7.1 provides an example of a congestion period which has $N = 12$ customers who must wait in queue, with a mat position at $M = 3$, and two mat cycles ($\Gamma = 2$).

Whenever $\Gamma > 0$, the congestion period can be broken down into $2\Gamma + 1$ congestion period partitions, each of which must be one of four distinct types. The first type of congestion period partition comprises the time $(0, d_1]$, i.e. the time from the beginning of the congestion period until the first time that there are M customers in queue (there must be at least $M - 1$ arrivals prior to time d_1). The second type of congestion period partition comprises the time $(d_j, r_j]$, $j = 1, \ldots, \Gamma$, a single mat cycle. This is the time between any depression of the mat and the subsequent release. Note that the queue can grow to any size greater than or equal to M during this congestion period partition. The third type of congestion period partition comprises the time $(r_j, d_{j+1}]$, $j = 1, \ldots, \Gamma - 1$, a single non-mat cycle (this type exists only when $\Gamma \geq 2$). This is the time within a single congestion period between any mat release and subsequent depression. The queue length can be anything between 0 and $M - 1$ during this congestion period partition, although no server may be made idle, as this would cause the end of the congestion period. Finally, the fourth type of congestion period partition comprises the time $(r_\Gamma, t_N]$, i.e. the time between the end of the last mat cycle and the end of the congestion period. Including t_N, there must be at least $M - 1$ departures during this congestion period partition, to empty out the $M - 1$

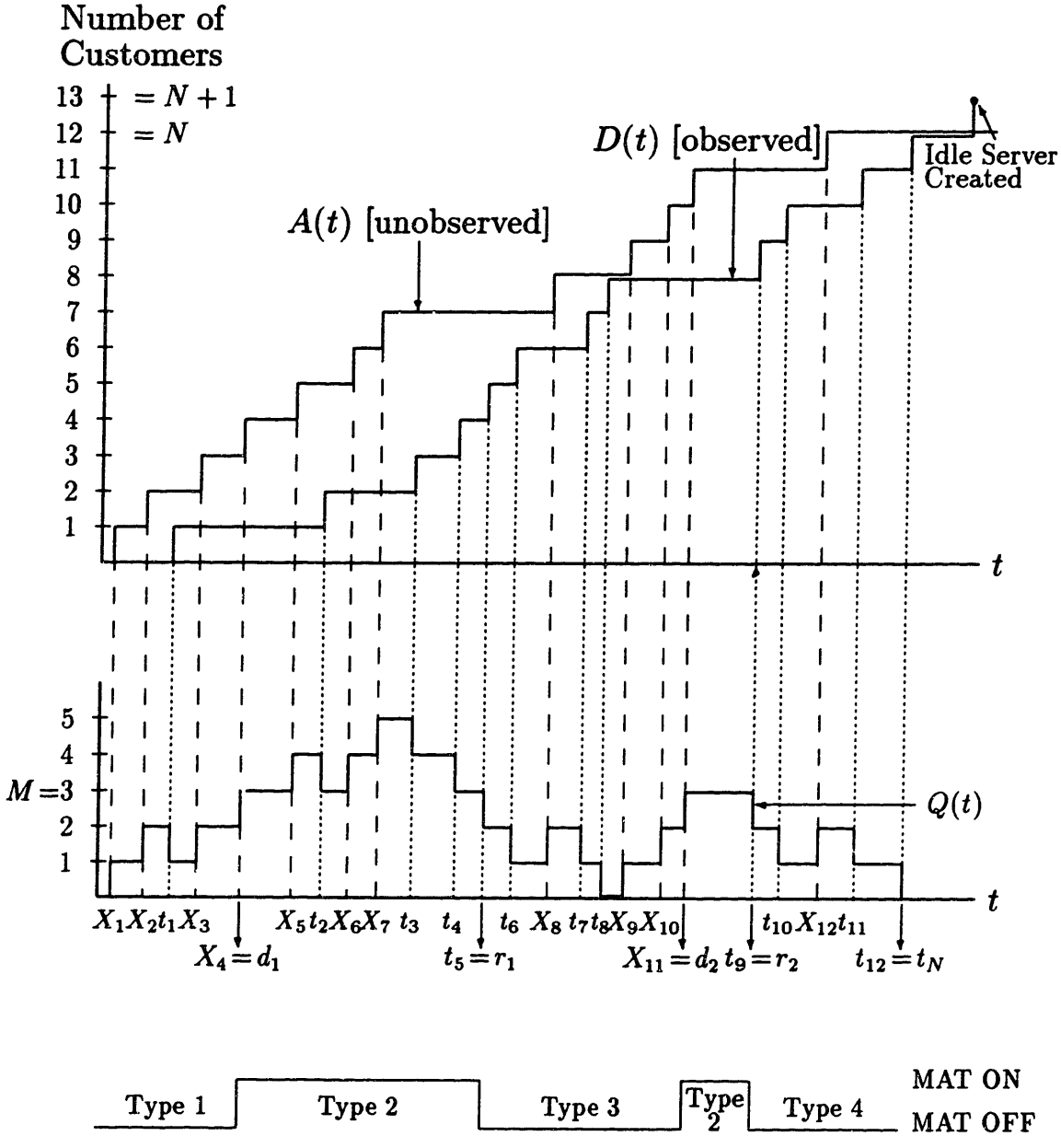


Figure 7.1: Sample Function for a Congestion Period with $N = 12$, $M = 3$, and $\Gamma = 2$

customers who are in queue at time r_Γ. Again looking at Figure 7.1, because there are two mat cycles ($\Gamma = 2$), there are two type 2 congestion period partitions, and one each of types 1, 3, and 4.

In the next section, we introduce a general algorithm which may be used to analyze all of the four types of congestion period partitions. Then, in each of the following four sections, we specialize the algorithm to each of the four types of partitions. As we analyze each congestion period partition, we will be partially filling in another β-matrix, this one with entries $\beta_{ki}^{\mathcal{M}}(\boldsymbol{\tau})$, where

$$\beta_{ki}^{\mathcal{M}}(\boldsymbol{\tau}) \equiv \Pr[A(\tau_i) \geq k | \mathcal{E}^S(\mathbf{t}), \mathcal{M}]$$

and where τ_i is any time at which information is available (i.e., the t_k's and the mat depression times, the d_j's). Here, $\mathcal{M}$ is used to denote the mat data, as contained in the telegraph wave, such as that depicted at the bottom of Figure 7.1. We also discuss the computational complexity of each of the specialized algorithms. Finally, we discuss how to complete the $\beta^{\mathcal{M}}$-matrix and how to use it to derive the queue statistics of interest. We call the algorithm which fills in the $\beta^{\mathcal{M}}$-matrix the QIE$^{\mathcal{M}}$ algorithm.

7.1 Algorithm to Find Arrival-Time Probabilities for Congestion Period Partitions

The fact that we have perfect information as to the queue length (and hence, the number of arrivals) at both the beginning and end of each of the congestion period partitions suggests that perhaps we could use the algorithm introduced in Section 5.3 in order to find the arrival-time probabilities of interest, the $\beta_{ki}^{\mathcal{M}}(\boldsymbol{\tau})$'s. However, there are four requirements for that algorithm which are not fulfilled by some or all of the congestion period partitions considered here. First, that algorithm requires that the cumulative number of arrivals at the beginning of the period to be analyzed

be zero: only the type 1 partition satisfies that requirement. Second, the algorithm requires that the number in queue at the end of the period to be analyzed also be zero: only the type 4 partition satisfies that requirement. The algorithm also requires that the number of arrivals during the period to be analyzed be equal to the number of departures: this is certainly not the case for the type 1 partition, since we start out the partition with zero customers in queue and end up with M in queue (so we must have had M more arrivals than departures). Finally, the algorithm assumes that we only have information at instants at which there were departures from the queue: in this case, we also have information at various arrival times. We now present a generalization of the algorithm presented in Section 5.3, which accounts for all of these discrepancies.

Consider a partition of an N-customer congestion period, such that we have perfect information as to the state of the queue at both the beginning and end of the segment or partition. Say that, during the partition, we have N_A arrivals, and let N_T represent the number of arrival times and departure times during the interval that we know with certainty. (In the standard algorithm, we have $N_A = N$ and $N_T = N$, i.e., we know only the N departure times with certainty.) We denote the beginning of the partition by τ_0 (note that this is not assumed to be zero) and each of the "times of interest" by $\tau_i, i = 1, 2, \ldots, N_T$. We also let $A_0 \equiv A(\tau_0)$ represent the cumulative number of arrivals to the system at the beginning of the interval. Here, we count only the arrivals who have had to wait in queue. Finally, we have that $A(\tau_{N_T}) = A_0 + N_A$. We would like to find the generalized β-quantities, $\beta^G_{ki}(\boldsymbol{\tau})$, defined as follows:

$$\begin{aligned} \beta^G_{ki}(\boldsymbol{\tau}) &\equiv \Pr[A(\tau_i) \geq k | \mathcal{E}^G(\mathbf{t})], \\ &\qquad i = 1, 2, \ldots, N_T, \quad k = A_0 + 1, A_0 + 2, \ldots, A_0 + N_A \\ \text{where} \quad \mathcal{E}^G(\mathbf{t}) &\equiv \bigcap_{j=0}^{N_T} \{l_j \leq A(\tau_j) \leq u_j\} \end{aligned}$$

where here the l_j's and u_j's are defined by the set of bounds G:

$$G \equiv \{l_0, l_1, \ldots, l_{N_T-1}, l_{N_T}, u_0, u_1, \ldots, u_{N_T-1}, u_{N_T}\}$$

$$= \{A_0, l_1, \ldots, l_{N_T-1}, A_0 + N_A, A_0, u_1, \ldots, u_{N_T-1}, A_0 + N_A\}$$

and the l_j's and u_j's must also conform to the last three equations in the set 5.1, with the boundary conditions as given above.

Then, we may proceed in a manner identical to that given in Section 5.3 to derive the values for $\beta^G_{ki}(\tau)$, using the following:

$$\begin{aligned}
\mathcal{E}^{A_0,N_A}(\tau) &\equiv \{A(\tau_0) = A_0\} \bigcap \{A(\tau_{N_T}) = A_0 + N_A\} \\
\alpha^G_{ki}(\tau) &\equiv \Pr[\mathcal{E}^{G\leq i}(\tau) | A(\tau_i) = k, \mathcal{E}^{A_0,N_A}(\tau)] \times \left(\frac{\tau_i - \tau_0}{\tau_{N_T} - \tau_0}\right)^{k-A_0} \\
\eta^G_{ki}(\tau) &\equiv \Pr[\mathcal{E}^{G\geq i}(\tau) | A(\tau_i) = k, \mathcal{E}^{A_0,N_A}(\tau)] \times \left(\frac{\tau_{N_T} - \tau_i}{\tau_{N_T} - \tau_0}\right)^{N_A-(k-A_0)}
\end{aligned}$$

The entire derivation will not be presented here since it is so close to the derivation of the algorithm in Section 5.3. Instead, we present the results for the three matrices. The α^G-matrix is given by:

$$\begin{aligned}
\alpha^G_{A_0,i}(\tau) &= 1, \quad \text{if } l_i = A_0, \quad i = 1, 2, \ldots, N_T - 1 \\
\alpha^G_{k1}(\tau) &= \left(\frac{\tau_1 - \tau_0}{\tau_{N_T} - \tau_0}\right)^{k-A_0}, \quad k = l_1, l_1+1, \ldots, u_1 \\
\alpha^G_{ki}(\tau) &= 0, \quad k = A_0, A_0 + 1, \ldots, l_i - 1 \text{ or } k = u_i+1, u_i+2, \ldots, A_0 + N_A, \\
&\quad i = 1, 2, \ldots, N_T \\
\alpha^G_{ki}(\tau) &= \sum_{j=\max(0,k-u_{i-1})}^{k-l_{i-1}} \alpha^G_{(k-j),(i-1)}(\tau) \times \binom{k - A_0}{j} \left(\frac{\tau_i - \tau_{i-1}}{\tau_{N_T} - \tau_0}\right)^j, \\
&\quad k = l_i, l_i+1, \ldots, u_i \text{ and } k > A_0, \quad i = 2, 3, \ldots, N_T
\end{aligned}$$

The η^G-matrix is defined by the following:

$$\begin{aligned}
\eta^G_{(A_0+N_A),i}(\tau) &= 1, \quad \text{if } u_i = A_0 + N_A, \quad i = 1, 2, \ldots, N_T - 1 \\
\eta^G_{k,(N_T-1)}(\tau) &= \left(\frac{\tau_{N_T} - \tau_{N_T-1}}{\tau_{N_T} - \tau_0}\right)^{N_A-(k-A_0)}, \quad k = l_{N_T-1}, l_{N_T-1}+1, \ldots, u_{N_T-1} \\
\eta^G_{ki}(\tau) &= 0, \quad k = A_0, A_0 + 1, \ldots, l_i - 1 \text{ or } k = u_i+1, u_i+2, \ldots, A_0 + N_A, \\
&\quad i = 0, 1, \ldots, N_T - 1
\end{aligned}$$

$$\eta^G_{ki}(\tau) = \sum_{j=\max(0,l_{i+1}-k)}^{u_{i+1}-k} \eta^G_{(k+j),(i+1)}(\tau) \times \binom{N_A-(k-A_0)}{j} \left(\frac{\tau_{i+1}-\tau_i}{\tau_{N_T}-\tau_0}\right)^j,$$
$$k = l_i, l_i+1, \ldots, u_i \text{ and } k < A_0 + N_A, \quad i = 0, 1, \ldots, N_T - 2$$

Finally, we have for the β^G-matrix:

$$\beta^G_{ki}(\tau) = 1, \quad k = A_0+1, A_0+2, \ldots, l_i, \; i = 1, 2, \ldots, N_T$$
$$\beta^G_{ki}(\tau) = 0, \quad k = u_i+1, u_i+2, \ldots, A_0+N_A, \; i = 1, 2, \ldots, N-1$$
$$\beta^G_{(A_0+N_A),i}(\tau) = \begin{cases} \dfrac{\alpha^G_{(A_0+N_A),i}(\tau)}{\alpha^G_{(A_0+N_A),N_T}(\tau)}, & \text{if } u_i = A_0+N_A \\ 0, & \text{if } u_i < A_0+N_A \end{cases}$$
$$i = 1, 2, \ldots, N_T - 1$$
$$\beta^G_{ki}(\tau) = \beta^G_{(k+1),i}(\tau) + \frac{1}{\alpha^G_{(A_0+N_A),N_T}(\tau)} \left\{ \binom{N_A}{k-A_0} \alpha^G_{ki}(\tau)\eta^G_{ki}(\tau) \right\},$$
$$k = l_i+1, l_i+2, \ldots, u_i \text{ and } k < A_0+N_A, \quad i = 1, 2, \ldots, N_T - 1$$

Note that the β^G-matrix generated by this algorithm has N_A rows and N_T columns.

We now proceed to analyze each of the four types of congestion period partitions in turn. For each type, we specify how to find A_0, N_A, N_T, and what the τ-vector is. We also specify the set of bounds l_i, $i = 1, 2, \ldots, N_T - 1$ and u_i, $i = 1, 2, \ldots, N_T - 1$. For all of these quantities, we indicate by a superscript of 1, $2j$, $3j$, or 4 whether the quantity applies to the type 1, j-th type 2, j-th type 3, or type 4 partition respectively. Hence, note that we will always have the following:

$$l^P_0 = u^P_0 = A^P_0$$
$$l^P_{N_T} = u^P_{N_T} = A^P_0 + N^P_A$$
$$P = 1, 2j, 3j, 4$$

where P denotes the type of partition being analyzed. Similarly, we let D_1, D_{2j}, D_{3j}, and D_4 denote the total number of departures during the type 1, j-th type 2, j-th type 3, or type 4 partition respectively. Note that:

$$D_1 + \sum_{j=1}^{\Gamma} D_{2j} + \sum_{j=1}^{\Gamma-1} D_{3j} + D_4 = N$$

These quantities allow us to fill in an N_A by N_T matrix of arrival-time probabilities, via the above algorithm. Then we calculate the computational complexity of the algorithm in each of the four cases. Finally, we describe how to combine all of the information to calculate queue statistics of interest. In all of the sections that follow, we define M to be the position of the mat, and we let Γ be the number of mat cycles, with mat depression times d_j and mat release times r_j, as described earlier. The set of times at which we have information as to arrivals and departures includes t_0 through t_N and all of the d_j's. In fact, we have perfect information as to the state of the queue both at time d_j (queue length equals M) and at time d_j^- (queue length equals $M-1$), so we will utilize both pieces of information.

7.2 Type 1 Congestion Period Partition Analysis: from 0 in Queue to M in Queue

During the type 1 congestion period partition, which runs over the time interval $(0, d_1]$, we know that the queue length starts out at zero, and then is less than M until the instant d_1, at which time the queue length increases from $M-1$ to M for the first time. Clearly, the boundaries of this partition, the times at which we know the state of the queue with certainty, are time $t_0 = 0$, at which time we know that the queue length is zero, and time d_1^-, when we know that the queue length equals $M-1$. Using the definitions of the last section, we also know that during the interval $(0, d_1]$, there were D_1 service completions, at times $t_1, t_2, \ldots, t_{D_1}$. Hence, N_T^1, the number of times of interest, for this type 1 congestion period partition, is $D_1 + 1$. Specifically, we have the following information as to the times of interest:

$$\begin{aligned} N_T^1 &= D_1 + 1 \\ \tau_i^1 &= t_i, \quad i = 0, 1, \ldots, N_T^1 - 1 \\ \tau_{N_T^1}^1 &= d_1^- \end{aligned}$$

We also know that at the beginning of the partition, there are zero customers who have waited in queue up to time τ_0^1, so A_0^1 for this type 1 partition is zero. Since we consider the partition to end the instant before the arrival which causes the queue length to increment to M, i.e., at time d_1^-, then N_A^1, the total number of arrivals during the partition, must be $D_1 + M - 1$, the total number of service completions during the partition, plus the increase in the number in queue during the partition (from 0 to $M-1$). We also know that we had continuous congestion throughout the partition, so the values of the l_i^1's are found from:

$$\begin{aligned} & A(t_k) \geq k \\ \Longrightarrow \quad & A(\tau_i^1) \geq i = l_i^1, \quad i = 0, 1, \ldots, N_T^1 - 1 \end{aligned}$$

Finally, to find the u_i^1's, we require that the queue length, at all times during the partition, be less than M. As pointed out in the last chapter, this is equivalent to requiring $Q(t_k^-) \leq M-1 \implies Q(t_k) \leq M-2$, which means for all t_k in the partition we have:

$$\begin{aligned} & Q(t_k) \leq M-2 \\ \Longrightarrow \quad & A(t_k) \leq k+M-2 \\ \Longrightarrow \quad & A(\tau_i^1) \leq i+M-2 = u_i^1, \quad i = 1, 2, \ldots, N_T^1 - 1 \end{aligned}$$

Note that these u_i^1's automatically satisfy $u_i^1 \leq u_{N_T^1}^1 = A_0^1 + N_A^1$. Summarizing, then, we have the following information as to the arrivals during the type 1 partition:

$$\begin{aligned} A_0^1 &= 0 \\ N_A^1 &= D_1 + M - 1 \\ l_i^1 &= i, \quad i = 0, 1, \ldots, N_T^1 - 1 \\ l_{N_T^1}^1 &= D_1 + M - 1 \\ u_0^1 &= 0 \\ u_i^1 &= i + M - 2, \quad i = 1, 2, \ldots, N_T^1 \end{aligned}$$

Using the two sets of data given above for the arrivals and the times of interest, we may generate a β^G-matrix, as described in the last section. This matrix will give all of the arrival-time probabilities of interest for the period $(0, d_1)$. It has $D_1 + M - 1$ rows and $D_1 + 1$ columns. The computational complexity of determining the β^G-matrix for this partition may be found by an analysis similar to that used in finding the computational complexity of the QIE^Q algorithm. In any column of the α^G- or η^G-matrix, there are at most $M - 1$ non-trivial values to compute. The number of terms contributing to each of these values is at most M, and the number of columns to be computed is $N_T^1 - 1 = D_1$. Hence, as in the QIE^Q algorithm, we calculate only a band of values in each of these two matrices, making their computational complexity $O(D_1 M^2)$. The β^G-matrix requires that $M - 2$ non-trivial values be calculated by an $O(1)$ operation in each of $N_T^1 - 1 = D_1$ columns, so that the computational complexity of just calculating the β^G-values is $O(D_1 M)$ (again, only a band of values need be calculated), and the computational complexity of the entire algorithm for the type 1 partition is $O(D_1 M^2)$.

We also could have analyzed the type 1 partition by using the QIE^Q algorithm directly, in the following manner, by making use of an artificial bulk departure of $M - 1$ customers at time d_1^-. Suppose that, rather than having an arrival at time d_1, we have $M - 1$ departures at d_1^-, with the assumption that these departures are the last ones of the congestion period. An analysis of this congestion period, using the QIE^Q algorithm with $Q = M - 1$, provides us with the probabilities of the various ways the Poisson arrivals could have occurred over the time interval $(0, d_1)$ while still obeying the queue length constraint and the usual arrival time inequalities, which is exactly what we are looking for. Of course, since we would have $\tau_{D_1+1}^1 = \tau_{D_1+2}^1 = \cdots = \tau_{D_1+M-1}^1 = d_1^-$, then entries in the last $M - 2$ columns of the three matrices of interest would be either zeroes or ones or redundant, so that the useful information would be contained in a $(D_1 + M - 1)$ by $(D_1 + 1)$ submatrix, identical to the matrix obtained above.

7.3 Type 2 Congestion Period Partition Analysis: A Single Mat Cycle

This type of congestion period partition runs over the time interval $(d_j, r_j], j = 1, 2, \ldots, \Gamma$. There are no restrictions on queue size during this congestion period partition, except that it be greater than or equal to M at all times during (d_j, r_j) and that it then decrease from M to $M-1$ at the instant r_j. At first glance, it might seem that the boundaries of this partition, the times at which we know the state of the queue with certainty, would be d_j and r_j. However, consider the service completion immediately prior to r_j. We claim that at this instant, the queue length must decrease from $M+1$ to M: if the queue length were any longer, there would be no way for it to decrease to M by time r_j; and if the queue length were shorter, the mat would have been released prior to r_j. This implies that there must have been zero arrivals between this prior service completion and r_j. This is similar to the situation we have in the standard QIE algorithm. Although the congestion period (all servers busy) technically lasts from time 0 to time t_{N+1}, we need only analyze the period $(0, t_N]$: we know a server was made idle at time t_{N+1}, so we know there were zero arrivals during $(t_N, t_{N+1}]$.

In order to determine some of the quantities of interest for this type 2 partition analysis, we first must calculate A_0^{2j}, the cumulative number of customers who have had to wait in queue by time d_j. This can be found by first finding the total number of service completions by time d_j, which is equal to $D_1 + \sum_{n=1}^{j-1}(D_{2n} + D_{3n})$, and then adding to it the number of customers in queue at time d_j, which is equal to M. This gives for A_0^{2j}:

$$A_0^{2j} = D_1 + \sum_{n=1}^{j-1}(D_{2n} + D_{3n}) + M, \quad j = 1, 2, \ldots, \Gamma$$

Since the type 2 partition runs from d_j to the service completion before r_j, with the times of interest comprising the service completions in this interval, then the following

should be obvious for $j = 1, 2, \ldots, \Gamma$:

$$\begin{aligned} N_T^{2j} &= D_{2j} - 1 \\ \tau_0^{2j} &= d_j \\ \tau_i^{2j} &= t_{A_0^{2j}-M+i}, \quad i = 1, 2, \ldots, N_T^{2j} \end{aligned}$$

We may find N_A^{2j}, the total number of arrivals during the partition, to be equal to the number of service completions during the interval, $N_T^{2j} = D_{2j} - 1$, since the queue length is equal to M both at the beginning and end of the partition. The upper bounds, u_i^{2j}, are easily found, since there are no upper constraints on queue length during this partition. The only implicit constraint is that the total cumulative number of arrivals at time τ_i^{2j} may not exceed the total cumulative number of arrivals by the end of the partition, i.e., we must have $A(\tau_i^{2j}) \leq A_0^{2j} + N_A^{2j}$. The lower bounds, l_i^{2j}, are found from the following observation about all times t_k during the partition:

$$\begin{aligned} & Q(t_k) \geq M \\ \Longrightarrow \quad & A(t_k) \geq M + k \\ \Longrightarrow \quad & A(\tau^{2j}_{M-A_0^{2j}+k}) \geq M + k \\ \Longrightarrow \quad & A(\tau_i^{2j}) \geq A_0^{2j} + i, \quad i = 1, 2, \ldots, N_T^{2j} \end{aligned}$$

Summarizing, then, we have the following information as to the arrivals during the type 2 partition, for $j = 1, 2, \ldots, \Gamma$:

$$\begin{aligned} A_0^{2j} &= D_1 + \sum_{n=1}^{j-1}(D_{2n} + D_{3n}) + M \\ N_A^{2j} &= D_{2j} - 1 \\ l_i^{2j} &= A_0^{2j} + i, \quad i = 0, 1, \ldots, N_T^{2j} \\ u_0^{2j} &= A_0^{2j} \\ u_i^{2j} &= A_0^{2j} + N_A^{2j}, \quad i = 1, 2, \ldots, N_T^{2j} \end{aligned}$$

Now we may use the data given above to generate a β^G-matrix for the j-th type 2 congestion period partition. This matrix will give all of the arrival-time probabilities

of interest for the period $(d_j, t_{A_0^{2j}-M+N_T^{2j}}]$. It is a square matrix, having $D_{2j}-1$ rows and columns. If we let A_0^{2j} be zero and N_A^{2j} be N, we see that we have exactly the same size and bounds as in the original QIE algorithm. Hence, the computational complexity of this algorithm must be the same as that of the original, which is $O(N^3)$ or $O(D_{2j}^3)$.

In fact, we could have analyzed this partition just by applying the standard QIE algorithm. Specifically, consider our original queueing system, with a waiting room added for the first M customers in queue; additional customers must wait outside. A "congestion period" for this new system begins when the waiting room becomes full (this occurs at time d_j in the original system, when the queue length goes from $M-1$ to M). The congestion period terminates when there is again space in the waiting room (this occurs at time r_j in the original system, when the queue length goes from M to $M-1$), which is equivalent to time t_{N+1}, so that we would analyze the period between d_j and the service completion prior to r_j, just as suggested above. Analysis of this new congestion period, using the original QIE algorithm, gives us a $(D_{2j}-1)$ by $(D_{2j}-1)$ matrix, with the desired arrival time probabilities, identical to that described above. Of course, we would have to shuffle indices and determine where in the $\beta^{\mathcal{M}}$-matrix these probabilities should go (this is handled automatically by our algorithm above). Note that, for $N \gg M$, one of these type 2 congestion period partitions could be very long, and hence its concomitant standard QIE analysis, with computational complexity of $O(D_{2j}^3)$ could be computationally burdensome.

7.4 Type 3 Congestion Period Partition Analysis: A Single Non-Mat Cycle

During the type 3 congestion period partition, which runs over the time interval $(r_j, d_{j+1}], 1 \leq j \leq \Gamma - 1$, there are many constraints. First, we know that at time r_j, there are exactly $M-1$ customers in queue. Second, we know that between r_j

and d_{j+1}, there are no more than $M-1$ customers in queue. Also, we know that no server is made idle over the same time period (although we may have 0 in queue). Finally, we know that at time d_{j+1}^-, there are exactly $M-1$ customers in queue again. Consider where the beginning of this partition should be defined. We know that at r_j, the queue length decreases from M to $M-1$. We claim that we also know that at the service completion immediately following r_j, the queue length decreases from $M-1$ to $M-2$. It can certainly be no greater than $M-1$ right before the completion, or we would have had another mat transition. It can also be no less than $M-1$, since there has been no other intervening service completion to deplete the queue. Hence, we know that there were zero arrivals between r_j and the subsequent service completion, and therefore we may start our analysis at this subsequent time, rather than at r_j.

In order to determine some of the quantities of interest for this type 3 partition analysis, we first must calculate A_0^{3j}, the cumulative number of customers who have had to wait in queue by τ_0^{3j}, the time of the service completion subsequent to r_j: this quantity is equal to the cumulative number of arrivals by r_j (since we just argued that there are zero arrivals in $(r_j, \tau_0^{3j}]$). This latter can be found by first finding the total number of service completions by time r_j, which is equal to $D_1 + \sum_{n=1}^{j-1}(D_{2n} + D_{3n}) + D_{2j}$, then adding to it the number of customers in queue at time r_j, which is equal to $M-1$. This gives for A_0^{3j}:

$$A_0^{3j} = D_1 + \sum_{n=1}^{j-1}(D_{2n} + D_{3n}) + D_{2j} + M - 1, \quad j = 1, 2, \ldots, \Gamma - 1$$

Since the type 3 partition runs from τ_0^{3j} to d_{j+1}^-, with the times of interest comprising the service completions in $(r_j, d_{j+1}]$, minus the service completion at τ_0^{3j}, plus the instant prior to the arrival at d_{j+1}, then the following should be obvious for $j = 1, 2, \ldots, \Gamma - 1$:

$$\begin{aligned} N_T^{3j} &= D_{3j} \\ \tau_i^{3j} &= t_{A_0^{3j} - M + 2 + i}, \quad i = 0, 1, \ldots, N_T^{3j} - 1 \end{aligned}$$

$$\tau_{N_T}^{3j} = d_{j+1}^-$$

We may find N_A^{3j}, the total number of arrivals during the partition, to be equal to the number of service completions during the interval $(r_j, d_{j+1}]$, which is D_{3j}, since the queue length is equal to $M-1$ both at the beginning and end of that interval, and there are no arrivals in $(r_j, \tau_0^{3j}]$. The lower bounds, l_i^{3j}'s, are found from the constraint that the busy period not end during the partition, i.e., that $A(t_k) \geq k$ for all t_k during the partition. We find:

$$\begin{aligned} A(t_k) &\geq k \\ \Longrightarrow \quad A(\tau^{3j}_{M-A_0^{3j}-2+k}) &\geq k \\ \Longrightarrow \quad A(\tau_i^{3j}) &\geq A_0^{3j} - M + 2 + i, \quad i = 1, 2, \ldots, N_T^{3j} - 1 \end{aligned}$$

Note that we also require that $A(\tau_i^{3j})$ be greater than or equal to the cumulative number of arrivals at the beginning of the partition, i.e., we require $A(\tau_i^{3j}) \geq A_0^{3j}$. The upper bounds are found by examining the constraint that $Q(t_k^-) \leq M-1$ which implies that $Q(t_k) \leq M-2$. Translating this into constraints on the $A(t_k)$'s, for all t_k during the partition:

$$\begin{aligned} A(t_k) &\leq k + M - 2 \\ \Longrightarrow \quad A(\tau^{3j}_{M-A_0^{3j}-2+k}) &\leq k + M - 2 \\ \Longrightarrow \quad A(\tau_i^{3j}) &\leq A_0^{3j} + i, \quad i = 1, 2, \ldots, N_T^{3j} - 1 \end{aligned}$$

Note that these u_i^{3j}'s automatically satisfy $u_i^{3j} \leq u_{N_T^{3j}}^{3j} = A_0^{3j} + N_A^{3j}$. Summarizing, then, we have the following information as to the arrivals during the type 3 partition, for $j = 1, 2, \ldots, \Gamma - 1$:

$$\begin{aligned} A_0^{3j} &= D_1 + \sum_{n=1}^{j-1}(D_{2n} + D_{3n}) + D_{2j} + M - 1 \\ N_A^{3j} &= D_{3j} \\ l_i^{3j} &= A_0^{3j}, \quad i = 0, 1, \ldots, M-2 \\ l_i^{3j} &= A_0^{3j} - M + 2 + i, \quad i = M-1, M, \ldots, N_T^{3j} - 1 \end{aligned}$$

$$\begin{aligned} l^{3j}_{N^{3j}_T} &= A^{3j}_0 + D_{3j} \\ u^{3j}_i &= A^{3j}_0 + i, \quad i = 0, 1, \ldots, N^{3j}_T \end{aligned}$$

Now we may use the data given above to generate a β^G-matrix for the j-th type 3 congestion period partition. This matrix will give all of the arrival-time probabilities of interest for the period $(t_{A^{3j}_0 - M+2}, d_{j+1})$. It is also a square matrix, having D_{3j} rows and columns. To find the computational complexity, first consider the α^G- and the η^G-matrices. Each column has $u^{3j}_i - l^{3j}_i + 1$ non-trivial values to compute, and with the values of the bounds given above, this number of values per column must be less than or equal to $M - 1$. Each of these values is calculated as a sum of at most $u^{3j}_i - l^{3j}_{i-1} + 1$ terms from an adjacent column, and this number must be less than or equal to M. Finally, there are $N^{3j}_T = D_{3j}$ columns to compute, so the complexity of calculating these two matrices is $\mathrm{O}(D_{3j}M^2)$. Since the β^G-matrix has D_{3j} columns to compute, each with $u^{3j}_i - l^{3j}_i$ non-trivial elements, then calculation of this matrix alone is $\mathrm{O}(D_{3j}M)$ and the entire algorithm has computational complexity $\mathrm{O}(D_{3j}M^2)$.

We also could have analyzed the type 3 partition by using the QIE^Q algorithm directly, in the following manner, by making use of both an artificial bulk departure of $M-1$ customers at time d^-_{j+1}, and artificial deterministic arrivals of $M-2$ customers at time τ^{3j}_0. That is, rather than having an arrival at time d_{j+1}, we have $M - 1$ departures at d^-_{j+1}, with the assumption that these departures are the last ones of the congestion period. Similarly, rather than having a departure at τ^{3j}_0, we claim that the "congestion period" begins at this instant, and we immediately have $M - 2$ arrivals to the queue. An analysis of this congestion period, using the QIE^Q algorithm with $Q = M-1$, provides us with the probabilities of the various ways the Poisson arrivals could have occurred over the time interval (τ^{3j}_0, d_{j+1}) while still obeying the queue length constraint and the usual arrival time inequalities, which is exactly what we are looking for. Of course, since the last $M - 1$ departures occur simultaneously at time d^-_{j+1}, then entries in the last $M - 2$ columns of the three matrices of interest would be either ones or zeroes or redundant. Similarly, since the first $M - 2$ arrivals are

deterministic, then entries in the first $M-2$ rows of the three matrices would also be ones or zeroes or redundant. Hence, the algorithm would generate a $(D_{3j}+M-2)$ by $(D_{3j}+M-2)$ square matrix, but only the lower left-hand square submatrix with D_{3j} rows and columns would contain any useful information, information identical to that generated by the algorithm above.

7.5 Type 4 Congestion Period Partition Analysis: from M in Queue to 0 in Queue

The type 4 congestion period partition runs over the time period $(r_\Gamma, t_N]$. We know that at time r_Γ, the queue length drops from M to $M-1$, and hence, at time r_Γ, there are exactly $M-1$ customers in queue. We also know that at time t_N, there are exactly 0 customers in queue. Finally, we know that during this congestion period partition, the queue length never exceeds $M-1$ customers. As in the case of the type 3 congestion period partition, we must consider where the beginning of this partition should be defined. We know that at r_Γ, the queue length decreases from M to $M-1$. We claim that we also know that at the service completion immediately following r_Γ, the queue length decreases from $M-1$ to $M-2$. It can certainly be no greater than $M-1$ right before the completion, or we would have had another mat transition. It can also be no less than $M-1$, since there has been no other intervening service completion to deplete the queue. Hence, we know that there were zero arrivals between r_Γ and the subsequent service completion, and therefore we may start our analysis at this subsequent time, rather than at r_Γ.

Since there are just the D_4 service completions during this partition, but the number included in N_T^4 does not include that at time τ_0^4, we have that $N_T^4 = D_4 - 1$. Note that we must have $D_4 - 1 \geq M-2$, so that all of the customers in queue at the beginning of the partition get emptied out by the end. We also know that the total number of service completions through time τ_0^4 is $D_1 + \sum_{j=1}^{\Gamma} D_{2j} + \sum_{j=1}^{\Gamma-1} D_{3j} + 1$,

which is more simply denoted $N - D_4 + 1$. Hence, we have the following:

$$\begin{aligned} N_T^4 &= D_4 - 1 \\ \tau_i^4 &= t_{N-D_4+1+i}, \quad i = 0, 1, \ldots, N_T^4 \end{aligned}$$

The cumulative number of arrivals to the system by time τ_0^4 is just the total number of service completions by that time plus $M - 2$, the number in queue at τ_0^4: hence, we have $A_0^4 = N - D_4 + M - 1$. Now, the total number of arrivals during $(\tau_0^4, t_N]$ must be $M - 2$ smaller than the number of service completions, so that the $M - 2$ customers in queue at τ_0^4 get emptied out by t_N, i.e., we have $N_A^4 = D_4 - M + 1$. This makes sense, since the total number of arrivals by the end of the partition, $A_0^4 + N_A^4$ must equal N. To find the values of l_i^4, we know that we must have $A(t_k) \geq k$ for all t_k during the partition, and $A(\tau_i^4) \geq A_0^4$. Consider the first constraint:

$$\begin{aligned} A(t_k) &\geq k \\ \Longrightarrow \quad A(\tau_{k-N+D_4-1}^4) &\geq k \\ \Longrightarrow \quad A(\tau_i^4) &\geq N - D_4 + 1 + i \end{aligned}$$

The second constraint may be expressed:

$$A(\tau_i^4) \geq N - D_4 + M - 1$$

The upper bounds are also found as the intersection of two constraints. First, we require that the queue length be less than M at all times during the partition, so that we have $Q(t_k^-) \leq M - 1 \Longrightarrow Q(t_k) \leq M - 2$. Continuing:

$$\begin{aligned} Q(t_k) &\leq M - 2 \\ \Longrightarrow \quad A(t_k) &\leq k + M - 2 \\ \Longrightarrow \quad A(\tau_{k-N+D_4-1}^4) &\leq k + M - 2 \\ \Longrightarrow \quad A(\tau_i^4) &\leq i + N - D_4 + M - 1 \end{aligned}$$

The second constraint is that the total number of arrivals by τ_i^4 be less than or equal to N, i.e.,

$$A(\tau_i^4) \leq N$$

Combining all of these results, then, we have for the arrivals during the type 4 partition:

$$\begin{aligned}
A_0^4 &= N - D_4 + M - 1 \\
N_A^4 &= D_4 - M + 1 \\
l_i^4 &= N - D_4 + M - 1, \quad i = 0, 1, \ldots, M-2 \\
l_i^4 &= N - D_4 + 1 + i, \quad i = M-1, M, \ldots, N_T^4 \\
u_i^4 &= i + N - D_4 + M - 1, \quad i = 0, 1, \ldots, D_4 - M + 1 \\
u_i^4 &= N, \quad i = D_4 - M + 2, D_4 - M + 3, \ldots, N_T^4
\end{aligned}$$

We use the data above to generate a β^G-matrix for the type 4 congestion period partition. This matrix will give all of the arrival-time probabilities of interest for the period $(t_{N-D_4+1}, t_N]$. This matrix has $D_4 - M + 1$ rows and $D_4 - 1$ columns. To find the computational complexity, we again begin with the α^G- and η^G-matrices. Each column of these matrices has $u_i^4 - l_i^4 + 1$ non-trivial values to compute, which, from the above, is at most $M - 1$. Each value is calculated as a sum of at most $u_i^4 - l_{i-1}^4 + 1$ terms from an adjacent column, which is at most M terms. Finally, there are $N_T^4 = D_4 - 1$ columns, so that the complexity of calculating these matrices is $O(D_4 M^2)$. The β^G-matrix also has $D_4 - 1$ columns, each of which has $u_i^4 - l_i^4 \leq M - 2$ non-trivial values to calculate (each of which is an $O(1)$ operation), so calculation of the β^G-matrix alone is $O(D_4 M)$ and the entire algorithm is $O(D_4 M^2)$.

Again, we could have analyzed the type 4 partition by using the QIE^Q algorithm directly, by making use of artificial deterministic arrivals of $M - 2$ customers at time τ_0^4. Specifically, rather than having a service completion at τ_0^4, we consider the "congestion period" to begin at that instant, with $M - 2$ customers immediately arriving to fill up the queue. An analysis of this congestion period, using the QIE^Q algorithm, with $Q = M - 1$, provides the probabilities of the ways the Poisson (non-deterministic) arrivals could have occurred over the time interval $(\tau_0^4, t_N]$, while still obeying the queue length constraint and the usual arrival-time inequalities, which

is exactly what we are looking for. Note that, since the first $M-2$ arrivals are deterministic, then entries in the first $M-2$ rows of the three matrices would either be ones or zeroes or redundant. Hence, the algorithm would generate a square matrix, with $D_4 - 1$ rows and columns, but only the lower $D_4 - M + 1$ rows would contain useful information, the same information generated by the algorithm above.

7.6 Completion of the $\beta^{\mathcal{M}}$-Matrix

We have shown how to fill in many parts of the $\beta^{\mathcal{M}}$-matrix by analysis of congestion period partitions. Filling in the rest of the matrix is easily accomplished in the manner described in this section. Recall that we defined $\beta^{\mathcal{M}}_{ki}(\tau)$ as follows:

$$\beta^{\mathcal{M}}_{ki}(\tau) \equiv \Pr[A(\tau_i) \geq k | \mathcal{E}^S(\mathrm{t}), \mathcal{M}], \quad k = 1, 2, \ldots, N$$

where τ_i is any time at which information is available. Specifically, τ_i could represent one of the t_k's, or it could represent d_j^- or d_j. Note that we must include both of these latter times in our $\beta^{\mathcal{M}}$-matrix, because we use this matrix to calculate the cumulative expected number of arrivals and expected queue length at time t during the congestion period. If we only included the point d_j, then the expected queue length function between the service completion prior to d_j (call it τ) and d_j would just be linearly increasing up to the value M. In fact, we know with certainty that there were exactly $M-1$ customers in queue just prior to d_j, and there was an arrival at d_j. Hence, we would like our expected queue length function between τ and d_j to be linearly increasing up to the value $M-1$, and then to make a step jump up to the value M. This is accomplished by including the time d_j^- in the $\beta^{\mathcal{M}}$-matrix.

Hence, although a standard β-matrix has N rows and N columns, this $\beta^{\mathcal{M}}$-matrix will have N rows (since the number of arrivals during the congestion period must still be N), and $N + 2\Gamma$ columns, one column each for $t_1, t_2, \ldots, t_N$, and one column for each d_j^- and d_j, for $j = 1, 2, \ldots, \Gamma$. We assume that the columns of the matrix are time-ordered, so that, for example, the column representing t_{D_1} is just to the left of

the column representing d_1^-, which is to the left of that representing d_1, which is to the left of that representing t_{D_1+1}, etc.

The algorithms presented in the last four sections allow submatrices of the $\beta^{\mathcal{M}}$-matrix to be filled in. We next consider how to fill in the remaining rows of those columns partially filled in by these algorithms. We claim the following for any column, τ_i, of the $\beta^{\mathcal{M}}$-matrix which is filled in by the type P congestion period partition algorithm:

Claim 7.1

$$\Pr[A(\tau_i) \geq k | \mathcal{E}^S(\mathrm{t}), \mathcal{M}] = \begin{cases} 1, & k \leq A_0^P \\ 0, & k > A_0^P + N_A^P \end{cases}$$
$$P = 1, 2j, 3j, 4$$

where P represents the congestion period partition algorithm used to fill in column τ_i of the $\beta^{\mathcal{M}}$-matrix.

Proof: Since we know that the cumulative number of arrivals to the system by the beginning of partition P is A_0^P, then the probability that the cumulative number of arrivals at a later time is greater than or equal to k, where k is less than or equal to A_0^P, must be one. Similarly, since we know that the cumulative number of arrivals to the system at the end of partition P is $A_0^P + N_A^P$, the the probability that the cumulative number of arrivals at an earlier time is greater than or equal to k, where k is greater than $A_0^P + N_A^P$, must be zero. ∎

This claim allows us to fill in all rows of the columns which are filled in by the four partition analysis algorithms. Next, we consider which columns remain to be completed, and how to complete them. First, we find the correspondence between the τ_i's (the $N + 2\Gamma$ columns of the $\beta^{\mathcal{M}}$-matrix), and the d_j's and t_k's. We know that by time d_j^-, $j = 1, 2, \ldots, \Gamma$, there has been one type 1 partition with D_1 service completions, plus $j - 1$ types 2 and 3 partitions, with a total of $\sum_{n=1}^{j-1} D_{2n} + D_{3n}$ service completions, plus $2(j-1)$ additional instants of information, the latter representing times $d_1^-, d_1, d_2^-, d_2, \ldots, d_{j-1}^-, d_{j-1}$. Hence, we have that the index on the τ_i

representing d_j^- (d_j) should be one (two) greater than the sum of the total number of service completions plus the number of additional instants of information, i.e.:

$$\begin{aligned} d_j^- &= \tau_i, \quad i = D_1 + \sum_{n=1}^{j-1} (D_{2n} + D_{3n}) + 2(j-1) + 1 \\ &\Longrightarrow i = D_1 + \sum_{n=1}^{j-1} (D_{2n} + D_{3n}) + 2j - 1 \\ d_j &= \tau_i, \quad i = D_1 + \sum_{n=1}^{j-1} (D_{2n} + D_{3n}) + 2(j-1) + 2 \\ &\Longrightarrow i = D_1 + \sum_{n=1}^{j-1} (D_{2n} + D_{3n}) + 2j \\ & j = 1, 2, \ldots, \Gamma \end{aligned}$$

So, for instance, we would have that $d_1^- = \tau_{D_1+1}$ and that $d_2^- = \tau_{D_1+D_{21}+D_{31}+3}$, which makes sense. If t_k is between the j-th and the $(j+1)$-th mat depression, then t_k represents the $(k+2j)$-th instant of information during the congestion period, i.e.:

$$\begin{aligned} t_k &= \tau_k, \quad t_0 \leq t_k < d_1 \\ t_k &= \tau_{k+2j}, \quad d_j < t_k < d_{j+1}, \quad j = 1, 2, \ldots, \Gamma - 1 \\ t_k &= \tau_{k+2\Gamma}, \quad d_\Gamma < t_k \leq t_N \end{aligned}$$

We combine all of the above into the following:

$$\begin{aligned} \tau_i &= t_i, \quad i = 0, 1, \ldots, D_1 \\ \tau_i &= d_j^-, \quad i = D_1 + \sum_{n=1}^{j-1} (D_{2n} + D_{3n}) + 2j - 1, \quad j = 1, 2, \ldots, \Gamma \\ \tau_i &= d_j, \quad i = D_1 + \sum_{n=1}^{j-1} (D_{2n} + D_{3n}) + 2j, \quad j = 1, 2, \ldots, \Gamma \\ \tau_i &= t_{i-2k}, \; i = \left(D_1 + \sum_{n=1}^{k-1} (D_{2n} + D_{3n}) + 2k + 1 \right), \ldots, \\ &\quad \left(D_1 + \sum_{n=1}^{k} (D_{2n} + D_{3n}) + 2k \right), \quad k = 1, 2, \ldots, \Gamma - 1 \\ \tau_i &= t_{i-2\Gamma}, \quad i = \left(D_1 + \sum_{n=1}^{\Gamma-1} (D_{2n} + D_{3n}) + 2\Gamma + 1 \right), \ldots, (N + 2\Gamma) \end{aligned}$$

Since the algorithm for the type P partition fills in the columns of the $\beta^{\mathcal{M}}$-matrix corresponding to τ_1^P through $\tau_{N_T^P}^P$ inclusive, we summarize here what those values are for the four types of partition and then determine how to fill in the remaining columns:

$$
\begin{aligned}
\tau_1^1 &= t_1 \\
\tau_{N_T^1}^1 &= d_1^- \\
\tau_1^{2j} &= t_k, \quad k = D_1 + \sum_{n=1}^{j-1} (D_{2n} + D_{3n}) + 1 \\
\tau_{N_T^{2j}}^{2j} &= t_k, \quad k = D_1 + \sum_{n=1}^{j-1} (D_{2n} + D_{3n}) + D_{2j} - 1 \\
&\qquad j = 1, 2, \ldots, \Gamma \\
\tau_1^{3j} &= t_k, \quad k = D_1 + \sum_{n=1}^{j-1} (D_{2n} + D_{3n}) + D_{2j} + 2 \\
\tau_{N_T^{3j}}^{3j} &= d_{j+1}^- \\
&\qquad j = 1, 2, \ldots, \Gamma - 1 \\
\tau_1^4 &= t_k, \quad k = D_1 + \sum_{n=1}^{\Gamma-1} (D_{2n} + D_{3n}) + D_{2\Gamma} + 2 \\
\tau_{N_T^4}^4 &= t_N
\end{aligned}
$$

First consider the gap between the type 1 partition and the first type 2 partition, or between $\tau_{N_T^1}^1 = d_1^-$ and $\tau_1^{21} = t_{D_1+1}$. It should be clear that the only column that needs to be filled in between these two is that representing d_1. Similarly, consider the gap between any type 3 partition and the subsequent type 2 partition, or between $\tau_{N_T^{3j}}^{3j} = dj + 1^-$ and $\tau_1^{2,(j+1)}$ for $j = 1, 2, \ldots, \Gamma - 1$. The service completion prior to d_{j+1}^- is at t_k, $k = D_1 + \sum_{n=1}^{j} (D_{2n} + D_{3n})$, so again the only column that needs to be filled in is that corresponding to d_{j+1}, $j = 1, 2, \ldots, \Gamma - 1$. In other words, we must find $\Pr[A(d_j) \geq k | \mathcal{E}^S(\mathbf{t}), \mathcal{M}]$, $j = 1, 2, \ldots, \Gamma$. But we claim this is easily found as follows:

Claim 7.2

$$\Pr[A(d_j) \geq k | \mathcal{E}^S(\mathrm{t}), \mathcal{M}] = \begin{cases} 1, & k = 1, 2, \ldots, \left(D_1 + \sum_{n=1}^{j-1} (D_{2n} + D_{3n}) + M \right) \\ 0, & k = \left(D_1 + \sum_{n=1}^{j-1} (D_{2n} + D_{3n}) + M + 1 \right), \ldots, N \end{cases}$$
$$j = 1, 2, \ldots, \Gamma$$

Proof: We know that the cumulative number of arrivals by d_j is equal to the number of departures by that time plus the number in queue (which we know has just increased to M): hence, we know

$$A(d_j) = D_1 + \sum_{n=1}^{j-1} (D_{2n} + D_{3n}) + M$$

Since we know the value of $A(d_j)$ to be exactly that above, the result of the claim follows immediately. ∎

Next, consider any columns missing between the last (Γ-th) type 2 partition and the type 4 partition, or between $\tau^{2\Gamma}_{N^{2\Gamma}_T}$ and τ^4_1. It should be clear that there are two missing columns between these, those corresponding to t_{h^Γ} and $t_{h^\Gamma+1}$, with $h^\Gamma = D_1 + \sum_{n=1}^{\Gamma-1} (D_{2n} + D_{3n}) + D_{2\Gamma}$. Similarly, consider which columns are missing between any of the first $\Gamma - 1$ type 2 partitions and the subsequent type 3 partitions, or between $\tau^{2j}_{N^{2j}_T}$ and τ^{3j}_1, with $j = 1, 2, \Gamma - 1$. It should be clear that there are two missing columns between these, those corresponding to t_{h^j} and t_{h^j+1}, with $h^j = D_1 + \sum_{n=1}^{j-1} (D_{2n} + D_{3n}) + D_{2j}$. In both cases just cited, we know that t_{h^j} corresponds to mat release time r_j, and t_{h^j+1} to the subsequent service completion. Hence, by previous arguments, and since we know there are $M - 1$ customers in queue at the mat release time, then we also know that

$$\begin{aligned} A(t_{h^j}) &= D_1 + \sum_{n=1}^{j-1} (D_{2n} + D_{3n}) + D_{2j} + M - 1 \\ &= A(t^j_{h+1}), \\ & \quad j = 1, 2, \ldots, \Gamma \end{aligned} \tag{7.1}$$

where the two values are equal because, by previous arguments, we can have no arrivals between a mat release and a subsequent service completion, unless that arrival coincides with a mat depression. Hence, we have the following:

Claim 7.3

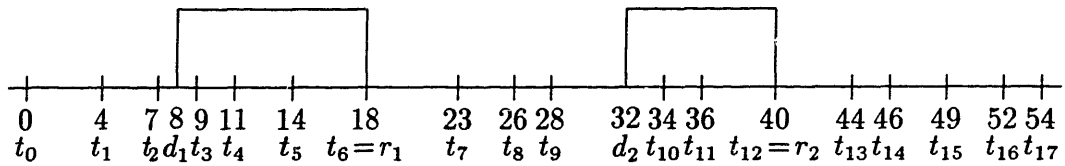
$$\begin{aligned}\Pr[A(t_{h^j}) \geq k | \mathcal{E}^S(\mathbf{t}), \mathcal{M}] &= \Pr[A(t_{h^j+1}) \geq k | \mathcal{E}^S(\mathbf{t}), \mathcal{M}] \\ &= \begin{cases} 1, & k = 1, 2, \ldots, \left(D_1 + \sum_{n=1}^{j-1} (D_{2n} + D_{3n}) + D_{2j} + M - 1 \right) \\ 0, & k = \left(D_1 + \sum_{n=1}^{j-1} (D_{2n} + D_{3n}) + D_{2j} + M \right), \ldots, N \end{cases} \\ h^j &= D_1 + \sum_{n=1}^{j-1} (D_{2n} + D_{3n}) + D_{2j}, \quad j = 1, 2, \ldots, \Gamma \end{aligned}$$

Proof: We know the exact values of $A(t_{h^j})$ and $A(t_{h^j+1})$ from Equation 7.1. Hence, the result of the claim follows immediately. ∎

We now have the means necessary to fill in the entire $\beta^{\mathcal{M}}$-matrix. We use the type 1 algorithm to fill in the upper $(D_1 + M - 1) \times (D_1 + 1)$ submatrix, filling in the entries below the submatrix with zeroes as per Claim 7.1. Then we fill in the column corresponding to d_1 via Claim 7.2. Next we fill in entries in the square submatrix comprising rows $(D_1 + M + 1)$ to $(D_1 + D_{21} + M - 1)$ and columns $(D_1 + 3)$ through $(D_1 + D_{21} + 1)$, according to the type 2 algorithm, and fill in above and below the submatrix using Claim 7.1. Next we fill in the columns corresponding to r_1 and the subsequent service completion, using Claim 7.3. We continue until the entire matrix is filled in. An example of this filling-in procedure is provided in Figure 7.2.

Once the matrix is completely filled in via the QIE$^{\mathcal{M}}$ algorithm, its values may be used to generate queue statistics. This is done in a method similar to that used in [Lars 90]. We calculate $\mathrm{E}[A(\tau_i) | \mathcal{E}^S(\mathbf{t}), \mathcal{M}]$ for all of the service completion times, as well as for the d_j^-'s and the d_j's, by adding up all of the values in the i-th column of the $\beta^{\mathcal{M}}$-matrix. To find $\mathrm{E}[Q(\tau_i) | \mathcal{E}^S(\mathbf{t}), \mathcal{M}]$, note the following:

$$\mathrm{E}[Q(t_k^-) | \mathcal{E}^S(\mathbf{t}), \mathcal{M}] = \mathrm{E}[A(t_k) | \mathcal{E}^S(\mathbf{t}), \mathcal{M}] - k + 1$$



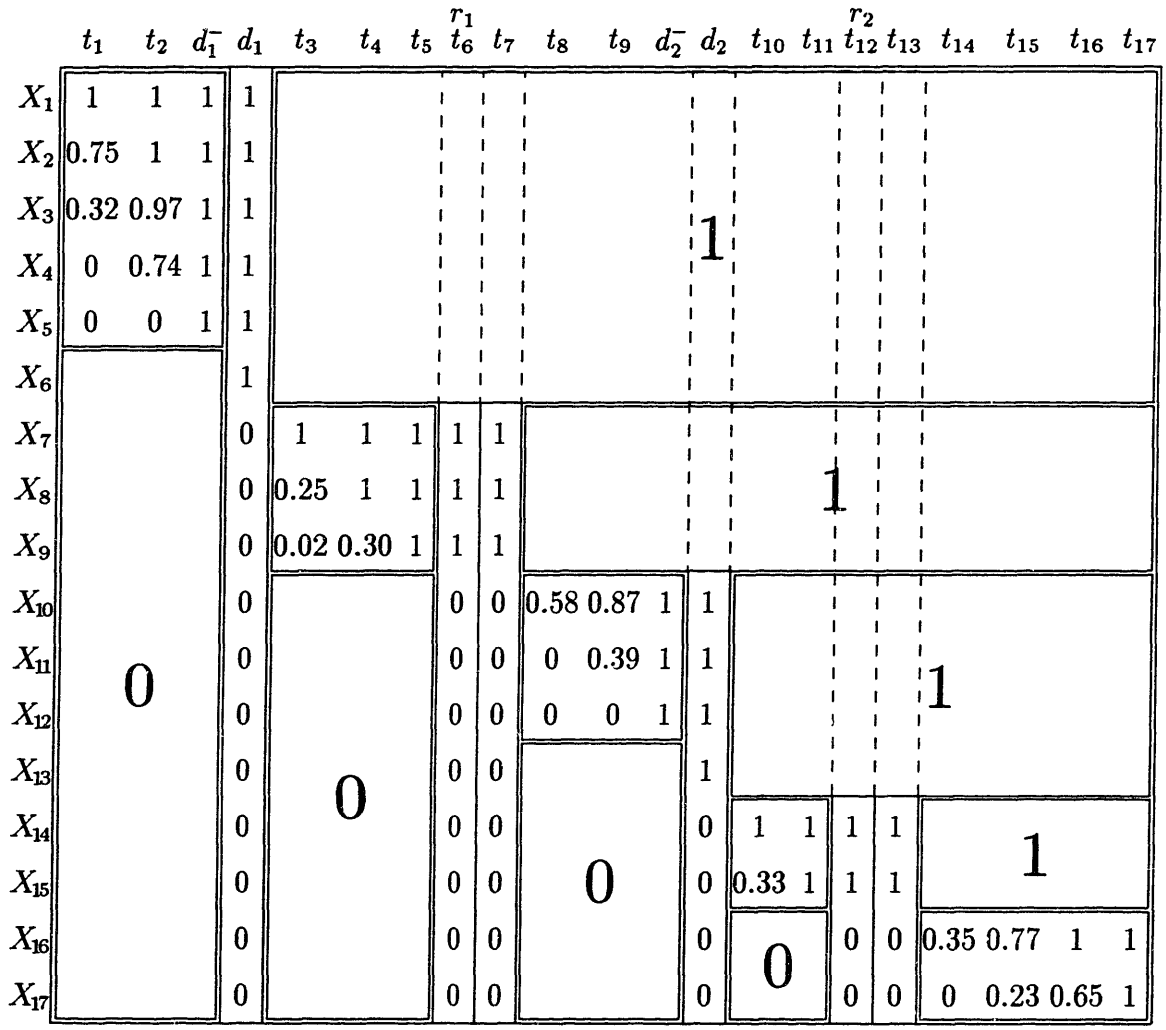


Figure 7.2: Sample Congestion Period with $M = 4, N = 17$, and $\Gamma = 2$; and Its $\beta^{\mathcal{M}}$-Matrix ($D_1 = 2$, $D_{21} = 4$, $D_{31} = 3$, $D_{22} = 3$, $D_4 = 5$)

$$\begin{aligned}
\mathrm{E}[Q(t_k)|\mathcal{E}^S(\mathbf{t}),\mathcal{M}] &= \mathrm{E}[A(t_k)|\mathcal{E}^S(\mathbf{t}),\mathcal{M}] - k \\
\mathrm{E}[Q(d_j^-)|\mathcal{E}^S(\mathbf{t}),\mathcal{M}] &= \mathrm{E}[A(d_j^-)|\mathcal{E}^S(\mathbf{t}),\mathcal{M}] - h^j \;=\; M-1 \\
\mathrm{E}[Q(d_j)|\mathcal{E}^S(\mathbf{t}),\mathcal{M}] &= \mathrm{E}[A(d_j)|\mathcal{E}^S(\mathbf{t}),\mathcal{M}] - h^j \;=\; M \\
& \qquad h^j \;=\; \text{index of service completion just prior to } d_j
\end{aligned}$$

Of course, both $\mathrm{E}[A(\tau)|\mathcal{E}^S(\mathbf{t}),\mathcal{M}]$ and $\mathrm{E}[Q(\tau)|\mathcal{E}^S(\mathbf{t}),\mathcal{M}]$ are linear between the values specified. We may then calculate the time-average queue length as the area under $\mathrm{E}[Q(\tau)|\mathcal{E}^S(\mathbf{t}),\mathcal{M}]$, divided by the total congestion period time:

$$\mathrm{E}[L_Q|\mathcal{E}^S(\mathbf{t}),\mathcal{M}] = \frac{1}{t_N}\sum_{i=1}^{N+2\Gamma}(\tau_i - \tau_{i-1})\left(\frac{\mathrm{E}[Q(\tau_i^-)|\mathcal{E}^S(\mathbf{t}),\mathcal{M}] + \mathrm{E}[Q(\tau_{i-1})|\mathcal{E}^S(\mathbf{t}),\mathcal{M}]}{2}\right)$$

where we assume that $d_j - d_j^- = 0$. The average wait in queue is still found to be:

$$\mathrm{E}[W_Q|\mathcal{E}^S(\mathbf{t}),\mathcal{M}] = \frac{t_N}{N}\mathrm{E}[L_Q|\mathcal{E}^S(\mathbf{t}),\mathcal{M}]$$

Incidence probabilities, $\Pi[k|\mathcal{E}^S(\mathbf{t}),\mathcal{M}]$, are found by creating a square submatrix of the $\beta^{\mathcal{M}}$-matrix, which only contains the columns corresponding to the service completion times and then proceeding exactly as described in [Lars 90], using this reduced matrix. $\mathrm{E}[\ell_Q|\mathcal{E}^S(\mathbf{t}),\mathcal{M}]$, the average queue length experienced by a randomly arriving customer, is also calculated exactly as in [Lars 90].

7.7 Computational Results: Comparison of the QIE and the QIE$^{\mathcal{M}}$ Algorithms

We include here results from simulation of an M/M/1 queue. These data were generated by three simulation runs with Poisson arrivals at rate 10 per hour, a single server, and exponential service times with expected values of 3 minutes for the first run (giving a value of $\rho = 0.5$) and 4 minutes for the last two runs (giving a value of $\rho = 0.67$). We compare the QIE$^{\mathcal{M}}$ algorithm to the standard QIE algorithm and also consider the effect of moving the mat to different positions. The statistics that are

used for comparison of the algorithms include: $E[L_Q|\ldots]$, the time-averaged number of customers in queue; $E[W_Q|\ldots]$, the average wait in queue; and ϵ, the time-average error, defined to be the absolute area between the actual queue length graph and the QIE expected queue length graph, divided by the total time of the congestion period. The run times to generate the beta-matrix for the different algorithms are also compared. Runs were on a 386/387-based Northgate Computer Systems PC. Each run time given below is an average of 1000 run times (for longer runs, presented to hundredths of a second) or 3000 run times (for shorter runs, presented to thousandths of a second) from different runs of the program on the same data. This averaging was necessary because the system clock is only updated every 0.0549254 seconds [Scan 83], so to get accuracy greater than 0.1 seconds, many runs must be averaged.

In Figures 7.3, 7.4, and 7.5, we compare the QIE and the QIE$^{\mathcal{M}}$ algorithms. In those figures, we present the six congestion periods with $N \geq 12$ from the first simulation run, as well as the longest congestion period (each with $N = 58$) from each of the last two runs. We compare the standard QIE performance (left set of graphs) with that of the QIE$^{\mathcal{M}}$ algorithm (right set of graphs), with a mat added at position $M = 3$ for the six shorter congestion periods and at position $M = 5$ for the two $N = 58$ congestion periods. Each graph on the left compares $E[Q(t)|\mathcal{E}^S(\mathbf{t})]$ to the actual queue length, as generated by the simulation run. Each graph on the right compares $E[Q(t)|\mathcal{E}^S(\mathbf{t}), \mathcal{M}]$ to the actual queue length. Note how the right-hand graphs are able to track the queue length exactly at the mat depression (queue length increasing from $M-1$ to M) and mat release (queue length decreasing from M to $M-1$) times. The values 3 and 5 were arbitrarily chosen for the mat position, although they were reasonable values for the selected congestion periods. The extent to which the QIE$^{\mathcal{M}}$ algorithm improves the queue estimates over the QIE algorithm, for all of the congestion periods considered, is detailed in Table 7.1. Note the considerable improvement both in accuracy (particularly as evidenced by ϵ) and in running time.

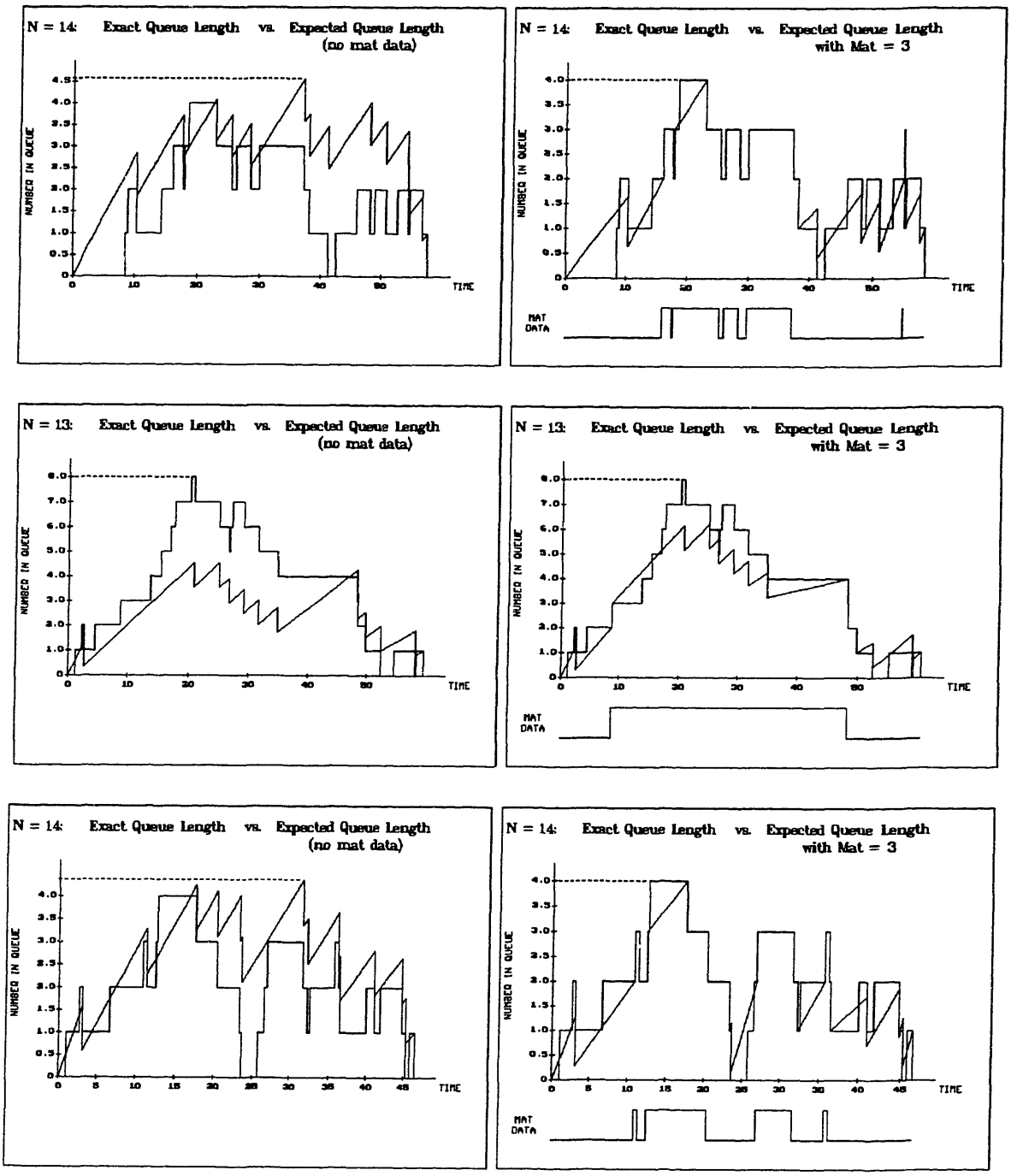


Figure 7.3: Exact vs. Expected Queue Length for Congestion Periods with $N = 14$, 13, and 14: Standard QIE (Left) and QIE^M, $M=3$ (Right)

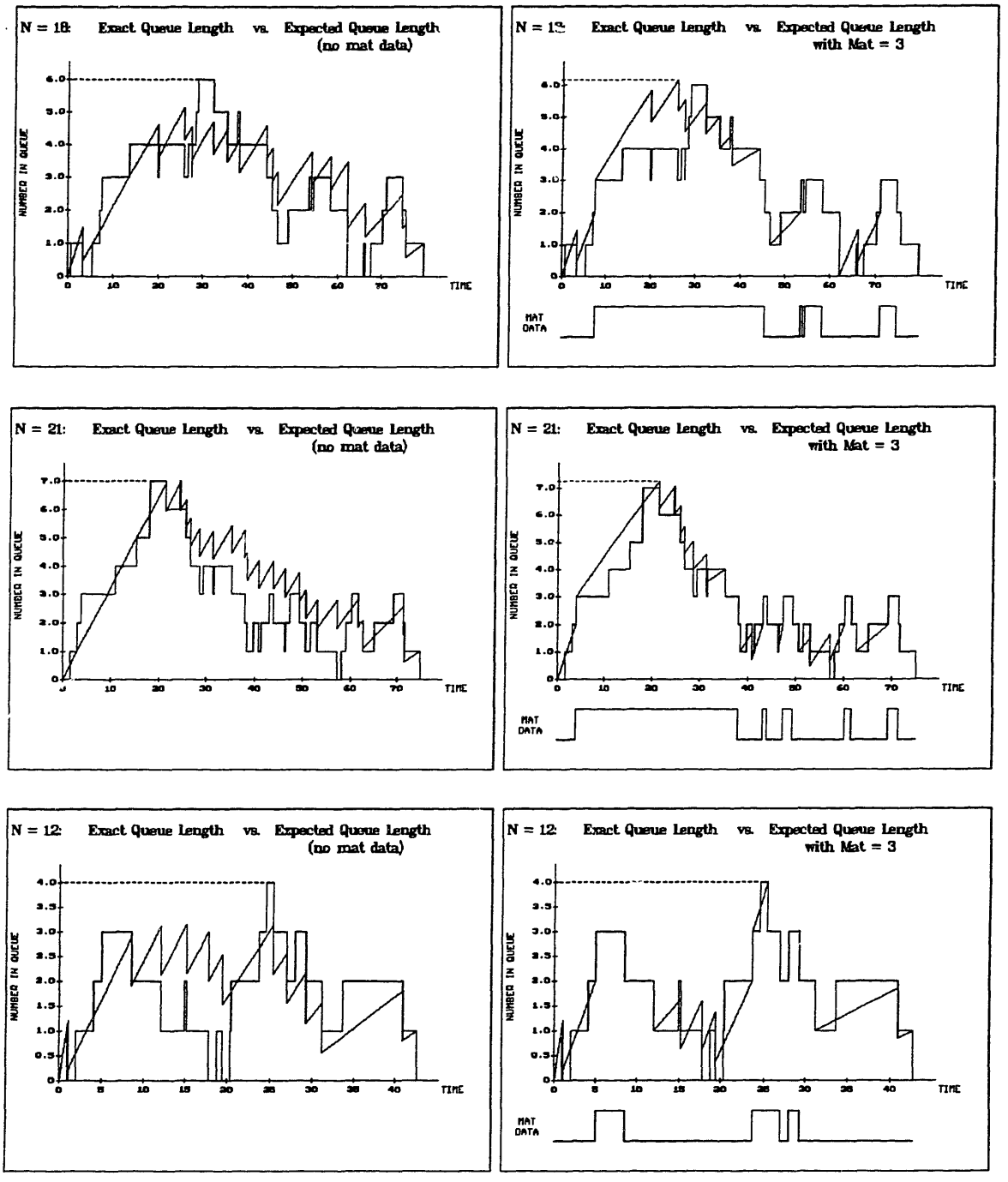


Figure 7.4: Exact vs. Expected Queue Length for Congestion Periods with $N = 18$, 21, and 12: Standard QIE (Left) and QIE^M, $M = 3$ (Right)

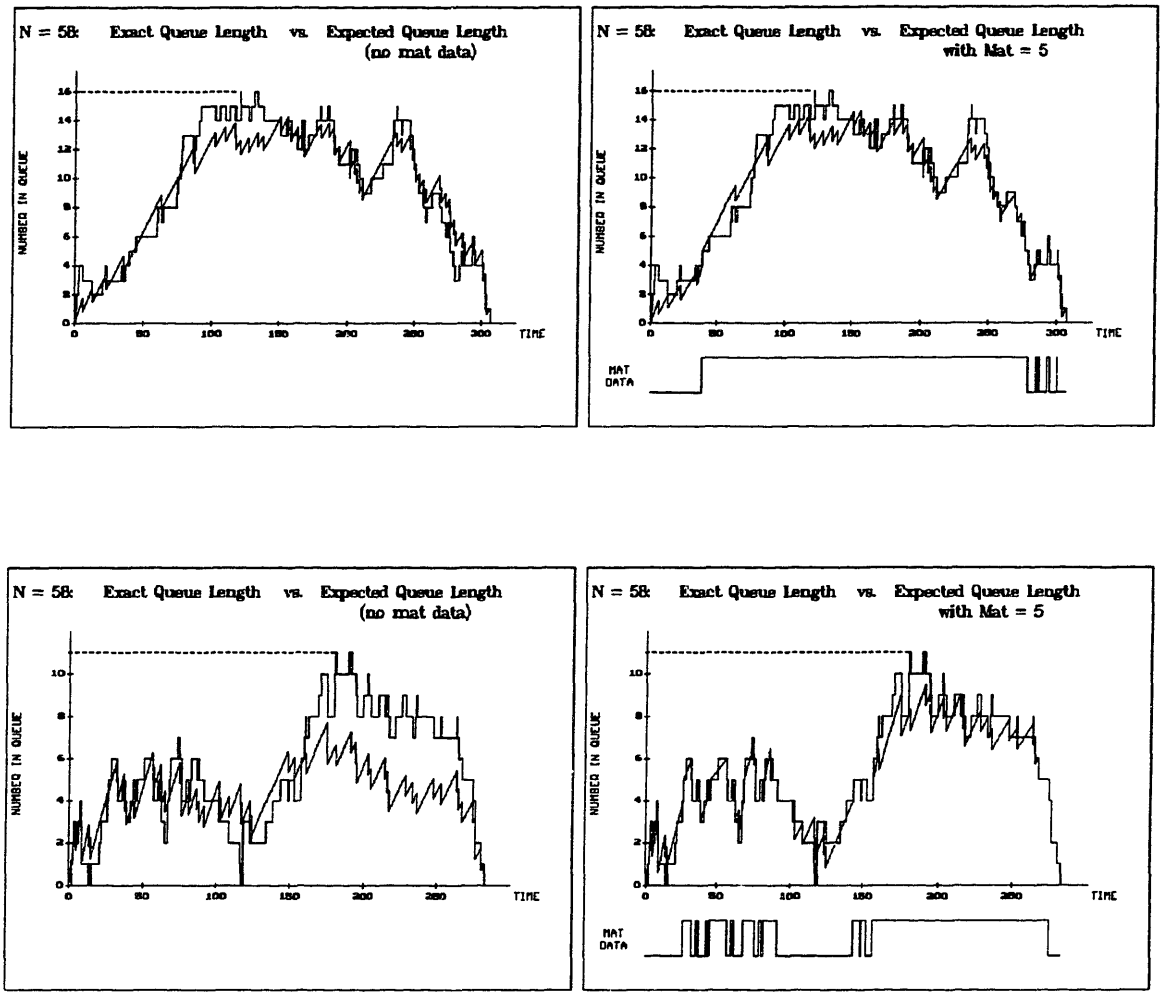


Figure 7.5: Exact vs. Expected Queue Length for Two Congestion Periods with $N = 58$: Standard QIE (Left) and QIE^M, $M = 5$ (Right)

Also note that the QIE$^{\mathcal{M}}$ algorithm does not always do better than the standard QIE (see, for instance, the queue statistics for $N = 18$); however, averaged over many runs, there will be a marked improvement. It is also interesting to compare the two $N = 58$ congestion periods, to see the difference in run times and in accuracy.

In Figure 7.6 we present a comparison of the performance of the QIE$^{\mathcal{M}}$ algorithm, as the value of M varies, for the congestion period of length $N = 18$. We also present the performance of the standard QIE algorithm, in the row labelled "NO MAT." As demonstrated by this figure, the values of M which provide the most accurate estimate of queue length are not necessarily predictable for a single congestion period. However, it is hypothesized that an optimal mat placement does exist in an ergodic sense for a queue with specified parameters. The queue statistics corresponding to this figure are provided in Table 7.2.

The issue of mat placement brings up the more general problem that if one does have some customer-tracking technology available, there are many practical decisions which must be made in order to use that technology to the best advantage. And if the technology is not available and one wishes to find full density functions as an improved performance measure, or to use bounds and approximations to speed up analysis, there are other practical decisions to be made as to which algorithm or algorithms to employ, in which situations. These issues are addressed in the next (and final) chapter.

Size of Cong. Period	Data Used	ϵ	E[L_Q|...]	E[W_Q|...] (minutes)	Run Time (secs)
$N = 14$	Actual	—	1.8149	7.4835	—
	QIE	1.0962	2.8067	11.5727	0.172
	QIE$^{\mathcal{M}}$, $M = 3$	0.2909	1.8123	7.4727	0.027
$N = 13$	Actual	—	3.6868	16.8793	—
	QIE	1.4417	2.5231	11.5515	0.133
	QIE$^{\mathcal{M}}$, $M = 3$	0.6742	3.2784	15.0092	0.039
$N = 14$	Actual	—	1.9843	6.5792	—
	QIE	0.8015	2.6239	8.6997	0.173
	QIE$^{\mathcal{M}}$, $M = 3$	0.2935	1.8419	6.1070	0.027
$N = 18$	Actual	—	2.7095	11.9771	—
	QIE	0.7426	2.8649	12.6644	0.404
	QIE$^{\mathcal{M}}$, $M = 3$	0.5127	3.0174	13.3384	0.040
$N = 21$	Actual	—	2.8922	10.2991	—
	QIE	0.8013	3.3871	12.0615	0.694
	QIE$^{\mathcal{M}}$, $M = 3$	0.4056	3.1268	11.1346	0.026
$N = 12$	Actual	—	1.7390	6.1643	—
	QIE	0.7353	1.8193	6.4488	0.103
	QIE$^{\mathcal{M}}$, $M = 3$	0.2582	1.6969	6.0151	0.021
$N = 58$ (1)	Actual	—	9.6573	51.0872	—
	QIE	1.0962	9.3605	49.5175	31.15
	QIE$^{\mathcal{M}}$, $M = 5$	0.9363	9.2485	48.9251	9.02
$N = 58$ (2)	Actual	—	5.3768	26.2675	—
	QIE	1.6965	4.4114	21.5511	31.22
	QIE$^{\mathcal{M}}$, $M = 5$	0.4516	5.1309	25.0664	0.781

Table 7.1: Comparison of QIE and QIE$^{\mathcal{M}}$ Algorithms for Six Congestion Periods with $M = 3$ and Two Congestion Periods with $M = 5$

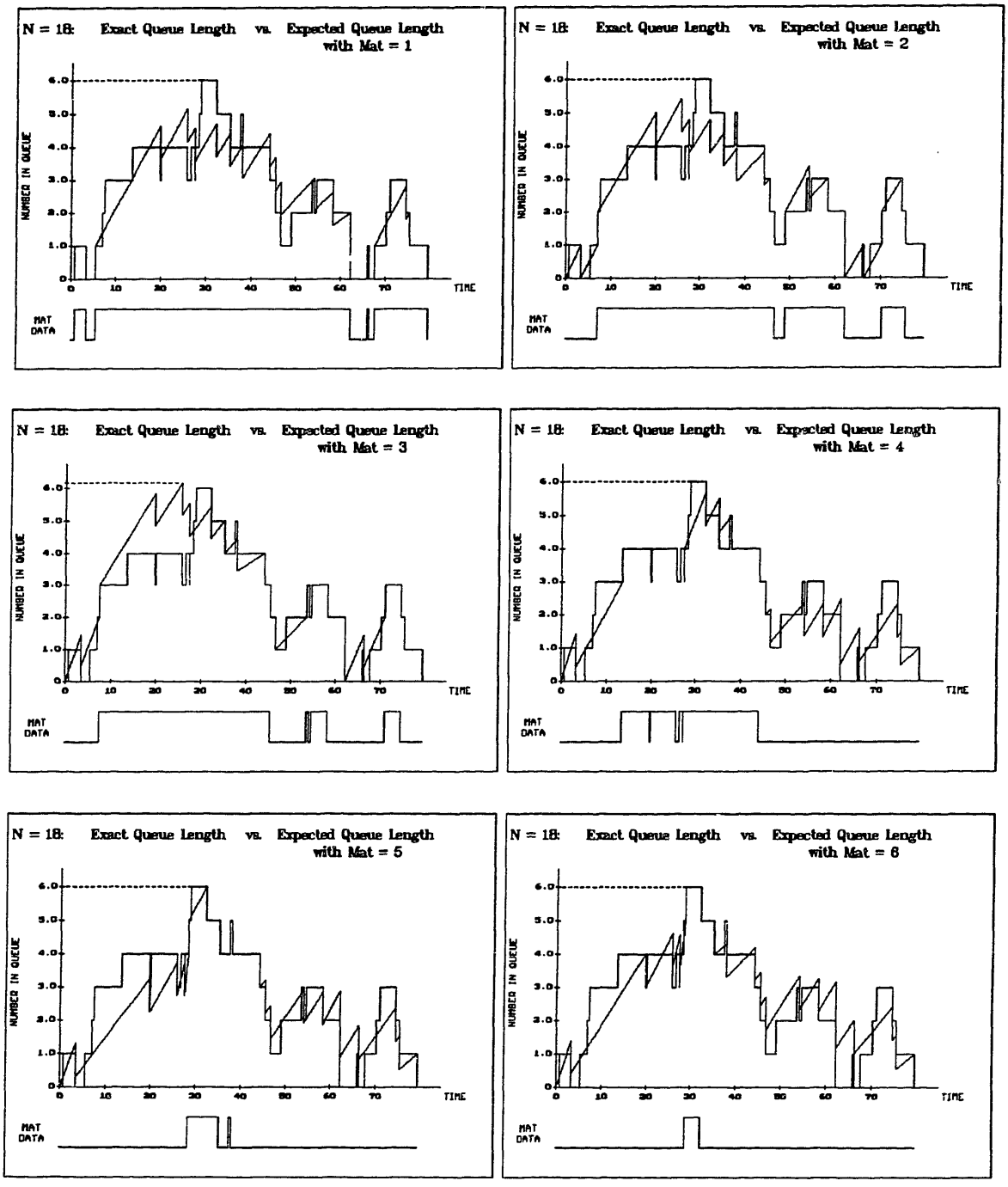


Figure 7.6: QIE$^{\mathcal{M}}$ Expected Queue Length for Congestion Period with $N = 18$ and Mat Placements at $M = 1, 2, 3, 4, 5$, and 6

Mat Placement	ϵ	E[L_Q|...] (actual = 2.7095)	E[W_Q|...] (minutes) (actual = 11.9771)
$M = 1$	0.4868	2.5744	11.3802
$M = 2$	0.4536	2.6432	11.6843
$M = 3$	0.5127	3.0174	13.3384
$M = 4$	0.4034	2.5925	11.4599
$M = 5$	0.6000	2.4720	10.9272
$M = 6$	0.5528	2.7509	12.1601
NO MAT	0.7426	2.8649	12.6644

Table 7.2: Queue Statistics for QIE^M Algorithm for a Congestion Period with $N = 18$ and $M = 1, 2, 3, 4, 5$, and 6

Chapter 8

Practical Implications and Future Research

We have found several ways to improve and extend the queue-length estimates provided by the QIE and to reduce the runtimes required. First, we provided a recursive method for finding the exact density functions for the arrival times and for the queue waiting times, conditioned on the arrival-time inequalities. Although the algorithm is slower than the original QIE algorithm, the exact density functions provide a much richer source of informaton than do the $\beta_{ki}(\mathbf{t})$ values. We gave two examples of when these more extensive performance measures might be used. In the case of ATM's, they could be used to provide customers with personalized information about their wait in queue, such as the time of day when they experienced their smallest wait, or the percentage of time that they had to wait more than 5 minutes. We also presented an airline example, in which the density functions could be used in lieu of the present method of one-month-a-year polling of customers, to generate flight-specific arrival-time density functions, which are then used to schedule ticket-counter staff.

Second, we found several alternative algorithms which give bounds and approximations to $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$, the expected cumulative number of arrivals to the system by time t, all of which have shorter runtimes than the original QIE algorithm. We

found an upper bound based on truncating the densities of the unordered arrivals. We also found a trapezoidal approximation which gives values demonstrably less than those given by the upper bound. We found several lower bounds: one was based on the concavity of the function $\mathrm{E}[A(t)|\mathcal{E}^S(\mathbf{t})]$; a second was found by omitting some of the arrival-time inequalities from the original set; and a third was found by adding the condition that the queue length be less than some specified value. These bounds and approximations can be used when large congestion periods, such as those experienced at downtown New York City ATM's at lunchtime (on the order of many hundreds of customers), have to be analyzed. The other use for bounds and approximations is to allow near-real-time analysis of congestion periods, so that, in the ATM example again, when queue lengths or queue waiting times begin to increase significantly, some ATM's may be switched over from being full-service machines to being express cash-only machines, thereby easing the traffic flow.

Finally, we found an algorithm which takes advantage of a specific type of partial queue-length information, namely, the times of all $(M-1)$-to-M and M-to-$(M-1)$ queue-length transitions, with M some pre-specified value. Because of the exact information available at these instants, we are able to break up the congestion period and analyze each partition individually, which results in more accuracy and shorter runtimes. This type of partial queue-length information could be obtained by sensor mats, electric eyes, or ultrasonic detectors. In some banks, in fact, sensor mats have been installed at position $M = 1$, i.e., at the head of the queue, so that this analysis may be applied to queue/no queue partitions of the congestion period. This mat position also helps to solve the "gap" problem which was mentioned at the beginning of Chapter 2. That is, knowing when a queue exists can help to determine when the system is actually in a congestion period, even though a gap between a service completion and the subsequent service commencement may have been sensed.

The QIE$^{\mathcal{M}}$ algorithm may be particularly useful in a setting in which the Poisson rate of arrivals varies slowly within a single congestion period, or in which there

is some semi-state-dependent balking. In the first case, since the QIE^M algorithm breaks the congestion period up into several congestion period partitions, then even if the Poisson rate varies significantly over the entire congestion period, it may be relatively stable over the course of any congestion period partition. Similarly, say that the system experiences balking such that the probability of a customer not joining the queue for queue lengths less than M is p_1, and the probability of a customer not joining the queue for queue lengths greater than or equal to M is p_2. In this case, if the Poisson rate of arrivals is approximately constant with rate λ, then during any type 1, 3, or 4 congestion period partition, arrivals are Poisson at rate λp_1, and during any type 2 congestion period partition, arrivals are Poisson at rate λp_2. If there were more than two balking rates, then, theoretically, one could add mats to correspond to the points at which the balking rate changed. This may be a simple way to capture the effects of balking on expected queue length.

All of these new algorithms can be useful to the growing number of industries with transactional data available, in order to help them manage their queueing environments to improve customer service and optimize operations. However, there are several areas of research which still need to be addressed and which would make these algorithms even more useful.

First, the densities for the queue waiting times were found only for the case of a first-come-first-served (FCFS) queueing discipline. Although this is the discipline observed in many human queueing situations, there are also some human queueing situations (large banks of ATM's, such as those at BayBank's Harvard Square branch) and other queueing environments (e.g., mobile communications) which might be better modelled as some other discipline, such as service in random order. It would be useful to try to find a simple algorithm for determining the queue waiting time densities under disciplines other than FCFS.

As was mentioned in Chapter 4, the usefulness of the bounds and approximations found there really depends on being able to guess how to combine either the uniform

upper bound or the trapezoidal approximation with the concavity lower bound. It should be fairly easy to come up with a heuristic that determines these proportions as a function of the size of the congestion period being analyzed. The nature of the **t**-vector might also have some impact on the chosen proportions, but since one of the points of using these bounds and approximations is that they are extremely fast, it would probably be defeating the purpose to do this sort of detailed analysis.

The two lower bound algorithms discussed in Chapter 6 are also in need of some heuristics. First, consider the multiple global conditions implementation of the QIE^R algorithm. Some improvements to that algorithm were already addressed in the computational section of Chapter 6: that is, we should always eliminate negative queue lengths and do concavity filtering, since so little computational time is added by these additional tasks. However, the question remains as to how many conditions and which conditions should be chosen to obtain some specified level of confidence in the tightness of the bound. Of course, the number of conditions would probably be primarily a function of the size of the congestion period. Which specific conditions to select might be more a function of the shape of the **t**-vector. The QIE^Q algorithm has the single parameter Q to be chosen: again, we would like to choose Q as a function of the size of the congestion period to ensure a relatively tight bound. The choice of Q might also depend on the shape of the **t**-vector: this is another area to be investigated.

Finally, there are several areas to be addressed with regard to the $\text{QIE}^{\mathcal{M}}$ algorithm. The first is how to ensure that person number M in queue is indeed the person who first depresses the mat; or, to turn the question around, how badly would the $\text{QIE}^{\mathcal{M}}$ estimates be thrown off if, say, the $(M-1)$-th person in queue were to stand on the mat, rather than the M-th (as assumed by the algorithm)? This is an issue of sensitivity analysis: if the algorithm is highly sensitive to which person is standing on the mat, then it may be preferable to run the standard QIE algorithm (i.e., no information may be better than bad information). If, on the other hand, the sensitivity

is not that high, then it may still be possible to use the QIE$^{\mathcal{M}}$ algorithm, although its performance will be degraded.

Another issue that must be addressed in a practical setting is that of queue-length propagation delays. In a human queueing situation, when a customer enters service, it takes some time for the entire queue to move forward to take up the slack. If M were relatively large, then in the analysis of the type 2 congestion period partition, it might make sense to use the high-frequency information provided by people stepping off and back onto the mat, as the queue moves forward, in place of the actual service-initiation times, the t_i's. This is because these service initiations might take a significant amount of time to propagate back to the mat, and analysis based on them may not provide accurate queue-length estimates.

As was seen in the previous chapter, the placement of the mat for a given queueing situation is critical. The placement may be optimized in terms of accuracy: i.e., how close do the queue estimates come to the actual values of those statistics? It may also be optimized in terms of computational complexity: how do we minimize the amount of computation to be done in the calculation of the $\beta^{\mathcal{M}}$-matrix? It is hypothesized that, in terms of accuracy, there is a single optimal location for the mat, for a given queue. Clearly, with $M = 0$ or $M = \infty$, we get zero additional information. It remains to be shown that, for a given set of queue parameters, the "best" mat location is at a single value of M.

Finally, the issue of multiple mats must be addressed. That is, say we have partial queue-length information that includes all $(M_1 - 1) \rightarrow M_1$ transitions (and back) *and* all $(M_2 - 1) \rightarrow M_2$ transitions (and back). This should be a fairly simple extension of the existing QIE$^{\mathcal{M}}$ algorithm, since we are simply breaking down the congestion period further into more congestion period partitions, now with two different sets of queue constraints. The number of mats to be used could, of course, exceed two; and the optimal number of mats to use would require a cost-benefit analysis.

One area which still remains to be investigated is how good the QIE and the other

algorithms presented in this thesis are, in a statistical sense. For example, we know that if we average the QIE estimate of the expected queue length over many congestion periods, it converges to the actual expected queue length for that particular queueing system. However, we do not know the variance of our estimates on any given run of the QIE. Similarly, we would like to know how close our bounds and approximations are to the exact QIE, both in a long-run average sense, and for any single congestion period analysis. We would also like to know how much better the $QIE^{\mathcal{M}}$ algorithm is than the standard QIE; we might decide on this basis whether implementation of some customer-tracking technology to provide partial queue-length information is worthwhile. This is a large area of research, which is vital if managers are to be convinced that the information generated by these algorithms is of value to them.

Finally, in a more general sense, as was noted in Chapter 1, the amount of transactional data which is available to operations managers is increasing exponentially. Finding ways of gleaning useful information from these data is an enormous and potentially critical area of research, in these days of global competition and shrinking profit margins. We have presented algorithms for divining queue estimates from service-time data, one small area of transactional data analysis. But there is a lot of data out there, waiting to be turned into useful information by operations researchers, to help managers fine-tune both their manufacturing and service operations.

Bibliography

[Barl 72] Barlow, R. E., D. J. Bartholomew, J. M. Brenner, and H. D. Brunk, *Statistical Inference under Order Restriction,* John Wiley and Sons, New York, NY, 1972.

[Bert 91] Bertsimas, D. J. and L. D. Servi, "Deducing Queueing from Transactional Data: the Queue Inference Engine Revisited," Technical Report OR 212-90, Operations Research Center, Massachusetts Instititute of Technology, Cambridge, MA, 1991.

[Chun 60] Chung, K. L., *Markov Chains with Stationary Transition Probabilities,* Springer-Verlag, Berlin, Germany, 1960.

[Dale 91] Daley, D. J. and L. D. Servi, "Exploiting Markov Chains to Infer Queue-Length from Transactional Data," submitted to *Journal of Applied Probability,* 1991.

[Davi 81] David, H. A., *Order Statistics,* John Wiley and Sons, New York, NY, 1981.

[Jone 91] Jones, L. K. and R. C. Larson, "Efficient Computation of Probabilities of Events Described by Order Statistics and Application to a Problem of Queues," Technical Report OR 249-91, Operations Research Center, Massachusetts Instititute of Technology, Cambridge, MA, May 1991.

[Lars 90] Larson, R. C., "The Queue Inference Engine: Deducing Queue Statistics from Transactional Data," *Management Science,* Vol. 36, No. 5, 1990, pp. 586–601.

[Lars 91] Larson, R. C., "The Queue Inference Engine: Addendum," *Management Science,* Vol. 37, No. 8, 1991.

[Lars 92] Larson, R. C., personal communication, January 1992.

[Ross 83] Ross, S. M., *Stochastic Processes,* John Wiley and Sons, New York, NY, 1983.

[Rudi 76] Rudin, W., *Principles of Mathematical Analysis, Third Edition,* McGraw-Hill Book Company, New York, NY, 1976.

[Scan 83] Scanlon, L. J., *IBM PC Assembly Language: A Guide for Programmers,* Robert J. Brady Company, Bowie, MD, 1983.

[Wolf 82] Wolff, R. W., "Poisson Arrivals See Time Averages," *Operations Research,* Vol. 30, No. 2, 1982, pp. 223-231.