University of Alberta

Essays on Stochastic Models of Service Systems

by

Fernanda Maria Campello de Souza

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Operations and Information Systems

Faculty of Business

Fall 2013

Edmonton, Alberta

To my parents.

ABSTRACT

We propose methods to plan capacity for two types of service systems: traditional multiserver systems, where customers wait in a single queue to be served by the first available service provider, in a single processing step (e.g., bank tellers); and case manager systems, where customers wait in a single queue to be assigned to a case manager who will handle all (multiple) processing steps required to complete service to that customer (e.g, emergency departments physicians).

Many researchers have addressed the problem of determining staffing requirements for traditional multiserver service systems. These requirements are often determined by segmenting time into periods and using a sequence of steady-state queueing models. The resulting requirements are approximate because nonstationary and transient effects are not considered. We propose using a non-stationary infinite-server model to determine staffing requirements for a finite-server model with the same arrival process. We prove that the resulting staffing requirements are necessary in the sense that the number of servers in a period must be greater than or equal to that period's staffing requirement in order to achieve the desired quality of service, regardless of how the system was staffed in previous periods. The requirements are exact in the sense that no steady-state approximation is used. We demonstrate the effectiveness of the requirements with numerical examples.

Comparatively few researchers have studied case manager systems, despite its ubiquity in real-world service systems. We propose a baseline stochastic model for this type of system, along with three stochastic models to aid in performance evaluation and capacity planning: a model that provides lower bounds on performance measures and approximates stability conditions for the baseline system, a model

that provides upper bounds on performance measures for the baseline system, and a model that approximates performance measures for the baseline system. We also examine how waiting times in case manager systems are affected by the imposition of an upper limit on the number of customers simultaneously handled by each case manager, and propose heuristic methods for choosing such a limit effectively.

ACKNOWLEDGEMENTS

My sincere thanks to my supervisor Prof. Armann Ingolfsson for giving so generously of his time and knowledge throughout this process, supporting me in all aspects of the PhD program. Thank you also to Prof. Robert Shumsky, co-author of the Queueing Models of Case Managers project in Chapter 5, for his great contribution to this work. I am also grateful to my supervisory committee members Prof. Bora Kolfal and Prof. Yonghua Ji, and to all members of the Operations and Information Systems research group at the University of Alberta for their valuable input throughout these years. Finally, I am grateful for all the encouragement from my family and friends. This dissertation would not have been possible without them. Thank you especially to my parents, Fernando and Tania Campello, for their love and for the emotional and financial support.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1

Introduction

This dissertation, written in partial fulfillment of the requirements for a Ph.D. degree in Operations and Informations Systems at the University of Alberta School of Business, studies the problem of matching supply to demand in a service system, while balancing customers waiting and operational costs. We examine two different service system settings: traditional multiserver queueing systems in which arriving customers are randomly assigned to any available server (e.g., bank tellers, call centers, and airport check-in); and case manager systems, where the service provided to a customer is composed of a random number of processing steps, all of which are handled by the same server (e.g, social workers, emergency departments physicians, and instant chat agents).

A typical problem in traditional multiserver systems, which many researchers have addressed (Green and Kolesar 1991, Green et al. 2001, 2007, Ingolfsson et al. 2010), is finding minimum staffing levels that guarantee a desired quality-of-service (QoS), defined in terms of customer waiting times. Since there is typically not only random, but also predictable variability in demand, most service systems are nonstationary and difficult to analyze. Most past work has focused on finding approximations to the numbers of servers that are both necessary and sufficient to ensure a particular QoS. Although approximate approaches provide good solutions in many cases, there are situations where they are not reliable, making the QoS fall below the desired level (Green et al. 2001), or providing expensive solutions. We address staffing for nonstationary systems in Chapters 2 – 4. In contrast to most past

work, our objective is obtaining lower bounds on staffing for the system that are exact (rather than approximate) and represent necessary (rather than approximately sufficient) conditions for ensuring the desired QoS.

In Chapter 2 we propose using a non-stationary infinite-server queueing model to determine staffing requirements for an otherwise identical finite-server queueing model, with the same arrival process, explicitly modeling the end-of-shift policy, which specifies what happens to a customer when his server is scheduled to leave but service is not yet completed. We prove that the staffing requirements proposed are necessary in the sense that the number of servers in a period must be greater than or equal to that period's staffing requirement in order to achieve the desired quality of service, regardless of how the system was staffed in previous periods. These proofs involve stochastic comparisons of performance measures of queueing systems with time-varying arrival rate and number of servers, general distributions for the interarrival and service times, and under both preemptive and exhaustive end-of-shift policies. We also prove a similar result for systems with abandonment.

In Chapter 3 we discuss in detail the computation of performance measures for the infinite-server queueing systems examined in Chapter 2. We discuss both numerical and simulation methods, outlining the necessary algorithms and their computational complexity. We focus on the computation of the typical performance measure, quality of service (QoS), which typically involves two steps: (1) Compute the systems occupancy probabilities and (2) compute the probability of the waiting time not exceeding an acceptable threshold τ , given the system occupancy. We discuss these two steps separately, and we compare their computational complexity.

In Chapter 4 we provide examples to illustrate the effectiveness of the staffing requirements proposed in Chapter 2 for different systems with a wide range of parameters. This chapter also includes real-world examples, where our lower bounds are used to tighten scheduling formulations.

Many service systems use case managers, servers who are assigned multiple customers and have frequent, repeated interactions with each customer until the customer's service is completed. Examples may be found in health care (emergency

department physicians), contact centers (agents handling multiple on-line chats simultaneously) and social welfare agencies (social workers with multiple clients). Although case managers are very common in service systems, they have received little attention in academia in comparison to standard multiserver systems. In Chapter 5 we propose a baseline stochastic model of a case manager system, formulate models that provide performance bounds and stability conditions for the baseline system, and formulate a birth-death process that approximates the baseline system's performance. Many systems place an upper limit on the number of customers simultaneously handled by each case manager. We examine the impact of these case-load limits on waiting time and describe effective, heuristic methods for setting these limits.

This dissertation contains two independent parts: Chapters 2 – 4 and Chapter 5. The common bibliography and appendices for both parts are collected at the end.

CHAPTER 2

Exact Necessary Staffing Requirements based on Stochastic Comparisons with Infinite-Server Models

2.1 Introduction

The issue of determining staffing levels is relevant to a variety of service systems, such as call centers and health care facilities. A typical objective is finding minimum staffing requirements that ensure a desired quality of service (QoS), taking into account the randomness and the predictable variability in the demand for service. Labor agreements typically limit how often staffing can be changed and therefore, although the demand rate may vary continuously, staffing must remain constant over periods that we refer to as *planning periods*. Although most service systems face nonstationary demand, most methods to determine staffing requirements are based on the idea of using a series of tractable stationary models to determine staffing requirements for each planning period.

A typical approach is to use formulas for stationary finite-server systems to find the minimum number of servers to ensure the desired QoS in each planning period (where the QoS is usually defined as the proportion of customers experiencing delays shorter than a given threshold). Two examples of this approach are the segmented pointwise stationary approximation (Segmented PSA, Green and Kolesar 1991) and the stationary independent period-by-period (SIPP) approach (Green et al. 2001). In Segmented PSA, first, the number of servers required at each epoch within a planning period is computed, as if the number of servers could be changed at any

time. Second, the staffing requirement for the planning period is set to the maximum of the staffing requirements within that period. In the SIPP approach, the average arrival rate for each planning period is used in stationary finite-server system formulas to find the required staffing. Instead of exact formulas, one can use approximate Square-Root-Staffing formulas (Gans et al. 2003), which decompose the required staffing into the offered load plus "safety staffing," proportional to the square-root of the offered load, to protect against random fluctuations in demand. One justification for this formula comes from approximating the number of customers in the finite-server system by the number of busy servers in a stationary infinite-server system with identical arrival process and service times, which can in turn be approximated by a normal distribution with both the mean and variance equal to the offered load (Jennings et al. 1996, Green et al. 2007). Another justification comes from an asymptotic approximation for the Erlang-C formula in the Quality and Efficiency Driven regime (Halfin and Whitt 1981), for large systems. Borst et al. (2004) showed how the optimal safety staffing can be computed given the tradeoff between server costs and QoS.

It is important to keep in mind that methods where staffing requirements are determined for each period independently are approximations of the real problem, because the queue that builds up creates a dependence between periods. Atlason et al. (2004) and Ingolfsson et al. (2010) provide detailed examples illustrating this dependence. The Segmented PSA and SIPP approaches typically perform well in cases with short service times and short planning periods, but their performance are likely to deteriorate as service times get longer (Green et al. 2001). In some cases the performance can be improved with a refinement where a time lag is introduced in the arrival rate function before the application of the Segmented PSA or SIPP approaches (Green et al. 2001). This time lag, which shifts the arrival rate function by the mean service time, accounts for the fact that when service times are longer (and thus customers stay longer in the system) there is a time lag in the congestion. Although there are many cases where the Segmented PSA or SIPP approaches (or one of their refinements) provide good solutions for the staffing problem, there are

other cases where these approximations are not reliable, making the QoS fall below the desired threshold, as shown in Green et al. (2001).

Like the Square-Root-Staffing Formula, the modified-offered-load (MOL) approximation (Massey and Whitt 1994) and the simulation-based iterative staffing algorithm (ISA, Feldman et al. 2008) use infinite-server models, with arrival and service processes identical to those of the finite-server system of interest, as tools to obtain staffing requirements. In the MOL approximation, which we discuss in Section 4.1, the mean number of busy servers in a nonstationary infinite-server system is used to construct an arrival rate function that is used as input for an approach similar to Segmented PSA and SIPP. The ISA starts by simulating the time-dependent distribution of the number of customers in an infinite-server system. This distribution is used to obtain a staffing function that satisfies the QoS target at all times. The system is then simulated again, using the staffing function constructed in the previous step, in order to obtain a new distribution for the number of customers in the system. The algorithm iterates between generating staffing functions and simulating the system until the staffing functions stabilizes. Since the ISA is based on simulation, it can be used for systems with general arrival and service processes and under different end-of-shift policies, which specify what happens to customers when their servers are scheduled to leave but the service is not yet completed. Feldman et al. (2008) prove the convergence of the algorithm (using stochastic ordering) for the special case of a nonhomogeneous Poisson arrival process and exponentially distributed service and abandonment times. The proof does not explicitly consider the end-of-shift policy. The result we prove here, on the other hand, is for a more general system, and we explicitly consider end-of-shift policies. Zeltyn et al. (2010) also use the simulation of an infinite-server system, keeping track of the number of busy servers, to estimate the offered load and later recommend staffing levels for a network of resources in an emergency department.

Staffing requirements are often used as inputs for staff scheduling, which incorporates constraints on available shifts and tours, in addition to the QoS constraints. Alternatively, the staffing and scheduling problems can be solved jointly—see for ex-

ample the methods proposed by Ingolfsson et al. (2010) and Atlason et al. (2008), both of which perform well when average service times are long and arrival rates are highly variable. Both methods employ lower bounds on the number of servers necessary to ensure the desired QoS, with tighter bounds being likely to increase their speed. Our infinite-server staffing requirements provide such bounds, as we discuss further in Section 2.4. Other scheduling methods (for example, Atlason et al. 2004, Cezik and L'Ecuyer 2008) assume the QoS is concave in the staffing levels. Since this is not true in general, these authors suggested adding lower limits on the staffing levels to limit themselves to the "concave region", but without proof that the limits chosen were necessary to achieve the desired QoS. Bounds that are provably necessary, such as the ones we provide, might improve the effectiveness of these methods.

Bounding a finite-server system with an infinite-server system, as we do, is useful because infinite-server systems are easier to analyze. For example, compare the solution of an $M(t)/M/s$ system with that of an otherwise identical $M(t)/M/\infty$ system, by numerically solving the forward differential equations. The former requires the solution of an infinite set of differential equations, so one must truncate the state space, at some system capacity K, chosen so that the probability of reaching state K is small. The infinite-server system requires less computational effort, because it will have fewer customers, so that one can truncate at a smaller capacity K. Furthermore, if the system starts empty, then the state probabilities for the infinite-server system follow a Poisson distribution at all times, so it suffices to solve a single differential equation, for the mean of the Poisson distribution (Eick et al. 1993b).

There is another reason that infinite-server bounds are useful. To compute or estimate the virtual waiting time distribution for a nonstationary system, one can often use the following approach: (1) compute the probability distribution for the number in the system at all times (using simulation or numerical evaluation) and (2) compute the waiting time distribution, conditional on the number in the system. Step (1) typically requires much more computation than step (2). If one wishes

to compare the virtual waiting time distributions (or functions thereof) for several staffing functions for a finite-server system, then one needs to perform both steps for every staffing function. But we will see that, depending on the end-of-shift policy, in order to compute our bounds on the virtual waiting time distribution for several staffing level functions, step (1) only needs to be performed once.

We make the following contributions in Chapters 2–4: (1) We prove that the virtual waiting time process in a system with a general arrival process, independent generally distributed service times, and time-varying number of servers with either a preemptive or an exhaustive end-of-shift policy is stochastically larger than a "pseudo virtual waiting time process" that we construct from an infinite-server but otherwise identical system. (2) We use the infinite-server system to compute lower bounds on staffing for the finite-server system in order to satisfy QoS constraints. In contrast to most work on staffing requirements for queueing systems, these lower bounds are exact (rather than approximate) and they represent necessary (rather than approximately sufficient) conditions for satisfying the QoS constraints. (3) We provide stochastic ordering results for queueing systems with a time-varying number of servers, explicitly modelling the end-of-shift policy—an aspect that we believe has not received sufficient attention. Most previous work allows the arrival rate and the service rate to vary with time, but not the number of servers. Therefore, these stochastic ordering results cannot be used to generate exact staffing requirements. Feldman et al. (2008) presented stochastic ordering results for systems with a time-varying number of servers, but they only considered Markovian systems and they did not address the end-of-shift policy in their proofs. (4) We compare the lower bounds to other methods to generate staffing requirements, in terms of computation time and accuracy. (5) We demonstrate that the lower bounds can be used to speed up previously-published shift scheduling algorithms.

In Section 2.2 we review stochastic ordering definitions and results. In Section 2.3 we define the pseudo virtual waiting time for infinite-server systems and obtain stochastic ordering results comparing it to the virtual waiting time in related finite-server systems. In Section 2.4 we use these results to obtain relationships between

various performance measures for the finite- and infinite-server systems and to obtain staffing requirements. In Chapter 3 we discuss how to evaluate the performance of infinite-server systems. In Chapter 4 we show the results of computational experiments comparing our lower bounds with the MOL approximation and the SIPP approach. We also demonstrate that our lower bounds can be used to speed up the method in Ingolfsson et al. (2010).

2.2 Stochastic Ordering and Notation

We start by defining the usual stochastic ordering between stochastic processes. A random variable X is stochastically larger than random variable Y, denoted $X \geq_{st} Y$, if $\Pr\{X > a\} \geq \Pr\{Y > a\}$, for all a (Ross 1996). This relation is generalized to say that a stochastic process $\{X(t), t \geq 0\}$ is stochastically larger than a process $\{Y(t), t \geq 0\}$ if $X(t) \geq_{st} Y(t)$ for all t (Muller and Stoyan 2002).

The basic idea behind most stochastic order relations between queueing systems is to show that, under specific circumstances, if the interarrival times decrease and the service times increase, the number of customers in the system increases. Whitt (1981) reviews such results for a variety of systems, including ones by Daley and Moran (1968), Jacobs and Schach (1972), Sonderman (1979a,b). Another example is Bhaskaran (1986). In general, it is harder to obtain stochastic ordering results when the assumptions of Poisson or renewal arrival processes and exponentially distributed service times are relaxed. This is especially true when we want to obtain an ordering between the virtual waiting times for systems with different number of servers, which is our goal in this project.

We use $G/G/s$ to denote a queueing system with a general arrival process, generally distributed service times, s servers, and infinite waiting room capacity. We add the argument (t) to indicate nonstationary processes. For systems with time-varying number of servers, we use EXH to denote an exhaustive end-of-shift policy, where servers that are scheduled to leave finish the jobs they had already started (if any) before leaving, and PRE to denote a preemptive end-of-shift policy, where cus-

tomers being served by servers that are scheduled to leave are sent back to the head of the queue. We use $+G$ to indicate systems with generally distributed abandonment times. We replace G with GI when the arrivals follow a renewal process and also when the service times or abandonment times are independent and generally distributed. Similarly, we replace G with M when the arrival process is Poisson and when the service times or abandonment times are independent and exponentially distributed. So, for example, $G(t)/M/s(t)///\text{EXH} + GI$ denotes a system with a nonstationary general arrival process, independent exponentially distributed service times, a time-varying number of servers following an exhaustive end-of-shift policy, and independent identically distributed abandonment times.

We denote by $N_j(t)$ the total number of customers and $W_j(t)$ the virtual waiting time in system j at time t. For system j, the interarrival time between customers $i-1$ and i is X_j^i, customer i's service time is S_j^i, and customer i's patience time is P_j^i. We use $A_j(t)$, $D_j(t)$, and $L_j(t)$ for the cumulative number of arrivals, departures, and abandonments up to time t in system j and P_j for the sequence of patience times $\left\{P_j^1, P_j^2, \dots\right\}$. Also, we use $A_j^m(t)$, $D_j^m(t)$, and $L_j^m(t)$ for the cumulative number of arrivals, departures, and abandonments up to time t in a modified version of system j with no arrivals after time m.

2.3 Comparing Finite and Infinite-Server Queues

Consider a $G(t)/G/s$ and a $G(t)/G/\infty$ queue with identical interarrival times $\{X_1, X_2, \dots\}$ and service times $\{S_1, S_2, \dots\}$ for all customers. We index the two systems with $j = F$ for "finite" and $j = I$ for "infinite" and we drop the system subscript j on the cumulative number of arrivals $A(t)$ because that process is identical for the two systems. The virtual waiting time for the finite-server system is (see, for example, Mandelbaum et al. 2002):

$$W_F(t) = \min\left\{r \geq 0 : D_F(t+r) \geq A(t) - (s-1)\right\}. \tag{2.1}$$

Since the virtual waiting time for a finite-server system at time t is not affected

by arrivals after t we can also (equivalently) define $W_F(t)$ in terms of a modified version of this system with arrivals stopped at time t:

$$W_F(t) = \min\left\{r \geq 0 : D_F^t(t+r) \geq A(t) - (s-1)\right\}. \tag{2.2}$$

Figure 2.1 illustrates this definition. It shows a customer graph (on the top) and the virtual waiting time (on the bottom) for a two-server system. Each horizontal bar on the customer graph represents one customer, with the vertical axis indicating the order of arrival. The left edge of a customer's bar marks her arrival time and the right edge marks her departure time, with the black portion of the bar representing the time interval where the customer waits for service and the shaded portion representing the interval where the customer is being served. So for example, the third customer arrives to the system at $t = 0.5$ hours, waits until service begins at $t = 0.8$ hours and leaves the system at $t = 0.9$ hours. More details about customer graphs can be found in Ingolfsson and Grossman (2002).

The true virtual waiting time in the infinite-server system is zero, but we compare the finite-server system virtual waiting time to what we call the *pseudo virtual waiting time* $W_I(t)$ in the infinite-server system, which we define, by applying the finite-server definition (2.1) to a modified version of the infinite-server system with arrivals stopped at time t, as follows:

$$W_I(t) = \min\left\{r \geq 0 : D_I^t(t+r) \geq A(t) - (s-1)\right\}. \tag{2.3}$$

Note that $W_I(t)$ is computed using the number of servers, s, in the finite-server system.

Figure 2.2 shows the customer graph and the pseudo virtual waiting time, as in (2.3), for the infinite-server system parallel to the two-server system in Figure 2.1. Note that the customer arrival times and service times in the infinite-server system are the same as in the finite server-system, but there is no waiting in the infinite-server system.

We show that the finite-server virtual waiting time is at least as large as the

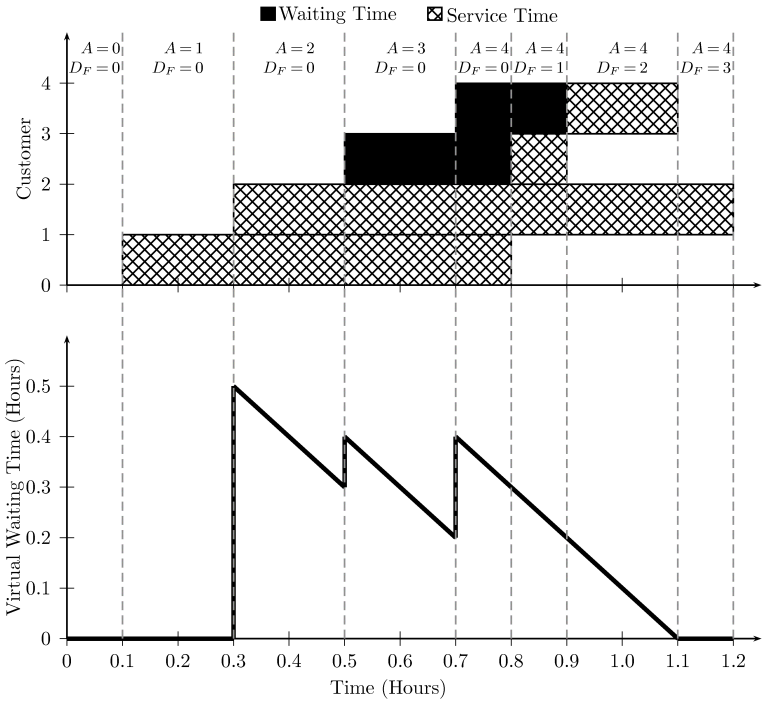

Figure 2.1: Customer graph and virtual waiting time for a $G(t)/G/2$ system.

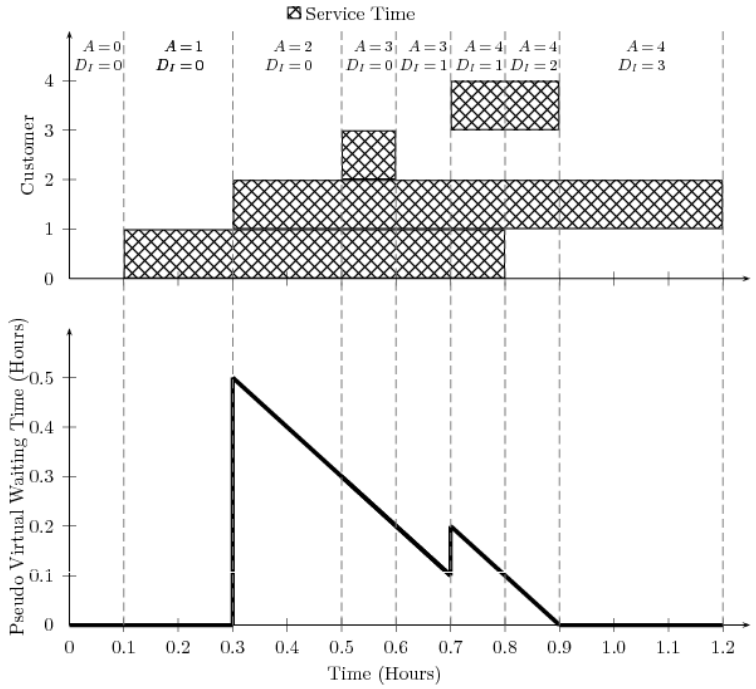

Figure 2.2: Customer graph and pseudo virtual waiting time for an infinite-server system with the same arrival process and service times as the two-server system in Figure 2.1.

infinite-server pseudo virtual waiting time for all sample paths, first for a fixed number of servers and later extending the basic result to a time-varying number of servers with an exhaustive or a preemptive end-of-shift policy, as well as systems with abandonment.

2.3.1 Queues with a Fixed Number of Servers

Theorem 2.1. *For a $G(t)/G/s$ system and a $G(t)/G/\infty$ system with identical interarrival and service times for all customers and $N_F(0) \geq N_I(0)$, it holds that:*

$$Pr\{W_F(t) \geq W_I(t)\} = 1. \tag{2.4}$$

Proof. The arrival epochs and service times for every customer are the same in both systems. Therefore, every customer's departure epoch in the infinite-server system is equal or earlier than in the finite-server system, which implies that $D_I^t(t+r) \geq D_F^t(t+r)$, for all t and all $r \geq 0$, and $\min\{r \geq 0 : D_I^t(t+r) \geq c\} \leq \min\{r \geq 0 : D_F^t(t+r) \geq c\}$, for any c. Using (2.2) and (2.3) it follows that $W_I(t) \leq W_F(t)$ for all t. □

Sample path ordering implies the usual stochastic ordering (Ross 1996), and therefore:

Corollary 2.2. *For the two systems defined in Theorem 2.1:*

$$W_F(t) \geq_{st} W_I(t) \tag{2.5}$$

Note the simplicity of the proof of Theorem 2.1, compared to most stochastic ordering proofs for queueing systems in the literature, which typically use coupling and thinning arguments. The simplicity stems from our use of the same arrival and service process in the two systems that we compare, as opposed to identically distributed but separate processes for the two systems. Our use of identical arrival and service processes does not limit the usefulness of our results, because we compare

a real finite-server system to a parallel but fictional infinite-server system whose only purpose is to aid in the analysis, as opposed to comparing two real systems.

2.3.2 Queues with a Time-Varying Number of Servers

When the number of servers $s(t)$ varies with time, we extend the virtual waiting time definitions to:

$$W_F(t) = \min\left\{r \geq 0 : D_F^t(t+r) \geq A(t) - (s(t+r) - 1)\right\}, \tag{2.6}$$

$$W_I(t) = \min\left\{r \geq 0 : D_I^t(t+r) \geq A(t) - (s(t+r) - 1)\right\}, \tag{2.7}$$

Let $R = \{t_1, t_2, ...\}$ be the set of all epochs where $s(t)$ changes, $\Delta = \{\delta_1, \delta_2, \dots\}$ be the number of servers scheduled to leave, and $\Gamma = \{\gamma_1, \gamma_2, \dots\}$ be the number of servers scheduled to arrive at each epoch in R. Note that both δ_i and γ_i can be positive for the same epoch t_i, if some servers are scheduled to leave and other servers are scheduled to arrive at the same epoch. Let t^- and t^+ denote the instants immediately before and after t, respectively.

In the interval $0 \leq t \leq t_1$, assuming $N_F(0) \geq N_I(0)$, Theorem 2.1 implies that:

$$\Pr\{W_F(t) \geq W_I(t)\} = 1 \quad \text{for } 0 \leq t \leq t_1.$$

Suppose that $s(t_1^-) = s_0$. When the γ_1 servers arrive, they begin serving the customers waiting in line, if any. As a result, these customers will depart earlier than if no new servers had arrived, but no earlier than in the infinite-server system, so the logic in the proof of Theorem 2.1 continues to hold.

When a server is scheduled to leave the system, we need to specify what will happen to the customer he is serving, if any. With a preemptive discipline, a customer could enter service multiple times (if preempted), in which case the virtual waiting time in (2.6) represents the time from arrival at t until service begins for the first time. A preemptive discipline may not be realistic if the customers and servers are human, in which case an exhaustive discipline could be more appropriate. We

treat both the preemptive and the exhaustive discipline.

2.3.2.1 Preemptive Discipline

The central argument in the proof of Theorem 2.1 relies on the fact that every customer leaves the infinite-server system no later than in the finite-server system. As noted above, this remains true when new servers arrive. It also remains true when servers depart in the case of a preemptive resume discipline, where service is continued from the point it stopped when the customer was preempted, because the return of a customer to the head of the line will add nonnegative waiting time to the customer's total time in the system and delay the departures even further. This is also true for a preemptive repeat discipline without re-sampling, where the customer's service time does not change, but is re-started from the beginning each time the customer enters service, thus adding not only waiting time, but also extra service (rework) time to the customer's total time in the system. Therefore, Theorem 2.1 holds when the number of servers varies with time, under a preemptive resume discipline and a preemptive repeat discipline without re-sampling.

2.3.2.2 Exhaustive Discipline

Under this discipline, although the physical number of customers waiting and receiving service at t_1^+ remains the same as at t_1^-, the number of customers affecting the delay of future customers changes.

Let δN_F be the number of servers who are scheduled to leave at t_1, but stay in the system to finish a service they started before t_1. We define $s_1 = s_0 + \gamma_1 - \delta_1$ as the number of scheduled servers for $t \in (t_1, t_2]$, excluding servers who stayed longer to finish service. We use s_1—the number of servers available to start new services—in the computation of the virtual waiting time in $(t_1, t_2]$. Also, we define $N_F(t_1^+)$ as the number of customers who are either in the queue or being served by servers scheduled to work in $t \in (t_1, t_2]$, i.e., we exclude customers who are being served by servers who were scheduled to leave the system but stayed to finish their jobs.

In other words, we model the server departures at time t_1 by "ejecting" from the system the δN_F customers that the δ_1 servers scheduled to leave were serving. In reality, the customers are not ejected; they stay in the system until their service is completed. But these customers do not impact the waiting times of future customers and therefore it is not necessary to model what happens to them. When we consider ejections, the number of customers in the finite-server system who impact the waiting times of future customers might change at t_1. If $N_F(t_1^-) \geq s_0$, all servers will be busy and therefore δ_1 customers and servers will be ejected from the system. If $N_F(t_1^-) < s_0$, the number of ejected customers will be between 0 and δ_1. As shown in Ingolfsson et al. (2007), if δ_1 servers are scheduled to leave at time t_1, with s_0 servers and $N_F(t_1^-)$ customers in the system at t_1^-, the probability ϕ of δN_F customers being ejected follows a hypergeometric distribution:

$$\phi(\delta N_F; \delta_1, s_0, N_F(t_1^-)) = \frac{\binom{N_F(t_1^-)}{\delta N_F}\binom{s_0 - N_F(t_1^-)}{\delta_1 - \delta N_F}}{\binom{s_0}{\delta_1}}, \quad \text{for } N_F(t_1^-) < s_0 \tag{2.8}$$

(Ingolfsson (2005) assumes a Markovian system but the formula above holds more generally, since we assume the servers scheduled to leave will be randomly selected from the servers currently on shift.) If we let $\pi^F(t) = (\pi_0^F(t), \pi_1^F(t), ...)$, where $\pi_i^F(t) = \Pr\{N_F(t) = i\}$, be the system's occupancy-probability vector at time t, then we have:

$$\pi^F(t_1^+) = \pi^F(t_1^-) B_F(t_1), \tag{2.9}$$

where $B_F(t_1)$ is a transition probability matrix with the following non-zero elements:

$$\begin{aligned} &b_{n_F, n_F - \delta_1} = 1, && \text{for } n_F = s_0, s_0 + 1, ... \\ &b_{n_F, n_F - \delta n} = \phi(\delta n; \delta_1, s_0, n_F), && \text{for } n_F = 0, 1, ..., s_0 - 1 \text{ and} \\ & && n_F - (s_0 - \delta_1)^+ \leq \delta n \leq \min(\delta_1, n_F). \end{aligned} \tag{2.10}$$

Figure 2.3 illustrates the computation of the virtual waiting time in a system that starts with 3 servers at time 0, and goes down to 2 servers at time $t_1 = 0.4$, because

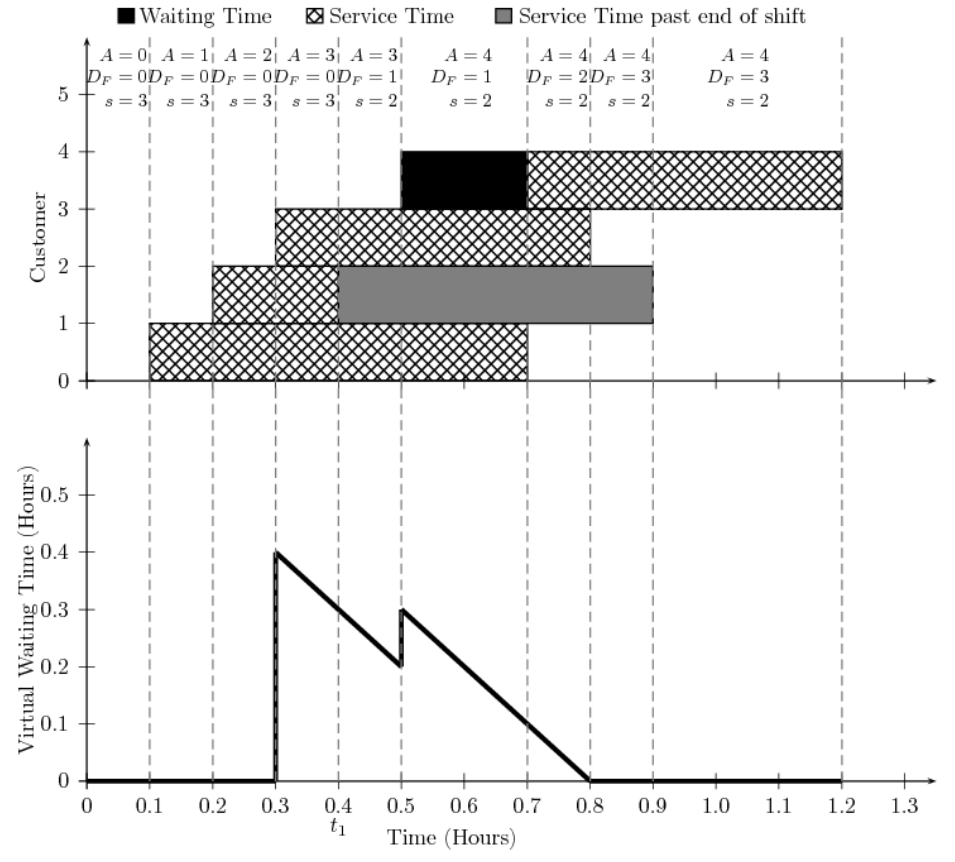

Figure 2.3: Customer graph and virtual waiting time for a finite-server system with time-varying number of servers.

$\delta_1 = 1$ server leaves and $\gamma_1 = 0$ servers arrive. In this example, the server scheduled to leave at time t_1 served Customer 2. Although both the server scheduled to leave and Customer 2 remain in the system until time $t = 0.9$, this service completion does not impact the virtual waiting time after t_1. In particular, note that the customer arriving at time 0.5 has to wait for the service completion of Customer 1, at $t = 0.7$ to begin service.

Figure 2.4 illustrates how the computation of the pseudo virtual waiting time in the infinite-server system parallel to the finite-server system in Figure 2.3 would be if, instead of redefining $N_I(t)$ and $D_I^t(t+r)$ as we do to obtain the result in Theorem 2.3, we defined $N_I(t)$ to be the total number of customers in the infinite-server system, with the cumulative number of departures $D_I^t(t+r)$ remaining unchanged right after servers leave the finite-server system. Note that in this case we would

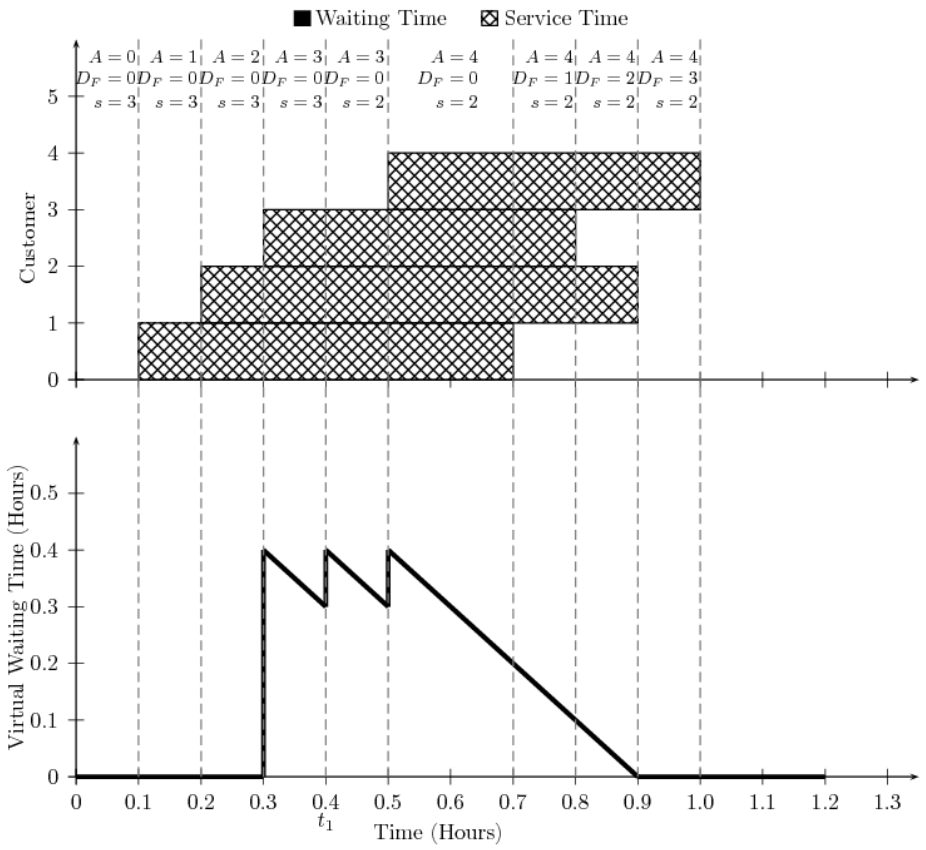

Figure 2.4: Customer graph and pseudo virtual waiting time for an infinite-server system parallel to the finite-server system in Figure 2.3, with $N_I(t)$ defined as the total number of customers in the infinite-server system, so that $D_I^t(t+r)$ is unchanged right after server departures.

have $W_I(t) \geq W_F(t)$ for $0.4 \leq t \leq 0.9$, so the desired ordering of the virtual waiting time and pseudo virtual waiting time does not hold.

The cumulative number of departures in the finite-server system increases after ejection, so in order to guarantee that the ordering $D_F^m(t_1^+) \leq D_I^m(t_1^+)$, $m < t_1^+$, is maintained, we need to eject customers in the infinite-server system as well. We discuss two possible ways of doing this. Let $\pi^I(t) = (\pi_0^I(t), \pi_1^I(t), ...)$, where $\pi_i^I(t) = \Pr\{N_I(t) = i\}$, be the infinite-server system's occupancy-probability vector at epoch t. First, suppose we eject $\delta N_I = \min(N_I(t_1^-), \delta_1)$ customers from the infinite-server system, that is, we eject one customer for every server that is scheduled to leave, up

to the total number of customers in the system. Thus we have:

$$\pi^I(t_1^+) = \pi^I(t_1^-)H(t_1), \tag{2.11}$$

where $H(t_1)$ is a transition probability matrix with the following non-zero elements:

$$\begin{aligned} h_{n_I,0} &= 1, && \text{for } n_I = 0, 1, \ldots, \delta_1 \\ h_{n_I,n_I-\delta_1} &= 1, && \text{for } n_I = \delta_1 + 1, \delta_1 + 2, ... \end{aligned} \tag{2.12}$$

We now show that the ordering between N_I and N_F, and W_I and W_F, is maintained after the ejection. Recall that the customer arrival epochs and service times are the same in the finite- and infinite-server systems. There are only two possibilities for the number of ejections in the infinite-server system:

1. If $N_I(t_1^-) \leq \delta_1$, all customers in the infinite-server system are ejected, $N_I(t_1^+) = 0$, and therefore the relationship $N_F(t_1^+) \geq N_I(t_1^+)$ is maintained. Furthermore, since the infinite-server system is empty at t_1^+, $W_F(t_1^+) \geq W_I(t_1^+)$.

2. If $N_I(t_1^-) > \delta_1$, $\delta N_I = \delta_1$. Since $0 \leq \delta N_F \leq \delta_1$, it follows that $\delta N_F \leq \delta N_I$ and therefore $N_F(t_1^+) = N_F(t_1^-) - \delta N_F \geq N_I(t_1^-) - \delta N_I = N_I(t_1^+)$. In order to guarantee that the relationship $W_F(t_1^+) \geq W_I(t_1^+)$ also holds it suffices to construct the ejections in the finite-server system from the ejections in the infinite-server system, such that the δN_F customers ejected from the finite-server system are a subset of the δN_I customers ejected from the infinite-server system.

This argument can be repeated to show that $\Pr\{W_I(t) \leq W_F(t)\} = 1$ for all t.

We summarize the comparison results for systems under preemptive and exhaustive disciplines in the following theorem.

Theorem 2.3. *Consider a finite-server system and an infinite-server system with identical interarrival and service times for every customer, with $N_F(0) \geq N_I(0)$. It follows that:*

1. *If the finite-server system is $G(t)/G/s(t)///PRE$ without resampling and the infinite-server system is $G(t)/G/\infty$,*

$$\Pr\{W_F(t) \geq W_I(t)\} = 1. \tag{2.13}$$

2. *If the finite-server system is $G(t)/G/s(t)///EXH$ and the infinite-server system is $G(t)/G/\infty$ with customers ejected according to matrix H in (2.12),*

$$\Pr\{W_F(t) \geq W_I(t)\} = 1. \tag{2.14}$$

We also examine an alternative way of ejecting customers from the system, with the occupancy-probabilities in the infinite-server system undergoing instantaneous transitions according to the matrix B_I, which has the following non-zero elements:

$$\begin{aligned} &b_{n_I,n_I-\delta_1} = 1, && \text{for } n_I = s_0, s_0+1, ... \\ &b_{n_I,n_I-\delta n} = \phi(\delta n; \delta_1, s_0, n_I), && \text{for } n_I = 0, 1, ..., s_0 - 1 \text{ and} \\ & && n_I - (s_0 - \delta_1)^+ \leq \delta n \leq \min(\delta_1, n_I). \end{aligned} \tag{2.15}$$

For this case we prove the following theorem (proof in Appendix A.1). We further discuss the difference between the two different types of ejection, using matrix H or B_I, in section 4.1.3.

Theorem 2.4. *Consider a $G(t)/G/s(t)///EXH$ finite-server system and a $G(t)/G/\infty$ infinite-server system with identical interarrival and service times for every customer and with customers ejected according to matrix B_I in (2.15), with $N_F(0) \geq N_I(0)$. It follows that:*

$$N_F(t) \geq_{st} N_I(t). \tag{2.16}$$

If we further assume that the service times are independent and identically distributed, according to an exponential distribution, it follows that:

$$W_F(t) \geq_{st} W_I(t). \tag{2.17}$$

Note that for both ordering results we have proven for systems under an exhaustive discipline, the associated infinite-server system depends on the staffing function $s(t)$ through the instantaneous transition matrices

2.3.3 Queues With Abandonment

Now we fix the number of servers at s but assume that the customers are impatient, with patience times $P = \{P_1, P_2, \dots\}$. Throughout this section, whenever we say "earlier" we mean "at the same time or earlier". Following Mandelbaum et al. (2002), we obtain the virtual waiting time at time m for an infinitely patient customer through a modified version of the original system with the arrival process interrupted at time m. Then the virtual waiting time for the modified system is given by:

$$W_F^m(t) = \min\left\{r \geq 0 : D_F^m(t+r) + L^m(t+r) \geq A^m(t) - (s-1)\right\}. \tag{2.18}$$

For the original system, we have $W_F(t) = W_F^t(t)$, for $t \geq 0$. We use the modified system with interrupted arrivals to define $W_F(t)$ to avoid situations where a customer arriving after time t abandons and causes $D_F(t+r)$ to increase for some $r \geq 0$, which could affect $W_F(t)$, even though the virtual waiting time at t should not change.

We show that the virtual waiting time in a $G(t)/G/s///+G$ system is bounded by the pseudo virtual waiting time in an infinite-server system with service time distribution equal to the minimum of the service time and patience time distributions of the finite-server system.

Theorem 2.5. *Consider a finite-server system $G(t)/G/s///+G$, with service time sequence S_F. Let an infinite-server system have the same arrival process as the finite-server system and set the service time sequence for the infinite-server system to the minimum of S_F and the patience time sequence for the finite-server system*

($S_I = \min\{S_F, P\}$). Assume that $N_F(0) \geq N_I(0)$. It follows that:

$$\Pr\{W_I(t) \leq W_F(t)\} = 1$$

Proof. Let the arrival epochs and service times for each customer be identical in the finite- and infinite-server systems. Given that $S_I = \min\{S_F, P\}$, we know that for every abandonment in the finite-server system there is an earlier departure due to service completion in the infinite-server system. We also know that for every departure due to service completion in the finite-server system there is an earlier departure due to service completion in the infinite-server system. Therefore we have $D_F(t) + L(t) \leq D_I(t)$ and also $D_F^t(t) + L^t(t) \leq D_I(t)$ for all t. Using (2.18) and (2.1) we have:

$$W_I(t) = \min\{r \geq 0 : D_I^t(t+r) \geq A(t) - (s-1)\} \leq$$
$$\min\{r \geq 0 : D_F^t(t+r) + L^t(t+r) \geq A(t) - (s-1)\} = W_F^t(t).$$

From this construction, we conclude that $\Pr\{W_I(t) \leq W_F(t)\} = 1$, for all t. □

The arguments in Sections 2.3.2.1 and 2.3.2.2 can be repeated to show that the relationship $\Pr\{W_I(t) \leq W_F(t)\} = 1$ also holds for $G(t)/G/s(t)///\text{PRE} + G$ and $G(t)/G/s(t)///\text{EXH} + G$ systems.

2.4 Comparing Performance Measures

In Section 2.3, we proved sample path ordering ($\Pr\{W_F(t) \geq W_I(t)\} = 1$) for several types of queueing systems. We now show how this implies orderings for various QoS measures and we point out one exception.

Perhaps the most common QoS measure is the proportion of customers experiencing delays less than or equal to some threshold, i.e., $\text{SL}(t) = P\{W_F(t) \leq \tau\}$. Stochastic ordering implies that $P\{W_F(t) \leq \tau\} \leq P\{W_I(t) \leq \tau\}$. Thus, the number of servers required to ensure that $P\{W_I(t) \leq \tau\} \geq \alpha$ is less than or equal to that required to ensure that $P\{W_F(t) \leq \tau\} \geq \alpha$, which leads to a lower bound on the

number of servers required to maintain a given QoS. The stochastic ordering also implies $\mathrm{E}[W_F(t)] \geq \mathrm{E}[W_I(t)]$ (Ross 1996), and therefore the pseudo virtual waiting time can be used to obtain lower bounds on staffing requirements if the expected waiting time is used as a performance measure.

The instantaneous measure $\Pr\{W(t) \leq \tau\}$ cannot be measured in a real queue, or in a simulation model. Instead, one typically uses averages over planning periods. Let $\bar{SL}_F^i$ and $\bar{SL}_I^i$ denote the time average QoS in planning period i in the finite- and infinite-server system. Since these time averages are weighted averages of instantaneous service levels, which are ordered at each instant, it follows that $\bar{SL}_F^i \leq \bar{SL}_I^i$.

Koole (2005) proposed an alternative QoS measure, called average excess (AE), defined as the average excess waiting time beyond an acceptable waiting threshold, given that the threshold is exceeded: $\mathrm{AE} = E[W(t) - \tau | W(t) > \tau]$. This measure has the advantage of eliminating the incentive for managers to give priority of service to customers who have not exceeded the waiting threshold yet (which would make the customers who exceeded the waiting threshold wait even longer). We have that:

$$E[W_F(t) - \tau | W_F(t) > \tau] = \int_0^\infty \frac{P\{W_F(t) > \tau + a\}}{P\{W_F(t) > \tau\}} da \tag{2.19}$$

and similarly for $E[W_I(t) - \tau | W_I(t) > \tau]$. In this case the sample path ordering of W_F and W_I does not necessarily imply the stochastic ordering of the QoS measure. In this case a (stronger) hazard rate order is required, where $P\{W_F(t) > \tau + a | W_F(t) > \tau\} \geq P\{W_I(t) > \tau + a | W_I(t) > \tau\}$ (Muller and Stoyan 2002). As shown in the example in Appendix A.2 it is possible for two variables to be ordered with probability one even though their distributions are not ordered in the hazard rate ordering.

For systems with abandonment, another commonly used QoS measure is the probability of abandonment $P^{ab}(t)$ at time t. Given $W_F(t) \geq_{st} W_I(t)$ (Theorem 2.5) and the probability density function (pdf) of the customer patience times, f_P,

(assumed to exist), we can write:

$$P_F^{ab}(t) = \int_0^\infty P\{W_F(t) > p\} f_P(p) dp \geq \int_0^\infty P\{W_I(t) > p\} f_P(p) dp = P_I^{ab}(t), \quad (2.20)$$

and we can obtain a lower bound for systems with abandonment when this QoS measure is used.

Our results in Section 2.3 also hold for systems operating for a finite time interval $[0, T]$. In these systems virtual waiting times are not relevant after the system shuts down, as new customers will not be admitted, but the time interval from T until all remaining customers have been served, C_F, can be an important QoS measure. Reducing this measure is typically in both the customers' and system operator's interest (to reduce overtime payments). Let C_F be:

$$\mathrm{C}_F = \inf\{r | D_F(T + r) = A(T)\}, \quad (2.21)$$

and similarly for its infinite-server counterpart C_I. Since $D_F(t) \leq_{st} D_I(t)$, then $\mathrm{C}_I(t) \leq_{st} \mathrm{C}_F(t)$. That is, the time until all the work is finished and all servers leave in the finite-server system is stochastically larger than in the infinite-server system.

CHAPTER 3

Computing Performance Measures for Infinite-Server Systems

To compute infinite-server based lower bounds, we need to evaluate the QoS in infinite-server systems efficiently. This computation can be decomposed into two steps: (1) Compute the system's occupancy probabilities, $\pi_n(t)$, and (2) compute the time-dependent QoS measure, conditional on the system occupancy n, for $n = 0, 1, ...$ We will discuss these two steps separately. We focus on the most common QoS measure, that is, $\text{SL}(t) = P\{W_F(t) \leq \tau\}$, the proportion of customers experiencing delays less than or equal to a threshold τ. Note that we define performance measures based on virtual waiting times as opposed to actual waiting times experienced. Therefore, even when using simulation, we cannot simply compute the probability that the waiting time exceeds the threshold in a single-step approach, as in the case of actual waiting times.

Closed-form solutions exist for the occupancy probabilities of $M(t)/G/\infty$ systems: if the system starts empty in the distant past, the number of customers in the system follows a Poisson distribution (Eick et al. 1993a,b), with a time-dependent mean $m(t)$ obtained from the differential equation in (3.1) for $M(t)/M/\infty$ systems and from the integral equation in (3.2) for $M(t)/G/\infty$ systems. We can evaluate (3.1) numerically using the Runge-Kutta the method (Shampine and Reichelt 1997)

and (3.2) using the Adaptive Simpson's method (Kuncir 1962), for example.

$$m'(t) = \lambda(t) - \frac{m(t)}{E[S]}, \text{ where } S \text{ is a service time.} \tag{3.1}$$

$$m(t) = \int_0^\infty \bar{G}(u)\lambda(t-u)du, \text{ where } \bar{G}(u) = 1 - G(u). \tag{3.2}$$

In cases where the system starts with a deterministic number (k) of customers at time $t = 0$ (rather than empty), Eick et al. (1993b) remark that the distribution of the number of customers in the system is the sum of two independent random variables: a Poisson random variable with mean $m(t)$ as in (3.2) and a binomial random variable with parameters k and $p = (1 - G(t))$. When the initial number of customers is a random variable (rather than deterministic) with Poisson distribution, the binomial distribution is replaced by a Poisson distribution (and the distribution of the number of customers in the system is Poisson).

Unfortunately, if the system does not start empty or with a random (Poisson) number of initial customers (for example, because of the type of instantaneous transition involved in our modeling of the exhaustive discipline) then the occupancy distribution is no longer Poisson. However, Nelson and Taaffe (2004) show how to obtain (exactly) the mean, variance and higher moments of the distribution of the number of customers in a $Ph(t)/Ph(t)/\infty$ system, where the interarrival and service times have time-dependent phase type distributions. The moments could be used to approximate the occupancy probability for any n, but this requires one to assume an approximate "closure" distribution, as in Rothkopf and Oren (1979).

In $M(t)/M/\infty$ systems, the occupancy probabilities can be computed directly by solving an infinite set of ordinary differential equations (ODEs) (the Chapman-Kolmogorov forward equations, see Kleinrock 1974b).

$$\pi_0'(t) = \mu\pi_1(t) - \lambda(t)\pi_0(t) \tag{3.3}$$

$$\pi_j'(t) = \lambda(t)\pi_{j-1}(t) + (j+1)\mu\pi_{j+1}(t) - (\lambda(t) + j\mu)\pi_j(t), \text{ for } j = 1, 2, \ldots, \tag{3.4}$$

subject to initial conditions $\pi_j(0) = q_j$ for $j = 0, \ldots, \infty$ (if the system starts empty, $q_0 = 1$ and $q_j = 0$ for all $j > 0$). This infinite set of equations can be approximated by the finite number $K+1$ of equations (with K large enough to ensure that $\pi_K(t)$ is small)

$$\pi_0'(t) = \mu\pi_1(t) - \lambda(t)\pi_0(t) \tag{3.5}$$

$$\pi_j'(t) = \lambda(t)\pi_{j-1}(t) + (j+1)\mu\pi_{j+1}(t) - (\lambda(t) + j\mu)\pi_j(t), \text{ for } j = 1, 2, \ldots, K-1 \tag{3.6}$$

$$\pi_K'(t) = \lambda(t)\pi_{K-1}(t) - K\mu\pi_K(t), \tag{3.7}$$

which can be solved using a general ODE solver or using the randomization method (Grassmann 1977).

When no analytical results are available, the occupancy probabilities can be estimated using simulation, by simulating m sample paths for the time-dependent number of customers in the system $N_i(t)$, $i \in \{1, \ldots, m\}$ and estimating the occupancy probabilities using $(1/m)\sum_{i=1}^{m} \mathbf{1}\{N_i(t) = n\}$.

The second step is to compute the distribution of the pseudo-virtual waiting time conditional on the system occupancy. Let $W_I^{n_I}(t)$ denote the pseudo-virtual waiting time for a customer that arrives at epoch t and finds n_I customers in the system. Then we can state the following theorem:

Theorem 3.1. *If s does not change in $[t, t+\tau]$, then $P(W_I^{n_I}(t) > \tau) = P(D_I'^t(\tau) \leq n_I - s)$, where $D_I'^t(\tau) = D_I^t(t+\tau) - D_I^t(t)$ is the number of departures between epochs t and $t+\tau$ given n_I customers in the system at epoch t, from a modified version of the infinite-server system with arrivals stopped at t.*

Proof. Given the definition of pseudo virtual waiting time in (2.3) we can write

$$\begin{aligned}
&P(W_I^{n_I}(t) > \tau) \\
&\quad = P(\min\left\{r \geq 0 : D_I^t(t+r) \geq A(t) - (s-1)\right\} > \tau | A(t) - D_I^t(t) = n_I) \\
&\quad = P(\min\left\{r \geq 0 : D_I^t(t+r) \geq D_I^t(t) + n_I - (s-1)\right\} > \tau | A(t) - D_I^t(t) = n_I) \\
&\quad = P(\min\left\{r \geq 0 : D_I^t(t+r) - D_I^t(t) \geq n_I - (s-1)\right\} > \tau | A(t) - D_I^t(t) = n_I) \\
&\quad = P\left(\min\left\{r \geq 0 : D_I'^t(r) \geq n_I - (s-1)\right\} > \tau\right).
\end{aligned} \tag{3.8}$$

Therefore $P(W_I^{n_I}(t) > \tau)$ is the probability that at time $t + \tau$, the number of departures from $D_I'^t(r)$ is $n_I - s$ or less. □

All n_I customers that are in the infinite-server system at time t are in service. Let $G_i^t(u)$ denote the distribution of the remaining service time after t for the customer with server $i \in \{1, \ldots, n_I\}$ and $\bar{G}_i^t(u) = (1 - G_i^t(u))$ denote the complementary distribution. Also, let $R_l^{n_I,i}$, $l = 1, \ldots, \binom{n_I}{i}$ represent the $\binom{n_I}{i}$ possible subsets of size i of the n_I busy servers. If the number of servers in the associated finite-server system $s(t)$ is constant in the interval $[t, t+\tau]$, then

$$P(W_I^{n_I}(t) > \tau) = \begin{cases} \sum_{i=0}^{n_I - s(t)} \sum_{l=1}^{\binom{n_I}{i}} \prod_{j \in R_l^{n_I,i}} G_j^t(\tau) \prod_{j \notin R_l^{n_I,i}} \bar{G}_j^t(\tau), & \text{if } n_I \geq s(t) \\ 0, & \text{otherwise} \end{cases} \tag{3.9}$$

Whitt (1999) points out that for systems with general i.i.d. service times S following a distribution G, the distribution of the remaining service time for a customer who has been in service for an unknown amount of time can be approximated by the stationary-excess distribution:

$$G_e(t) = \frac{1}{E[S]} \int_0^t [1 - G(u)] du, \qquad t \geq 0. \tag{3.10}$$

If there are n_I customers in the infinite-server system at time t, all customers are in service and the distribution of the remaining service time for each customer can be approximated by $G_e(t)$ in (3.10). We can then approximate $P(W_I^{n_I}(t) > \tau)$

for a case where the number of servers in the associated finite-server system $s(t)$ is constant in the interval $[t, t+\tau]$, by

$$P(W_I^{n_I}(t) > \tau) = \begin{cases} \sum_{i=0}^{n_I - s(t)} \binom{n_I}{i} G_e(\tau)^i \bar{G}_e(\tau)^{n_I - i}, & \text{if } n_I \geq s(t) \\ 0, & \text{otherwise.} \end{cases} \tag{3.11}$$

If the number of servers increases by δs at epoch $t + \epsilon < t + \tau$, then the pseudo virtual waiting time will be greater than τ if there are no more than $n_I - s(t) - \delta s$ departures from the system in the interval $[t, t+\tau]$ and

$$P(W_I^{n_I}(t) > \tau) = \begin{cases} \sum_{i=0}^{n_I - s(t) - \delta s} \binom{n_I}{i} G_e(\tau)^i \bar{G}_e(\tau)^{n_I - i}, & \text{if } n_I \geq s(t) + \delta s \\ 0, & \text{otherwise.} \end{cases} \tag{3.12}$$

If the number of servers decreases by δs at epoch $t + \epsilon < t + \tau$ under an exhaustive discipline and $n_I \geq s(t)$, then δs customers are ejected from the system at epoch $t + \epsilon$. Then the pseudo virtual waiting time will be greater than τ if there are no more than $n_I - \delta s - (s(t) - \delta s) = n_I - s(t)$ departures from the system in the interval $[t, t+\tau]$ and

$$P(W_I^{n_I}(t) > \tau) = \begin{cases} \sum_{i=0}^{n_I - s(t)} \binom{n_I}{i} G_e(\epsilon)^i \bar{G}_e(\epsilon)^{n_I - i} \times \\ \sum_{j=0}^{n_I - s(t) - i} \binom{n_I - \delta s - i}{j} G_e(\tau - \epsilon)^j \bar{G}_e(\tau - \epsilon)^{n_I - \delta s - i - j}, & \text{if } n_I \geq s(t) \\ 0, & \text{otherwise.} \end{cases} \tag{3.13}$$

If the number of servers decreases by δs at epoch $t + \epsilon < t + \tau$ under a preemptive discipline and $n_I \geq s(t)$, then δs customers are re-inserted in the queue at epoch $t + \epsilon$. Then the pseudo virtual waiting time will be greater than τ if there are no more than $n_I - s(t)$ departures in the interval $[t, t+\epsilon]$ and no more than $n_I - s(t) + \delta s$

departures in the interval $[t, t+\tau]$, and

$$P(W_I^{n_I}(t) > \tau) = \begin{cases} \sum_{i=0}^{n_I - s(t)} \binom{n_I}{i} G_e(\epsilon)^i \bar{G}_e(\epsilon)^{n_I - i} \times \\ \sum_{j=0}^{n_I - s(t) + \delta s - i} \binom{n_I - i}{j} G_e(\tau - \epsilon)^j \bar{G}_e(\tau - \epsilon)^{n_I - i - j}, & \text{if } n_I \geq s(t) \\ 0, & \text{otherwise} \end{cases} \tag{3.14}$$

Notice that $s(t)$ changing in $[t, t+\tau]$ would not be a concern for a wide variety of systems, where τ is small in comparison to the staffing periods, such as call centers, where τ might be 20 seconds but staffing periods are typically at least 15 minutes long, or emergency departments, where τ might be on the order of 1 hour, but staffing levels are constant for entire shifts of 8 hours or 12 hours.

Since in a $G/M/\infty$ system where service times are exponentially distributed with service rate μ, the remaining service time for a customer that was already in service at time t is also (exactly) exponentially distributed with rate μ, (3.11) becomes (3.15) and (3.12)–(3.14) are modified in the same way.

$$P(W_I^{n_I}(t) > \tau) = \begin{cases} \sum_{i=0}^{n_I - s(t)} \binom{n_I}{i} \left[1 - e^{-\mu\tau}\right]^i \left[e^{-\mu\tau}\right]^{(n_I - i)}, & \text{if } n_I \geq s(t) \\ 0, & \text{otherwise} \end{cases} \tag{3.15}$$

Once we've computed $P(W_I^n(t) > \tau)$ we can compute the distribution of the pseudo virtual waiting time (a lower bound for the QoS in the parallel finite-server system) as follows:

$$P(W_I(t) \leq \tau) = 1 - P(W_I(t) > \tau) = 1 - \sum_{n_I = s(t)}^{+\infty} P(W_I^{n_I}(t) > \tau) \pi_{n_I}(t). \tag{3.16}$$

For systems under a preemptive discipline we can prove that $P(W_I(t) \leq \tau)$ is increasing in τ. First note that $P(W_I^{n_I}(t) > \tau)$ in (3.9) is the cumulative distribution function of a binomial process with sample size n_I and probability of success $G_e(\tau)$, evaluated at $n_I - s(t)$, which is decreasing in $G_e(\tau)$ (the higher the probability of success in a binomial process, the smaller the probability that the number of

successes in n_I trials will stay below the threshold $n_I - s(t)$). Therefore, provided that $\pi_{n_I}(t)$ remains constant when τ changes, the summation term in (3.16) is decreasing in τ, and hence $P(W_I(t) \leq \tau)$ is increasing in τ. Different choices of τ will typically produce different staffing requirements in our lower bound computations. For systems under a preemptive discipline the $\pi_{n_I}(t)$ do not depend on staffing choices and thus remain constant when we vary τ. Therefore, in the preemptive discipline case, $P(W_I(t) \leq \tau)$ is increasing in τ and our lower bounds for each period are decreasing in τ. In systems under an exhaustive discipline, on the other hand, different staffing choices can produce different instantaneous transition matrices B_I (or H) to model customer ejections in the computation of the $\pi_{n_I}(t)$. In the first period, before any customer ejections occur, a smaller τ would yield a higher staffing requirement for that period. But that could also increase the chances of customer ejections at the beginning of the second period, which would reduce the number of customers in the system. Therefore, in the exhaustive discipline case, it is not clear how changing τ impacts staffing requirements.

In practice we need to truncate the infinite summation in (3.16) at some upper limit $M-1$, chosen so that he error term $E(M) = \sum_{n_I=M}^{\infty} P(W_I^{n_I}(t) > \tau)\pi_{n_I}(t)$ is very small. When $\pi_{n_I}(t)$ follows a Poisson distribution, as in $M(t)/G/\infty$ systems, we can choose M based on the Poisson distribution, in the same way as suggested by Grassmann (1977) for the randomization method. If $\pi_{n_I}(t)$ follows a Poisson distribution, and since $P(W_I^{n_I}(t) > \tau) \leq 1$ for all n_I and t, $E(M)$ is bounded by the complementary cumulative Poisson distribution

$$E(M) \leq 1 - \sum_{n_I=0}^{M-1} \pi_{n_I}(t). \tag{3.17}$$

For small $m(t)$ we can compute the Poisson probabilities for several M until the desired threshold is reached. For large $m(t)$ we can approximate the Poisson distribution with a normal distribution and set $M = m(t) + a\sqrt{m(t)} + b$. Grassmann (1977) found that $a = 4$ and $b = 5$ guaranteed $E(M) \leq 10^{-4}$. For more general systems where state probabilities do not follow a Poisson distribution we could choose

bigger values of a and b to be more conservative. Table 3.1 summarizes the methods for computing state probabilities and service levels for various types of systems.

Table 3.1: Computational methods for evaluating service level.

System	State Probabilities	Service Level
$M(t)/M/\infty///$PRE	differential equation for $m(t)$; state probabilities from Poisson distribution with mean $m(t)$	(3.11),(3.12),(3.14) with $G_e(u) = (1 - e^{-\mu u})$
$M(t)/G/\infty///$PRE	integral equation for $m(t)$; state probabilities from Poisson distribution with mean $m(t)$	(3.11),(3.12),(3.14)
$M(t)/M/\infty///$EXH	Randomization Method	(3.11),(3.12),(3.13) with $G_e(u) = (1 - e^{-\mu u})$
$Ph(t)/Ph(t)/\infty///$PRE	Nelson and Taaffe (2004)	(3.11),(3.12),(3.14)
$G(t)/G/\infty///$PRE	Simulation	(3.11),(3.12),(3.14)
$G(t)/G/\infty///$EXH	Simulation	(3.11),(3.12),(3.13)

To have a sense for the computational effort required to evaluate service level in a infinite-server system we can examine, for example, an $M(t)/M/\infty///EXH$ system, where state probabilities undergo instantaneous transitions following matrix B_I in (2.15). If we use the randomization method, with a truncation limit M, to compute state probabilities at times $T = \{t_1, t_2, \ldots, t_n\}$, we need, for each $t_i \in T$, $(M-1)$ vector-matrix multiplications, $(M-1)$ vector-scalar multiplications, $(M-1)$ vector additions, $(M-1)$ scalar multiplications, $(M-1)$ scalar divisions, and one exponential function evaluation (see Grassmann (1977)). Additionally, for each time in $R = \{t_i \in T | \text{number of servers changes}\}$, we need one vector-matrix multiplication. To compute the service level for each $t_i \in T$ conditional on the number of customers in the system $n_I(t_i)$, in (3.11) with $G_e(\tau) = (1 - e^{-\mu\tau})$, starting from $i = 0$ and computing the subsequent terms in the summation recursively, we need 1 exponential evaluation, $[n_I(t_i) - s(t_i)]$ scalar additions, $2[n_I(t_i) - s(t_i)]$ scalar divisions, and $n_I + 2[n_I(t_i) - s(t_i)]$ scalar multiplications where $s(t_i)$ is the number of servers in the associated finite-server system at time t_i. Since we need to compute $P(W_I^{n_I}(t) > \tau)$ for $n_I = s(t_i), \ldots, K$, we need a total of 1 exponential evaluation, $\sum_{n_I=s(t_i)}^{K-1}[n_I - s(t_i)] = [K - s(t_i)][s(t_i) + K - 1]/2 - [K - s(t_i)]s(t_i) =$

$[K - s(t_i)][K - s(t_i) - 1]/2$ scalar additions, $[M - s(t_i)][K - s(t_i) - 1]$ scalar divisions, and $\sum_{n_I=s(t_i)}^{K-1}[3n_I - 2s(t_i)] = 3[K - s(t_i)][s(t_i) + K - 1]/2 - 2[K - s(t_i)]s(t_i) = [K - s(t_i)][3K - s(t_i) - 3]/2$ scalar multiplications. To compute the unconditional service level at time $t_i \in T$, in (3.16), we need $[K - s(t_i)]$ scalar multiplications and $[K - s(t_i) + 1]$ scalar additions. Table 3.2 details the number of operations needed to evaluate the service level at two epochs ($T = \{t_1, t_2\}$) in an $M(t)/M/\infty///EXH$ system with $s(t) = 4$ for $t = [0, t_1]$ and $s(t) = 2$ for $t = (t_1, t_2]$, with capacity truncated at $K = 19$, and with a truncation limit for the randomization method of $M = 100$. If $K = 19$, the vectors in the randomization method are 1×20 and the matrices are 20×20, but very sparse, with only the diagonal, upper diagonal, and lower diagonal having non-zero elements. A vector-matrix multiplication requires $(3 \times 20 - 2) = 58$ scalar multiplications and $(2 \times 20 - 2) = 38$ scalar additions, a vector-scalar multiplication requires 20 scalar multiplications, and a vector addition requires 20 scalar additions. Matrix B_I is also 20×20 and very sparse, with only 26 non-zero elements. Note that even though we did not account for the operations required to build matrix B_I in Table 3.2, the number of operations needed to compute state probabilities is more than 15 times higher than the number needed to compute conditional service levels, which is more than 24 times higher than the number needed to compute unconditional service levels.

Table 3.2: Number of operations to evaluate performance at $T = \{t_1, t_2\}$ in an $M(t)/M/\infty///EXH$ system with $s(t) = 4$ for $t \in [0, t_1]$ and $s(t) = 2$ for $t \in (t_1, t_2]$, and truncation limit $M = 100$.

	Scalar Additions	Scalar Multiplications	Scalar Divisions	Exponential Evaluation
Randomization Method				
$2(M-1) = 198$ Vector-Matrix Multiplications	7,524	11,484		
$2(M-1) = 198$ Vector-Scalar Multiplications		3,960		
$2(M-1) = 198$ Vector Additions	3,960			
Scalar Operations		198	198	2
Customer Ejections at t_1				
Vector-Matrix Multiplication	6	26		
Total for State Probabilities	**11,490**	**15,668**	**198**	**2**
Conditional SL in (3.9)				
At t_1 ($s(t_1) = 4$)	120	424	240	
At t_2 ($s(t_2) = 2$)	153	495	306	
Total	**273**	**919**	**546**	**1**
Unconditional SL in (3.16)				
At t_1 ($s(t_1) = 4$)	17	16		
At t_2 ($s(t_2) = 2$)	19	18		
Total	**36**	**34**		

CHAPTER 4

Effectiveness of Staffing Requirements: Numerical Examples

4.1 Computational Experiments

We performed computational experiments to investigate the following issues: (1) how close to feasibility (that is, satisfying QoS targets at all times) are the lower bounds we proposed in Chapter 2, (2) what is the impact of using an exhaustive rather than preemptive end-of-shift policy, (3) can our lower bounds be "repaired" to make them feasible, (4) the suitability of our lower bounds for a real-world emergency department, and (5) can our lower bounds be used to enhance the efficiency of a previously proposed staff scheduling algorithm.

We ran tests on a 3.16 GHz Windows 64-bit server with 32 Gb of RAM. We wrote the code in Matlab and used the randomization method to compute transient state probabilities. We solved the staff scheduling optimization problems using the Tomlab optimization environment (Holmström 1999), with CPLEX 11 used to solve linear and integer programs.

4.1.1 Comparison of Staffing Requirements: SIPP, MOL and Our Lower Bounds

In this subsection we compare our lower bounds to SIPP and MOL staffing requirements. We compare to the SIPP approach because it is commonly used in practice and we compare to the MOL approach because it has been found to be

consistently effective in situations with a high QoS standard (Jennings et al. 1996, Green et al. 2007) and therefore the MOL approach is an appropriate benchmark. We used the 27 test cases presented in Ingolfsson et al. (2010), which represent situations where the SIPP approach performs poorly (Green et al. 2003). A sinusoidal arrival rate of the form $\lambda(t) = \lambda\{1 + \gamma \sin(\pi t/4)\}$ was used to define the nonhomogeneous Poisson arrival process. The 12-hour planning horizon was divided into smaller calculation periods where the arrival rate was assumed to be constant. The parameter γ was set to 1 and the test problems were generated by varying the service rate ($\mu = 1, 2$ and 4), the average offered load ($r = \bar{\lambda}/\mu = 16, 32$ and 64, where the average arrival rate over a cycle is $\bar{\lambda} = (1/12)\int_0^{12} \lambda(t)dt = \lambda(1 + 2\gamma/(3\pi))$, and the length of the planning period ($\delta = 0.25, 0.5$ and 1 hours). To facilitate comparison with previous work by Green et al. (2001), we assume a waiting time threshold τ equal to zero and a preemptive discipline. The target QoS requirement considered was $P\{W_F(t) = 0\} \geq 80\%$. We used Matlab to compute the average number of customers (or number of busy servers) $m(t)$ in the $M(t)/M/\infty$ system (see Appendix A.3 for details on the MOL requirements computations). We computed our lower bound and the QoS for all test cases assuming a preemptive end-of-shift policy (See Appendix A.4 for details on the computation of our lower bounds.). We obtained SIPP requirements for each planning period by using the Queueing Toolpak, Version 4.0 (Ingolfsson and Gallop 2003) to compute the minimum number of servers need to satisfy the desired service level in a stationary $M/M/s$ system with arrival rate equal to the average arrival rate in that period.

Figure 4.1 shows our lower bound and the SIPP and MOL staffing requirements for a test case with $\mu = 2$, $r = 16$, and $\delta = 0.25$. We see that the MOL requirements are at all times above and the SIPP requirements are often below our lower bound, which means that the QoS for the SIPP approach is below the target in some of the planning periods. This pattern was repeated in all 27 test cases. The MOL requirements guaranteed the desired QoS in all test cases, with an average minimum QoS of 83.1%, while our lower bound only guaranteed it in one case and the SIPP requirements did not guarantee it in any of the test cases. The average minimum

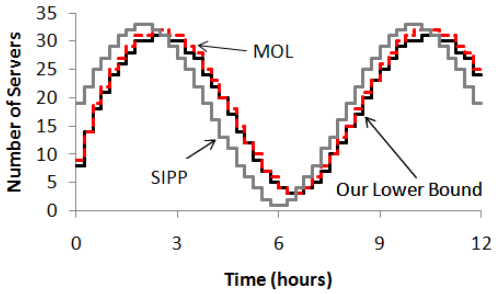

Figure 4.1: MOL requirements, SIPP requirements, and our lower bounds for the case with $\mu = 2$, $r = 16$, and $\delta = 0.25$.

QoS for the SIPP requirements was 2.0%, with the QoS being below 80% on average 52.2% of the time. For our lower bound, the average minimum QoS was 77.0%, with the QoS being below 80% on average 17.5% of the time. Since our lower bound is not intended to be sufficient, we would expect the QoS to dip below 80% in some periods. What should be noted is that our lower bounds are very close to being sufficient, despite using on average 2.9% fewer server-hours than the MOL requirements.

4.1.2 Repairing Our Lower Bounds

We used the method in Ingolfsson et al. (2010) (hereinafter referred to as the ICWC method) to "repair" our lower bounds, that is, increase the staffing requirements until the QoS target is achieved at all times. The ICWC method alternates between a schedule generator and a schedule evaluator to find low cost feasible solutions to the staffing problem, starting with a lower bound on the number of servers needed to ensure the minimum QoS in each period (see Appendix A.5 for more details). In the experiments in this section we used our lower bound as a starting point for the ICWC method. Our goal was to obtain staffing requirements and therefore we did not include any shift constraints. We assumed a preemptive end-of-shift policy to permit comparison to the MOL staffing requirements. Since the set of test problems used here was the same as in Ingolfsson et al. (2010), our parameter settings for the ICWC method follow their recommendations.

Figure 4.2 shows the requirements obtained when our lower bound was used as

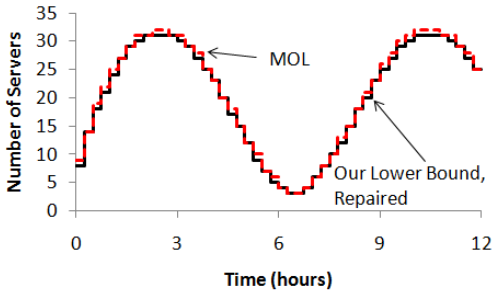

Figure 4.2: MOL requirements and our lower bounds, repaired by the ICWC method, for the case with $\mu = 2$, $r = 16$, and $\delta = 0.25$.

a starting point for the ICWC method (we refer to these requirements as "our lower bound, repaired") along with the MOL staffing requirements for the same test case as in Figure 4.1 ($\mu = 2$, $r = 16$, and $\delta = 0.25$). Our lower bounds, repaired by the ICWC method, were always less than or equal to the MOL requirements, while still ensuring the target QoS of 80%.

Table 4.1 summarizes the results for the 27 test cases. We see that our repaired lower bounds have lower costs than the MOL requirements in all test cases (average of 1.8% decrease in cost). Also, both sets of requirements guaranteed the desired QoS of 80% at all times. It is important to note that when we start the ICWC method from our lower bound the feasible staffing requirements are found very quickly (the total time for computing our lower bound and running the ICWC method was on average 30 seconds per test case), making this method of finding staffing requirements competitive with approximations such as MOL and SIPP.

4.1.3 Preemptive vs. Exhaustive Discipline

If the servers had followed an exhaustive (instead of a preemptive) end-of-shift policy, we would expect our lower bounds to be lower or equal to the ones in the previous section, because the servers would sometimes stay longer in the system to finish their services. We compared our lower bounds for the 27 test cases under a preemptive and an exhaustive discipline, both when customers are ejected from the infinite-server system according to matrix H in (2.12) and according to matrix B_I

Table 4.1: Results of the ICWC using our lower bounds as starting point for the 27 test cases, under a preemptive discipline.

			MOL		ICWC with Our Lower Bound – Requirements				% Decrease in Cost with
μ	ρ	δ	Cost	Min SL(t)	Cost	Min SL(t)	Time (min)	No. Iter.	ICWC + Our Lower Bound
1	16	0.25	239.0	83.1%	234.8	80.2%	0.36	4	1.8%
1	16	0.5	248.0	83.7%	243.0	80.3%	0.07	2	2.0%
1	16	1	265.0	84.1%	261.0	80.6%	0.06	2	1.5%
1	32	0.25	439.0	82.8%	431.5	80.0%	0.85	9	1.7%
1	32	0.5	457.0	83.3%	448.5	80.1%	0.13	2	1.9%
1	32	1	491.0	83.7%	482.0	80.1%	0.10	2	1.8%
1	64	0.25	829.3	83.5%	814.5	80.1%	1.88	9	1.8%
1	64	0.5	865.0	84.4%	848.0	80.3%	0.37	5	2.0%
1	64	1	933.0	83.9%	914.0	80.1%	0.23	3	2.0%
2	16	0.25	252.3	83.0%	246.5	80.3%	0.26	2	2.3%
2	16	0.5	264.5	84.6%	256.5	80.2%	0.07	1	3.0%
2	16	1	285.0	84.8%	276.0	80.8%	0.06	0	3.2%
2	32	0.25	465.3	82.5%	457.8	80.3%	0.85	7	1.6%
2	32	0.5	486.0	82.9%	479.0	80.8%	0.22	5	1.4%
2	32	1	526.0	83.0%	517.0	80.2%	0.11	1	1.7%
2	64	0.25	880.8	83.0%	866.8	80.0%	1.93	10	1.6%
2	64	0.5	923.0	83.5%	905.5	80.2%	0.37	4	1.9%
2	64	1	998.0	83.6%	980.0	80.2%	0.27	3	1.8%
4	16	0.25	256.8	81.5%	254.3	80.2%	0.57	8	1.0%
4	16	0.5	268.5	82.4%	263.0	80.0%	0.15	4	2.0%
4	16	1	290.0	81.9%	286.0	80.7%	0.09	3	1.4%
4	32	0.25	477.8	82.4%	470.0	80.4%	0.82	6	1.6%
4	32	0.5	498.0	82.5%	489.5	80.4%	0.24	4	1.7%
4	32	1	540.0	83.5%	533.0	80.8%	0.16	3	1.3%
4	64	0.25	901.8	82.0%	891.0	80.2%	2.36	10	1.2%
4	64	0.5	945.0	82.1%	930.0	80.1%	0.54	6	1.6%
4	64	1	1026.0	82.4%	1012.0	80.6%	0.37	4	1.4%

in (2.15). For the exhaustive discipline computations, if the scheduled number of servers increased by δs at time t, we assumed that this occurred because δs servers began work, and conversely, if the scheduled number of servers decreased by δs at time t, we assumed that this was because δs servers were scheduled to end their shift. In other words, we assumed that there were no epochs where some servers were scheduled to begin and others were scheduled to end work. Figure 4.3 shows our lower bound under preemptive and exhaustive disciplines for the same test case as in Figure 4.1. We see that the exhaustive discipline lower bounds are never above the preemptive discipline lower bounds. The exhaustive discipline lower bounds are far below the preemptive discipline lower bounds during time intervals when the number of servers is decreasing, which is when servers are more likely to stay beyond their scheduled end time. As expected, the total costs of our lower bounds under exhaustive discipline are always lower than under a preemptive discipline (on average 12.4% lower when we use matrix H and 10.6% lower when we use matrix B_I). Moreover, in each period, the required number of servers under a preemptive discipline is always at least as large as the number required under an exhaustive discipline. The lower bounds that use the H matrix have lower costs (on average 476.8 versus 485.5), lower minimum service levels (on average 39.6% versus 69.4%) and higher fractions of time below the target QoS (on average 48.3% versus 37.5%) than the ones that use the B_I matrix. This was expected, because when we use matrix H the number of costumers ejected from the infinite-server system is greater than or equal to the number ejected when we use matrix B_I. We note that the results change considerably for different end-of-shift policies, indicating that it is important to explicitly model this aspect, which has received little attention in the previous literature.

Table 4.2 shows the cost and minimum service level of the MOL requirements and of the requirements from the ICWC method using our lower bound as a starting point for the 27 test cases, under an exhaustive discipline. The MOL requirements in Table 4.2 are the same as in Table 4.1, since the MOL approach doesn't take into account specific end-of-shift policies. The service level, on the other hand, was computed

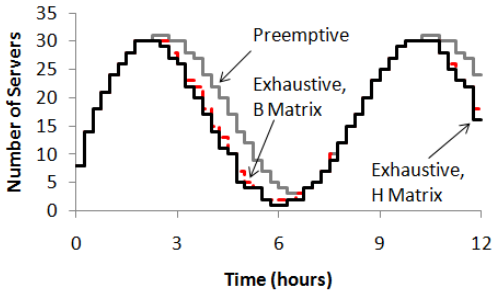

Figure 4.3: Our lower bounds under preemptive and exhaustive disciplines for the case with $\mu = 2$, $r = 16$, and $\delta = 0.25$.

under an exhaustive discipline and is higher in Table 4.2 than in Table 4.1 for every test case, as expected. Our repaired lower bounds have costs on average 10.3% lower than the MOL requirements (lower in every case). Also, both sets of requirements guaranteed the desired QoS of 80% at all times. The total time for computing our lower bound and running the ICWC method was on average 43.6 seconds per test case, higher than the average of 30 seconds for the preemptive discipline, since the computation of state probabilities under an exhaustive discipline requires extra operations (customer ejections). Even with the increase in computational time, this method of finding staffing requirements is still competitive with approximations such as MOL and SIPP for systems under an exhaustive discipline, providing even bigger cost savings than for systems under a preemptive discipline.

4.1.4 Real-World Example: Emergency Department

In this Section we compare our lower bound with MOL requirements for an emergency department (ED) in the Inwood neighborhood of northern Manhattan, studied by Green et al. (2006). Figure 4.4 shows the hourly arrival rates for weekdays we used to compute our lower bound, obtained through visual inspection of the graph in Figure 1 of Green et al. (2006). Following Green et al. (2006) we assumed an $M(t)/M/s(t)$ model with arrival rate as in Figure 4.4 and average service time of 30 minutes, and used a performance target that 80% of the ED patients were seen within 1 hour. Note that modeling an ED as a traditional multiserver system

Table 4.2: Results of the ICWC method using our lower bounds as starting point for the 27 test cases, under an exhaustive discipline (matrix B_I used in the computations of lower bounds).

			MOL		ICWC with Our Lower Bound – Requirements				% Decrease in Cost with
μ	ρ	δ	Cost	Min SL(t)	Cost	Min SL(t)	Time (min)	No. Iter.	ICWC + Our Lower Bound
1	16	0.25	239.0	84.5%	202.0	80.2%	1.02	20	15.5%
1	16	0.5	248.0	84.8%	215.5	80.2%	0.13	2	13.1%
1	16	1	265.0	84.9%	227.0	80.5%	0.08	2	14.3%
1	32	0.25	439.0	83.9%	375.0	80.4%	1.01	7	14.6%
1	32	0.5	457.0	84.4%	386.5	80.0%	0.29	6	15.4%
1	32	1	491.0	83.7%	422.0	80.2%	0.14	1	14.1%
1	64	0.25	829.3	84.4%	710.8	80.1%	2.26	5	14.3%
1	64	0.5	865.0	85.3%	732.0	80.3%	0.48	2	15.4%
1	64	1	933.0	84.2%	802.0	80.0%	0.30	3	14.0%
2	16	0.25	252.3	83.3%	228.8	80.1%	1.00	17	9.3%
2	16	0.5	264.5	84.6%	237.5	80.3%	0.17	3	10.2%
2	16	1	285.0	84.8%	253.0	80.9%	0.09	1	11.2%
2	32	0.25	465.3	82.5%	427.3	80.4%	1.05	6	8.2%
2	32	0.5	486.0	82.9%	440.0	80.1%	0.35	7	9.5%
2	32	1	526.0	83.0%	470.0	80.2%	0.19	4	10.6%
2	64	0.25	880.8	83.0%	795.5	80.0%	2.51	7	9.7%
2	64	0.5	923.0	83.5%	836.0	80.5%	0.58	3	9.4%
2	64	1	998.0	83.6%	887.0	80.2%	0.33	2	11.1%
4	16	0.25	256.8	81.5%	243.0	80.1%	0.88	9	5.4%
4	16	0.5	268.5	82.4%	249.5	80.0%	0.22	4	7.1%
4	16	1	290.0	81.8%	269.0	80.3%	0.14	4	7.2%
4	32	0.25	477.8	82.9%	447.8	80.5%	1.18	6	6.3%
4	32	0.5	498.0	83.2%	479.0	80.2%	0.37	4	3.8%
4	32	1	540.0	83.5%	498.0	80.4%	0.23	4	7.8%
4	64	0.25	901.8	82.0%	845.8	80.1%	3.19	9	6.2%
4	64	0.5	945.0	83.6%	880.0	80.1%	0.86	6	6.9%
4	64	1	1026.0	82.4%	946.0	80.6%	0.56	4	7.8%

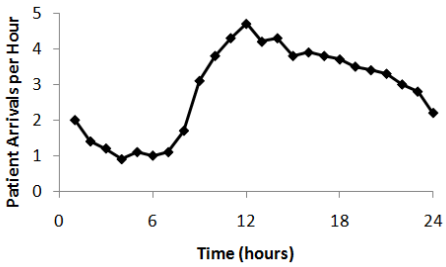

Figure 4.4: Hourly patient arrivals rates in the ED.

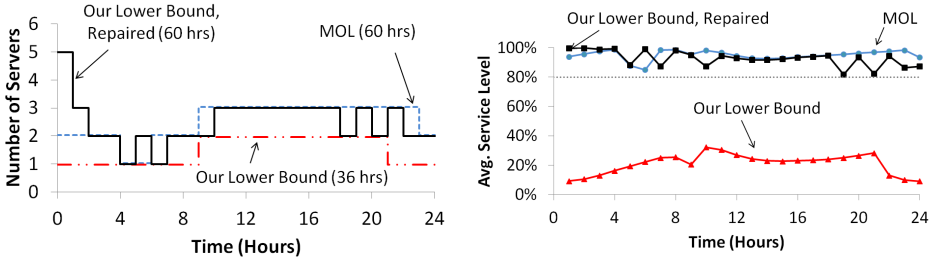

Figure 4.5: Number of servers and QoS for the MOL requirements, our lower bounds, and our lower bounds repaired to feasibility.

is a rough approximation. In Chapter 5 we propose more detailed models for this setting, which incorporate repeated patient-physician interactions.

Figure 4.5 shows the MOL requirements, our lower bound, and our lower bound repaired to feasibility, assuming a preemptive end-of-shift policy and an 1-hour staffing period, along with the hourly average QoS associated with each set of requirements.

We see that the MOL requirements meet the QoS target at all times while our lower bound falls below the target (recall that our lower bounds are not intended to be sufficient). But we can repair our lower bound (to obtain "our lower bound, repaired") using the following simple heuristic: Increase the number of servers in Staffing period 1 until the QoS target is met in that period, and repeat this procedure, in sequence, for the remaining 23 periods. Our lower bound, repaired, meets the QoS at all times and has the same cost as the MOL requirements. When we computed the requirements for 2-hour and 3-hour staffing periods, however, the cost of our lower bound was a bit higher (6.9% and 4.8% for 2-hour and 3-hour staffing

periods) than the cost of the MOL requirements.

4.1.5 Tightening Staff Scheduling Formulations with Our Lower Bounds

Another benefit of our lower bounds is decreased computation times for scheduling methods that use a lower bound as the starting point, such as the ICWC method. Note that finding tighter lower bounds for the staffing requirements as we do here is similar to finding better valid inequalities in order to tighten LP relaxations for an integer program.

We used both the original ICWC method and the ICWC method with our lower bounds as a starting point to find solutions for the set of 27 test cases, including the shift constraints in Appendix A of Ingolfsson et al. (2010). The solution costs are on average 0.4% lower with our lower bounds. However, the larger impact is on the computation times, which are reduced on average by 86.7% using our lower bounds. Our lower bounds are not only faster to compute, but they also drastically reduce the number of iterations required by the ICWC method from an average of 31.1 to an average of 2.2 iterations. This makes the ICWC method computationally competitive with the typical approach, which uses staffing requirements (for example, ones generated with the SIPP approach or one of its variants) as right-hand-sides in scheduling constraints in an integer program that is solved once.

CHAPTER 5

Queueing Models of Case Managers[1]

5.1 Introduction

Many service systems employ *case managers*: customer service agents in a contact center who manage multiple on-line chats at once; parole officers and social workers who meet with clients in crisis; and emergency department (ED) physicians who treat multiple patients simultaneously. Case manager systems are popular because they can provide highly customized service and can avoid errors and delays due to handoffs.

We define a case manager as a server who is assigned multiple customers and repeatedly interacts with those customers. Interactions between an individual customer and the case manager are usually interspersed by *external delays* that do not require the manager's attention, e.g., the delay while an on-line chat customer composes a message, the time a parole officer's client stays out of trouble, and the wait for a test result to be returned to the ED physician. Many of these systems place an upper limit on the number of customers assigned to each case manager at one time, and this leads to the formation of a *pre-assignment queue* for customers who have not yet been assigned to a case manager.

Despite the use of case managers in a wide variety of service systems, when compared to the analysis of standard multi-server systems there has been relatively little work on case manager systems in academia (we review the important existing

[1]This chapter is a joint work with Robert A. Shumsky (shumsky@dartmouth.edu), Tuck School of Business, Dartmouth College, Hanover, New Hampshire 03755.

literature in Section 5.3). In practice, the analysis and management of case manager systems is often rudimentary. For example, one method for setting caseloads proposed in the academic literature on social work is a simple deterministic calculation: divide the number of hours a case manager is available per month by the average time required per case per month (Yamatani et al. 2009). Professional organizations such as the Child Welfare League of America (CWLA) publish caseload standards, e.g., that child and family social workers handle "no more than 17 active families" (CWLA 1999). The rationale behind these standards, however, is unclear and the standards include the qualification that "every agency should conduct a workload analysis to determine the appropriate workload standards." (CWLA 1999). On their web site, the CWLA adds that "Although the field could benefit from a standardized caseload/workload model, currently there is no tested and universally accepted formula ... Yet, the CWLA standards most requested are those that provide recommended caseload and/or workload sizes." (CWLA 2013) Our models are intended to fill this need. In particular, existing standards and models do not capture the variable and unpredictable nature of the work (Yamatani et al. 2009). Our models incorporate this randomness and can be used to assess the impact of caseload limits on throughput and pre-assignment delay.

In this chapter we make the following contributions: (1) We define a model of a baseline case manager system (the 'S' system), discuss challenges with its exact analysis, and discuss tractable special cases. (2) We define random routing (R) and pooled (P) systems that we numerically show provide lower and upper bounds on the S system and we provide proofs for special cases. (3) We analyze the stability of the S, R, and P systems. (4) We define a simple balanced system (B) approximation for the waiting times in the S system. (5) We use numerical experiments to investigate the impact of changing various system parameters on the performance of the four systems, using a base case that corresponds to published data from an emergency department. (6) We identify situations in which the S system approaches the R system or the P system. (7) We investigate the tradeoff between pre-assignment delay and internal delay when the caseload limit is varied and identify methods that

may be used, in practice, to set reasonable caseloads.

5.2 Definitions and Models

In our system the service provided to a given customer, which we refer to as a *case*, is composed of a random number of processing steps, all of which are handled by the same case manager (server). When a processing step is finished either the case is completed and leaves the system or the case waits for the completion of an external delay that does not require the case manager's attention before the next processing step can begin. In an ED, for example, the processing steps are encounters with the patient's assigned physician, the external delays are diagnostic tests or requests for other information, and a particular case is completed when the patient is either discharged or admitted to the hospital.

Figure 5.1 shows our baseline model. Customers arrive according to a Poisson process with rate Λ to a pre-assignment queue where they wait to be assigned to one of N case managers who each have a maximum caseload M. When a case manager completes a case, then another case, if available, is assigned from the pre-assignment queue to that case manager. If the case manager is busy, the new case joins a FCFS *internal queue*. Otherwise, the new case immediately begins the first processing step with the case manager. The duration of each processing step is exponentially distributed with mean $1/\mu$. The probability that a case is completed after each processing step is γ. Otherwise, with probability $1-\gamma$, the case moves to an exponentially distributed external delay with mean $1/\lambda$.

If multiple case managers are below their case limits when a case arrives, then that case is immediately sent to a manager with the smallest caseload. We refer to this scheme as the join-the-smallest-caseload (JSC) routing policy. Note that the JSC policy may not be the optimal policy, although Tezcan (2011) finds that the JSC policy is asymptotically optimal for a similar system. We refer to the baseline system as the S system because of this Smallest-caseload policy.

Figure 5.2 shows the state space and transition directions of a Markov model

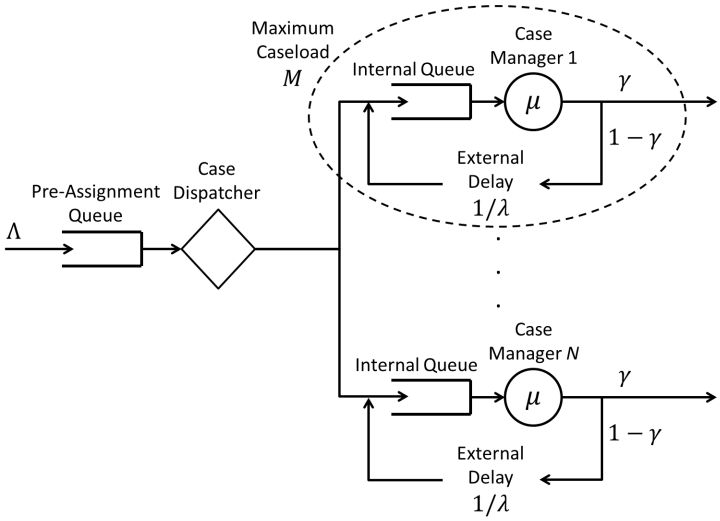

Figure 5.1: The baseline case manager S system.

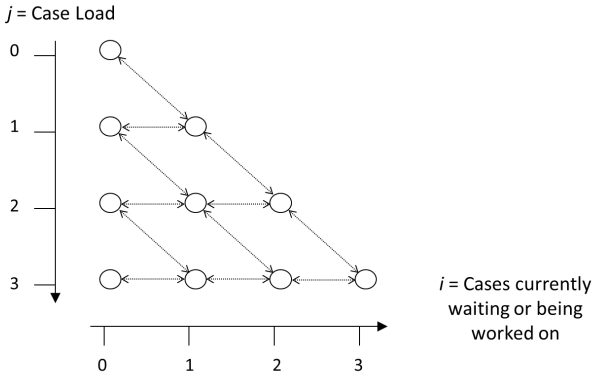

Figure 5.2: Markov model for an individual case manager with maximum caseload $M = 3$.

for an individual case manager, assuming a maximum caseload of $M = 3$. The state of this Markov model is described by the caseload j and the number of cases currently waiting or being worked on, i. The state space of a Markov model of the entire organization with N individual managers can be represented by the caseload $j_k \in \{0, \ldots, M\}$, the number of cases $i_k \in \{0, \ldots, j_k\}$ currently waiting for or being worked on by each manager $k \in \{0, \ldots, N\}$, and the number of cases waiting for assignment $c \geq 0$. If we limit the size of the pre-assignment queue to C, then $c \in \{0, \ldots, C\}$ and the state space size is

$$\left[\frac{(M+2)(M+1)}{2}\right]^N + C(M+1)^N.$$

The state space grows exponentially with the number of case managers, which makes the Markov chain representation of organizations with a large number of case managers computationally challenging, even if there is a limit on the size of the pre-assignment queue (for example, the Children, Youth and Families Department of Pittsburgh described in Yamatani et al. (2009) has $N = 112$ case managers). Even for systems where $\gamma = 1$ (the case managers are parallel exponential servers) and $N > 2$, the computation of performance measures under join-shortest-queue (JSQ) routing (equivalent to our JSC) requires various approximations (Lin and Raghavendra 1996, Nelson and Philips 1989). In Section 5.8 we use simulation to analyze the S system. We also formulate three systems that are substantially easier to analyze and generate interesting insights into system performance: two that seem to provide bounds on the S system (R = random and P = pooled) and one that approximates the S system (B = balanced).

In the R system (Figure 5.3), new case arrivals are routed randomly to one of the N case managers, so that new cases arrive to each case manager according to a Poisson process with rate Λ/N. If the manager's caseload equals M, then a new arrival to that case manager waits in a pre-assignment queue associated with that particular manager. The term "pre-assignment queue" is used here to match the analogous queue in the S system.

In the P system (Figure 5.4), cases are not assigned to a particular server; they may use any server for each processing step. If the total number of customers in service, in the internal queue, and in external delay is greater than NM, then an arriving customer waits in a pre-assignment queue. Otherwise, if all servers are busy the customer waits in a first-come-first-served internal queue that is common to all N case managers. As we will see in Section 5.3, the P system has frequently been used to describe hospital ward operations.

In the B system (Figure 5.5), we assume that a case manager handling m cases functions as an exponential server with service rate $\phi(m)$ equal to the steady state service completion rate in a related single-server finite-source ($M/M/1//m$) queueing model. We assume that arrivals are routed and cases are transferred between

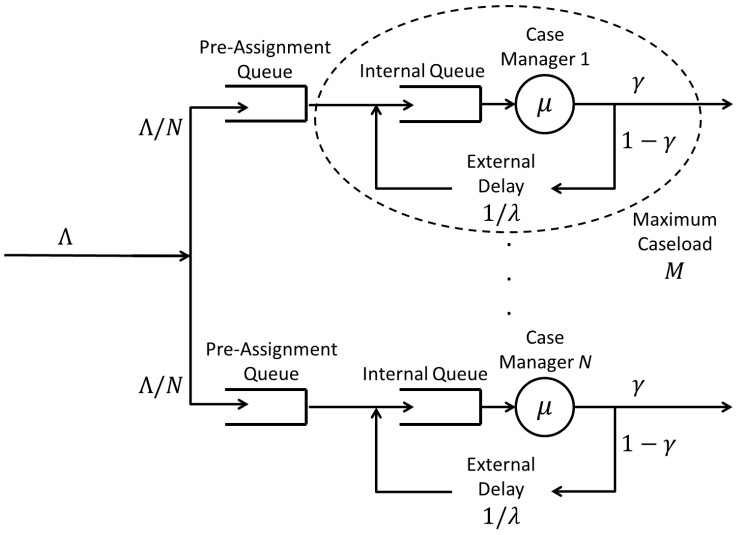

Figure 5.3: The R system.

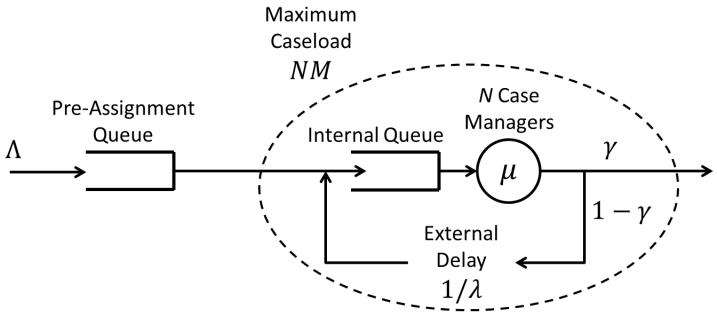

Figure 5.4: The P system.

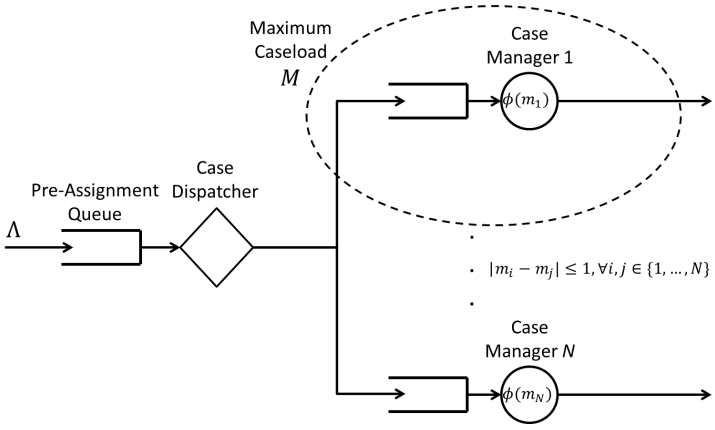

Figure 5.5: The B system.

case managers so that managers always have caseloads that are within 1 case of each other. This enables us to model the system as a simple birth-death process, as we discuss in Section 5.6.

5.3 Literature Review

There is a rich and growing literature on health-care operations that is closely related to our models. In particular, several researchers have proposed and analyzed models that are similar to our P system. Yom-Tov and Mandelbaum (2011) propose solutions to ED nurse and physician staffing problems based on the application of time-varying fluid and diffusion approximations to a pooled system with unlimited caseload. To support capacity planning decisions in an oncology ward, Yom-Tov (2010) uses a pooled model with a finite caseload, where patients are blocked when the system reaches the caseload limit. de Véricourt and Jennings (2011) examine the efficiency of nurse-to-patient ratio policies for nurse staffing using a closed $M/M/s//n$ queueing system (which is similar to our pooled system, but with a fixed number of customers and no pre-assignment queue) to model medical units. Yankovic and Green (2011) examine a finite-source queueing model with two sets of servers: nurses and beds. The variable population size allows them to include the potential change in the number of patients during a work shift. de Véricourt and Zhou (2005) describe a general model of a call center in which a

customer may revisit the system if the customer's problem is not resolved on the first call. As in our P system (and distinct from our S system), all of these models assume that any customer can be treated by any server. On the other hand, in Apte et al. (1999), case managers receive independent streams of jobs, as in the R system.

Primary care physicians may also be seen as case managers: they have their own patients (their 'panel') who repeatedly visit the physician for examination or treatment. Green and Savin (2008) model a single physician using a single-server queueing model, where the arrival rate to the physician is proportional to the panel size. This is a reasonable model because panel sizes are large (in the thousands) and the probability of arrival for any particular patient on any particular day is small. Our model, however, is designed for systems where the servers have small caseloads (1-30 customers rather than thousands) and customers may return relatively quickly to the case manager. In addition, we model the process of assigning a customer to one of multiple case managers when a customer first enters the system, while Green and Savin (2008) focus on a single physician.

Models closest to our S system may be found in Saghafian et al. (2011), Saghafian et al. (2012), Dobson et al. (2013), Tezcan (2011), and Luo and Zhang (2013). Saghafian et al. (2011) model an ED as a case worker system, as we define it, and disaggregate the analysis to "Phase 1" (similar to our pre-assignment queue) and "Phase 2" (with repeated testing and interactions with a physician). They model Phase 1 as a priority $M/G/1$ queue and focus on the triage decision, that is, whether to prioritize patients with simple or complex conditions. They analyze Phase 2 as a Markov Decision Process and focus on how a physician chooses the next patient. In our S model, we integrate Phases 1 and 2, but assume that all patients are homogeneous. Saghafian et al. (2012) use a model similar to that in Saghafian et al. (2011) to examine how patients should be routed (or "streamed") through an ED, depending on whether the patient is likely to be discharged or admitted to the hospital.

Dobson et al. (2013) (hereafter DTT) examine a case manager system that is

also motivated by an ED. Their model allows for limited capacity to serve customers in external delay, service interruptions from customers in external delay, and distinct service time distributions for the initial vs. subsequent customer-case manager encounters. Both DTT and this paper use simulation to analyze systems with separate (non-pooled) case managers. This paper differs from DTT in terms of both methodology and focus. This paper models the bounding systems as quasi-birth-death (QBD) processes, while DTT use high-caseload asymptotic analysis to examine the performance of single-server and pooled systems. DTT focus on the optimal control of the system—whether the case manager should prioritize new customers or returning customers—while we focus on system stability and the determination of caseload limits.

The models in Tezcan (2011) and Luo and Zhang (2013) are motivated by customer service chat and instant messaging systems in which each agent simultaneously serves multiple customers. In both papers, the system is approximated with a processor sharing model, that is, each agent's capacity is infinitely divisible and all customers are served simultaneously. Tezcan (2011) focuses on the optimal routing policy, and he finds that under certain conditions the optimal policy is similar to our JSC policy for system S. Luo and Zhang (2013) focus on the transient and steady state behavior of the system, given a routing policy. Both papers derive their results using a many-server asymptotic analysis. These processor sharing models are built upon general functions that describe each manager's case completion rate, given caseloads. Our models instead describe the specific interactions between customers and case managers. Our approach allows us to obtain a specific case completion rate function and to predict the impact of changes in customer or manager behavior (such as average duration of external delays or probability of service completion) on system performance.

The B system approximation is related to Gilbert's (1996) "perpetual backlog" system—a finite-source model of a single case manager that assumes the manager is always at the caseload limit. Finally, Kc (2013) empirically examines the effect of caseload levels (or "multitasking") on the productivity and service quality of ED

physicians, and we will return to his results in Section 5.8.

5.4 Analysis of the Bounding Systems

In the remainder of the paper, we use the superscripts R, S, B, and P on performance measures and other quantities to distinguish among the four systems that we discuss. In this section, we focus on the R and P systems, which we believe provide lower and upper bounds, respectively, on S system performance. Our numerical studies support this hypothesis. In addition, these easy-to-analyze systems enable us to quickly determine ranges of parameters for which the case manager system is stable, as well as the range of performance measures we could expect to find in the S system. In particular, the R and P system bounds dramatically reduce the number of simulations needed to analyze the S system. The bounds also help us to understand the dynamics of the case manager system, identifying when there is considerable advantage in the pooling effect from routing to the server with the smallest caseload, and when this advantage is small and the case manager system performs close to a random routing system.

5.4.1 Random Routing and Pooled Systems

We formulate the subsystem for each individual case manager in the R system as a QBD process (Latouche and Ramaswami 1999), with state variables i and j, where i is the total number of cases in the system (in the pre-assignment queue or assigned to the case manager) and j the number of cases in the internal queue or in service. These two state variables are sufficient to determine the pre-assignment queue length $l_a \equiv (i - M)^+$, the caseload $q = \min(i, M)$, the internal queue length $(j - 1)^+$, and an indicator variable $s = \min(j, 1)$ that equals one if the manager is busy and zero otherwise. The state space is $\Omega = \{(i, j) : i \geq 0, 0 \leq j \leq \min(i, M)\}$. We order the states (i, j) lexicographically and we treat j as the phase, with the level equal to 0 when $i < M$ and equal to $l_a + 1$ otherwise. The possible transitions are:

- Arrival of a new case: $(i,j) \to (i+1, j+1)$ with rate Λ/N, when $i < M$, and $(i,j) \to (i+1, j)$ with rate Λ/N, when $i \geq M$.
- Service completion that results in case completion: $(i,j) \to (i-1, j-1)$ with rate $s\gamma\mu$ when $i \leq M$, and $(i,j) \to (i-1, j)$ with rate $s\gamma\mu$, when $i > M$.
- Service completion that does not result in case completion: $(i,j) \to (i, j-1)$ with rate $s\,(1-\gamma)\,\mu$.
- Completion of external delay: $(i,j) \to (i, j+1)$ with rate $(q-j)\lambda$.

The general form for a QBD infinitesimal generator is:

$$Q = \begin{bmatrix} B_1 & B_0 & & & \\ B_2 & A_1 & A_0 & & \\ & A_2 & A_1 & A_0 & \\ & & A_2 & A_1 & \ddots \\ & & & \ddots & \ddots \end{bmatrix}. \tag{5.1}$$

Following QBD convention, the diagonal matrix blocks correspond to transitions where the level does not change whereas the off-diagonal blocks correspond to transitions where the level increases (above the diagonal) or decreases (below the diagonal) by one. The R and P systems both have infinitesimal generators with this general form. Appendix B.1 defines the matrix blocks B_0^R, B_1^R, and B_2^R for transitions out of, within, and into the $(M+1)M/2$ boundary states. The R system repeating matrix blocks A_0^R, A_1^R, and A_2^R are square matrices of order $M+1$ as follows (using Δ for generic diagonal elements in A_1^R and A^R):

$$A_0^R = (\Lambda/N)I, A_1^R = \begin{bmatrix} \Delta & M\lambda & & & \\ (1-\gamma)\mu & \Delta & (M-1)\lambda & & \\ & \ddots & \ddots & \ddots & \\ & & (1-\gamma)\mu & \Delta & \lambda \\ & & & (1-\gamma)\mu & \Delta \end{bmatrix}, \tag{5.2}$$

$$A_2^R = \gamma\mu \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}, \tag{5.3}$$

$$A^R = A_0^R + A_1^R + A_2^R = \begin{bmatrix} \Delta & M\lambda & & & \\ (1-\gamma)\mu & \Delta & (M-1)\lambda & & \\ & \ddots & \ddots & \ddots & \\ & & (1-\gamma)\mu & \Delta & \lambda \\ & & & (1-\gamma)\mu & \Delta \end{bmatrix}. \tag{5.4}$$

The matrix A^R is the infinitesimal generator for the Markov chain of a finite-source single-server queue with M customers that we will analyze in Section 5.5 when we investigate the stability of the R system.

We define the P system similarly to the R system, with the same state variables i and j, for the total number of customers in the system and the total number of customers in service or waiting in an internal queue, respectively. The auxiliary state variables are computed as $l_a = (i - NM)^+$, $q = \min(i, NM)$, and $s = \min(j, N)$. The possible transitions are the same as for the R system and the matrix blocks (shown in Appendix B.1) have similar structures. The sum A^P of the repeating matrix blocks corresponds to the Markov chain of a finite-source N-server queue with NM customers, which will play a role in our analysis of the stability of the P system in Section 5.5.

Let $\pi_0^k, k = R, P$ be a column vector of stationary probabilities for the boundary states, and let $\pi_n^k, k = R, P$ be a column vector of stationary probabilities for level $n, n \geq 1$ (with $l_a = n - 1$ customers in the pre-assignment queue). The probability vectors π_n^k satisfy the matrix-geometric recursion

$$\pi_{n+1}^k = \pi_n^k R^k, \quad n \geq 1, \tag{5.5}$$

where the rate matrix R^k is the minimal nonnegative solution of the nonlinear matrix equation

$$A_0^k + R^k A_1^k + (R^k)^2 A_2^k = 0, k = R, P. \tag{5.6}$$

We compute R^k using the modified SS method (Gun 1989) and we compute π_0^k and π_1^k through standard QBD analysis, as detailed in Appendix B.1.

Table 5.1 shows the performance measures that we focus on. Expressions (5.7)-(5.8) provide formulas to compute the average pre-assignment queue length, $L_a^k, k = R, P$, for the R and P systems (the queue length is aggregated over all case managers for the R system, for easier comparison to the other systems). Appendix B.1 provides similar closed-form expressions for the other performance measures, for the R and P systems.

Table 5.1: Performance measure definitions for systems $k = P, S, B, R$.

	Expected Number	Expected Time
Pre-Assignment:	L_a^k	W_a^k
Internal Queue:	L_q^k	W_q^k
External Delay:	L_e^k	$T_e^k = (1/\lambda)(1/\gamma - 1)$
Service:	$N\rho^k$	$(1/\mu)(1/\gamma)$
Total in System:	L^k	T^k

$$L_a^R = N \sum_{n=1}^{\infty} (n-1)\pi_n^R e = N\pi_1^R R^R (I - R^R)^{-2} e, \tag{5.7}$$

$$L_a^P = \sum_{n=1}^{\infty} (n-1)\pi_n^P e = \pi_1^P R^P (I - R^P)^{-2} e, \tag{5.8}$$

where e is a column vector of ones.

5.4.2 Comparing the R, S, and P systems

In the P system there is no fixed customer-server assignment and a customer at the head of the internal queue is served by the first available server. The customer does not need to wait for a particular server to be free. Therefore, a given server

is less likely to be idle due to an empty internal queue in the P system than in the S system, where there is a fixed customer-server assignment. For this reason we expect queue lengths and waiting times to be smaller in the P system than in the S system. Pooling resources that work at the same rate is known to be beneficial in many settings. For example, Smith and Whitt (1981) show that pooling two $M/M/s$ loss systems with the same service time distribution is beneficial (but pooling might not be beneficial if the service time distributions are different). Based on these considerations, we conjecture the following:

Conjecture 5.1. *For an S and a P system with the same parameters (N, M, Λ, λ, μ, and γ), $T^S \geq T^P$*

The routing in the S system is state-dependent, using dynamic caseload information for each manager in an attempt to achieve a more balanced distribution of caseloads among managers than in the R system. In a system with better balanced caseloads, the chances of having an idle server should be smaller, so we expect performance measures such as queue lengths and waiting times to be smaller in the S system than in the R system. Therefore, we conjecture the following:

Conjecture 5.2. *For an S and an R system with the same parameters (N, M, Λ, λ, μ, and γ), $T^R \geq T^S$*

These relationships have been established for the special case where $\gamma = 1$ and $M \to \infty$. In this case, the R system corresponds to N parallel, independent, and identical $M/M/1$ queues, the S system corresponds to a join-the-shortest-queue system with N parallel exponential servers and the P system corresponds to an $M/M/N$ system. Nelson and Philips (1989) argue that in this situation the number of customers in the S system is stochastically larger than number of customers in the P system, and the S system has a lower expected response time than the R system. This relationship between S and R also holds true for more general service time distributions with non-decreasing hazard rate (Weber 1978). (Whitt (1986) discusses service time distributions for which JSQ is not optimal, however.) The

bounds that we conjecture hold true for all computational experiments we have done so far, up to simulation error.

5.5 Stability Conditions

Let Λ^k_{lim} be the largest external arrival rate that system $k = R, S, P$ can accommodate without the expected length of the pre-assignment queue growing without bound. We will refer to $[0, \Lambda^k_{\text{lim}})$ as the system k stability region. Intuitively, we expect the limit on the external arrival rate to be the product of three components:

1. The number of case managers, N,
2. The rate at which a case manager clears cases when busy, $\gamma\mu$,
3. The probability that a case manager is busy, if the external arrival rate is sufficiently high to not limit the case manager's busy probability.

The product of the first two components, $N\gamma\mu$, is the rate at which the system could clear cases if all case managers were always busy. The product of the first and third components can be viewed as $E[B^k_{\text{lim}}]$, the steady state expected number of busy servers in a limiting system where all case managers have a full caseload (for the P system, this means a system caseload of NM). We expect that the P system will have a larger stability region than the R and S systems, because the P system avoids situations where a case manager is idle, while at the same time a case is waiting in internal delay.

In this section, we first demonstrate that the stability regions for the three systems coincide in the special case when $M = 1$ and in the limiting case when M approaches infinity. Then we formally prove that the limit on the external arrival rate for the R and P systems can be expressed as the product of the three components that we have mentioned and that P has a larger stability region than R. We conjecture that the R and S systems have the same stability regions and we provide numerical support for this conjecture for systems with two case managers.

When $M = 1$, a case will never wait for a case manager—its entire time with the case manager will consist of processing steps and external delays, without any internal delays. The average total time that a case is assigned to a case manager is $1/(\gamma\mu)+(1/\gamma-1)(1/\lambda)$ and out of this total, the average time that the case manager is busy is $1/(\gamma\mu)$. It follows that the proportion of time that a case manager is busy, if she has a case assigned at all times, is

$$\frac{\frac{1}{\gamma\mu}}{\frac{1}{\gamma\mu}+(\frac{1}{\gamma}-1)(\frac{1}{\lambda})} = \frac{1}{1+\frac{\gamma\mu(1-\gamma)}{\gamma\lambda}} = \frac{1}{1+\mu(1-\gamma)/\lambda} = \frac{1}{1+x}, \tag{5.9}$$

where $x = \mu(1-\gamma)/\lambda$. Therefore, the external arrival rate limit is $\Lambda^k_{\lim} = N\gamma\mu/(1+x)$ for all three systems.

When M approaches infinity, then the R and P systems can be viewed as open Jackson networks and straightforward analysis of these networks (included in Appendix B.2) shows that $\Lambda^k_{\lim} = N\gamma\mu$, that is, the external arrival rate limit equals the rate at which the system can clear cases if all case managers are busy at all times.

We provide general expressions for the external arrival rate limits for the R and P systems in Theorem 5.3. We use a general QBD ergodicity condition (Latouche and Ramaswami 1999) to prove the validity of these expressions.

Theorem 5.3. *The R and P systems are stable if and only if $\Lambda < \Lambda^k_{\lim}$ for $k = R, P$, where*

$$\Lambda^k_{\lim} = \gamma\mu E[B^k_{\lim}],\ k = R, P \tag{5.10}$$

and $B^k_{\lim}$ is the steady state number of busy servers in a limiting system for system $k = R, P$.

The limiting system $R_{\lim}$ for R is a collection of N independent and identical single-server finite-source Markovian queueing systems ($M/M/1/./M$) with population size M. The limiting system $P_{\lim}$ for P is an N-server finite-source Markovian queueing system ($M/M/N/./NM$) with population size NM. The service rate is $(1-\gamma)\mu$ and the average time until arrival is $1/\lambda$ for each customer in the popula-

tion, for both limiting systems. The steady state expected number of busy servers in these two systems can be expressed as follows:

$$E[B^R_{\text{lim}}] = N\left(\sum_{i=0}^{M} \min\{i,1\}\omega_i^R\right), \tag{5.11}$$

$$E[B^P_{\text{lim}}] = N\left(\sum_{i=0}^{NM} \min\{i/N,1\}\omega_i^P\right), \tag{5.12}$$

where ω_i^k is the steady state probability of state i in the Markov chain corresponding to matrix block A^k, for $k = R, P$.

Proof. The general QBD ergodicity condition that we use (Latouche and Ramaswami 1999, pg. 155) is that $\omega A_0 e < \omega A_2 e$, where ω is the steady state probability vector corresponding to the transition matrix $A = A_0 + A_1 + A_2$, satisfying $\omega A = 0$ and $\omega e = 1$; A_0, A_1, and A_2 are the repeating matrix blocks for the QBD.

Using the matrix blocks from (5.2)-(5.3) for the R system, $\omega^R A_0^R e < \omega^R A_2^R e$ reduces to $\Lambda < N\gamma\mu(1-\omega_0^R)$, where ω_0^R is the steady state probability of the first state in the Markov chain corresponding to matrix block A^R. Inspection of the matrix block A^R in (5.4) reveals that it corresponds to a birth-death process, whose transition diagram is illustrated in Figure 5.6. The system can be viewed as an $M/M/1/./M$ finite-source queueing system. With this interpretation, the sum of the probabilities of all but the leftmost state in Figure 5.6 equals the probability that the single server in this queueing system is busy. We refer to a collection of N such systems as R_{lim}, because this collection of single-server finite-source queueing systems describes how the R system would work if the external arrival rate was sufficiently large to ensure that all N case managers had a full caseload of M at all times. This proves (5.10) for $k = R$ and $E[B^R_{\text{lim}}]$ as given in (5.11).

The proof of (5.10) for $k = P$ and (5.12) follows the same steps. Inspection of the matrix A^P in (B.22) reveals that it is the transition matrix for an $M/M/N/./NM$ system, as illustrated in Figure 5.7. We refer to this system as P_{lim} and note that it corresponds to how the P system would operate if the external arrival rate was large enough to ensure that the system had a full caseload of NM at all times. The

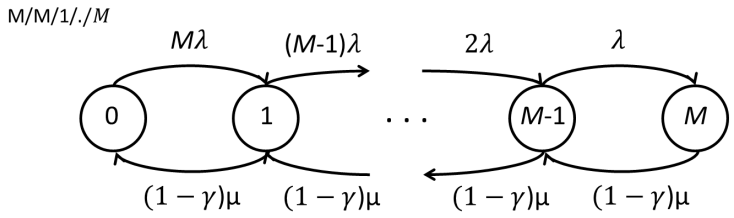

Figure 5.6: State transition diagram for the A^R matrix and the R_{lim} system.

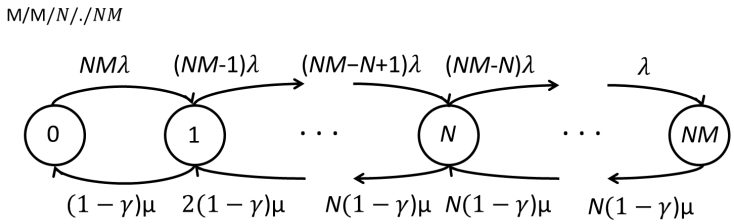

Figure 5.7: State transition diagram for the A^P matrix and the P_{lim} system.

ergodicity condition $\omega^P A_0^P e < \omega^P A_2^P e$ reduces to $\Lambda < N\gamma\mu(\sum_{i=0}^{NM} \min\{i/N, 1\}\omega_i^P)$, where ω_i^P is the steady state probability of state i in the P_{lim} system shown in Figure 5.7. The summation in parentheses is the steady state expected proportion of busy servers in the P_{lim} system.

□

Figure 5.8 shows that P has a larger stability region than R for caseload limits M between 1 and ∞ and confirms that their stability regions coincide when $M = 1$ and when $M \to \infty$. This figure was generated by using the expressions in Theorem 1 to compute $\Lambda_{\text{lim}}^k, k = R, P$ for systems with $N = 2$ case managers, with parameters $\mu = 7.5$, $\gamma = 1/3$, $\lambda = 2.1, 5.1$, and 9.6, and maximum caseload limits varying from 1 to 10. The stability limits increase when λ increases, because less time in external delay leads to less forced server idleness.

It is possible to formulate the S system as a QBD process, by combining the state variables for the caseload and queue length of each case manager into a single state variable with finite (but large) range. We did this for $N = 2$ case managers (details on the possible transitions are in Appendix B.2), in order to numerically compute stability limits for the S system (Λ_{lim}^S). We only need to generate the repeating

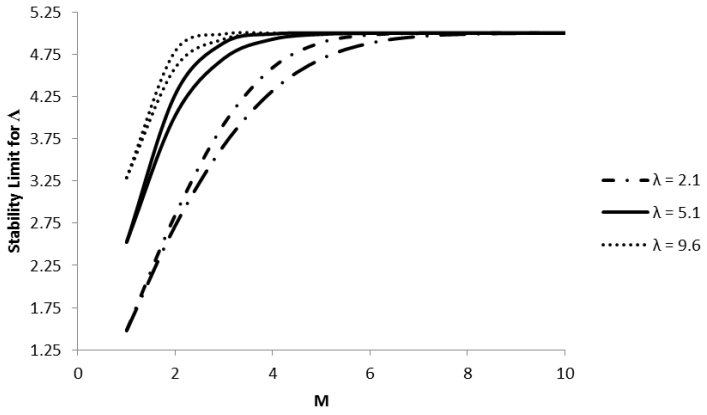

Figure 5.8: New-case arrival rate stability limits for maximum caseloads of 1 to 10 cases, for random routing (bottom curves) and pooled (top) systems with $N = 2$ case managers, $\mu = 7.5$, $\gamma = 1/3$, $\lambda = 2.1, 5.1$, and 9.6.

matrix blocks (not the boundary matrix blocks) to compute stability limits. The S system repeating matrix blocks are square matrices of order $(M+1)^N$. We verified numerically that the R and S systems have exactly the same stability limits for all values of M and λ that are shown in Figure 5.8 (as well as for many other cases that we tried, all with $N = 2$). This numerical evidence leads us to the following:

Conjecture 5.4. *For an S and an R system with the same parameters (N, M, Λ, λ, μ, and γ), $\Lambda^S_{\lim} = \Lambda^R_{\lim}$.*

In addition to the numerical evidence, we observe that if the arrival rate of new cases is sufficiently high, one would expect the internal queues of the R and S systems to behave in the same way. For such highly loaded systems, each case manager would operate, most of the time, as a single-server M-customer finite-source queue, in both the R and the S systems. The numerical results that we report in Section 5.8 (in particular, see the right panels of Figures 5.10-5.12) are consistent with these arguments.

We conclude this section by proving that $\Lambda^P_{\lim} \geq \Lambda^R_{\lim}$ in general.

Theorem 5.5. *Let $B^k_{\lim}(t)$ be the number of busy servers and $Q^k_{\lim}(t)$ be the number of customers waiting for service at time t in a $k_{\lim}$ system, where $k = R, P$. If both the $R_{\lim}$ and the $P_{\lim}$ systems start empty ($B^R_{\lim}(0) = Q^R_{\lim}(0) = B^P_{\lim}(0) = Q^P_{\lim}(0)$),*

then $B^P_{\lim} \geq_{st} B^R_{\lim}$*, which implies that* $\Lambda^P_{\lim} = \gamma\mu E[B^P_{\lim}] \geq \gamma\mu E[B^R_{\lim}] = \Lambda^R_{\lim}$.

Proof. For $t = 0$ it is true that $B^P_{\lim}(t) \geq_{st} B^R_{\lim}(t)$. Assume that $B^P_{\lim}(t) \geq_{st} B^R_{\lim}(t)$ for $t \in [0, t']$ and that $B^P_{\lim}(t') = B^R_{\lim}(t') = b' > 0$. We will prove, using a coupling argument, that the desired order will continue to hold after the next event after time t'.

If $Q^P_{\lim}(t') > 0$, then the $P_{\lim}$ system has one or more waiting customers, which implies that all of the servers in that system are busy, or $B^P_{\lim}(t') = B^R_{\lim}(t') = N$. Therefore, an arrival to either $P_{\lim}$ or $R_{\lim}$ will not change the number of busy servers. A departure from $P_{\lim}$ will not change $B_{\lim}$ (because there is at least one waiting customer in that system) and a departure from $R_{\lim}$ will either leave $B^R_{\lim}$ unchanged or reduce it by one, depending on whether the server that completes service has a waiting customer or not. Thus, the desired ordering of $B^P_{\lim}$ and $B^R_{\lim}$ is maintained regardless of what the subsequent event is.

If $Q^P_{\lim}(t') = 0$, then it follows that $Q^R_{\lim}(t') \geq 0 = Q^P_{\lim}(t')$, which implies that $P_{\lim}$ has at least as many customers in external delay $(NM - b')$ as $R_{\lim}$ $(NM - b' - Q^R_{\lim}(t'))$. We have the following distributions for the time until the next event after t' of each type:

$$\text{Next arrival to } P_{\lim} \text{ after } t'\text{: } a^P(t') \sim \exp\left\{(NM - b')\lambda\right\} \tag{5.13}$$

$$\text{Next arrival to } R_{\lim} \text{ after } t'\text{: } a^R(t') \sim \exp\left\{[NM - b' - Q^R_{\lim}(t')]\lambda\right\} \tag{5.14}$$

$$\text{Next departure from } P_{\lim} \text{ after } t'\text{: } d^P(t') \sim \exp\left\{b'(1-\gamma)\mu\right\} \tag{5.15}$$

$$\text{Next departure from } R_{\lim} \text{ after } t'\text{: } d^R(t') \sim \exp\left\{b'(1-\gamma)\mu\right\} \tag{5.16}$$

Note that immediately after t', customers arrive to the queue in $P_{\lim}$ at the same or a higher rate than they arrive to a queue in $R_{\lim}$. Therefore, we can couple $P_{\lim}$ and $R_{\lim}$ as follows. After t' we let $P_{\lim}$ run freely. If the next event after t' in $P_{\lim}$ is a departure, then we let a departure occur in $R_{\lim}$ with probability 1. If the next event after t' in $P_{\lim}$ is an arrival, then we let an arrival occur in $R_{\lim}$ with probability $p = (NM - b' - Q^R_{\lim}(t'))/(NM - b')$. This construction ensures the

proper distributions for $d^R(t')$ and $a^R(t')$ and keeps the sample path of the number of busy servers in P_{lim} at or above the sample path of the number of busy servers in R_{lim} with probability 1 at all times. Therefore, $B^P_{\text{lim}} \geq_{st} B^R_{\text{lim}}$, which implies that $E[B^P_{\text{lim}}] \geq E[B^R_{\text{lim}}]$ (Ross 1996, Lemma 9.1.1). □

5.6 The Balanced System Approximation

In the B system, we make three assumptions that allow us to model the case manager system as a birth-death process:

1. **Balanced caseloads:** We assume that cases are transferred between case managers to ensure that the caseloads m_i and m_j of any two case managers i and j are equal, if possible, and otherwise differ by at most one case. Appendix B.4 describes a case transfer mechanism that achieves this objective.

2. **Markovian case completion rates:** We assume that if a case manager has a caseload m at time t, then she will complete a case in $(t, t + dt]$ with probability $\phi(m)dt + o(dt)$, where $\lim_{dt\to 0} o(dt)/dt = 0$, independent of all other case managers.

3. **Stationary finite-source case completion rates:** We assume that the case completion rate $\phi(m)$ of a case manager with caseload m equals the steady-state case completion rate in system $B_{SS}(m)$: A single-server finite-source Markovian queueing system with m customers $(M/M/1/./m)$, with service rate $(1 - \gamma)\mu$ (the rate at which cases cycle back) and average time until arrival $1/\lambda$ for any customer in the population—identical to the limiting system R_{lim} that we used in the stability analysis for the R system, except for the population size. We also assume that the expected internal wait, given a caseload of m, can be computed using the same $M/M/1/./m$ system.

It follows from these assumptions that the total number of customers in the system, i, evolves as a Markovian birth-death process. The birth rate b_i in any state i is the rate Λ of new case arrivals. In order to obtain the death rates, we decompose the

total number of customers in the system as

$$i = n(i) + (N - u(i))m_{\min}(i) + u(i)(m_{\min}(i) + 1), \tag{5.17}$$

where $n(i) = (i - NM)^+$ is the length of the pre-assignment queue, $m_{\min}(i) = (i - n(i) - u(i))/N$ is the minimum caseload of any manager, and $u(i) = (i - n(i)) \mod N$ is the number of managers with $m_{\min}(i)+1$ cases. That is, $N-u(i)$ managers have a caseload of $m_{\min}(i)$ and the remaining $u(i)$ managers have a caseload of $m_{\min}(i) + 1$. Given Assumption 2, it follows that the death rate d_i in state i equals

$$d_i = (N - u(i))\phi(m_{\min}(i)) + u(i)\phi(m_{\min}(i) + 1),\ i = 1, 2, \ldots \tag{5.18}$$

The death rate saturates at $d_i = N\phi(M)$ for $i > MN$, which implies that the birth-death process has a geometrically-decaying tail, and the B system is stable if $\Lambda < N\phi(M)$.

To compute $\phi(m)$, let $\omega_0(m)$ be the steady-state probability that the server is idle in system $B_{SS}(m)$. Then the steady-state server case completion rate equals $\phi(m) = \mu\gamma(1-\omega_0(m))$—the case completion rate while the server is busy, times the server utilization. With this expression for $\phi(m)$ and the fact that $\omega_0(M) = \omega_0^R$, we see that the stability limit for the B system is the same as for the R system.

Define $r^B = \Lambda/(N\phi(M))$. If the system is stable, that is, if $r^B < 1$, then standard birth-death process calculations reveal that the steady-state probability of states 0 and NM, p_0 and p_{NM}, and the average pre-assignment queue length, L_a^B, can be calculated as

$$p_0 = \left(1 + \sum_{i=1}^{NM-1} \frac{\Lambda^i}{\prod_{j=1}^{i} d_j} + \frac{\Lambda^{NM}}{\prod_{i=1}^{NM} d_i} \frac{1}{1 - r^B}\right)^{-1}, \tag{5.19}$$

$$p_{NM} = \frac{\Lambda^{NM}}{\prod_{i=1}^{NM} d_i} p_0, \tag{5.20}$$

$$L_a^B = p_{NM} \frac{r^B}{(1 - r^B)^2}. \tag{5.21}$$

Using Little's Law, the average pre-assignment wait is $W_a^B = L_a^B/\Lambda$.

To approximate the expected wait in the internal queues, we first calculate the expected queue length $L_{SS}(m)$ in the finite-source system $B_{SS}(m)$ for $m = 1, \ldots, M$. The overall expected number in internal queues is:

$$L_q^B = \sum_{i=1}^{\infty} p_i \left\{ (N - u(i)) L_{SS}(m(i)) + u(i) L_{SS}(m(i)+1) \right\} \tag{5.22}$$

$$= p_0 \left\{ \sum_{i=1}^{MN-1} \left(\frac{\Lambda^i}{\prod_{j=1}^{i} d_j} \right) \left[(N - u(i)) L_{SS}(m(i)) + u(i) L_{SS}(m(i)+1) \right] \right\} \tag{5.23}$$

$$+ p_{NM} N \frac{L_{SS}(M)}{1 - r^B} \tag{5.24}$$

By Little's Law, the expected internal wait is $W_q^B = L_q^B / \Lambda$.

In Section 5.8 we will test the accuracy of this approximation as well as its ability to determine optimal caseload limits. Note that by adjusting $\phi(m)$, the model can be extended to include case manager service rates that vary with the caseload, as well as reneging or balking from the queues.

5.7 Deterministic Approach for Setting Caseload Limits

Yamatani et al. (2009) propose a simple method for setting caseload limits: Divide the time, χ, that a case manager is available per month by the time per month that each case requires. We reinterpret this advice in the context of our model. The amount of time each case requires per month from the case manager is χ multiplied by the proportion of time that a case requires from its case manager while assigned, that is, $\chi \times [(1/\mu)/(1/\mu + 1/\lambda)]$. The recommended caseload limit is therefore:

$$M^D = \frac{\chi}{\chi(1/\mu)/(1/\mu + 1/\lambda)} = \frac{1/\mu + 1/\lambda}{1/\mu}. \tag{5.25}$$

This approach implicitly assumes (i) that there is no variability in the system and (ii) that the case manager is always working on the maximum possible caseload. In Section 5.8.4 we will compare this method with other approaches we propose.

5.8 Calibrating and Using the Models

In this Section, we solve the R, S, B, and P models for several problem instances, to generate insights and to illustrate how the models can be used in practice. We programmed the QBD calculations for the R and P models and the birth-death process calculations for the B model in Matlab. The computation time per instance was less than a second for each of the R and P models and negligible for the B system. We simulated the S system using the Arena simulation software. For each instance, we simulated 100 replications, each of which had a 500-hour warmup period, followed by 2,000 simulated hours. These simulations required roughly 12 minutes of computation time per instance.

We begin, in Section 5.8.1, by estimating base-case parameters for the models, using published data for an Emergency Department (ED). In Section 5.8.2, we explore how the system behavior changes as we vary the base-case parameters, one at a time. In Section 5.8.3, we discuss situations in which the S system behavior approaches that of the R or P systems. In Section 5.8.4, we compare methods for setting maximum caseloads.

5.8.1 Calibrating a Base Case from Partial Information

In practice, administrative data and observational studies for case manager systems may not capture sufficient information for direct estimation of all system parameters (M, N, Λ, λ, μ, and γ). For example, in an ED, administrative data might track a patient's total length of stay (LOS) and the times of consultations with physicians but might not include information about when a patient's external delay (a diagnostic imaging test, for example) ends and internal delay (waiting for a consultation with the assigned physician) begins. In this section, we illustrate how one might address these potential difficulties.

We use information from a time study of emergency physician workload by Graff et al. (1993). We view physicians as case managers. Graff et al. (1993) studied how physician service time varies with patient service category, length of stay,

and intensity of service. The physicians in their study (from a university-affiliated community teaching hospital) recorded the beginning and ending times of each interaction with a patient, as well as the LOS—the time between patient registration in the ED and patient release.

Table 5.2 lists statistics from Graff et al. for five patient types. The aggregate patient averages in Table 5.2 permit direct estimation of the average number of processing steps and the average service time per processing step, as follows:

$$\text{Average number of processing steps} = \frac{1}{\gamma} = 1.86 \;\Rightarrow \gamma = 0.54 \tag{5.26}$$

$$\text{Average physician service time} = \frac{1}{\mu} = \frac{\text{total service time}}{\text{average number of steps}} = \frac{0.32 \text{ hrs.}}{1.86} \tag{5.27}$$

$$= 0.17 \text{ hrs.} = 10.3 \text{ minutes} \;\Rightarrow \mu = 5.91/\text{hr.}$$

Table 5.2: Data from Graff et al. (1993). All times are in hours.

Patient type	Number	Avg. service time (T_s)	Avg. # of steps ($1/\gamma$)	γ	LOS (T)	Avg. # of ext. delays (N_e)	$T - T_s$
Nonselected	514	0.40	2.20	0.45	2.17	1.20	1.76
Walk-in	637	0.16	1.30	0.77	0.98	0.30	0.82
Obs.	52	0.93	6.30	0.16	12.41	5.30	11.48
Lac. repair	102	0.42	1.10	0.91	1.60	0.10	1.18
Critical	42	0.53	2.60	0.38	2.92	1.60	2.39
Total	1347						
Wtd. avg.		0.32	1.86	0.54	1.98	0.86	1.67

The data do not allow direct estimation of the external arrival rate (Λ) and the average external delay ($1/\lambda$). We can use the S model, however, to determine values for (λ, Λ) that are consistent with the 1.98-hour average total LOS from Graff et al. We decompose the total LOS as follows:

$$\text{Total LOS} = \text{Pre-assignment delay} + \text{internal delay} + \text{service time} \tag{5.28}$$
$$+ \text{ external delay} = 1.98 \text{ hours.}$$

After substituting direct estimates for the average total LOS and the average service

time, we are left with

$$\text{Pre-assignment delay + internal delay + external delay} = \tag{5.29}$$
$$W_a(\Lambda, \lambda) + W_q(\Lambda, \lambda) + T_e(\Lambda, \lambda) = 1.67 \text{ hours.}$$

We can use the S model to identify (λ, Λ) pairs that satisfy (5.29) and are, therefore, consistent with the data in Graff et al. (1993), but first we must set base-case values for N and M. We assume $N = 3$ physicians (typical for a small to medium-sized ED) with a maximum caseload of $M = 5$ patients (based on the empirical study by Kc (2013), which found that when caseloads climb above 5, physician performance declined significantly).

After fixing N, M, μ, and γ, we first varied λ and computed the stability limits for the R and P systems, as shown in Figure 5.9. Then we simulated the S system for several (λ, Λ) pairs that fell within the R system stability region. Figure 5.9 shows several such pairs that satisfy (5.29), up to simulation error. These pairs form an approximate contour along which (5.29) is satisfied, and we see that this contour lies entirely within the R system stability region. The complete set of values corresponding to the (λ, Λ) pair that we chose for our base case are $\Lambda = 8.6$/hour, $\lambda = 1.8$/hour, $\mu = 5.91$/hour, $\gamma = 0.54$, $M = 5$, and $N = 3$. With the S model, these values result in a physician utilization of 90%, average pre-assignment wait of 0.6 hours, average internal wait of 0.62 hours, and average external delay of 0.47 hours—values that appear plausible for an ED.

5.8.2 Variations from the Base Case

In Figure 5.10 we allow Λ to approach the R system stability limit ($\Lambda/\Lambda^R_{\text{lim}}$ approaches 1), where $\Lambda^R_{\text{lim}} = 9.44$ and $\Lambda^P_{\text{lim}} = 9.57$ per hour. Recall our Conjecture 5.4, that $\Lambda^S_{\text{lim}} = \Lambda^R_{\text{lim}}$, which justifies the use of $\Lambda/\Lambda^R_{\text{lim}}$ as a measure of congestion for the S system. The pre-assignment wait grows quickly while the internal wait increases more slowly. The pre-assignment queue in a case manager system is analogous to an infinite-capacity multi-server queue, and it's length grows without bound as the ar-

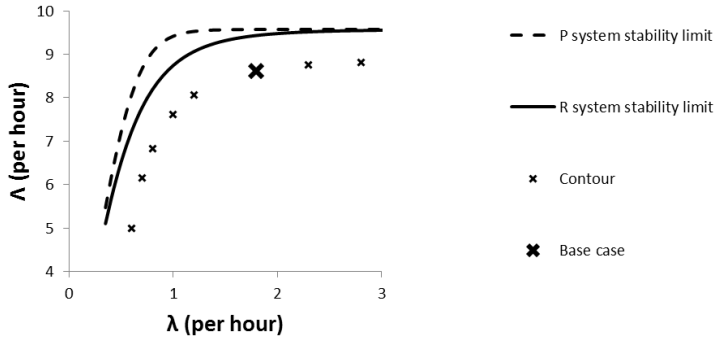

Figure 5.9: Contour of cases satisfying (5.29) along with the stability limits

.

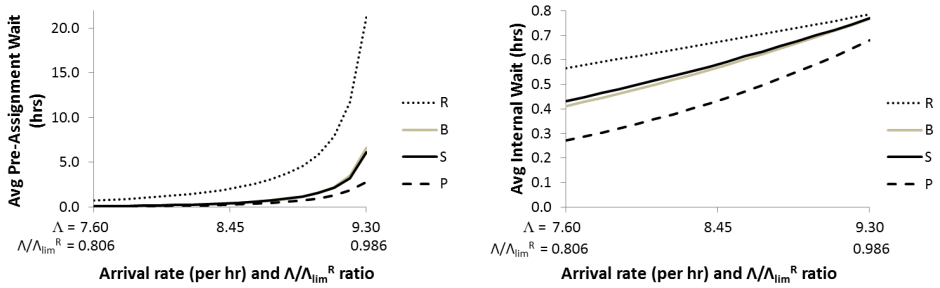

Figure 5.10: Average waits for the R, S, B, and P systems when the new case arrival rate Λ varies from 7.6 to 9.3 per hour.

rival rate approaches the system capacity. When $\Lambda = 9.3$ (99% of Λ^R_{lim}), the pooling benefits of the P system reduce the average pre-assignment delay eightfold compared to the R system (from 21.23 to 2.81 hours). The state-dependent routing in the S system achieves most of this benefit, with a 6.12-hour average pre-assignment delay, while maintaining the benefits of continuity of care. In these experiments, as in most of the experiments that we discuss in this subsection, the B system results are almost identical to the S system simulation results.

The ratio $\Lambda/\Lambda^R_{\text{lim}}$ can also be varied by changing μ, λ, or γ (see equations (5.10) and (5.11)). In Figures 5.11 and 5.12, we see that varying λ or γ has mostly the same qualitative effect as varying Λ, as does the effect of varying μ (not shown). The exception is the effect of changes in $1/\lambda$, the average external delay, on internal wait, as seen in the right panel of Figure 5.11. On the one hand, increasing $1/\lambda$ decreases effective capacity, thereby increasing $\Lambda/\Lambda^R_{\text{lim}}$ and the pre-assignment delay

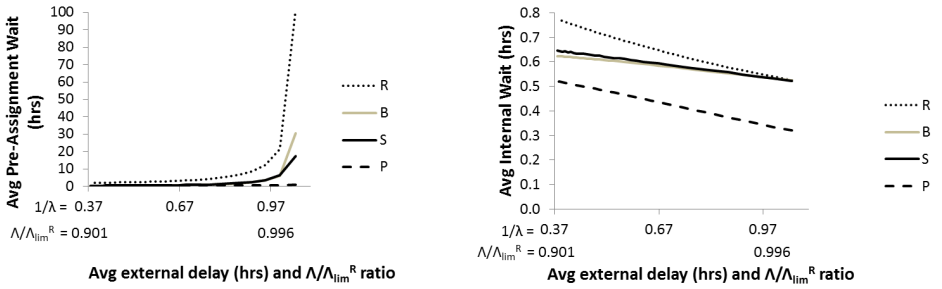

Figure 5.11: Average waits for the R, S, B, and P systems when the average external delay $1/\lambda$ varies from 0.37 to 0.97 hours.

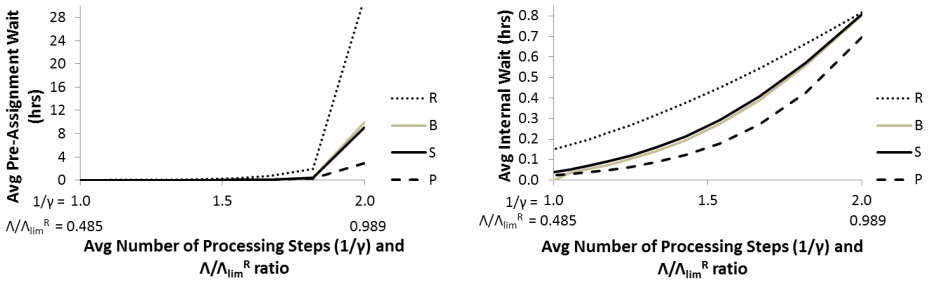

Figure 5.12: Average waits for the R, S, B, and P systems when the average number of processing steps $1/\gamma$ varies from 1 to 2.

(Figure 5.11, left panel). On the other hand, in heavily loaded systems where the case managers operate close to their caseload limit, a longer average external delay results in a shorter average internal wait, because the total number of cases in external delay and the internal queue is almost constant (Figure 5.11, right panel). The effect of varying μ is similar to the effect of varying γ.

5.8.3 When Does the S System Approach the P or R System?

In all of our experiments, the S-system pre-assignment delay is closer to the P-system pre-assignment delay than the R-system pre-assignment delay, again demonstrating that the S system provides most of the benefits of pooling. This was also true for the total wait, because the total wait is dominated by the pre-assignment wait.

For the internal wait, however, as Λ, $1/\lambda$ and $1/\gamma$ increase so that $\Lambda/\Lambda_{\text{lim}}^R$ approaches 1, the S system's performance approaches that of the R system (see the right panels of Figures 5.10-5.12). As $\Lambda/\Lambda_{\text{lim}}^R$ approaches 1, both the R and S sys-

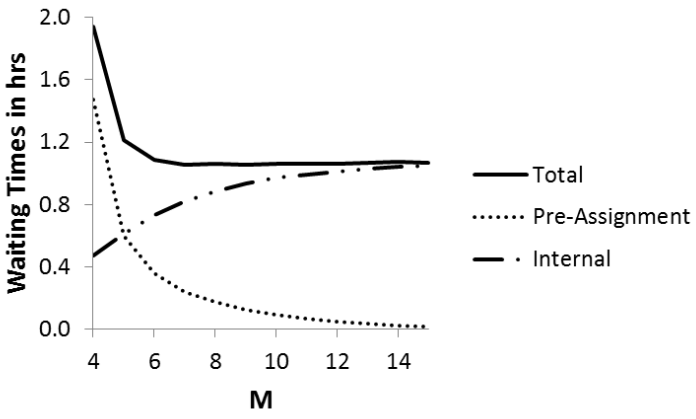

Figure 5.13: Average total, internal, and pre-assignment waits for the S system, varying the caseload limit from $M = 4$ to 15. The deterministic caseload limit for the base case is $M^D = 4$.

tems become heavily loaded, with most new cases waiting in the pre-assignment queue and then being routed to the first available case manager, thus removing the benefits of state-dependent routing.

5.8.4 Setting Caseloads

Varying the caseload limit M adjusts the tradeoff between pre-assignment delay and internal delay. On the one hand, with a higher M, the case manager is more likely to be busy, so that the internal delay increases. On the other hand, the case manager's increased utilization increases the system capacity, which decreases the pre-assignment delay. Figure 5.13 illustrates this tradeoff and shows that the impact of changes in M on pre-assignment delay tend to dominate the impact on internal delay, so that the total delay declines as M rises. This was true for all of our numerical experiments. Therefore, we define W^{∞} as the average total wait when there is no caseload limit ($M = \infty$) and we hypothesize that this is the minimum possible average total wait in an S system.

From the literature on multitasking, however, we know that increased caseloads can have a negative impact on service quality (Kc 2013). Therefore it would be useful to identify reasonable caseload limits that reduce the impact of multitasking while keeping the average total wait below a target.

We ran simulation experiments to identify $M^S_{10\%}$, defined as the smallest caseload limit such that the average total wait in the S system is at most 10% above the minimum, W^∞. Let M^P_{lim} and M^R_{lim} be the smallest caseload limits for which a pooled system and a random routing system are stable, respectively. To find $M^S_{10\%}$, we simulate the S system with $M = M^P_{\text{lim}}$ and then increment M by one case at a time until $W^S/W^\infty \leq 1.1$. We use a similar procedure to identify $M^B_{10\%}$, the smallest caseload limit that brings the average total waiting time in the B system below $1.1W^\infty$. We also compute the deterministic caseload limit M^D, using (5.25).

We ran two series of experiments: Series A, with lightly loaded systems and low recommended caseload limits and Series B, with heavily loaded systems and high recommended caseload limits. We controlled the system load via the ratio $\Lambda/(N\gamma\mu)$, which corresponds to the case manager utilization for a system with $M = \infty$. The experiments covered a wide range of parameter values that might be seen in health care settings, for example, $1/\lambda$ varied from 23 minutes to 1 hour in Series A and from 2 to 4 hours in Series B. The parameter sets were primarily constructed using a full factorial design, but with unstable systems eliminated and a few experiments added to widen the range of recommended caseloads. Appendix B.3 lists all parameter settings for Series A and B.

Table 5.3 and Figure 5.14 summarize the results of the experiments. The fourth and fifth lines of Table 5.3 and the clustering of the B-system caseload limit recommendations on the diagonal in Figure 5.14 show that $M^B_{10\%}$ provides us with an accurate method for setting caseload limits. The balanced model caseload limits usually match the exact $M^S_{10\%}$ (75% of cases in Series A and 88% of cases in Series B) and they differ from $M^S_{10\%}$ by at most 1 in all cases. The deterministic approach, on the other hand, is a poor approximation. The deterministic caseload limit M^D matches $M^S_{10\%}$ in only 10% of the Series A cases and 4% of the Series B cases and M^D is often an overestimate, by up to 10 cases. Figure 5.14 also shows that $M^P_{10\%}$ often significantly underestimates the recommended caseload limit.

The B system is less successful at providing precise performance measure estimates, given the recommended caseload. From Table 5.3, the B-system average

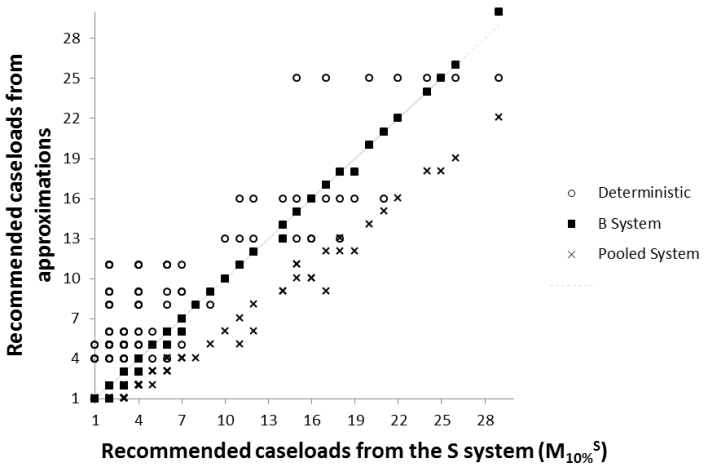

Figure 5.14: Recommended caseloads from the S simulation ($M_{10\%}^S$) versus caseload limits from the deterministic model (M^D), the balanced model ($M_{10\%}^B$), and the stability limit of the pooled model (M_{lim}^P)

and maximum absolute errors for total wait, compared to the S-system simulation, were 9% and 34% in Series A, respectively. The performance of the approximation was much better in Series B (1%, 6%). Note, however, that in Series A the absolute waiting times were extremely small, so that the absolute total waiting time error produced by the B system was also small, averaging 0.9 minutes.

Table 5.3: Summary of numerical experiments.

	Series A	Series B
Number of cases	81	24
Average for M_{lim}^P	1.8	11.8
Average for $\Lambda/(N\gamma\mu)$	0.56	0.92
% cases $M_{10\%}^B = M_{10\%}^S$	75%	88%
Max $\lvert M_{10\%}^B - M_{10\%}^S \rvert$	1	1
Avg. abs % system time error by B, given $M_{10\%}^B$	2%	0.4%
Max. abs. % system time error by B, given $M_{10\%}^B$	7%	3%
Avg. % waiting time error by B, given $M_{10\%}^B$	9%	1%
Max. abs. % waiting time error by B, given $M_{10\%}^B$	34%	6%
% cases $M^D = M_{10\%}^S$	10%	4%
Max $\lvert M^D - M_{10\%}^S \rvert$	9	10

CHAPTER 6

Conclusion

In this dissertation we explored capacity planning in two different service system settings: traditional multiserver systems and case manager systems. We were particularly concerned with systems where servers and customers are humans, although the results can be applied more generally. This lead us to take into account specific end-of-shift policies and consider the cognitive load of multitasking in our discussion.

For traditional nonstationary multiserver systems, we obtained an exact lower bound on the staffing needed to ensure the desired QoS at all times, with time-varying arrival processes and time-varying number of servers. In Chapter 2 we constructed this lower bound based on stochastic ordering results we proved between the virtual waiting time in the system of interest and the pseudo virtual waiting time in an otherwise identical infinite-server system. We showed how these lower bounds can be constructed for systems with both preemptive and exhaustive end-of-shift policy and for systems with abandonment. We also showed that the stochastic ordering between the virtual waiting time and the pseudo virtual waiting time can be used to construct lower bounds on staffing requirements if the expected waiting time or the average QoS in each period are used as performance measures (instead of the QoS).

In Chapter 3 we discussed how to evaluate time-dependent state probabilities and service levels (based on the pseudo virtual waiting time defined in Chapter 2) for

different types of nonhomogeneous infinite-server systems. When computing service levels for infinite-server systems with Poisson arrivals, where the state probabilities follow a Poisson distribution, we can choose a truncation limit for the summation in (3.16) based on the Poisson distribution that keeps the error within a specified threshold, as suggested by Grassmann (1977) and discussed in Chapter 3. A future research topic would be to recommend such truncation limits for systems where state probabilities do not follow a Poisson distribution.

In Chapter 4 we reported results from computational experiments to compare our lower bound with the SIPP approximation (commonly used in practice) and with the requirements from an approach to choosing staffing levels which is also based on an infinite-server system, the MOL approximation. The test cases we used correspond to situations where the SIPP approximation has been shown to perform poorly, and we confirm this in our results. The MOL staffing requirements ensured the target QoS at all times in all test cases, but they had high costs. Our lower bounds only ensured the QoS target at all times in one test case (which was expected, since they were not meant to be sufficient), but the minimum QoS was very close to the target in all test cases (average minimum QoS of 77% for a target of 80%). Thus, we showed that our lower bound can perform very close to the target despite having significantly lower costs than the MOL requirements (our lower bounds were on average 2.9% cheaper). We also showed that our lower bound can be used as a starting point for the ICWC method (a staff scheduling optimization method previously proposed by Ingolfsson et al. (2010)) to obtain staffing requirements which guarantee the desired QoS at all times, but are cheaper than the MOL requirements in all test cases (on average 1.8% cheaper). Note that the ICWC method found feasible staffing requirements very quickly when our lower bound was used as a starting point, making this method competitive with approximate approaches such as MOL and SIPP. Furthermore, we showed that our lower bound can be used to considerably increase the speed with which the scheduling method in Ingolfsson et al. (2010) finds a low cost feasible solution to the scheduling problem. The method is on average 86.7% faster when our lower bound is used. We would

expect to obtain time savings in other methods that start the search for low cost staffing solutions from a lower bound.

The lower bound we proposed is easy to compute and can be very useful as a starting point for finding low-cost staffing schedules which satisfy the QoS target. Furthermore, in situations where it is not essential that the QoS target be met at all times, our lower bound can be used in an approximate approach to find staffing schedules that provide QoS close to the target.

In Chapter 5 we developed a stochastic model of a case manager system. Exact analysis of this baseline Markov chain model, which has two state variables for every case manager, is difficult because of the curse of dimensionality. This motivated us to formulate two simpler-to-analyze models, which we believe provide lower and upper performance bounds, as well as a birth-death process approximation. We provided expressions to determine stability limits for the bounding models, which can help in planning simulation experiments for the baseline model.

Analysis and numerical experiments with these systems generated insights that may be used to design and operate case manager systems. We showed that for special cases, the stability limit of the baseline S system is equal to that of the R system with independent case managers. The average performance of the S system in terms of overall delay, however, is consistently closer to that of the P system, with entirely pooled case managers.

We also found that as the arrival rate, average number of processing steps, and average service time rise, both pre-assignment and internal delay rise. As the average external delay rises, pre-assignment delay also rises but internal delay falls. The effects of all these parameters on pre-assignment delay can be dramatic, exhibiting typical queueing congestion behavior as the system approaches the stability limit. Internal delay, however, varies inside a limited range.

Experiments with caseload limits demonstrated that managers may trade-off pre-assignment and internal delay. The optimal caseload limit will depend upon the relative costs of these delays, as well as upon other costs not modeled directly here, such as the impact of caseloads on service quality (Kc 2013). In our computational

experiments, we used our models to find the minimum caseload that satisfies a delay criterion. We found that the birth-death process approximation provided caseload limits that differ by at most one case from caseload limits obtained by simulating the baseline model. A deterministic caseload limit calculation, proposed in the social work literature, performs poorly. This calculation ignores the impact of system parameters (such as the external delay) and may recommend caseload limits that are either unreasonably high or are so low that the system is unstable. Finally, another advantage of the birth-death approximation is that it is easily adapted to incorporate particular relationships between the manager's caseload and the case completion rate, as documented in Kc (2013).

BIBLIOGRAPHY

BIBLIOGRAPHY

Apte, U. M., C.M. Beath, C. Goh. 1999. An analysis of the production line versus the case manager approach to information intensive services. *Decision Sciences* **30**(4) 1105–1129.

Atlason, J., M. A. Epelman, S. G. Henderson. 2004. Call center staffing with simulation and cutting plane methods. *Annals of Operations Research* **127** 333–358.

Atlason, J., M. A. Epelman, S. G. Henderson. 2008. Optimizing call center staffing using simulation and analytic center cutting-plane methods. *Management Science* **54**(2) 295–309.

Bhaskaran, B. G. 1986. Almost sure comparison of birth and death processes with application to $M/M/s$ queueing systems. *Queueing Systems* **1** 103–127.

Borst, S., A. Mandelbaum, M. I. Reiman. 2004. Dimensioning large call centers. *Operations Research* **52**(1) 17–34.

Cezik, M. T., P. L'Ecuyer. 2008. Staffing multiskill call centers via linear programming and simulation. *Management Science* **54**(2) 310–323.

CWLA. 1999. *CWLA Standards of Excellence for Services for Abused or Neglected Children and Their Families*. Washington DC.

CWLA. 2013. Recommended caseload standards. `http://www.cwla.org/newsevents/news030304cwlacaseload.htm`.

Daley, D. J., P. A. P. Moran. 1968. Two-sided inequalities for waiting time and queue size distributions in $GI/G/1$. *Theory of Probability and its Applications* **13**(2) 338–341.

de Véricourt, F., O. B. Jennings. 2011. Nurse staffing in medical units: A queueing perspective. *Operations Research* **59**(6) 1320–1331.

de Véricourt, F., Y-P Zhou. 2005. Managing response time in a call-routing problem with service failure. *Operations Research* **53**(6) 968–981.

Dobson, Gregory, Tolga Tezcan, Vera Tilson. 2013. Optimal workflow decisions for investigators in systems with interruptions. *Management Science* doi:10.1287/mnsc.1120.1632. URL `http://mansci.journal.informs.org/content/early/2013/01/08/mnsc.1120.1632.abstract`.

Eick, S. G., W. A. Massey, W. Whitt. 1993a. $M_t/G/\infty$ queues with sinusoidal arrival rates. *Management Science* **39**(2) 241–252.

Eick, S. G., W. A. Massey, W. Whitt. 1993b. The physics of the $M_t/G/\infty$ queue. *Operations Research* **41**(4) 731–742.

Feldman, Z., A. Mandelbaum, W.A. Massey, W. Whitt. 2008. Staffing of time-varying queues to achieve time-stable performance. *Management Science* **54**(2) 324–338.

Gans, N., G. Koole, A. Mandelbaum. 2003. Telephone call centers: Tutorial, review, and research prospects. *Manufacturing & Service Operations Management* **5**(2) 79–141.

Graff, L.G., S. Wolf, R. Dinwoodie, D. Buono, D. Mucci. 1993. Emergency physician workload: A time study. *Annals of Emergency Medicine* **22**(7) 1156–1163.

Grassmann, W. 1977. Transient solutions in Markovian queuing systems. *Computers & Operations Research* **4** 47–53.

Green, L. V., P. J. Kolesar, J. Soares. 2003. An improved heuristic for staffing telephone call centers with limited operating hours. *Production and Operations Management* **12**(1) 46–61.

Green, L. V., S. Savin. 2008. Reducing delays for medical appointments: A queueing approach. *Operations Research* **56**(6) 1526–1538.

Green, L. V., J. Soares, J. F. Giglio, R. A. Green. 2006. Using queueing theory to increase the effectiveness of emergency department provider staffing. *Academic Emergency Medicine* **13** 61–68.

Green, L.V., P. Kolesar. 1991. The pointwise stationary approximation for queues with nonstationary arrivals. *Management Science* **37**(1) 84–97.

Green, L.V., P.J. Kolesar, J. Soares. 2001. Improving the SIPP approach for staffing service systems that have cyclic demand. *Operations Research* **49** 549–556.

Green, L.V., P.J. Kolesar, W. Whitt. 2007. Coping with time-varying demand when setting staffing requirements for a service system. *Production and Operations Management* **16**(1) 13–39.

Gun, L. 1989. Experimental results on matrix-analytical solution techniques—extensions and comparisons. *Stochastic Models* **5**(4) 669–682.

Halfin, S., W. Whitt. 1981. Heavy-traffic limits for queues with many exponential servers. *Operations Research* **29**(3) 567–588.

Holmström, K. 1999. The TOMLAB optimization environment in Matlab. *Advanced Modeling and Optimization* **1** 47–69.

Ingolfsson, A. 2005. Modeling the $M(t)/M/s(t)$ queue with an exhaustive discipline. Working paper, Department of Finance and Management Science, School of Business, University of Alberta, Edmonton, Alberta, Canada, `http://apps.business.ualberta.ca/aingolfsson/documents/PDF/MMs_note.pdf`.

Ingolfsson, A., E. Akhmetshina, S. Budge, Y. Li, X. Xu. 2007. A survey and experimental comparison of service-level-approximation methods for nonstationary $M(t)/M/s(t)$ queueing systems with exhaustive discipline. *INFORMS Journal on Computing* **19**(2) 201–214.

Ingolfsson, A., F. Campello, X. Wu, E. Cabral. 2010. Combining integer programming and the randomization method to schedule employees. *European Journal of Operations Research* **202** 153–163.

Ingolfsson, A., F. Gallop. 2003. Queueing toolpak (QTP) version 4.0. `http://apps.business.ualberta.ca/aingolfsson/qtp/`.

Ingolfsson, A., T. A. Grossman. 2002. Graphical spreadsheet simulation of queues. *INFORMS Transactions on Education* **2**(2) 27–39. `http://archive.ite.journal.informs.org/Vol2No2/IngolfssonGrossman/`.

Jackson, J. R. 1957. Networks of waiting lines. *Operations Research* **5**(4) 518–521.

Jacobs, D. R., S. Schach. 1972. Stochastic order relationships between $GI/G/k$ systems. *The Annals of Mathematical Statistics* **43**(5) 1623–1633.

Jennings, O. B., A. Mandelbaum, W. A. Massey, W. Whitt. 1996. Server staffing to meet time-varying demand. *Management Science* **42**(10) 1383–1394.

Kc, D.S. 2013. Does multitasking improve performance? Evidence from the emergency department. Available at SSRN: `http://ssrn.com/abstract=2261757` or `http://dx.doi.org/10.2139/ssrn.2261757`.

Kleinrock, L. 1974a. *Queueing Systems, Volume 2: Computer Applications*. Wiley, New York.

Kleinrock, L. 1974b. *Queueing Systems, Volume 2: Theory*. Wiley, New York.

Klenke, A., L. Mattner. 2010. Stochastic ordering of classical discrete distributions. *Advances in Applied Probability* **42**(2) 392–410.

Koole, G. 2005. Redefining the service level in call centers. Department of Mathematics, Vrije Universiteit Amsterdam, The Netherlands, `http://www.math.vu.nl/~koole/articles/report03b/art.pdf`.

Kuncir, G. F. 1962. Algorithm 103: Simpson's rule integrator. *Communications of the ACM* **5**(6) 347.

Latouche, G, V Ramaswami. 1999. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. Society for Industrial and Applied Mathematics, Philadelphia PA. doi:10.1137/1.9780898719734. URL `http://epubs.siam.org/doi/abs/10.1137/1.9780898719734`.

Lin, H-C, C.S. Raghavendra. 1996. An approximate analysis of the join the shortest queue (JSQ) policy. *IEEE Transactions on Parallel and Distributed Systems* **7**(3) 301–307.

Luo, Jun, Jiheng Zhang. 2013. Staffing and control of instant messaging contact centers. *Operations Research* **61**(2) 328–343.

Mandelbaum, A., W. Massey, M. Reiman, A. Stolyar, B. Rider. 2002. Queue lengths and waiting times for multiserver queues with abandonment and retrials. *Telecommunication Systems* **21**(2-4) 149–171.

Massey, W.A., W. Whitt. 1994. An analysis of the modified offered load approximation for the nonstationary Erlang loss model. *Annals of Applied Probability* **4** 1145–1160.

Muller, A., D. Stoyan. 2002. *Comparison methods for stochastic models and risks*. 3rd ed. Wiley, New York.

Nelson, B.L., M.R. Taaffe. 2004. The $Ph_t/Ph_t/\infty$ queueing system: Part I – the single node. *INFORMS Journal on Computing* **16**(3) 266–274.

Nelson, R.D., T.K. Philips. 1989. An approximation to the response time for shortest queue routing. *Performance Evaluation Review* **1**(1) 181–189.

Ross, S. 1996. *Stochastic Processes*. 2nd ed. Wiley, New York.

Rothkopf, M. H., S. S. Oren. 1979. A closure approximation for the nonstationary $m/m/s$ queue. *Management Science* **25**(6) 522–534.

Saghafian, S., W. J. Hopp, M. P. Van Oyen, J. S. Desmond, S. L. Kronick. 2012. Patient

streaming as a mechanism for improving responsiveness in emergency departments. *Operations Research* **60**(5) 1080–1097.

Saghafian, Soroush, Wallace Hopp, Mark Van Oyen, Jeffrey Desmond, Steven Kronick. 2011. Complexity-based triage: A tool for improving patient safety and operational efficiency. *Ross School of Business Paper* (1161).

Shampine, L. F., M. W. Reichelt. 1997. The MATLAB ODE suite. *SIAM Journal on Scientific Computing* **18**(1) 1–22.

Smith, D. R., W. Whitt. 1981. Resource sharing for efficiency in traffic systems. *Bell System Technical Journal* **60**(13) 39–55.

Sonderman, D. 1979a. Comparing multi-server queues with finite waiting rooms, I: Same number of servers. *Advances in Applied Probability* **11**(2) 439–447.

Sonderman, D. 1979b. Comparing multi-server queues with finite waiting rooms, II: Different number of servers. *Advances in Applied Probability* **11**(2) 448–455.

Tezcan, Tolga. 2011. Design and control of customer service chat systems. *Available at SSRN 1964434* .

Weber, R. R. 1978. On the optimal assignment of customers to parallel servers. *Journal of Applied Probability* **15**(2) 406–413.

Whitt, W. 1981. Comparing counting processes and queues. *Advances in Applied Probability* **13**(1) 207–220.

Whitt, W. 1986. Deciding which queue to join: Some counter examples. *Operations Research* **34**(1) 55–62.

Whitt, W. 1999. Predicting queueing delays. *Management Science* **45**(6) 870–888.

Yamatani, H., R. Engel, S. Spjeldnes. 2009. Child welfare worker caseload: What's just right? *Social Work* **54**(4) 361–368.

Yankovic, N., L. V. Green. 2011. Identifying good nursing levels: A queueing approach. *Operations Research* **59**(4) 942–955.

Yom-Tov, G. B. 2010. Queues in hospitals: Queueing networks with reentering customers in the QED regime. *PhD thesis, Technion - Israel Institute of Technology* .

Yom-Tov, G B, A Mandelbaum. 2011. Erlang-R: A time-varying queue with ReEntrant customers, in support of healthcare staffing. *Preprint* .

Zeltyn, S., B. Carmeli, O. Greenshpan, Y. Mesika, S. Wasserkrug, P. Vortman, Y. N. Marmor, A. Mandelbaum, A. Shtub, T. Lauterman, D. Schwartz, K. Moskovitch,

S. Tzafrir, F. Basis. 2010. Simulation-based models of emergency departments: Operational, tactical and strategic staffing. http://ie.technion.ac.il/serveng/References/ED_simulation_modeling_rev.pdf.

APPENDICES

APPENDIX A

Exact Necessary Staffing Requirements Based on Infinite-Server Systems

A.1 Alternative Lower Bound for the Exhaustive Discipline Case

We examine an alternative way of using an infinite-server system to obtain lower bounds on the number of servers and the virtual waiting time for a finite-server system in which the number of servers varies with time under an exhaustive discipline. In this case, given that δ_1 servers are scheduled to leave the finite-server system at time t_1, the state probabilities in the infinite-server system undergo instantaneous transitions according to the matrix B_I in (2.15).

Since we assume that $N_F(t_1^-) \geq N_I(t_1^-)$, there are three possibilities for the distribution of the number of customers in the two systems after the ejection:

1. If $N_F(t_1^-) \geq N_I(t_1^-) \geq s_0$,

$$P(N_F(t_1^+) = n) = \begin{cases} 1, & \text{for } n = N_F(t_1^-) - \delta_1 \\ 0, & \text{otherwise} \end{cases} \tag{A.1}$$

$$P(N_I(t_1^+) = n) = \begin{cases} 1, & \text{for } n = N_I(t_1^-) - \delta_1 \\ 0, & \text{otherwise} \end{cases} \tag{A.2}$$

2. If $N_F(t_1^-) \geq s_0 \geq N_I(t_1^-)$,

$$P(N_F(t_1^+) = n) = \begin{cases} 1, & \text{for } n = N_F(t_1^-) - \delta_1 \\ 0, & \text{otherwise} \end{cases} \tag{A.3}$$

$$P(N_I(t_1^+) = n) = \begin{cases} \phi(N_I(t_1^-) - n; \delta_1, s_0, N_I(t_1^-)), & \text{for } L_I \leq n \leq U_I \\ 0, & \text{otherwise} \end{cases} \tag{A.4}$$

where $L_I = [N_I(t_1^-) - \delta_1]^+$ and $U_I = \min(N_I(t_1^-), s_0 - \delta_1)$.

3. If $s_0 \geq N_F(t_1^-) \geq N_I(t_1^-)$,

$$P(N_F(t_1^+) = n) = \begin{cases} \phi(N_F(t_1^-) - n; \delta_1, s_0, N_F(t_1^-)), & \text{for } L_F \leq n \leq U_F \\ 0, & \text{otherwise} \end{cases} \tag{A.5}$$

where $L_F = [N_F(t_1^-) - \delta_1]^+$ and $U_F = \min(N_F(t_1^-), s_0 - \delta_1)$.

$$P(N_I(t_1^+) = n) = \begin{cases} \phi(N_I(t_1^-) - n; \delta_1, s_0, N_I(t_1^-)), & \text{for } L_I \leq n \leq U_I \\ 0, & \text{otherwise.} \end{cases} \tag{A.6}$$

In cases 1 and 2 it is straightforward to show that $P(N_F(t_1^+) > a) \geq P(N_I(t_1^+) > a)$, for all a, and therefore $N_F(t_1^+) \geq_{st} N_I(t_1^+)$. For case 3, the same ordering can be proven using the method proposed by Klenke and Mattner (2010) to prove the likelihood ratio ordering between $N_F(t_1^+)$ and $N_I(t_1^+)$. We say that $N_F(t_1^+)$ is greater than $N_I(t_1^+)$ in the monotone likelihood ratio order, $N_F(t_1^+) \geq_{lr} N_I(t_1^+)$, if the likelihood ratio $l(n) = P(N_I(t_1^+) = n)/P(N_F(t_1^+) = n)$ is nonincreasing in n. According to Remark 2.1 and Proposition 2.1 in Klenke and Mattner (2010), in order to prove that $N_F(t_1^+) \geq_{lr} N_I(t_1^+)$, it is enough to show that $l(n)$ is monotone decreasing in the interval $L_F \leq n \leq U_I$, $l(L_I) \geq 1$ (left-tail condition), and $l(U_F) \leq 1$ (right-tail condition). Since the likelihood ratio ordering implies the conventional stochastic ordering (Ross 1996), we conclude that $N_F(t_1^+) \geq_{st} N_I(t_1^+)$ in case 3. The argument can be repeated to show that $N_I(t) \leq_{st} N_F(t)$ for all t.

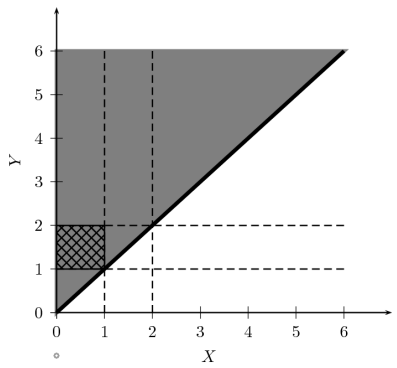

Figure A.1: Joint pdf for X and Y.

If the service times are exponentially distributed, then $N_I(t) \leq_{st} N_F(t)$ implies $W_I(t) \leq_{st} W_F(t)$ (based on a minor modification of Theorem 4 in Bhaskaran (1986)).

A.2 Sample Path versus Hazard Rate Ordering: Counter Example

Let X and Y be two random variables with joint probability density function (pdf):

$$f_{X,Y}(x,y) = \begin{cases} \frac{10}{27}, & \text{for } 1 \leq y < 2,\ 0 \leq x < 1 \\ \frac{1}{27}, & \text{for } 2 \leq y < 6,\ 0 \leq x < y \text{ or } 1 \leq y < 2,\ 1 \leq x < y, \\ & \text{or } 0 \leq y < 1,\ 0 \leq x < y \\ 0, & \text{otherwise} \end{cases} \tag{A.7}$$

This pdf is only different from 0 in the gray region in Figure A.1, where $0 \leq y \leq 6$ and $x \leq y$. Furthermore, the probability density is higher in the crosshatched area where $1 \leq y \leq 2$ and $0 \leq x \leq 1$.

It is easy to see that the sample path ordering of X and Y holds, as $\Pr\{X \leq Y\} = 1$, but we show that the hazard rate ordering does not hold in this case. The random variable Y is said to be larger than the random variable X in a hazard rate ordering sense if $\Pr\{Y > \tau + a | Y > \tau\} \geq \Pr\{X > \tau + a | X > \tau\}$ for all τ, a. Assume that $\tau = 1$ and $a = 1$. In this case we have:

$$\Pr\{X > \tau + a | X > \tau\} = \Pr\{X > 2 | X > 1\} = \frac{\Pr\{X > 2\}}{\Pr\{X > 1\}} = \frac{\frac{1}{27}\frac{(6-2)^2}{2}}{\frac{1}{27}\frac{(6-1)^2}{2}} = \frac{16}{25} = 0.64 \tag{A.8}$$

$$\Pr\{Y > \tau + a | Y > \tau\} = \frac{\Pr\{Y > 2\}}{\Pr\{Y > 1\}} = \frac{\frac{1}{27}\frac{(2+6)(6-2)}{2}}{\frac{1}{27}\frac{(2+6)(6-2)}{2} + \frac{1}{27}\frac{(2-1)^2}{2} + \frac{10}{27}(2-1)(1-0)} \tag{A.9}$$

$$= \frac{32}{53} \approx 0.60 \tag{A.10}$$

Since $\Pr\{Y > 2 | Y > 1\} < \Pr\{X > 2 | X > 1\}$ we conclude that although $\Pr\{X \leq Y\} = 1$, Y is not larger than X in a hazard rate ordering sense, and therefore the sample path ordering does not necessarily imply the hazard rate ordering.

A.3 Computation of MOL Requirements

We computed MOL requirements for each $M(t)/M/s(t)$ system in Chapter 4 as follows:

1. We computed the average number of customers (or number of busy servers) $m(t)$ in an otherwise identical $M(t)/M/\infty$ system (System I), using the following differential equation (Corollary 4 in Eick et al. (1993b)), where $\lambda(t) = \lambda\left[1 + \gamma \sin\left(\frac{\pi t}{4}\right)\right]$:

$$m'(t) = \lambda(t) - \mu m(t) \Rightarrow m'(t) = \lambda\left[1 + \gamma \sin\left(\frac{\pi t}{4}\right)\right] - \mu m(t), \quad 0 \leq t \leq 12 \tag{A.11}$$

   We solved (A.11) using the Runge-Kutta numerical integration method implemented in the ode45 function in Matlab (Shampine and Reichelt 1997), recording the value of $m(t)$ each calculation period of $\delta_{\text{calc}} = 5$ minutes.

2. For each planning period j, spanning time interval $[t_j^i, t_j^f]$ we found the minimum number of servers needed to ensure the desired QoS in a $M/M/s$ system

with arrival rate $\lambda = \mu \max\left\{m(t), t \in [t_j^i, t_j^f]\right\}$, using formulas for the stationary state probabilities in an $M/M/s$ system (Kleinrock 1974a) implemented in the Queueing Toolpak, Version 4.0 (Ingolfsson and Gallop 2003).

A.4 Computation of Our Lower Bounds

We computed our lower bound for each $M(t)/M/s(t)$ system in Chapter 4 as follows:

1. We computed the state probabilities for an otherwise identical $M(t)/M/\infty$ system (System I), using the randomization method (Grassmann 1977). We implemented the randomization method in the same fashion as Ingolfsson et al. (2010), approximating the continuous arrival rate $\lambda(t)$ by a piecewise constant function $\lambda(t) = \tilde{\lambda}_l = \int_{(l-1)\delta_{\text{calc}}}^{l\delta_{\text{calc}}} \lambda(s)ds/\delta_{\text{calc}}$ for $t \in ((l-1)\delta_{\text{calc}}, l\delta_{\text{calc}}]$, for calculation periods of $\delta_{\text{calc}} = 5$ minutes. For systems under a preemptive discipline, we used the state probabilities at the end of each calculation period as the initial state probabilities for the subsequent calculation period. For systems under an exhaustive discipline, we multiplied the state probabilities at the end of each calculation period by the appropriate ejection matrix (matrix H in (2.12) or B_I in (2.15)) and then used the resulting state probabilities as the initial state probabilities for the subsequent calculation period. In each calculation period we used an $M(t)/M/K/K$ system to approximate the infinite-server infinite-capacity system $M(t)/M/\infty$, with K chosen so that $\pi_K(t) < 10^{-4}$ for all t. Note that for systems under a preemptive discipline we could use (A.11) to compute the time-dependent average number of customers in the system and then compute time-dependent state probabilities, which follow a Poisson distribution (Eick et al. 1993a,b). We chose to use a method that works for both end-of-shift policies.

2. For each planning period j, spanning time interval $[t_j^i, t_j^f]$, we found the minimum number of servers s_j needed to ensure the desired QoS, which in our experiments was $P(W_I(t) = 0) = \sum_{i=0}^{s_j-1} \pi_i \geq 0.8$ for $t \in [t_j^i, t_j^f]$. Note that the

probability of no delay is computed in the same way for preemptive and exhaustive disciplines. If the chosen QoS were of the form $P(W_I(t) < \tau) \geq \alpha$ the calculations would be different for different end-of-shift policies, as discussed in Chapter 3.

A.5 ICWC Staffing

The ICWC method alternates between a schedule generator and a schedule evaluator to find low cost feasible solutions to the staffing problem. It starts with strict lower bounds on the number of servers needed to ensure the minimum QoS in each period. An integer program is solved to find the minimum cost staffing plan satisfying the constraint that the number of servers in each period is greater than this strict lower bound. This solution is then evaluated and the infeasible intervals, where the QoS goes below the minimum required, are identified. In the next step the additional number of servers needed in each infeasible interval to bring the QoS up to the minimum required level is estimated (approximately). This information is used to build new constraints on the number of servers in each infeasible interval to be added to the original integer program. Since the additional number of servers needed is an approximation, a parameter $\beta \in [0, 1]$ is included to scale down the added constraints, in order to avoid eliminating the optimal solution from the feasible set of the integer program. The higher β is, the tighter the new constraints are. The process of solving the integer program, evaluating the QoS for the solution obtained, and adding new constraints to the integer program is repeated until a feasible solution is reached. Note that the ICWC method allows for constraints on the set of shifts that can be used, I, with a shift $i \in I$ being represented as a binary row vector a_{ij} of length n (the number of planning periods), where $a_{ij} = 1$ if shift i includes planning period j and $a_{ij} = 0$ otherwise.

In Chapter 4.1 experiments we used our lower bound as a starting point for the ICWC method. The ICWC method was run with a maximum duality gap of 0.5% and a maximum number of simplex iterations of 5,000. Also, the parameter

β was set to 0.7 and the calculation period δ_{calc} was set to 5 minutes in all of the experiments.

APPENDIX B

Queueing Models of Case Managers

B.1 Computing Steady State Probabilities and Performance Measures for the R and P Systems

B.1.1 R System

The R system the boundary matrix blocks are:

$$B_0^R = \Lambda/N \begin{bmatrix} 0_{(M-1)M/2,M+1} \\ 0_{M,1}|I_M \end{bmatrix}, \tag{B.1}$$

$$B_1^R = \begin{bmatrix} \Delta & U_1 & & & \\ L_1 & D_1 & U_2 & & \\ & L_2 & D_2 & \ddots & \\ & & \ddots & \ddots & U_{M-1} \\ & & & L_{M-1} & D_{M-1} \end{bmatrix}, \text{ where } U_n^R = \Lambda/N \left[0_{n,1}|I_n\right], \tag{B.2}$$

$$L_n^R = \gamma\mu \begin{bmatrix} 0_{1,n} \\ I_n \end{bmatrix}, \text{ and } D_n^R = \begin{bmatrix} \Delta & n\lambda & & & \\ (1-\gamma)\mu & \Delta & (n-1)\lambda & & \\ & \ddots & \ddots & \ddots & \\ & & (1-\gamma)\mu & \Delta & \lambda \\ & & & (1-\gamma)\mu & \Delta \end{bmatrix}, \quad \text{(B.3)}$$

$$B_2^R = \gamma\mu \begin{bmatrix} 0_{1,x} \\ 0_{M,x-M}|I_M \end{bmatrix}. \quad \text{(B.4)}$$

The vectors π_0^R and π_1^R can be obtained from the boundary conditions

$$\pi_0^R B_1^R + \pi_1^R B_2^R = 0, \quad \text{(B.5)}$$

$$\pi_0^R B_0^R + \pi_1^R A_1^R + \pi_2^R A_2^R = 0, \quad \text{(B.6)}$$

and the normalization condition

$$\pi_0^R e + \sum_{n=1}^{\infty} \pi_n^R e = \pi_0^R e + \pi_1^R \sum_{n=1}^{\infty} (R^R)^{n-1} e = \pi_0^R e + \pi_1^R (I - R^R)^{-1} e = 1, \quad \text{(B.7)}$$

where A_0^R, A_1^R, and A_2^R are defined in Section 5.4.1. Let i_0^R be the column vector of the number of customers assigned to a manager and j_0^R be the column vector of the number of customers in internal queue or in service in the boundary states. We can obtain the state probabilities using (5.5) and we can compute performance measures as:

- Average caseload:

$$\begin{aligned} L_c^R &= \pi_0^R i_0^R + \sum_{n=1}^{\infty} M\pi_n^R e = \pi_0^R i_0^R + M\pi_1^R \sum_{n=1}^{\infty} (R^R)^{n-1} e \quad \text{(B.8)} \\ &= \pi_0^R i_0^R + M\pi_1^R (I - R^R)^{-1} e, \end{aligned}$$

- Average internal queue length:

$$L_q^R = \pi_0^R (j_0^R - e)^+ + \sum_{n=1}^{\infty} \pi_n^R \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ M-1 \end{bmatrix} \tag{B.9}$$

$$= \pi_0^R (j_0^R - e)^+ + \pi_1^R (I - R^R)^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ M-1 \end{bmatrix}$$

- Average utilization:

$$\rho^R = \pi_0^R \min\{j_0^R, 1\} + \sum_{n=1}^{\infty} \pi_n^R \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \pi_0^R \min\{j_0^R, 1\} + \pi_1^R (I - R^R)^{-1} \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{B.10}$$

- Average number of cases in external delay:

$$L_e^R = L_c^R - L_q^R - \rho^R \tag{B.11}$$

- Average pre-assignment queue length, aggregated over all case managers to allow comparisons with S and P systems:

$$L_a^R = N \sum_{n=1}^{\infty} (n-1) \pi_n^R e = N \pi_1^R R^R (I - R^R)^{-2} e \tag{B.12}$$

- Average total system time:

$$T^R = \frac{L_c^R}{\Lambda/N} + \frac{L_a^R}{\Lambda}, \tag{B.13}$$

  where the first term is the average time spent assigned to a case manager (in the internal queue, external queue, or in service) obtained using Little's Law (each case manager receives an arrival rate of Λ/N) and the second term is the average time spent in the pre-assignment queue, also obtained using Little's Law.

Note that L_c^i, L_q^i, L_e^i are all measured "per case manager" (for $i = R, P$), whereas L_a^i is a measure for the system as a whole.

B.1.2 P System

In the P system the boundary matrix blocks B_0^P, B_1^P, and B_2^P are:

$$B_0^P = \Lambda \begin{bmatrix} 0_{(NM-1)NM/2,NM+1} \\ 0_{NM,1}|I_{NM} \end{bmatrix}, \tag{B.14}$$

$$B_2^P = \gamma\mu \left[\begin{array}{c|cccc} & 0 & \dots & \dots & 0 \\ 0_{NM+1,(NM-1)NM/2} & \min\{1,N\} & & & \\ & & \min\{2,N\} & & \\ & & & \ddots & \\ & & & & \min\{NM,N\} \end{array}\right], \tag{B.15}$$

$$B_1^P = \begin{bmatrix} \Delta & U1 & & & \\ L_1 & D_1 & U_2 & & \\ & L_2 & D_2 & \ddots & \\ & & \ddots & \ddots & U_{M-1} \\ & & & L_{M-1} & D_{M-1} \end{bmatrix}, \text{ where } U_n = \Lambda\left[0_{n,1}|I_n\right], \tag{B.16}$$

$$L_n = \gamma\mu \begin{bmatrix} 0 & \dots & \dots & 0 \\ \min\{1,N\} & & & \\ & \min\{2,N\} & & \\ & & \ddots & \\ & & & \min\{n,N\} \end{bmatrix}, \text{ and} \tag{B.17}$$

$$D_n = \begin{bmatrix} \Delta & n\lambda & & & & \\ \min\{1,N\}(1-\gamma)\mu & \Delta & (n-1)\lambda & & & \\ & \min\{2,N\}(1-\gamma)\mu & \ddots & \ddots & & \\ & & \ddots & \Delta & \lambda & \\ & & & \min\{n,N\}(1-\gamma)\mu & \Delta \end{bmatrix}. \tag{B.18}$$

The repeating matrix blocks are (using Δ for generic diagonal elements in A_1^P and A^P):

$$A_0^P = \Lambda I, \tag{B.19}$$

$$A_1^P = \begin{bmatrix} \Delta & NM\lambda & & & & & & \\ (1-\gamma)\mu & \Delta & & & & & & \\ & \ddots & \ddots & & & & & \\ & & (N-1)(1-\gamma)\mu & \Delta & 3\lambda & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & N(1-\gamma)\mu & \Delta & \lambda & \\ & & & & & N(1-\gamma)\mu & \Delta \end{bmatrix} \tag{B.20}$$

$$A_2^P = \gamma\mu \begin{bmatrix} 0 & & & & & & \\ & \ddots & & & & & \\ & & (N-2) & & & & \\ & & & (N-1) & & & \\ & & & & N & & \\ & & & & & \ddots & \\ & & & & & & N \end{bmatrix} \tag{B.21}$$

$$A^P = A_0^P + A_1^P + A_2^P = \tag{B.22}$$

$$= \begin{bmatrix} \Delta & NM\lambda & & & & & & \\ (1-\gamma)\mu & \Delta & & & & & & \\ & \ddots & \ddots & & & & & \\ & & (N-1)(1-\gamma)\mu & \Delta & 3\lambda & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & N(1-\gamma)\mu & \Delta & \lambda & \\ & & & & & N(1-\gamma)\mu & \Delta & \end{bmatrix}$$

The vectors π_0^P and π_1^P can be obtained from the boundary conditions as in (B.5) and (B.6) and the normalization condition as in (B.7). Let i_0^P be the column vector of the number of customers assigned to a manager and j_0^P be the column vector of the number of customers in internal queue or in service in the boundary states. We can then obtain the state probabilities using Equation (5.5) and compute performance measures:

- Average Caseload per Manager(L_c^P)

$$L_c^P = \frac{1}{N}\left[\pi_0^P i_0^P + \sum_{n=1}^{\infty} NM\pi_n^P e\right] = \frac{1}{N}\left[\pi_0^P i_0^P + NM\pi_1^P \sum_{n=1}^{\infty} R^{P^{n-1}} e\right] = \tag{B.23}$$

$$L_c^P = \frac{1}{N}\left[\pi_0^P i_0^P + NM\pi_1^P (I - R^P)^{-1} e\right] \tag{B.24}$$

- Average Length of Internal Queue per Manager (L_q)

$$L_q^P = \frac{1}{N}\left[\pi_0^P \max\{j_0^P - e, 0\} + \sum_{n=1}^{\infty} \pi_n^P \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ NM-1 \end{bmatrix}\right] \tag{B.25}$$

$$L_q^P = \frac{1}{N}\left[\pi_0^P \max\{j_0^P - e, 0\} + \pi_1^P (I - R^P)^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ NM-1 \end{bmatrix}\right] \tag{B.26}$$

- Average utilization (ρ^P)

$$\rho^P = \frac{1}{N}\left[\pi_0^P \min\{j_0^P, N\} + \sum_{n=1}^{\infty} \pi_n^P \begin{bmatrix} 0 \\ \vdots \\ N \\ \vdots \\ N \end{bmatrix}\right] \tag{B.27}$$

$$= \frac{1}{N}\left[\pi_0^P \min\{j_0^P, N\} + \pi_1^P (I - R^P)^{-1} \begin{bmatrix} 0 \\ \vdots \\ N \\ \vdots \\ N \end{bmatrix}\right] \tag{B.28}$$

- Average Number of Cases in External Delay per Manager (L_d^P)

$$L_e^P = L_c^P - L_q^P - \rho^P \tag{B.29}$$

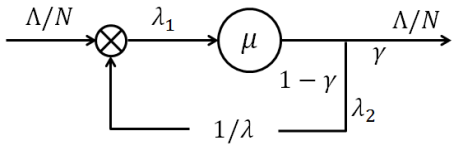

Figure B.1: Jackson network for a single manager in a random routing system with unlimited caseload.

- Average Length of Pre-Assignment Queue (L_a^P)

$$L_a^P = \sum_{n=1}^{\infty}(n-1)\pi_n^P e = \pi_1 R^P (I - R^P)^{-2} e \tag{B.30}$$

- Average Total Time in the System (T^P)

$$T^P = \frac{NL_c^P}{\Lambda} + \frac{L_a^P}{\Lambda} \tag{B.31}$$

B.2 Stability Limits in Special Cases

B.2.1 Stability Limits for R and P Systems with Infinite Caseload Limit

A single case manager in an R system with N case managers and unlimited caseload can be represented by the Jackson Network (Jackson 1957) in Figure B.1. In this Jackson network flow balance requires $\lambda_1\gamma = \Lambda/N$, where λ_1 is the arrival rate to the case manager. In order for the network to be stable, every node in the network needs to be stable. The external delay node has infinitely many servers, so it will always be stable. In order for the service node to be stable we need $\lambda_1/\mu = \Lambda/(N\gamma\mu) < 1 \Rightarrow \Lambda < N\gamma\mu$. The stability limit $\Lambda_{\text{lim}}^R = N\gamma\mu$ is the rate with which cases leave the system if the case managers are never idle. When M is infinite, a case manager's capacity is never reduced because of forced idleness while there are cases available to work on.

A P system with N case managers and unlimited caseload can be represented by the Jackson Network in Figure B.2. In this Jackson network $\lambda_1\gamma = \Lambda$. In order for the service node to be stable we need $\lambda_1/(N\mu) = \Lambda/(N\gamma\mu) < 1 \Rightarrow \Lambda < N\mu\gamma$. We conclude that as M tends to infinity, the stability conditions for the R and P

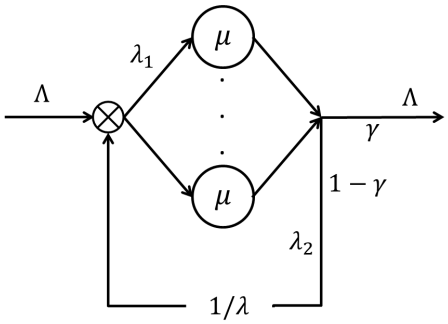

Figure B.2: Jackson network for a pooled system with unlimited caseload.

systems converge to the same value.

B.2.2 Stability Limits for the S System with Two Case Managers

To formulate an S system with $N = 2$ case managers as a QBD we need 5 state variables: l (total number of customers in the system), l_i (number of customers assigned to case manager i, $i = 1, 2$), and q_i (number of customers assigned to case manager i, $i = 1, 2$, that are service or in internal queue). We order the states lexicographically and define the level as 0 for the states where $l < NM$ and as $l - NM + 1$ for the states where $l \geq NM$. We use l_1, l_2, q_1, and q_2 to define the phase. The possible transitions are:

- Arrival of a new case:

$$(l, l_1, l_2, q_1, q_2) \rightarrow \begin{cases} (l+1, l_1+1, l_2, q_1+1, q_2), & \text{when } l_1 < l_2 \leq M \text{ (rate } \Lambda) \text{ or} \\ & l_1 = l_2 < M \text{ (rate } \Lambda/2) \\ (l+1, l_1, l_2+1, q_1, q_2+1), & \text{when } l_2 < l_1 \leq M \text{ (rate } \Lambda) \text{ or} \\ & l_1 = l_2 < M \text{ (rate } \Lambda/2) \\ (l+1, l_1, l_2, q_1, q_2), & \text{when } l_1, l_2 \geq M \text{ (rate } \Lambda) \end{cases} \tag{B.32}$$

- Service completion that results in case completion:

$$(l, l_1, l_2, q_1, q_2) \to \begin{cases} (l-1, l_1-1, l_2, q_1-1, q_2), & \text{when } q_1 > 0 \text{ and } l \le 2M \\ & \text{(rate } \gamma\mu\text{)} \\ (l-1, l_1, l_2-1, q_1, q_2-1), & \text{when } q_2 > 0 \text{ and } l \le 2M \\ & \text{(rate } \gamma\mu\text{)} \\ (l-1, l_1, l_2, q_1, q_2), & \text{when } q_1, q_2 > 0 \text{ and } l > 2M \\ & \text{(rate } 2\gamma\mu\text{), or } \min\{q_1, q_2\} = 0, \\ & \max\{q_1, q_2\} > 0, \text{ and } l > 2M \\ & \text{(rate } \gamma\mu\text{)} \end{cases} \tag{B.33}$$

- Service completion that does not result in case completion:

$$(l, l_1, l_2, q_1, q_2) \to \begin{cases} (l, l_1, l_2, q_1-1, q_2), & \text{when } q_1 > 0 \text{ (rate } [1-\gamma]\mu\text{)} \\ (l, l_1, l_2, q_1, q_2-1), & \text{when } q_2 > 0 \text{ (rate } [1-\gamma]\mu\text{)} \end{cases} \tag{B.34}$$

- Completion of external delay:

$$(l, l_1, l_2, q_1, q_2) \to \begin{cases} (l, l_1, l_2, q_1+1, q_2), & \text{when } [l_1 - q_1] > 0 \text{ (rate } [l_1 - q_1]\lambda\text{)} \\ (l, l_1, l_2, q_1, q_2+1), & \text{when } [l_2 - q_2] > 0 \text{ (rate } [l_2 - q_2]\lambda\text{)} \end{cases} \tag{B.35}$$

The S system with $N = 2$ repeating matrix blocks A_0^S, A_1^S, and A_2^S are square matrices of order $(M+1)^2$.

B.3 Parameters for Caseload Experiments

We list the parameters for the caseload experiments in Section 5.8.4 in Tables B.1 (Series A) and B.2 (Series B).

Table B.1: Parameters for Series A ($N = 3$ case managers and $M = 5$ cases in all experiments).

Exp. #	λ	γ	Λ	μ	Exp. #	λ	γ	Λ	μ
1	0.95	0.54	7.60	5.91	42	1.80	0.75	8.60	9.00
2	0.95	0.54	7.60	7.00	43	1.80	0.75	9.30	5.91
3	0.95	0.54	7.60	9.00	44	1.80	0.75	9.30	7.00
4	0.95	0.54	8.60	5.91	45	1.80	0.75	9.30	9.00
5	0.95	0.54	8.60	7.00	46	1.80	0.95	7.60	5.91
6	0.95	0.54	8.60	9.00	47	1.80	0.95	7.60	7.00
7	0.95	0.54	9.30	5.91	48	1.80	0.95	7.60	9.00
8	0.95	0.54	9.30	7.00	49	1.80	0.95	8.60	5.91
9	0.95	0.54	9.30	9.00	50	1.80	0.95	8.60	7.00
10	0.95	0.75	7.60	5.91	51	1.80	0.95	8.60	9.00
11	0.95	0.75	7.60	7.00	52	1.80	0.95	9.30	5.91
12	0.95	0.75	7.60	9.00	53	1.80	0.95	9.30	7.00
13	0.95	0.75	8.60	5.91	54	1.80	0.95	9.30	9.00
14	0.95	0.75	8.60	7.00	55	2.65	0.54	7.60	5.91
15	0.95	0.75	8.60	9.00	56	2.65	0.54	7.60	7.00
16	0.95	0.75	9.30	5.91	57	2.65	0.54	7.60	9.00
17	0.95	0.75	9.30	7.00	58	2.65	0.54	8.60	5.91
18	0.95	0.75	9.30	9.00	59	2.65	0.54	8.60	7.00
19	0.95	0.95	7.60	5.91	60	2.65	0.54	8.60	9.00
20	0.95	0.95	7.60	7.00	61	2.65	0.54	9.30	5.91
21	0.95	0.95	7.60	9.00	62	2.65	0.54	9.30	7.00
22	0.95	0.95	8.60	5.91	63	2.65	0.54	9.30	9.00
23	0.95	0.95	8.60	7.00	64	2.65	0.75	7.60	5.91
24	0.95	0.95	8.60	9.00	65	2.65	0.75	7.60	7.00
25	0.95	0.95	9.30	5.91	66	2.65	0.75	7.60	9.00
26	0.95	0.95	9.30	7.00	67	2.65	0.75	8.60	5.91
27	0.95	0.95	9.30	9.00	68	2.65	0.75	8.60	7.00
28	1.80	0.54	7.60	5.91	69	2.65	0.75	8.60	9.00
29	1.80	0.54	7.60	7.00	70	2.65	0.75	9.30	5.91
30	1.80	0.54	7.60	9.00	71	2.65	0.75	9.30	7.00
31	1.80	0.54	8.60	5.91	72	2.65	0.75	9.30	9.00
32	1.80	0.54	8.60	7.00	73	2.65	0.95	7.60	5.91
33	1.80	0.54	8.60	9.00	74	2.65	0.95	7.60	7.00
34	1.80	0.54	9.30	5.91	75	2.65	0.95	7.60	9.00
35	1.80	0.54	9.30	7.00	76	2.65	0.95	8.60	5.91
36	1.80	0.54	9.30	9.00	77	2.65	0.95	8.60	7.00
37	1.80	0.75	7.60	5.91	78	2.65	0.95	8.60	9.00
38	1.80	0.75	7.60	7.00	79	2.65	0.95	9.30	5.91
39	1.80	0.75	7.60	9.00	80	2.65	0.95	9.30	7.00
40	1.80	0.75	8.60	5.91	81	2.65	0.95	9.30	9.00
41	1.80	0.75	8.60	7.00					

Table B.2: Parameters for Series B ($N = 3$ case managers and $M = 5$ cases in all experiments).

Exp. #	λ	γ	Λ	μ
1	0.25	0.20	3.40	5.91
2	0.40	0.20	3.40	5.91
3	0.50	0.20	3.40	5.91
4	0.25	0.30	5.00	5.91
5	0.40	0.30	5.00	5.91
6	0.50	0.30	5.00	5.91
7	0.25	0.40	6.90	5.91
8	0.40	0.40	6.90	5.91
9	0.50	0.40	6.90	5.91
10	0.25	0.50	6.90	5.91
11	0.40	0.50	6.90	5.91
12	0.50	0.50	6.90	5.91
13	0.25	0.20	3.01	5.91
14	0.40	0.20	3.01	5.91
15	0.50	0.20	3.01	5.91
16	0.25	0.30	4.52	5.91
17	0.40	0.30	4.52	5.91
18	0.50	0.30	4.52	5.91
19	0.25	0.40	6.03	5.91
20	0.40	0.40	6.03	5.91
21	0.50	0.40	6.03	5.91
22	0.25	0.50	7.54	5.91
23	0.40	0.50	7.54	5.91
24	0.50	0.50	7.54	5.91

B.4 B System

The following is one mechanism that ensures that caseloads for any two case managers differ by at most one case. If the pre-assignment queue is empty and case manager i completes a job and is left with m_i jobs, compare m_i with the caseload of the case manager k with the largest number of cases, m_k. If $m_k - m_i > 1$, then move one case from case manager k to case manager i. If the pre-assignment queue is occupied when a case manager completes a job, then she pulls a case from the pre-assignment queue. If a new case arrives and finds the pre-assignment queue empty, then assign the case to a server with the smallest caseload. If all caseloads $m_i = M$, then an arriving case waits in the pre-assignment queue.