INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

A Survey and Experimental Comparison of Service-Level-Approximation Methods for Nonstationary M(t)/M/s(t) Queueing Systems with Exhaustive Discipline

Armann Ingolfsson, Elvira Akhmetshina, Susan Budge, Yongyue Li, Xudong Wu,

To cite this article:
Armann Ingolfsson, Elvira Akhmetshina, Susan Budge, Yongyue Li, Xudong Wu, (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. https://doi.org/10.1287/ijoc.1050.0157

INFORMS Journal on Computing
Vol. 19, No. 2, Spring 2007, pp. 201–214
ISSN 1091-9856 | EISSN 1526-5528 | 07 | 1902 | 0201

DOI 10.1287/ijoc.1050.0157

A Survey and Experimental Comparison of Service-Level-Approximation Methods for Nonstationary $M(t)/M/s(t)$ Queueing Systems with Exhaustive Discipline

Armann Ingolfsson, Elvira Akhmetshina, Susan Budge, Yongyue Li
School of Business, University of Alberta, Edmonton, Alberta T6G 2R6, Canada
{armann.ingolfsson@ualberta.ca, elvira@ualberta.ca, sbudge@ualberta.ca, yongyue@ualberta.ca}

Xudong Wu
Department of Computing Science, University of Alberta, Edmonton, Alberta T6G 2R6, Canada,
xudong@cs.ualberta.ca

We compare the performance of seven methods in computing or approximating service levels for nonstationary $M(t)/M/s(t)$ queueing systems: an exact method (a Runge-Kutta ordinary-differential-equation solver), the randomization method, a closure (or surrogate-distribution) approximation, a direct infinite-server approximation, a modified-offered-load infinite-server approximation, an effective-arrival-rate approximation, and a lagged stationary approximation. We assume an exhaustive service discipline, where service in progress when a server is scheduled to leave is completed before the server leaves. We used all of the methods to solve the same set of 640 test problems. The randomization method was almost as accurate as the exact method and used about half the computational time. The closure approximation was less accurate, and usually slower, than the randomization method. The two infinite-server-based approximations, the effective-arrival-rate approximation, and the lagged stationary approximation were less accurate but had computation times that were far shorter and less problem-dependent than the other three methods.

Key words: queues, nonstationary; queues, algorithms; queues, approximations
History: Accepted by Edward P. C. Kao, Area Editor for Computational Probability and Analysis; received June 2002, revised June 2003, December 2004, May 2005; accepted July 2005.

1. Introduction

Systems where time-variable and stochastic customer demand is served by a time-variable number of servers are pervasive in modern society; for example, call centers, retail stores, and emergency services. Planning for such systems involves a difficult trade-off between the cost associated with the servers (which we will call *labor cost*—although more generally some services could be provided by machines rather than humans) and the customer *service level*. Labor cost is usually easy to quantify, at least compared to the level of service. A common quantification of the level of service is the fraction of customers that wait less than some threshold amount of time, τ, before being served. While one can measure service level after the fact if appropriate data have been collected, it is more difficult to predict what level of service will result from making a certain number of servers available at different times. The aim of this paper is to compare the computational effort and accuracy of seven proposed methods for calculating or approximating the service level, as defined above, for $M(t)/M/s(t)$ queueing systems (with a nonstationary Poisson arrival process, exponentially distributed service times, and a time-varying number of servers). To complete the specification of the $M(t)/M/s(t)$ model, we assume an *exhaustive* policy, where service in progress when a server is scheduled to leave is completed before the server leaves. This is in contrast to a *pre-emptive* policy, which has been assumed in most previous related work, where service is interrupted and the customer rejoins the queue when the server is scheduled to leave.

We are primarily motivated by the problem of evaluating the time-varying service level resulting from a given staffing schedule. The staffing schedule could be generated manually or by a scheduling algorithm that calls a subroutine to evaluate service levels for various schedules, as in Ingolfsson et al. (2002a, b), and Atlason et al. (2004). In either case, utilization (offered load divided by number of servers) may vary widely over time and can exceed 100% temporarily, for example, when the number of servers is limited by

the size of a physical facility, service-level evaluation becomes challenging.

A problem that is related to service-level evaluation is finding staffing requirements that, if followed, will provide a specified level of service. Such requirements are often used as input to a scheduling algorithm. Service-level evaluation is related to finding staffing requirements, because a reliable method for service-level evaluation could be used as a subroutine for a staffing-requirements algorithm. Indeed, some of the approximations we consider (e.g., the effective-arrival-rate and the lagged stationary approximations) were originally developed to generate staffing requirements, but they imply an underlying service-level-approximation method.

We focus on the computational effort and accuracy of approximation methods. When a service-level approximation is used to generate staffing requirements, additional considerations become relevant. For example, instead of insisting on staffing requirements that guarantee a specified minimum service level at all times, it may be more appropriate to allow the service level to drop slightly below a target (for example, by at most 10%; see Green et al. 2001, 2003) but these and other considerations relevant to the evaluation of methods that generate staffing requirements are outside our scope.

While our primary motivation is performance evaluation for service systems in which the number of servers varies over time, our results may be of interest for other application areas, such as airport-capacity evaluation (Bookbinder 1986, Koopman 1972, Barnhart et al. 2003), allocation of helicopters for fighting forest fires (Bookbinder and Martell 1979), and telephone trunk line performance evaluation (Jennings and Massey 1997).

The methods we compare are (1) numerical solution of the Chapman-Kolmogorov forward equations using the Runge-Kutta method, (2) randomization (or uniformization) (Jensen 1953, Grassmann 1977), (3) closure (or surrogate-distribution) approximation (Rothkopf and Oren 1979, Clark 1981, Taaffe and Ong 1987), (4) direct infinite-server approximation, (5) infinite-server modified-offered-load (MOL) approximation (Jagerman 1975, Jennings et al. 1996, Massey and Whitt 1997), (6) effective arrival rate approximation (Thompson 1993), and (7) lagged stationary independent period-by-period (SIPP) approximation (Green et al. 2001, 2003). We will use method (1) as a baseline and refer to it as the *exact solution*, as it involves no approximations other than those needed to solve numerically an infinite set of differential equations.

Our list of methods is not exhaustive. Notable omissions are the pointwise stationary approximation (PSA, see Green and Kolesar 1991) and the (nonlagged) stationary independent period-by-period approximation (SIPP, see Green et al. 2001, 2003). Both methods use a sequence of stationary $M/M/s$ models to approximate performance. The methods differ in how the arrival rates to the stationary models are determined, with SIPP using an average arrival rate over a period of nonzero duration (during which the number of servers is constant) to approximate performance during that period and PSA using the arrival rate at each epoch to approximate performance at that epoch. Green and Kolesar (1991) demonstrate empirically that PSA provides tight upper bounds for the probability of delay and expected delay, and Whitt (1991) shows that PSA is asymptotically correct as the rates increase. Implementing SIPP and PSA is straightforward and involves minimal computation. When they are sufficiently accurate, these are the methods of choice, because of their simplicity. However, as Green et al. (2001, 2003) point out, the SIPP approximation results in large errors in many commonly occurring situations. A lagged version of SIPP reduces these errors, and we include this version in our comparison.

Other methods we chose not to compare include transform-inversion techniques and Krylov subspace methods (see Grassmann (2000) and Stewart (1994) for a description of these methods and of some of the methods that we compare).

Most of the methods we compare have been known for several years, and some have been used for various applications, but they do not seem to be generally known in certain obvious fields of application. For example, workforce-management software for call centers typically use Erlang formulas based on stationary $M/M/s$ models—in effect using the nonlagged SIPP approximation—even though these models often provide poor approximations. When choosing a performance-evaluation method (exact or approximate) for the applications we have outlined, a practitioner will likely attempt to answer some of the following questions: (1) Are the assumptions of the method appropriate for the system? (2) Is software that implements the method available, and if not, how difficult is it to implement the method? (3) How accurate is the method? (4) How much computation time does the method require? Our goal is to help practitioners answer these questions. While we focus on $M(t)/M/s(t)$ systems, we summarize (in the Online Supplement to this paper on the journal's website) the extent to which each method can model more general systems. We implemented all of the methods on a common platform (Matlab). We comment on implementation issues and compare computation speed and accuracy of the different methods for test problems designed to span a wide range of situations encountered in practice.

Other researchers have compared some of these methods but usually in a context where the arrival

rate or the number of servers are constant. For example, Reibman and Trivedi (1988) compared Runge-Kutta, the randomization method, and a stable implicit ordinary differential equation (ODE) solver for calculating transient probabilities for stationary Markovian systems. Of the three methods, they concluded that randomization would be preferred in most situations. Leese and Boyd (1966) compared several methods that included a stable predictor-corrector ODE solver, a method similar to the randomization method, and simulation, for an instance of an $M(t)/M/1$ model. They concluded that a method involving a Volterra-type integral equation (Wragg 1963) was the most useful for the application they studied, in terms of the computational effort required for a given precision. Unfortunately, this method is not applicable to multi-server systems. Green et al. (2001, 2003) used Runge-Kutta to evaluate the probability of delay that results when the number of servers is set using the SIPP approximation and various modifications of it.

We focus on the $M(t)/M/s(t)$ model for two reasons:

1. It is the simplest model that seems realistic for labor scheduling, incorporating both randomness and time variation in the arrival and service processes.

2. Although the methods we compare can be used for more general queueing systems, the types of possible generalizations depend on the method. All of these methods can be applied to approximate $M(t)/M/s(t)$ systems.

The two components of our model that vary with time are the arrival rate $\lambda(t)$ and the number of servers $s(t)$. The number of servers changes discontinuously from one constant level to another. Although it seems more natural to expect the true arrival rate to vary continuously with time, it is often modeled as piecewise constant, for various reasons, including (1) the only available arrival data may be aggregated over time intervals of some length, (2) estimating a continuous arrival-rate function may be difficult, and (3) some methods (e.g., randomization) require the arrival rate to be piecewise constant. As the cost of data collection and storage decreases, the first reason should become less compelling. Research continues on how to estimate arrival-rate functions either from period counts (for example, Massey et al. 1996) or transactional data (for example, Arkin and Leemis 2000). The relevance of research on estimating continuous arrival-rate functions depends, in part, on whether such functions lead to different results than using a piecewise-constant arrival-rate function. Our comparison of the exact method, with a continuous arrival-rate function, and randomization, with a piecewise-constant arrival-rate function over five-minute intervals, leads to almost identical service-level curves in all cases.

The service rate μ may also change with time, because of time variability in either server productivity or the customer mix. However, we expect that the service rate will typically change more slowly than the arrival rate so we assume it to be constant, to simplify the exposition and to limit the number of factors in our computational experiment. All of the computational methods we discuss can be extended to account for time-varying service rates.

We focus on service level because this seems to be the most common measure used to evaluate performance of staffing plans (e.g., Cleveland and Mayben 1997). Some approximations (for example, closure approximations) may perform better in approximating other performance measures, such as the mean number of customers in the system.

Our methodological contributions focus on epochs when the number of servers $s(t)$ changes. We demonstrate (in Section 2) how one can model servers finishing their current task before going off duty (rather than pre-empting service, as assumed in previous related work). Two main empirical conclusions can be drawn from our work. First, if accuracy is important, then randomization may be preferable to the exact method because it gave almost identical results to the exact method but required about half the computation time. Second, when computation speed is important and the relevant performance measure is the probability of zero delay, the modified-offered-load approximation may be preferred, providing a median time-average relative error of 16%, with median computation times of 1 second to evaluate performance for a 24-hour span. If a more appropriate performance measure is the probability of delay being less than or equal to a positive threshold, then an infinite-server approximation is more accurate, with a median time-average relative error of about 50% and computation times similar to the modified-offered-load approximation.

Section 2 presents the model formally, defines notation, and develops formulas for calculating service levels; Section 3 describes each method; Section 4 describes our experimental design; and Section 5 presents computational results. Section 6 summarizes the results and our conclusions.

The Online Supplement contains additional material on flexibility and implementation issues for each method and more computational results.

2. Model Definition and Common Notation

We are interested in calculating the service level as a function of time over an interval $(0, T]$. We assume customers are served FCFS. Let $W(t)$ be the virtual waiting time at time t and let τ be an "acceptable-waiting-time" threshold. We define the service level

to be $\mathrm{SL}(t) = \Pr\{W(t) \le \tau\}$ and the "complementary service level" as $\overline{\mathrm{SL}}(t) = \Pr\{W(t) > \tau\}$. We suppress the dependence of the service level on τ to simplify notation. Informally, the service level is the probability that a (fictitious) customer that arrives at time t would have to wait τ time units or less.

We divide the time interval $(0, T]$ into n_p "planning periods" of length δ_p and assume that the number of servers $s(t)$ can change only at the beginning of a planning period. Most past work that has numerically solved $M(t)/M/s(t)$ systems has implicitly assumed a *pre-emptive* discipline, where a customer in service is sent back to the queue when the server is scheduled to leave. Instead, we assume an *exhaustive* discipline, where the service is completed before the server leaves, which we believe is more realistic. We outline below how this discipline can be modeled. The approach is extended and described in greater detail in Ingolfsson (2005). We have implemented this approach with the exact and randomization methods, the two that calculate state-probability vectors directly. In comparing our computational results to previous work, note that two of the approximations (the modified-offered-load and lagged stationary approximations) were tested against an exact solution of a system with pre-emptive discipline. The effective-arrival-rate approximation was tested against a simulation model with exhaustive discipline. Closure approximations have, to our knowledge, been used only for systems where the number of servers is constant, in which case the distinction between exhaustive and pre-emptive service disappears.

When considering an exhaustive discipline, we distinguish between customers receiving service from servers that are scheduled to be on duty (Type 1 customers) and those receiving service from servers that are working past their scheduled end time (Type 2 customers). Type 2 customers have no impact on how long future customers will have to wait, because when such a customer completes service, the server leaves. With this in mind, we define the state variable $N(t)$ for an $M(t)/M/s(t)$ queue as the sum of the number of Type 1 customers in the system and the number of customers in queue. Let $B(t) = \min(N(t), s(t))$ be the number of busy servers at time t (excluding servers working beyond their scheduled time).

$N(t)$ is a continuous-time Markov chain (CTMC) except when some servers are scheduled to go off duty. At these epochs, the process experiences an instantaneous transition, "ejecting" Type 2 customers (in reality, the customers remain in the system until served, but in the model, it is no longer necessary to keep track of them, because they have no impact on future waiting times). Ingolfsson (2005) calls this a *mixed discrete-continuous Markov chain* and shows that it can be viewed as a semi-Markov process. Let $\pi_i(t) = \Pr\{N(t) = i\}$ and let $\pi(t)$ be the vector $(\pi_0(t), \pi_1(t), \ldots)$. In the interior of each planning period (i.e., for $t \in ((i-1)\delta_p, i\delta_p)$ for some i), these probabilities evolve according to the Chapman-Kolmogorov forward equations (Kleinrock 1974, for example). At epochs where some servers might be scheduled to go off duty (i.e., $t = i\delta_p$ for some i), the state-probability vector experiences an instantaneous transition

$$\pi(t^+) = \pi(t^-)P(t), \tag{1}$$

where t^+ and t^- are the epochs just after and just before epoch t, respectively. If no servers are scheduled to go off duty at epoch t, then $P(t) = I$. If u servers are scheduled to go off duty at epoch t, consider two cases. If $N(t^-) \ge s(t^-)$, then all servers are busy just before epoch t, and u Type 2 customers will be "ejected" as a result of u servers being scheduled to go off duty. If $N(t^-) < s(t^-)$, then some servers are idle. The number of customers to be ejected will then equal the number of servers that are busy *and* scheduled to go off duty. We assume that an arriving customer is randomly assigned to a server when multiple servers are available. Then the number of customers to be ejected follows a hypergeometric distribution (Johnson et al. 1993), corresponding to an urn with a total of $s(t^-)$ balls (i.e., servers), u of them white (i.e., scheduled to go off duty), with a sample of $N(t^-)$ balls drawn without replacement. The probability that δN customers are ejected (i.e., δN white balls are drawn) is therefore

$$\phi(\delta N; u, s(t^-), N(t^-)) = \frac{\binom{N(t^-)}{\delta N}\binom{s(t^-) - N(t^-)}{u - \delta N}}{\binom{s(t^-)}{u}}. \tag{2}$$

In this situation, the number of customers to be ejected is at most $\min(N(t^-), u)$ and at least $(N(t^-) - (s(t^-) - u))^+$, so the transition-probability matrix $P(t)$ has the following nonzero entries:

$$\begin{aligned} &p_{n,\,n-u} = 1 \quad \text{for } n = s(t^-), s(t^-)+1, \ldots \\ &p_{n,\,n-\delta n} = \phi(\delta n; u, s(t^-), n) \\ &\qquad\qquad \text{for } n = 0, 1, \ldots, s(t^-) - 1 \quad \text{and} \\ &\qquad\qquad (n - (s(t^-) - u))^+ \le \delta n \le \min(u, n). \end{aligned} \tag{3}$$

Note that u will not necessarily be the decrease in the number of scheduled servers. As an example, if 5 servers are scheduled to continue, 5 servers are scheduled to leave, and 3 servers are scheduled to start at epoch t, then $s(t^-) = 10$, $s(t^+) = 8$, and $u = 5$. Allowing u to be different from $\max(s(t^-) - s(t^+), 0)$ adds no complexity to the calculations.

We now calculate service levels as a function of the state probabilities. Our basic approach of conditioning on $N(t)$ is standard for $M/M/s$ queues, but we

need to worry about whether the number of servers will change in the next τ time units. Let $D(t, t+\tau]$ be the number of service completions for Type 1 customers during $(t, t+\tau]$ and assume, initially, that the number of servers is constant during this interval. If $N(t) = s(t) + i$, $i \geq 0$, then a fictitious customer that arrives at t will wait for $i+1$ service completions before commencing service. As long as all $s(t)$ servers are busy, service completions occur according to a Poisson process with rate $\mu s(t)$. The event $W(t) > \tau$, conditional on $N(t) = s(t) + i$, $i \geq 0$ is equivalent to $D(t, t+\tau] \leq i$. The probability of this latter event is the sum of terms from a Poisson distribution with mean $a \equiv \mu \int_t^{t+\tau} s(u)\,du = \mu\tau s(t)$. Combining all of these facts leads to

$$\begin{aligned} \mathrm{SL}(t) &= \Pr\{W(t) \leq \tau\} = 1 - \Pr\{W(t) > \tau\} \\ &= 1 - \sum_{i=0}^{\infty} \pi_{s(t)+i}(t) \sum_{j=0}^{i} \frac{a^j e^{-a}}{j!}. \end{aligned} \tag{4}$$

Next, suppose that the number of servers increases from $s(t)$ to $s(t) + \delta s$ at epoch $t + \epsilon < t + \tau$ and assume that $N(t) = s(t) + \delta s + i$, $i \geq 0$. Then the first δs waiting customers will commence service before $t + \tau$ with certainty. After taking this into account, the argument in the previous paragraph proceeds in the same way as before, and we have that

$$\mathrm{SL}(t) = 1 - \sum_{i=0}^{\infty} \pi_{s(t)+\delta s+i}(t) \sum_{j=0}^{i} \frac{a^j e^{-a}}{j!} \tag{5}$$

where now $a = \mu(\epsilon s(t) + (\tau - \epsilon)(s(t) + \delta s))$. Finally, suppose that the number of servers decreases from $s(t)$ to $s(t) - \delta s$ at epoch $t + \epsilon < t + \tau$. Then (4) holds, if a is set to $\mu(\epsilon s(t) + (\tau - \epsilon)(s(t) - \delta s))$.

An expression equivalent to (4) appeared in Ingolfsson et al. (2002b), but that paper failed to consider the end-of-shift issues discussed above. Expression (5) was developed independently by Green and Soares (2007), who also treat cases where the number of servers changes more than once in $(t, t+\tau]$, assuming a pre-emptive discipline. As shown in Ingolfsson (2005), (5) extends readily, under an exhaustive discipline, to cases where the number of servers changes multiple times in $(t, t+\tau)$.

For one of the methods (randomization), we approximate the arrival-rate function $\lambda(t)$ with a piecewise-constant function $\tilde{\lambda}(t) = \tilde{\lambda}_i$ for $t \in ((i-1)\cdot\delta_{\mathrm{calc}}, i\delta_{\mathrm{calc}})$, where δ_{calc} is the length of a "calculation period." We assume that the length of a planning period is an integer multiple of the length of a calculation period. We denote the average arrival rate by $\bar{\lambda} = \int_0^T \lambda(t)\,dt/T$ and the average number of servers by $\bar{s} = \int_0^T s(t)\,dt/T$. When we need to approximate the infinite-capacity $M(t)/M/s(t)$ system with a finite-capacity system, we denote the system capacity (the sum of the number of servers and the queue capacity) by K.

3. Computational Methods

In this section, we briefly describe the computational methods. In the Online Supplement, we comment on the extent to which each method can be used for more general systems, implementation issues, and evidence regarding accuracy and speed from other researchers.

3.1. Exact Method

We refer to the numerical solution of the forward equations using general purpose ODE solvers as the "exact method." This requires approximation of the infinite set of forward equations with the first $K+1$ equations and the approximations inherent in any numerical solution of ODEs. However, setting solver parameters appropriately can control the error from these approximations, so we refer to this method as exact. It is commonly used as a benchmark; see, for example, Green et al. (1991) and Odoni and Roth (1983).

We used the ode45 Runge-Kutta ODE solver from the Matlab ODE suite (Shampine and Reichelt 1997). We called the solver separately for each planning period, applying the matrix multiplication in (1) at the end of each period, and using the resulting probability vector as the initial solution for the next planning period. We started with a system capacity $K = \max(100, \max\{s(t)\colon t \in [0, T]\})$ and checked whether the solution satisfied $\pi_K(t) \leq \epsilon_K = 10^{-6}$ for all t. If not, we increased K in steps of 50% until the condition was met.

3.2. Randomization Method

The randomization method (also known as the uniformization method) is originally due to Jensen (1953). Grassmann (1977), Gross and Miller (1983), and Reibman and Trivedi (1988) provide further details on the method and its probabilistic interpretation, applications, and implementation. The method relies on the fact that if the total transition rate out of every state in a homogeneous CTMC is the same, then the total number of state transitions in any time interval is Poisson. The total transition rate is not, in general, the same for all states, but can be made the same ("uniformized") by introducing fictional self-transitions.

Randomization, as usually presented, applies only to homogeneous CTMCs. In a homogeneous $M/M/s$ system, the total transition rate out of state i is $\lambda + \min(i, s)\mu$ so the maximum total transition rate is $L = \lambda + s\mu$, and states $s, s+1, \ldots$ all have this rate. States $0, 1, \ldots, s-1$ can be uniformized by adding self-transitions to state i at rate $(s-i)\mu$. Then, state transitions (including self-transitions) will occur according to a Poisson process with rate L. When a transition occurs, the next state is chosen according to the transition probability matrix $P = Q/L - I$, where Q is the

infinitesimal generator matrix for the process. One can view P as a transition-probability matrix for a discrete-time Markov chain, with the times of transitions being "randomized" according to a Poisson process with rate L; hence the name "randomization." Letting $p_J(j)$ be a Poisson PMF with mean Lt, the transient state probabilities at time t can be calculated by conditioning on the number of transitions in $(0, t]$ using the law of total probability:

$$\pi(t) = \sum_{j=0}^{\infty} p_J(j)\pi(0)P^j. \quad (6)$$

To implement randomization, we truncate the infinite series (6) after the first m terms. We used $m = \lceil Lt + 5\sqrt{Lt} + 4.9 \rceil$ as suggested by Grassmann (1989). More generally, m can be chosen to control the truncation error by using a normal approximation to the Poisson distribution and adding a constant term (the 4.9 in the expression we used) for situations where the normal approximation is poor. Efficient calculation computes the terms in (6) recursively. The first term involves $p_J(0) = \exp(-Lt)$. If Lt is large, this calculation will result in zero with finite-precision arithmetic and will also cause $p_J(1), p_J(2), \ldots$ to evaluate to zero. There are various ways to avoid this. We chose to limit the size of the time step t to prevent Lt from getting too large, limiting t to be at most $\ln(1/\epsilon_t)/L$ with $\epsilon_t = 10^{-30}$.

For homogeneous CTMCs, randomization involves only two approximations: truncation of the infinite series in (6) and truncation of the state space if it is infinite. For inhomogeneous CTMCs, such as our system, one must approximate the time-varying elements in Q with piecewise-constant functions and apply the method separately over each interval where the generator matrix is constant. For our purposes, this involved approximating the arrival-rate function $\lambda(t)$ with a piecewise-constant function $\tilde{\lambda}(t)$.

3.3. Closure Approximations

Rothkopf and Oren (1979) presented a closure approximation for $M(t)/M/s$ queues, approximating the forward equations (an infinite set of differential equations) with two differential equations—one for $\mathrm{E}[N(t)]$ and another for $\mathrm{var}[N(t)]$. Unfortunately, these equations cannot be solved without knowing the transient state probabilities $\pi(t)$. To get around this, Rothkopf and Oren assumed that $\pi(t)$ could be approximated with a negative binomial distribution so the transient state probabilities can be expressed in terms of $\mathrm{E}[N(t)]$ and $\mathrm{var}[N(t)]$, "closing" the set of differential equations for the mean and variance. One could refer to the negative binomial distribution as the *closure distribution*. Rothkopf and Oren found that this choice worked well for single-server systems, but was less accurate with multiple servers.

Clark (1981) extended Rothkopf and Oren's work, approximating the forward equations with five differential equations for the first two moments of the number of customers in the system and the number of busy servers, and for the probability that no customers are waiting in queue. Clark's closure approximation was more accurate, especially with multiple servers. Taaffe and Ong (1987) generalized Clark's approach to $Ph(t)/M(t)/s/K$ systems with a phase arrival process and referred to the general approach as a *surrogate distribution approach*. Taaffe and Ong's approach, when specialized to $M(t)/M/s(t)$ systems, is slightly different from Clark's original approach. We implemented both approaches, and found Taaffe and Ong's approach to be more accurate and faster. In the Online Supplement, we describe the details, modified to allow for infinite system capacity. This approach uses a Polya Eggenberger (PE) closure distribution (Johnson et al. 1993).

We found two aspects of the implementation of this method to be crucial: ensuring valid parameter values for the PE distribution (see the Online Supplement) and determining how to "restart" the method at epochs when the number of servers $s(t)$ changes. The quantities integrated under Taffee and Ong's approach are "partial moments" (see the Online Supplement for definitions). The definitions of the partial moments change when the number of servers changes. Suppose the number of servers changes at epoch t. It seems natural to convert the final values of the partial moments with the previous number of servers (just before epoch t) into an approximate state probability distribution using the assumed closure distribution, apply an instantaneous transition as in (1), if appropriate, calculate new partial moments using their definition and the new number of servers, and use these partial moments as initial values at epoch t. However, experimentation indicated that this increased computation time and reduced accuracy. Simply using the final partial moments with the previous number of servers as initial conditions, when the numerical integration is restarted with a new number of servers, worked much better. Figure 1 in the Online Supplement illustrates this.

The benefit of closure approximations is that the number of differential equations to be solved remains constant regardless of the system parameters, whereas the exact approach requires the solution of K differential equations, and the value of K needed to approximate an infinite capacity system adequately varies depending on the system parameters. The disadvantage is that the forward equations (a set of linear differential equations) are replaced by a set of nonlinear differential equations. The equations are nonlinear because the probabilities for states $s(t) - 1$ and $s(t)$ must be expressed as functions of the variables being

integrated (the partial moments), in order to "close" the set of differential equations. The resulting expressions for $\pi_{s(t)-1}(t)$ and $\pi_{s(t)}(t)$ are nonlinear. Although the number of equations to be integrated remains constant, the computational effort to solve them depends on system parameters, as our results show.

3.4. Infinite-Server Approximations

The closure approximations described in Section 3.3 start with the forward equations and derive from them a smaller set of differential equations. The approaches described in this section can also be viewed as closure approximations, but they involve an additional level of approximation, whereby an $M(t)/M/s(t)$ system is approximated with an $M(t)/M/\infty$ system. This leads to a simple differential equation for the expected number of busy servers in the system (Eick et al. 1993a):

$$\mathrm{E}[B(t)]' = \lambda(t) - \mu \mathrm{E}[B(t)]. \tag{7}$$

We used the ode45 ODE solver for this, and its solution served as the basis for the two approximations described in this section. Note that (7) is already closed—no knowledge of the state probabilities is required to solve it.

In an infinite-server system, the number of customers in the system equals the number of busy servers, so (7) holds with $B(t)$ replaced by $N(t)$. These two viewpoints lead to different approximations. The number of customers in an $M(t)/M/\infty$ system follows a Poisson distribution with mean determined by (7), if the system started empty in the distant past (Eick et al. 1993a) so if one views (7) as describing the evolution of the mean number in system, then one can use a Poisson PMF to approximate the state probabilities and then calculate service levels using (4) (or (5) when appropriate). We refer to this as a *direct infinite-server approximation*.

In contrast, viewing (7) as describing the number of busy servers motivates a different choice of distribution to approximate $\pi(t)$. In a stationary $M/M/s$ system, Little's law applied to the servers obtains the expected number of busy servers as λ/μ. The MOL approximation uses the stationary distribution for an $M/M/s$ system to approximate $\pi(t)$, with the number of busy servers chosen to match $\mathrm{E}[B(t)]$ as obtained from solving (7). The matching is done by solving $\lambda/\mu = \mathrm{E}[B(t)]$ for λ, which gives $\lambda = \mathrm{E}[B(t)]\mu$. Thus, we approximate $\pi(t)$ with the stationary distribution for an $M/M/s$ system with arrival rate $\mathrm{E}[B(t)]\mu$, service rate μ, and $s(t)$ servers (or equivalently, arrival rate $\mathrm{E}[B(t)]$, service rate 1, and $s(t)$ servers, since this stationary distribution is insensitive to multiplication of both the arrival and service rates with the same constant). The MOL approximation was originally used for $M(t)/M/s/s$ loss systems (Jagerman 1975) and has been used for $M(t)/M/s$ systems in Massey and Whitt (1997).

The MOL approximation can be expected to work best when utilization is low enough that the solution to (7) provides a good approximation to the number of busy servers over time. Note that (7) allows solutions where $\mathrm{E}[B(t)] > s(t)$ and when this happens, the MOL approximation can be expected to be poor. When the utilization $\lambda(t)/(s(t)\mu)$ exceeds 100%, the MOL approximation may break down in the sense that the approximating stationary system is unstable. When this happened, we set the service level to zero.

As shown in Eick et al. (1993a), the expected number of busy servers in an $M(t)/G/\infty$ system is

$$\mathrm{E}[B(t)] = \int_{-\infty}^{t} G^c(t-u)\lambda(u)\,du, \tag{8}$$

where $G^c(u)$ is the complementary service-time distribution. The MOL approximation uses $\mathrm{E}[B(t)]\mu$ as an arrival rate for a stationary $M/M/s$ system. One can view this quantity as an "effective arrival rate" $\lambda_{\mathrm{eff}}(t)$, as suggested by Sze (1984). By specializing (8) to $M(t)/M/\infty$ systems, the effective arrival rate can be expressed as an exponentially weighted moving average of the arrival rate:

$$\lambda_{\mathrm{eff}}(t) = \mathrm{E}[B(t)]\mu = \int_{-\infty}^{t} \mu e^{-\mu(t-u)}\lambda(u)\,du. \tag{9}$$

This will help put the next approximation method we discuss into context.

3.5. Effective-Arrival-Rate Approximation

Thompson (1993) developed an effective-arrival-rate approximation to generate staffing requirements that take linkages between planning periods into account. Like the MOL approximation, this method develops a time-varying effective arrival rate and uses it, along with the service rate and the number of servers, as input to a stationary $M/M/s$ model.

We present a slightly modified version of Thompson's approximation, which is simpler to implement. Thompson assumes the arrival rate $\lambda(t)$ to be constant over each planning period. His method is designed to generate an effective arrival rate $\lambda_{\mathrm{eff}}(t)$ that is also constant over each planning period. This requirement is natural when the method is used to generate staffing requirements, but is not necessary in performance evaluation. Allowing the effective arrival rate to vary continuously simplifies presentation of the method and facilitates comparison to the infinite-server approximations.

The method is based on two simplifying assumptions: (1) service times are deterministic and (2) waiting times in queue are deterministic and denoted Wq. Thompson suggested estimating the waiting time in

queue Wq as the expected wait in queue in a stationary $M/M/s$ system with arrival rate $\bar{\lambda}$, service rate μ, and number of servers $\tilde{s}$, where $\tilde{s}$ is chosen as the minimum number of servers required to provide a prespecified level of service in this stationary system. We want to use the method to *predict* service levels, *given* the number of servers, so we let $\tilde{s}$ be the rounded-up time-average number of servers $\lceil \bar{s} \rceil$.

Under the two simplifying assumptions, the customers in service at time t will be the ones that arrived during the interval $[t - Wq - 1/\mu, t - Wq]$. The effective arrival rate is then calculated as the average of the true arrival rate during this interval:

$$\lambda_{\text{eff}}(t) = \int_{t-Wq-1/\mu}^{t-Wq} \mu\lambda(r)\, dr. \tag{10}$$

Comparing (9) and (10), the effective arrival rate under the MOL approximation is an exponentially weighted moving average of the arrival rate over the interval $(-\infty, t]$, whereas the effective arrival rate under the present approximation is a moving average over the window $[t - Wq - 1/\mu, t - Wq]$.

Like the MOL approximation, this method may result in an effective arrival rate that is larger than μ times the number of servers. When this happened, we set the service level to zero.

3.6. Lagged Stationary Approximation

Green et al. (2001, 2003) evaluated the reliability of a commonly used approach to generate staffing requirements they refer to as *stationary independent period by period* (SIPP). With this approach, the average arrival rate over a planning period $((i-1)\delta_p, i\delta_p)$ and the service rate are used as input to a stationary $M/M/s$ model to determine the minimum number of servers needed in the planning period to provide a specified level of service. By comparison with the numerical solution of an $M(t)/M/s(t)$ model, they found that this often recommends staffing levels that are insufficient to guarantee the specified level of service, but a pair of simple modifications replacing the average arrival rate by a lagged maximum or lagged average of the arrival rate worked much better. Green et al. (2001, 2003) referred to these two approaches as *Lag Max* and *Lag Avg*. We refer to them, together with guidelines from Green et al. (2003) for when to use each approach, as the *lagged stationary approximation*.

Lag Max and Lag Avg can be considered in the same framework as the effective-arrival-rate approximation, with the effective arrival rates

$$\text{Lag Avg: } \lambda_{\text{eff}}(t) = \frac{1}{\delta_p} \int_{(i-1)\delta_p - 1/\mu}^{i\delta_p - 1/\mu} \lambda(r)\, dr \quad \text{for } t \in ((i-1)\delta_p, i\delta_p]$$

$$\text{Lag Max: } \lambda_{\text{eff}}(t) = \max\{\lambda(u) \mid (i-1)\delta_p - 1/\mu \le u \le i\delta_p - 1/\mu\} \quad \text{for } t \in ((i-1)\delta_p, i\delta_p].$$

In words, we shift the planning period one average service time into the past and take either the average or maximum of the arrival rate during the resulting interval. Consequently, the effective arrival rate is piecewise constant over each planning period and so is the approximated service level. The lag of one average service time was chosen because it approximates the time lag between a peak in the arrival rate and the resulting peak in the expected number in system, when the number of servers is constant. Green et al. (2003) recommend the use of Lag Avg when the relative amplitude of the arrival-rate function is low (i.e., when the maximum arrival rate is at most 50% higher than the average arrival rate) and planning periods are short (i.e., 0.25 or 0.5 hour), and Lag Max in all other situations.

4. Experimental Design

We applied the seven methods to a set of 640 test problems. We included a sufficiently wide range of conditions to permit inferences regarding the relative merits of the methods. In this section, we discuss how we specified $\lambda(t)$, $s(t)$, and other problem parameters. Then we describe the combinations of parameter values that specified our test problems. Finally, we describe how we measured the accuracy and speed of each method.

Service systems often experience cyclical demand, with cycles recurring on various time scales, for example, daily, weekly, monthly, and yearly. We focus on daily cycles because the amplitude of demand variation is often greatest on this time scale. All of our experiments assume the facility operates continuously for $T = 24$ hours per day.

Similarly to Green et al. (1991) and Eick et al. (1993b), we used a sinusoidal arrival-rate function with a 24-hour cycle, $\lambda(t) = \bar{\lambda}(1 + \alpha \sin(2\pi t/24))$ where $\bar{\lambda}$ is the average arrival rate and $\alpha \in [0, 1]$ is the relative amplitude.

We assume that the system will reach *periodic steady state*, which Heyman and Whitt (1984) discuss and prove that the periodic $M(t)/M/s$ queue reaches periodic steady state (it remains to be proven that the $M(t)/M/s(t)$ queue with exhaustive discipline reaches periodic steady state). To approximate the periodic steady state, we evaluate performance for n days, until performance on day $n-1$ is sufficiently similar to performance on day n. Specifically, we continue to compute service levels until $|\text{SL}(24(n-1)+t) - \text{SL}(24n+t)| \le 0.01$ for all $t \in (0, 24]$.

Ideally, the number of servers over time will be chosen to match the variation in demand. However,

the match may be less than ideal for various reasons, including (1) cost savings from reducing variability in staffing, (2) an upper limit on staffing, (3) limitations on when servers can begin and end work, and (4) lack of planning. We are interested in the performance of these methods for situations where the number of servers is not necessarily well matched to the arrival rate for two reasons: (1) such situations occur in reality, and (2) a scheduling algorithm that calls one of the methods we are comparing as a subroutine may need to do so for some "poor" schedules on its way to the "optimal" schedule.

We model the number of servers as a sinusoidal function discretized to be constant over each planning period and may have a phase shift relative to the arrival-rate function. Thus, we start with the continuous function $s_c(t) = \bar{s}_c(1 + \beta \sin(2\pi(t-\gamma)/24))$ where β is the relative amplitude, γ is a phase shift, and $\bar{s}_c$ is the average number of servers. The number of servers peaks γ time units after the arrival rate peaks. We included the phase shift to create test problems where variation in the number of servers is poorly matched to variation in the arrival rate, to challenge the methods. Then, for each planning period $((i-1)\delta_p, i\delta_p]$, we replace $s_c(t)$ with its rounded-up average $s(t) = \lceil \int_{(i-1)\delta_p}^{i\delta_p} s_c(u)\,du/\delta_p \rceil$. We set the number of servers scheduled to leave at the end of a planning period to $\max(s(t^-) - s(t^+), 0)$, the amount by which the scheduled number of servers decreases. This assumption is easily relaxed (see Section 2).

Table 1 shows the parameters we varied and their values. All combinations of these values resulted in 640 different test problems. The average utilization together with the average offered load determined the average number of servers $\bar{s}_c = r/\bar{\rho}$, which ranged from 2.1 to 64.

We calculated service levels at five-minute intervals, starting at time zero and ending at time 24, so the output from each method for each test problem was a service-level vector with 289 elements. We calculated errors by comparing to the exact method, and we measured the CPU time used by each method for each test problem. We calculated time averages and maxima of the errors for each test problem and each method.

Table 1 Parameter Values for Computational Experiments

Factor	Low value	High value
Service rate (μ)	2/hour	32/hour
Offered load ($r = \bar{\lambda}/\mu$)	2	32
Arrival rate relative amplitude (α)	0.1	0.9
Servers relative amplitude (β)	0.1	0.9
Average utilization ($\bar{\rho}$)	0.5	0.95
Phase shift (γ)	0 hours	3 hours
Planning period (δ_p)	0.5 hours	8 hours
Maximum acceptable waiting time (τ)	$0, 0.25/\mu, 0.5/\mu, 0.75/\mu, 1/\mu$	

In calculating errors, it is important to consider both how computed service levels are used and the imperfections of our standard—the exact method. Policy goals for service level are often expressed in the form "at least X percent of customers should wait less than τ time units." For example, "80% in 20 seconds" is a common service level goal for call centers. When performance-evaluation methods are used to compare the estimated consequences of different plans, it is typically more important to know whether, say, 79% or 80% wait less than τ (because this could be the difference between meeting a goal or not) than it is to know whether 4% or 5% wait less than τ (in either case, the goal is far from being met). These considerations suggest the use of the *relative* error for the *complementary* service level, because this measure magnifies errors when the complementary service level is close to zero, i.e., when the service level is close to one. Letting $\overline{\mathrm{SL}}_{\mathrm{EXT}}(t)$ be a value for the complementary service level computed by the exact method and $\overline{\mathrm{SL}}_{\mathrm{APPROX}}(t)$ a value computed by some other method, the relative error is

$$\frac{|\overline{\mathrm{SL}}_{\mathrm{EXT}}(t) - \overline{\mathrm{SL}}_{\mathrm{APPROX}}(t)|}{\overline{\mathrm{SL}}_{\mathrm{EXT}}(t)}. \tag{11}$$

Unfortunately, the relative error can sometimes be misleading. As we illustrate in the Online Supplement, there can be substantial relative differences in situations when the complementary service level is close to zero that, although unlikely to be important in practice, will be magnified by the relative error formula (11). We contend that when the absolute error is small, the relative error is not important, even when it is large. The limits on the accuracy of the exact method are another reason not to assign much importance to situations where the absolute error is small but the relative error is large. As illustrated in the Online Supplement, when the complementary service level is below the default "absolute error tolerance" of 10^{-6}, the computed values from the exact method do not look plausible and equal zero in some cases. We modified the relative error calculation as follows, to reduce the impact of these difficulties:

$$\frac{\max(0, |\overline{\mathrm{SL}}_{\mathrm{EXT}}(t) - \overline{\mathrm{SL}}_{\mathrm{APPROX}}(t)| - 10^{-3})}{\overline{\mathrm{SL}}_{\mathrm{EXT}}(t) + 10^{-3}}. \tag{12}$$

As this expression shows, we ignore the relative error (that is, set it to zero) when the absolute error is less than one in a thousand, and we limit the magnification of the absolute error to be at most thousand-fold (by adding 10^{-3} to the denominator). Absolute errors of less than one in a thousand seem unlikely to be important in practice and magnifying absolute errors

Table 2 Summary of Results, by Wait Threshold (WT or τ)

	Method						
WT	EXT	RND	CLS	ISA	MOL	EAR	LST
			Computation time: median (sec.)				
0	31.27	14.67	29.53	1.06	0.97	0.62	0.71
$0.25/\mu$	31.27	14.67	29.60	1.07	0.93	0.62	0.70
$0.50/\mu$	31.27	14.67	29.18	1.05	0.93	0.62	0.70
$0.75/\mu$	31.27	14.67	28.72	1.07	0.93	0.62	0.70
$1.00/\mu$	31.27	14.67	28.08	1.05	0.93	0.62	0.70
			Computation time: mean (sec.)				
0	2,985.0	258.5	4,874.1	1.17	0.99	0.56	0.65
$0.25/\mu$	2,985.1	258.6	4,772.6	1.17	0.92	0.56	0.65
$0.50/\mu$	2,985.1	258.6	4,816.1	1.05	0.89	0.56	0.65
$0.75/\mu$	2,985.1	258.5	4,630.2	1.03	0.89	0.56	0.65
$1.00/\mu$	2,984.6	258.2	4,656.7	1.00	0.91	0.56	0.65
			Time-average relative error: median				
0	N/A	0%	40%	32%	16%	16%	35%
$0.25/\mu$	N/A	0%	20%	42%	50%	52%	69%
$0.50/\mu$	N/A	0%	11%	50%	67%	66%	86%
$0.75/\mu$	N/A	0%	16%	56%	71%	72%	99%
$1.00/\mu$	N/A	0%	18%	58%	74%	73%	122%
			Time-average relative error: mean				
0	N/A	0.021%	1,132%	43%	38%	37%	1,304%
$0.25/\mu$	N/A	0.011%	52%	52%	76%	75%	1,006%
$0.50/\mu$	N/A	0.006%	48%	55%	115%	115%	1,174%
$0.75/\mu$	N/A	0.004%	48%	56%	158%	160%	1,359%
$1.00/\mu$	N/A	0.002%	54%	55%	225%	231%	1,584%
			Maximum relative error: median				
0	N/A	0%	287%	94%	116%	113%	236%
$0.25/\mu$	N/A	0%	104%	96%	171%	168%	489%
$0.50/\mu$	N/A	0%	100%	98%	269%	262%	543%
$0.75/\mu$	N/A	0%	110%	98%	354%	354%	735%
$1.00/\mu$	N/A	0%	116%	98%	520%	535%	942%
			Maximum relative error: mean				
0	N/A	0.316%	5,108%	4,024%	4,610%	4,537%	13,048%
$0.25/\mu$	N/A	0.179%	1,917%	3,284%	6,501%	6,468%	12,400%
$0.50/\mu$	N/A	0.114%	1,546%	2,565%	7,680%	7,664%	13,677%
$0.75/\mu$	N/A	0.081%	1,227%	1,898%	8,883%	8,943%	15,632%
$1.00/\mu$	N/A	0.058%	1,138%	1,472%	10,556%	10,690%	18,115%

by more than one thousand does not seem likely to produce useful information. All errors reported in the next section are relative errors, computed using (12).

5. Computational Results

In this section, we will sometimes use abbreviations for the seven methods: EXT for the exact method, RND for the randomization method, CLS for the closure approximation, ISA for the direct infinite-server approximation, MOL for the modified-offered-load approximation, EAR for the effective-arrival-rate approximation, and LST for the lagged stationary approximation. All computations were performed on a 900 MHz Unix workstation.

Table 2 shows summary statistics (means and medians) for computation times, time-average relative errors, and maximum relative errors. The statistics are shown separately for each value of the maximum acceptable waiting time τ. Figure 1 uses box-and-whisker plots on a logarithmic scale to summarize the distribution of computation times for each method across the 640 test problems. Each box-and-whisker plot is centered on the median computation time per test problem, which ranged from 0.6 second for the EAR approximation to 30 seconds for the EXT method. RND had a median computation time of 14 seconds, for about 55% savings over the exact method—comparable to the 75% savings observed by Reibman and Trivedi (1988) for homogenous test problems.

Figure 2 summarizes the distribution of time-average relative errors, compared to the exact method, for the six approximation methods. The RND method is always most accurate but the ranking of the other

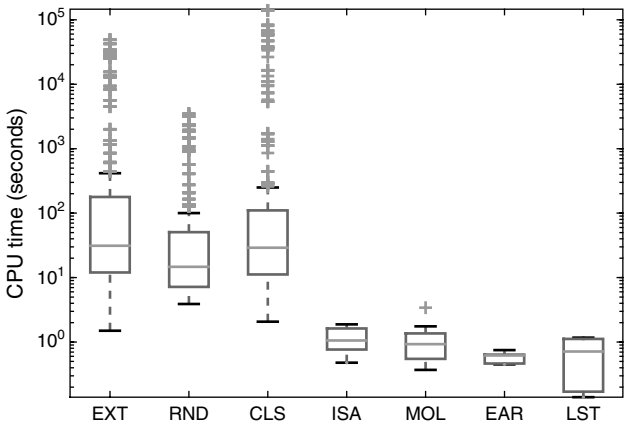

Figure 1 Box-and-Whisker Plots of the Distribution of Computation Times for the Seven Methods Across 640 Test Problems

Notes. EXT, exact method; RND, randomization; CLS, closure approximation; ISA, direct infinite-server approximation; MOL, modified-offered-load approximation; EAR, effective-arrival-rate approximation; LST, lagged stationary approximation.

methods depends on whether τ is zero or positive. When $\tau = 0$, using the median time-average relative error as a yardstick, the methods are ranked as follows, in order of decreasing accuracy (see Table 2): RND, MOL/EAR, ISA, LST, CLS. RND has zero relative error in most cases, MOL and EAR are off by about 16%, ISA by about 32%, LST by about 35%, and CLS by 40%. When $\tau > 0$, the ranking changes to RND, CLS, ISA, MOL/EAR, LST. RND again has zero relative error in most cases, CLS is off by 10%–20%, ISA by 42%–58%, MOL/EAR by 50%–74%, and LST by 69%–122%. The rankings change if one uses means instead of medians or maxima instead of time-averages as yardsticks, but the following patterns emerge: RND is always most accurate, MOL and EAR always have similar accuracy, ISA is always third best or better, and LST always has the worst or second-worst accuracy.

We investigated the accuracy of RND further by taking a closer look at test problems where it was least accurate. Figure 4 in the Online Supplement shows the results of applying the exact method and the randomization method to a test problem where RND resulted in a time-average relative error of 0.45%. RND resulted in time-average relative error of less than 0.4% for all other test problems, so this Figure illustrates the worst-case accuracy for RND, among the problems we solved. Even in the worst case, it is difficult to distinguish between the two service level curves visually.

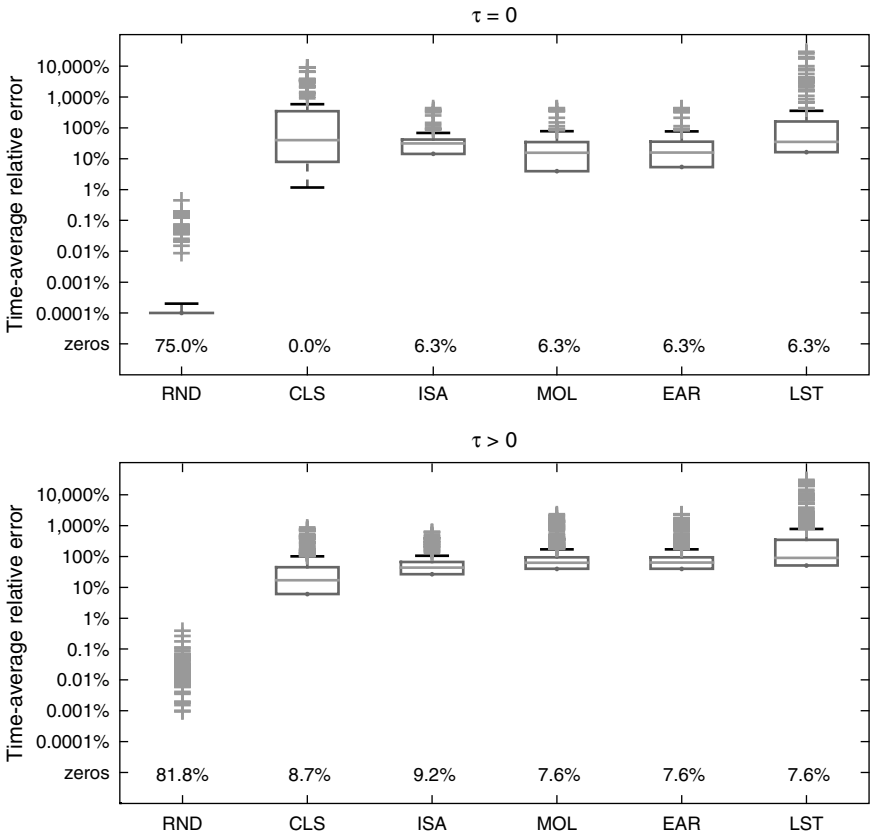

Figure 2 Box-and-Whisker Plots of the Distribution of Time-Average Relative Errors for Six Approximation Methods (Using the Exact Method as a Standard) for $\tau = 0$ (128 Test Problems) and $\tau > 0$ (512 Test Problems)

Note. The fraction of test problems with zero time-average relative error is shown separately because zeros cannot be displayed on the logarithmic scale.

The ISA, MOL, EAR, and LST approximations have significantly lower and more consistent computation times than the EXT, RND, and CLS methods. Of the former four, the EAR and LST approximations are consistently faster than the ISA and MOL methods, and the ISA approximation was usually the most accurate (in 409 of 640 cases, based on time-average relative error). LST and EAR's speed advantage was fairly large in relative terms (about 30%, on average) but small in absolute terms (never more than 3.5 seconds). ISA was slower than the other three methods because it requires approximation of the service level through truncation of the series in (4), while MOL, EAR, and LST use a closed-form expression (from the steady-state $M/M/s$ model) to evaluate the service level.

We examined the impact of the individual factors and their interactions on the computation time and accuracy of each method. Two factors, the service rate μ and the average offered load $r = \bar{\lambda}/\mu$, have by far the largest impact on computation time, for all methods. Figure 3 illustrates this for the exact method. The computation times for this method are on the order of 4 seconds when $\mu = r = 2$, on the order of 40 seconds when one of μ or r is 32 and the other is 2, and on the order of 4,000 seconds when $\mu = r = 32$. The other methods exhibit similar behavior, except that computation times grow more slowly with increasing service rate or increasing average offered load.

The average offered load is a useful measure of system size (it equals the average number of busy servers for stationary $M/M/s$ systems) and the higher the average offered load, the larger the system capacity K needed by EXT and RND to approximate adequately an $M(t)/M/s(t)$ infinite capacity system. The service rate is an indication of the *event frequency* (Green et al. 1991), i.e., the total frequency of arrivals and service completions. With our parameterization, the event frequency can be expressed as $\lambda(t)+\mu s(t) = \mu(r(t)+s(t))$ where $r(t) = \lambda(t)/\mu$, i.e., the event frequency is proportional to the service rate for fixed $r(t)$ and $s(t)$. The higher the event frequency, the faster the number in system will grow during time intervals when the utilization is close to or above 100%. Hence, higher service rates cause the system capacity K required by EXT and RND to be higher. This explains why computation time for the EXT and RND methods increases with average offered load and service rate.

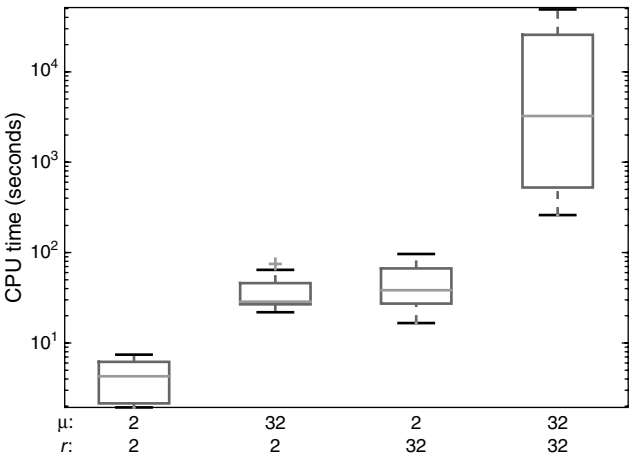

Figure 3 **Distribution of Calculation Times for the Exact Method as a Function of the Service Rate μ and the Average Offered Load r**

Overall, the performance of CLS was disappointing. For 42 of 640 test problems, CLS was dominated, i.e., it was both the slowest and the least accurate, as measured by time-average relative error. No other method was dominated in this sense, for any test problem. CLS did especially poorly when τ equaled zero, being dominated in 31 of 128 test problems. In one test problem, we terminated CLS after it failed to converge to periodic steady after 40 hours of computation time.

With the exception of CLS, the methods generally needed a similar number of cycles to converge to periodic steady state. RND always needed the same number of cycles to converge as EXT. MOL and ISA needed at most one more or one fewer cycles to converge than EXT, for all but one test problem, where MOL required five more cycles. EAR and LST require only the evaluation of one cycle, as mentioned in the Online Supplement.

Given the speed advantage of the MOL approximation, we were interested in identifying domains where this method provided acceptable accuracy. As Table 2 shows, MOL is more accurate when $\tau = 0$. Besides τ, the average utilization and the relative amplitude of the server function seem to be the primary determinants of the accuracy of MOL. When both of these factors took their low values (0.5 and 0.1, respectively) and $\tau = 0$, the time-average relative error for MOL was 2.2% on average and 12% in the worst case. The influence of the relative amplitude of the server function on the accuracy of MOL is consistent with the observations made in the Online Supplement about systematic errors from the MOL approximation just after the number of servers changes.

The Online Supplement contains additional computational results, including a comparison of "balanced systems" (where $\alpha = \beta$ and $\gamma = 0$) and unbalanced systems, an analysis of whether a more sophisticated implementation of the randomization method might reduce computation time, and an illustration of systematic errors with the MOL and EAR methods just after epochs when the number of servers changes.

6. Conclusions

Before conducting the experimental comparison of the seven methods, we expected that they would fall into three categories: (1) methods that are slow but highly accurate, (2) methods that are fast but often inaccurate, and (3) an intermediate category with modest computation times and reasonably good accuracy. We expected the exact and randomization methods (EXT and RND) to fall in the first category; the infinite-server-based approximations (ISA and MOL), the effective-arrival-rate approximation (EAR), and the lagged stationary approximation (LST) to fall in the second category; and the closure approximation (CLS) to fall in the third category. Our expectations were generally confirmed, with some notable exceptions. In particular, the closure approximation performed worse than expected, being both the slowest and least accurate method for 42 of the 640 test problems and failing to converge in one problem.

Two caveats regarding the poor performance of the closure approximation are in order. First, the method is designed to approximate the mean number in system, not the service level. The literature on closure approximations shows them to be less accurate in approximating the variance and the second moment of the number in system (Rothkopf and Oren 1979, Clark 1981, Taaffe and Ong 1987) and our experiments indicate that they are even less accurate in approximating the service level, which depends on tail probabilities for the number in system. Second, the closure approximation has a computational advantage over the exact method for systems that require a large system capacity to approximate the system sufficiently well. We did not see such an advantage over the randomization method for the test problems we solved, but closure approximations may be faster than randomization for problems that are larger than the ones we solved.

In the "slow but accurate" category, we demonstrated that the randomization method provides essentially the same results as the exact method with half the computational cost, even though application of the randomization method requires approximation of the arrival rate with a piecewise-constant function. The speed advantage of randomization over the exact method is consistent, except for systems with low service rates, low arrival rates, and long planning periods, where the exact method is consistently faster than randomization.

In the "fast-but-sometimes-inaccurate" category, the modified-offered-load approximation is, on average, the most accurate of the methods. The lagged stationary approximation is faster, but the speed advantage is small in absolute terms. The direct infinite-server approximation may be useful in some applications to generate upper bounds quickly on the service level. The effective-arrival-rate approximation performed similarly to the modified-offered-load approximation. The accuracy of the lagged stationary approximation should be viewed in light of the fact that it was designed for a pre-emptive discipline, but we compared it to a model with an exhaustive discipline.

Acknowledgments

This research was supported by Grant 203534 from the Natural Sciences and Engineering Research Council of Canada. Geert Jan Franx and Ger Koole contributed to the modeling of end-of-shift effects in Section 2 and made other useful suggestions. The authors thank the referee and Associate Editor for constructive comments that significantly improved this paper.

References

Arkin, B. L., L. M. Leemis. 2000. Nonparametric estimation of the cumulative intensity function for a nonhomogeneous Poisson process from overlapping realizations. *Management Sci.* **46** 989–998.

Atlason, J., M. A. Epelman, S. G. Henderson. 2004. Call center staffing with simulation and cutting plane methods. *Ann. Oper. Res.* **127** 333–358.

Barnhart, C., P. Belobaba, A. R. Odoni. 2003. Applications of operations research in the air transport industry. *Transportation Sci.* **37** 368–391.

Bookbinder, J. H. 1986. Multiple queues of aircraft under time-dependent conditions. *INFOR* **24** 280–288.

Bookbinder, J. H., D. L. Martell. 1979. Time-dependent queuing approach to helicopter allocation for forest fire initial-attack. *INFOR* **17** 58–72.

Clark, G. M. 1981. Use of Polya distributions in approximate solutions to nonstationary $M/M/s$ queues. *Comm. ACM* **24** 206–217.

Cleveland, B., J. Mayben. 1997. *Call Center Management on Fast Forward*. Call Center Press, Annapolis, MD.

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

Eick, S. G., W. A. Massey, W. Whitt. 1993b. The physics of the $M_t/G/\infty$ queue. *Oper. Res.* **41** 731–742.

Grassmann, W. K. 1977. Transient solutions in Markovian queueing systems. *Comput. Oper. Res.* **4** 47–53.

Grassmann, W. 1989. Numerical solutions for Markovian event systems. *Quantitative Methoden in den Wirtschaftswissenschaften*. Springer-Verlag, Berlin, Germany, 73–87.

Grassmann, W. K., ed. 2000. *Computational Probability*. Kluwer Academic Publishers, Boston, MA.

Green, L. V., P. J. Kolesar. 1991. The pointwise stationary approximation with nonstationary arrivals. *Management Sci.* **37** 84–97.

Green, L. V., J. Soares. 2007. Computing time-dependent waiting time probabilities in $M(t)/M/s(t)$ queueing systems. *Manufacturing Service Oper. Management* **2** 54–61.

Green, L. V., P. J. Kolesar, J. Soares. 2001. Improving the SIPP approach for staffing service systems that have cyclic demands. *Oper. Res.* **49** 549–564.

Green, L. V., P. J. Kolesar, J. Soares. 2003. An improved heuristic for staffing telephone call centers with limited operating hours. *Production Oper. Management* **12** 1–16.

Green, L. V., P. J. Kolesar, A. Svoronos. 1991. Some effects of nonstationarity on multiserver Markovian queueing systems. *Oper. Res.* **39** 502–511.

Gross, D., D. R. Miller. 1983. The randomization technique as a modeling tool and solution procedure for transient Markov processes. *Oper. Res.* **32** 343–361.

Heyman, D. P., W. Whitt. 1984. The asymptotic behavior of queues with time-varying arrival rates. *J. Appl. Probab.* **21** 143–156.

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://www.bus.ualberta.ca/aingolfsson/working_papers.htm.

Ingolfsson, A., E. Cabral, X. Wu. 2002a. Combining integer programming and the randomization method to schedule employees. Research Report 02-1, Department of Finance and Management Science, Faculty of Business, University of Alberta, Edmonton, Alberta, Canada.

Ingolfsson, A., M. A. Haque, A. Umnikov. 2002b. Accounting for time-varying queueing effects in tour scheduling. *Eur. J. Oper. Res.* **139** 585–597.

Jagerman, D. L. 1975. Nonstationary blocking in telephone traffic. *Bell Systems Tech. J.* **54** 625–661.

Jennings, O. B., W. A. Massey. 1997. A modified offered load approximation for nonstationary circuit switched networks. *Telecomm. Systems* **7** 229–251.

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

Jensen, A. 1953. Markoff chains as an aid in the study of Markoff processes. *Skandinavisk Aktuarietidskrift* **36** 87–91.

Johnson, N. L., S. Kotz, A. W. Kemp. 1993. *Univariate Discrete Distributions*. Wiley, New York.

Kleinrock, L. 1974. *Queueing Systems, Volume 1: Theory*. Wiley, New York.

Koopman, B. O. 1972. Air terminal queues under time-dependent conditions. *Oper. Res.* **20** 1089–1114.

Leese, E. L., D. W. Boyd. 1966. Numerical methods of determining the transient behaviour of queues with variable arrival rates. *INFOR* **4** 1–13.

Massey, W. A., W. Whitt. 1997. Peak congestion in multi-server systems with slowly varying arrival rates. *Queueing Systems* **25** 157–172.

Massey, W. A., G. A. Parker, W. Whitt. 1996. Estimating the parameters of a nonhomogeneous Poisson process with linear rate. *Telecomm. Systems* **5** 361–388.

Odoni, A. R., E. Roth. 1983. An empirical investigation of the transient behavior of stationary queueing systems. *Oper. Res.* **31** 432–455.

Reibman, A., K. Trivedi. 1988. Numerical transient analysis of Markov models. *Comput. Oper. Res.* **15** 19–36.

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

Shampine, L. F., M. W. Reichelt. 1997. The Matlab ODE suite. *SIAM J. Sci. Comput.* **18** 1–22.

Stewart, W. J. 1994. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, Princeton, NJ.

Sze, D. Y. 1984. A queueing model for telephone operator staffing. *Oper. Res.* **32** 229–249.

Taaffe, M. R., K. L. Ong. 1987. Approximating nonstationary $Ph(t)/M(t)/s/c$ queueing systems. *Ann. Oper. Res.* **8** 103–116.

Thompson, G. M. 1993. Accounting for the multi-period impact of service when determining employee requirements for labor scheduling. *J. Oper. Management* **11** 269–287.

Whitt, W. 1991. The pointwise stationary approximation for $M_t/M_t/s$ queues is asymptotically correct as the rates increase. *Management Sci.* **37** 307–314.

Wragg, A. 1963. The solution of an infinite set of differential-difference equations occurring in polymerization and queueing problems. *Proc. Cambridge Philos. Soc.* **59** 117–124.