Predicting Departure Times in Multi-Stage Queueing Systems

Kevin R. Gue and Hyun Ho Kim

Department of Industrial & Systems Engineering
Auburn University
Auburn, Alabama 36849
kevin.gue@auburn.edu, sfhyunhokim@gmail.com

Abstract

We develop an approximation model for the state-dependent sojourn time distribution of customers or orders in a multi-stage, multi-server queueing system, when interarrival and service times can take on general distributions. The model can be used to make probabilistic statements about the departure time of a customer or order, given the number and location of customers currently in process or waiting, and these probabilities can be recomputed while waiting at any point during the sojourn time. The model uses phase-type distributions and a new method to estimate the remaining processing times of customers in service when the sojourn time distribution is computed.

Keywords: Queueing, Markov processes, phase-type distributions, sojourn time distributions

1. Motivation

Service systems often make explicit statements—perceived by customers, effectively, as promises—about how long a wait will be. For example, call centers may claim, "expected wait time is 3 minutes," or an internet bookseller may promise, "order in the next 2 hours and receive it tomorrow." To make such promises, a firm must estimate both processing times and waiting times, whose sum is the sojourn time. In stochastic environments, the result must be a distribution, from which service estimates or promises are made. When offered or called upon in real time, these service promises must also consider the number of customers already in the system, that is, the system *state*. In the examples above, the state of the system is defined by other callers or orders already in queue.

The relevance of state-dependent waiting times for call centers is well-recognized (Whitt, 1999). These problems are typically modeled with a single, multi-server queue, but many service systems require a multi-stage representation. For example, the motivation for our research is a large distribution center, which operates as a series of multi-server queues corresponding to the picking, packing, and shipping operations. To make real-time decisions about workforce allocation, we sought the probability that a particular order would complete its processing before the last truck departed for the day (Kim, 2009). This probability can only be computed with a state-dependent sojourn time distribution.

Models such as the one we develop here seem all the more important when one considers that simulation is an ineffective means of developing these distributions. For general service times (which we assume), the state of the system must include not only queue lengths and numbers of busy servers, but also the remaining processing time for each order being served. To our knowledge, there is no accepted method of simulating the "remaining times" of orders already in process. The brute force simulation method, which we describe and use below, observes the state of the system upon each arrival and records results only for orders encountering the system state of interest. Such simulations can take hours or even days to generate a single distribution. We believe our models offer a better approach.

Research on sojourn time distributions can roughly be classified into two categories: steady state waiting time and state-dependent waiting time. A steady state sojourn time distribution only has meaning when a customer or order arrives to the system, whereas a state-dependent sojourn time distribution can be computed at any time while in the system, not just upon arrival. Our interest in this paper is a state-dependent sojourn time distribution, which includes the waiting time and service time distributions.

There is a rich literature on steady state sojourn time distributions. Neuts (1981) and Luh and Zheng (2005) showed how to generate the sojourn time distribution of a single stage queue with a single server using a matrix geometric method. Sengupta (1989) developed a "continuous analog" of the matrix-geometric method. Asmussen and O'Cinneide (1998) and Asmussen and Møller (2001) extended Sengupta's analysis of the GI/PH/1 queue to the multi-server case. They showed the steady state waiting time distribution in a GI/PH/c queue is also phase-type.

Shanthikumar and Sumita (1988) and You et al. (2002) suggested an approximation model for queueing networks of single servers. Shanthikumar and Sumita (1988) approximated the sojourn time distribution as a phase-type distribution based on the "service index," and You et al. (2002) introduced an approximation using the convolution property of the phase-type distribution. Steady state sojourn time distributions for queueing networks of multiple servers was studied by Mandelbaum et al. (1998) and Gue and Kim (2009). Mandelbaum et al. (1998) addressed diffusion approximations for M/M/c queueing networks, and Gue and Kim (2009) developed an approximation model for G/G/c queueing networks based on the characteristics of the phase-type distribution.

A state-dependent sojourn time distribution provides managers or system controllers with the ability to post real-time information to customers or to take real-time actions to improve system performance. Call centers are one of the most active research areas on this subject. Given the system state at the time of estimation, Whitt (1999) proposed a method of estimating the waiting time distribution of each customer in a single stage G/G/c queue. He estimated the waiting time distribution using a Normal approximation for a large call center with many servers. Nakibly (2002) suggested an approximation model of the waiting time distribution in a multi-server queue by calculating iteratively the waiting time of each customer in the queue. To increase customer satisfaction by announcing expected waiting time to a customer, Jouini and Dallery (2006) investigated the waiting time distribution for multiclass, multi-server call centers with exponential arrival and service times.

Our work differs from existing research in important ways. First, for single-stage systems, our method is effective for small and medium sized systems; whereas the method of Whitt (1999) is effective only for large systems. Second, existing research addresses only single stage systems, whereas we extend our methodology to address multi-stage systems and even simple, acyclic networks.

We focus on the sojourn time instead of waiting time to extend our interest to manufacturing and warehousing systems. Our approximation model is based on the phase-type distribution. In contrast to the steady state sojourn time distribution, we do not need to consider the arrival process for a state-dependent sojourn time distribution, because it does not affect the sojourn time distribution of a order in the system. Throughout this paper, we allow processing times to follow general distributions, and we model them with phase type distributions. We also assume that all servers in a workstation are homogeneous; that is, they have the same processing time distributions.

The rest of this paper is organized as follows: in Section 2, we introduce characteristics of the phase-type distribution, which is the fundamental concept behind our models. Also, we introduce the method of fitting a general distribution as a corresponding phase-type distribution. In Section 3, we introduce an approximation model of the sojourn time distribution for a multi-server queueing system with exponential service times. In Section 4, we present an approximation model for general service times and apply it to some numerical examples. In Section 5, we compare an approximation model for queueing networks with simulation; in Section 6, we discuss the results and implications of our work.

2. Preliminaries

2.1. Phase-type distributions

The phase-type distribution is composed of a finite sum or a finite mixture of exponentially distributed components, or a combination of both. When the interarrival time and service time follow the exponential distribution, we call it a Markovian queueing system and solve the system using Markov processes. In addition, if we approximate the interarrival time and service time as a corresponding phase-type distribution, we can analyze the system using the Markov property.

The continuous phase-type distribution used in this paper was defined by Neuts (1981). The following relationships are developed in that book and are repeated here for clarity of exposition. A phase-type distribution is the distribution of time to reach absorbing state $m+1$ in a finite Markov process having infinitesimal generator

$$\mathbf{Q} = \begin{bmatrix} \mathbf{T} & \mathbf{T^0} \\ \mathbf{0} & 0 \end{bmatrix},$$

where $\mathbf{0}$ is a row vector of zeros, the $m \times m$ matrix $\mathbf{T}$ satisfies $T_{ii} < 0$, for $1 \leq i \leq m$, and $T_{ij} \geq 0$, for $i \neq j$. The equation $\mathbf{Te} + \mathbf{T^0} = \mathbf{0}$ is satisfied, where $\mathbf{0}$ is a column vector of zeros and $\mathbf{e}$ is a column vector of ones. The initial probability vector of $\mathbf{Q}$ is $(\boldsymbol{\alpha},\ \alpha_{m+1})$, with $\boldsymbol{\alpha}\mathbf{e} + \alpha_{m+1} = 1$. The pair $(\boldsymbol{\alpha}, \mathbf{T})$ specifies the phase-type representation.

Given initial probability vector $\boldsymbol{\alpha}$, the cumulative distribution function of the time to reach state $m+1$ is

$$F(x) = 1 - \boldsymbol{\alpha} e^{\mathbf{T}x}\mathbf{e}. \tag{1}$$

The density function is

$$f(x) = \boldsymbol{\alpha}\mathbf{e}^{\mathbf{T}x}\mathbf{T^0} = \boldsymbol{\alpha}\mathbf{e}^{\mathbf{Tx}}(-\mathbf{T})\mathbf{e},$$

and the moments are defined by

$$E(X^k) = (-1)^k k! \boldsymbol{\alpha}\mathbf{T}^{-k}\mathbf{e}. \tag{2}$$

Neuts (1981) also introduced the convolution property of the phase-type distribution. If $H(\cdot)$ and $I(\cdot)$ are both continuous phase-type distributions with representations $(\boldsymbol{\alpha}, \mathbf{T})$ and $(\boldsymbol{\beta}, \mathbf{S})$ of orders m and n, then the convolution of two distributions, $H * I(\cdot)$ is also a phase-type distribution with representation $(\boldsymbol{\gamma}, \mathbf{P})$ and the infinitesimal generator and the initial probability vector are given by

$$\mathbf{P} = \begin{bmatrix} \mathbf{T} & \mathbf{T^0}\boldsymbol{\beta} \\ 0 & \mathbf{S} \end{bmatrix},$$

$$\boldsymbol{\gamma} = [\boldsymbol{\alpha}, \alpha_{m+1}\boldsymbol{\beta}].$$

2.2. Fitting distribution

In order to analyze our system as a Markovian queueing system, we approximate a general service time as a corresponding phase-type distribution. Mapping general distributions to phase-type distributions has been studied by Sauer and Chandy (1975), Marie (1980), Tijms (1994), You et al. (2002) and Osogami and Harchol-Balter (2003). Phase-type distributions comprise a dense set in the set of all distributions, and so can be made to approximate a general distribution with any degree of accuracy. We follow the relatively simple matching two moments method of Tijms (1994) and You et al. (2002), for reasons of simplicity and computational time. As we show below, this fitting method provides good results for the systems we investigate; other methods could be used.

We fit a general distribution as one of three phase-type distributions: Erlang-k, balanced hyper-exponential, and the exponential distribution, based on the squared coefficient of variation $C^2 = \sigma^2/\mu^2$. If $C^2 < 1$, a general distribution is approximated as an Erlang distribution, Erlang (k, λ), of order k. The density function is given by

$$f(x) = \lambda^k \frac{x^{k-1}}{(k-1)!} e^{-\lambda x}, x \geq 0.$$

The shape parameter k is computed by $\left\lceil \frac{1}{C^2} \right\rceil$ and $\lambda = k/E[X]$, where $E[X]$ is the mean of the general distribution.

If $C^2 > 1$, a general distribution is converted to a balanced hyper-exponential distribution, HE_2. The density function of HE_2 is described by

$$f(x) = p_1 \lambda_1 e^{-\lambda_1 x} + p_2 \lambda_2 e^{-\lambda_2 x}, x \geq 0,$$

where $p_1 = \frac{1}{2}\left(1 + \sqrt{\frac{C^2-1}{C^2+1}}\right)$, $p_2 = 1 - p_1$, $\lambda_1 = 2p_1/E[X]$ and $\lambda_2 = 2p_2/E[X]$.

If $C^2 = 1$, we use the exponential distribution, with density function,

$$f(x) = \lambda e^{-\lambda x}, x \geq 0,$$

where $\lambda = 1/E[X]$.

3. Exponential service times

In this section, we assume all servers in a workstation are identical and have the same exponential distribution of processing times. Whitt (1999) suggested that the waiting time distribution in such a system follows an Erlang-$(k+1)$ distribution with mean $1/c\mu$ when the system has c servers and there are k customers ahead. We are interested in the complete sojourn time distribution, so we must add processing time to the waiting time.

3.1. An exact model

The state dependent sojourn time distribution does not rely on the inter-arrival time distribution because the waiting time distribution is determined only by the number of servers, the number of orders ahead and the service rate. Also, we do not need to model the remaining service time for exponential service, due to the memoryless property. Every order arriving to the system has Erlang waiting time in queue and is processed in exponential service time.

We compute the sojourn time distribution of an order in the system based on the convolution property of the phase-type distribution, introduced by Neuts (1981). Because service times are exponential (for now), both the waiting time and service time distributions are phase-type. Therefore the sojourn time distribution is also phase-type, and the model is exact.

We construct the initial probability vector and the infinitesimal generator of the system based on this result. The initial probability vector and the infinitesimal generator are composed of the waiting time representation $(\boldsymbol{\alpha}, \mathbf{W})$ and service time representation $(\boldsymbol{\beta}, \mathbf{S})$. The sizes of the infinitesimal generator $\mathbf{W}$ and the initial probability vector $\boldsymbol{\alpha}$ are determined by the number of orders ahead in the queue. For example, if there are k orders ahead, the size of the infinitesimal generator and initial probability vector of waiting time are given by $\mathbf{W}_{(k+1)\times(k+1)}$ and $\boldsymbol{\alpha}_{1\times(k+1)}$:

$$\mathbf{W} = \left[\begin{array}{cccccc} -c\mu & c\mu & 0 & \cdots & 0 & 0 \\ 0 & -c\mu & c\mu & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -c\mu & c\mu \\ 0 & 0 & 0 & \cdots & 0 & -c\mu \end{array}\right],$$

$$\boldsymbol{\alpha} = [1, 0, \cdots, 0].$$

Thus, the phase-type representation of the sojourn time distribution is determined by

$$\mathbf{K} = \left[\begin{array}{cc} \mathbf{W} & \mathbf{W^0}\boldsymbol{\beta} \\ 0 & \mathbf{S} \end{array}\right],$$

$$\boldsymbol{\gamma} = [1, 0, \cdots, 0].$$

We can generate the cumulative distribution function (CDF) of the system based on the phase-type representation $(\boldsymbol{\gamma}, \mathbf{K})$

$$F(t) = P(T \leq t) = 1 - \boldsymbol{\gamma} e^{\mathbf{K}t}\mathbf{e}, (t \geq 0).$$

The probability density function (PDF) is given by

$$f(t) = \boldsymbol{\gamma} e^{\mathbf{K}t}\mathbf{K^0} = \boldsymbol{\gamma} e^{\mathbf{K}t}(-\mathbf{K})\mathbf{e},\ (t \geq 0).$$

4. General service times

In addition to the exponential service time case, Whitt (1999) developed an approximation model for general service times in single stage systems. For a large number of servers c, a Normal approximation can be used. The mean waiting time is given by

$$E[W] \approx \frac{k+1}{\mu c}\left(1+\frac{1}{2c}\right),$$

and the full distribution of waiting time is described by a Normal approximation. Whitt states that his approximation is appropriate only when the number of servers is substantially larger than the number of customers ahead, as is commonly the case in call centers.

In contrast to Whitt (1999), we approximate the sojourn time distribution using phase-type distributions. The procedure is similar to the exponential service times case, except for considering the remaining service times of the orders in service.

The procedure is:

1. Approximate the service time distribution as a corresponding phase-type representation $(\boldsymbol{\beta}, \mathbf{S})$ based on the C_s^2.
2. Approximate the first waiting time distribution as a corresponding phase-type representation $(\boldsymbol{\alpha}_1, \mathbf{W}_1)$ based on a Markov process we define below.
3. Approximate the second waiting time distribution as a corresponding phase-type representation $(\boldsymbol{\alpha}_2, \mathbf{W}_2)$ based on the Markov process and the first initial probability vector $\boldsymbol{\alpha}_1$.
4. Approximate successive waiting time distributions as corresponding phase-type representations $(\boldsymbol{\alpha}_i, \mathbf{W}_i)$, $i \geq 3$, according to the same procedure.
5. Generate the initial probability vector and infinitesimal generator $(\boldsymbol{\gamma}, \mathbf{K})$ for the system using the convolution property of the phase-type distribution.
6. Solve $F(t) = P(T \leq t) = 1 - \boldsymbol{\gamma} e^{\mathbf{K}t}\mathbf{e}, (t \geq 0)$ to obtain the CDF of the sojourn time distribution.

To analyze a system using continuous time Markov chains, we need to approximate a general service time distribution as a phase-type distribution.

We generate the phase-type representation of service time $(\boldsymbol{\beta}, \mathbf{S})$ using the method in Section 2.2. The number of phases and the transition rate are determined by the squared coefficient of variation (SCV) and the number of servers.

If a server is idle, an arriving order enters service immediately, and we do not need to consider waiting time to calculate the sojourn time—the sojourn time is equal to the service time. If there is no idle server and the arriving order finds k orders ahead in the queue, its sojourn time consists of three times — the waiting time for the first order to depart, the waiting time for the next k orders to depart, and the service time. In this condition, if one of the servers finishes its order, the order can go one step forward and then all servers are working immediately. So the order waits $k+1$ "sub-waiting times" to enter service and departs the system after receiving service. Hereafter, we refer to a "sub-waiting time" as an *epoch*.

4.1. Approximating the first epoch

Each epoch is the time an arriving order spends in queue until one of the servers finishes its order. That is, an epoch starts when all servers are busy (*all-busy*) and ends when one of the servers finishes its order (*partial-busy*).

To estimate the distribution of each epoch, we introduce a continuous time Markov process $\{N(t); t \geq 0\}$ with some absorbing states to model the all-busy period, where the system-state of the Markov chain is the number of servers in each phase.

Asmussen and O'Cinneide (1998) showed that the waiting time distribution of an epoch in a GI/PH/c queue is phase-type, and that the number of phases is

$$\left(\begin{array}{c} m+c-1 \\ c \end{array} \right),$$

where c is the number of servers and m is the number of phases of each server.

Suppose there are c homogeneous servers, each with m phases, and an order finds k orders ahead in the queue. There are $m+1$ server-states, and each server can be in one of those states. An order arrives to find k orders ahead and servers in one of m states (none are idle). We describe the system-state as an m-vector of server-states. The i^{th} element in the vector records the number of servers in state i.

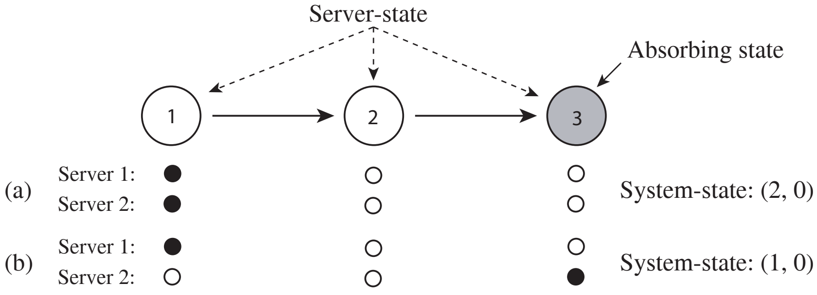

Figure 1: System-state and server-state.

Example 1. An order arrives to a system with 2 homogeneous servers and finds 3 orders ahead in the queue. We wish to estimate the sojourn time distribution of this order given $E[T_s] = 2$ and $C_s^2 = 0.8$.

First, we approximate the general service time distribution as a corresponding distribution using the fitting distribution method in Section 2.2. We approximate with an Erlang$(2, 1)$ because $m = \left\lceil \frac{1}{C_s^2} \right\rceil = 2$ and $\mu = \frac{m}{E[T_s]} = 1$.

Figure 1 shows two examples of the relationship between server-state and system-state: (a) shows two servers working in the first server-state, and this is system-state (2, 0); (b) shows server 1 working in the first server-state and server 2 has finished its order, and this is system-state (1, 0). There are 5 system-states: (2, 0), (1, 1), (0, 2), (1, 0), (0, 1). We call (2, 0), (1, 1), (0, 2) *all-busy* states and (1, 0), (0, 1) *partial-busy* states. The time between partial-busy states is an epoch, except for the first epoch which begins upon arrival to the system and therefore to an all-busy state (otherwise a server was empty and the order entered service immediately).

We are interested in the distribution of the first epoch, which is the time until the process enters the first partial-busy state, given the process started from an all-busy state. The infinitesimal generator $\mathbf{W}_1$ of the first epoch includes state changes directly among all-busy states. In Example 1,

$$\mathbf{W}_1 = \begin{bmatrix} -2 & 2 & 0 \\ 0 & -2 & 1 \\ 0 & 0 & -2 \end{bmatrix}.$$

We also need to approximate the initial probability vector $\boldsymbol{\alpha}_1$ of the first epoch, for which we use the stationary distribution of the all-busy states, because an arriving order finds the system in one of these all-busy states. In our approximation model, if the system reaches a partial-busy state, the system-state is changed immediately to an all-busy state because there is another order in the queue. Thus we should consider two kinds of state changes for $\boldsymbol{\alpha}_1$. One is state changes from the all-busy states to all-busy states, and the other is state changes from all-busy states to partial-busy states instantaneously followed by state changes from partial-busy states to all-busy states. If we sequence the states such that the all-busy states precede the partial-busy states, the former state changes have infinitesimal generator W_1, which has zero value for the lower triangle below the diagonal. The latter state changes have transition rate matrix H, which has the same size as W_1 and has zero value for the upper triangle and diagonal. The transition rate matrix $\mathbf{Q} = \mathbf{W_1} + \mathbf{H}$ includes all possible state changes when an order arrives at a certain position in queue. In Example 1, $\mathbf{H}$ and $\mathbf{Q}$ are

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} -2 & 2 & 0 \\ 1 & -2 & 1 \\ 0 & 2 & -2 \end{bmatrix}.$$

Now we can compute $\boldsymbol{\alpha}_1$ using the transition rate matrix $\mathbf{Q}$. The stationary distribution is given by

$$\pi\mathbf{Q} = \mathbf{0}, \pi\mathbf{e} = 1,$$

where $\mathbf{e}$ is a column vector of ones. That this is an approximation for $\boldsymbol{\alpha}_1$ and not an exact expression can be see by considering a simple M/E_2/1 system, for which the probability that an order is in the second stage of service is less than half, whereas our method would give one half. However, for more interesting systems with multiple servers and more orders in queue, the approximation is more accurate.

In Example 1, the initial probability vector $\boldsymbol{\alpha}_1$ of the all-busy states in the first epoch is

$$\boldsymbol{\alpha}_1 = \pi = (\pi_{20}, \pi_{11}, \pi_{02}) = (0.25, 0.5, 0.25).$$

4.2. Approximating the remaining epochs

The infinitesimal generators $\mathbf{W}_k$ of the following epochs are the same as the infinitesimal generator of the first epoch, because the state changes from the all-busy to partial-busy states are the same. However, the initial probability vectors $\boldsymbol{\alpha}_k$ of the following epochs are not the same. Rather, they must be derived successively from the initial probability in the previous epoch, because these affect which partial-busy state was reached. As we mentioned above, if the system reaches a partial-busy state, the system state is changed immediately to an all-busy state. This means the initial probability vectors $\boldsymbol{\alpha}_k$ of the all-busy states in epoch k come directly from the stationary distribution $\boldsymbol{\beta}_{k-1}$ of the partial-busy states in the former epoch $k-1$.

Now we introduce a method to compute the stationary distribution $\boldsymbol{\beta}_{k-1}$ of partial-busy states in epoch $k-1$. Every epoch starts from one of the all-busy states and ends in one of the partial-busy states, so if we know the stationary distribution of the all-busy states and the absorbing probability from the all-busy states to partial-busy states, then the stationary probability of the partial-busy state j

$$\beta_j = \sum_{all-busy\ \ i} \pi_i u_{ij},$$

where π_i is stationary probability of all-busy state i and u_{ij} is the absorbing probability from all-busy state i to partial-busy state j. From the Chapman-Kolmogorov equations,

$$u_{ij} = \sum_{all-busy\ \ h} P_{ih} u_{hj}.$$

where P_{ih} is the transition probability from all-busy state i to all-busy state h.

Finally, we get the elements of $\boldsymbol{\alpha}_k$ directly from the corresponding elements in $\boldsymbol{\beta}_{k-1}$. Because the dimensions of $\boldsymbol{\alpha}_k$ and $\boldsymbol{\beta}_{k-1}$ are different, we must append to $\boldsymbol{\beta}_{k-1}$ a sufficient number of zeros to reach the dimension of $\boldsymbol{\alpha}_k$. In Example 3.1, the initial probability vector $\boldsymbol{\alpha}_2$ of the all-busy states in the second epoch is

$$\boldsymbol{\alpha}_2 = (\beta_{10}, \beta_{01}, 0) = (0.375, 0.625, 0).$$

Note that elements of the initial probability vector $\boldsymbol{\alpha}_k$ converge to the same value as the epoch k increases.

4.3. Approximating the sojourn time distribution of the system

The sojourn time of an order with $c + k$ orders ahead in the system has $k + 1$ different sub-waiting times and one service time. So, the state-dependent sojourn time distribution is given by the convolution of these $k + 2$ distributions.

We construct the initial probability vector and the infinitesimal generator of the queueing system based on the work of Neuts (1981). The infinitesimal generator is composed of the waiting time representation $(\boldsymbol{\alpha}_k, \mathbf{W})$ of each order's waiting time in the system and the service time representation $(\boldsymbol{\beta}, \mathbf{S})$. If there are k orders ahead in the queue, the infinitesimal generator and initial probability vector are given by

$$\mathbf{K} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_1^0 \boldsymbol{\alpha}_2 & 0 & \cdots & 0 & 0 \\ 0 & \mathbf{W}_2 & \mathbf{W}_2^0 \boldsymbol{\alpha}_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \mathbf{W}_{k+1} & \mathbf{W}_{k+1}^0 \boldsymbol{\alpha}_{k+2} \\ 0 & 0 & 0 & \cdots & 0 & \mathbf{S} \end{bmatrix},$$

$$\boldsymbol{\gamma} = [1, 0, \cdots, 0].$$

The CDF and PDF are given by

$$F(t) = P(T \leq t) = 1 - \boldsymbol{\gamma} e^{\mathbf{K}t} \mathbf{e}, (t \geq 0),$$

$$f(t) = \boldsymbol{\gamma} e^{\mathbf{K}t} \mathbf{K^0} = \boldsymbol{\gamma} e^{\mathbf{K}t} (-\mathbf{K}) \mathbf{e}, (t \geq 0).$$

4.4. Normal Approximation Model

One drawback of our approximation model is computation time for very large systems. For example, models with up to 30 servers and fewer than, say, 20 orders ahead, generally solve within 1 minute; problems with 50 servers can take 3 minutes; problems with 100 servers can take 8.5 hours and problems with 200 servers, more than a day. To address these larger problems, we introduce a second approximation called the Normal Approximation Model (NAM).

The NAM uses the same phase-type representation of waiting and processing times as the approximation we have already developed, except that the CDF and PDF are not calculated with the matrix exponential expression in Equation 1. Instead, we assume the final distribution is Normally distributed, and that we can compute the required first and second moments with Equation 2, which requires only a matrix power computation.

4.5. Numerical results

To test the approximation model from Section 4.3, we consider two factors—the number of servers c and the number of orders ahead k. We use

- $c =\{$ 2, 3, 5, 10, 20, 50, 100, 200$\}$, and
- $k =\{5, 10, 20\}$ for $c \leq 100$, $\{40, 60, 80\}$ for $c = 200$.

We test possible combinations of these two factors under the same capacity (mean processing time $E[T] = 5$ and $C^2 = 0.5$). For the comparison, we add the mean processing time $E[T] = 5$ hours to the waiting time of Whitt's results because he computes only mean waiting time. In the simulation model, we used the Gamma distribution for the service time because it is often an appropriate model for task completion time (Law and Kelton, 2000).

Table 1 shows the mean sojourn time results from the approximation model, Whitt's model and the simulation model. As expected, Whitt's model shows good results when the number of servers is high (in this case, greater than 30). With a few exceptions, the approximation model appears to perform well over a wide range of problem instances. We should note however, that the largest problems in this table are very difficult to compute using our approximation model. For example, a problem with 100 servers and 20 orders ahead takes 8.3 hours, and 200 server problems take about 1 day. We record computation time using a 2.4 GHz Intel Core 2 Duo processer, and most of the problems except for those mentioned above are computed in a few seconds. Therefore, for single-stage systems, one might view Whitt's method and ours as complements—ours performs well for small and medium sized systems, his for large.

Figures 2–4 compare the PDF and CDF of the approximation model with the simulation model under different numbers of servers and orders ahead in the queue. We use the Anderson-Darling (A-D) test to check the agreement between distributions. As shown in Table 2, all test statistics are significant with $\alpha = 5\%$.

4.6. Testing the Normal Approximation Model

Table 3 shows the percent differences of percentiles between the NAM and the simulation when $c = 2, 20$ and 100. (We could not compare these results with Whitt, because he models only the waiting time.) We see that differences at the 95^{th} percentile are greater than 5 percent for cases $k <$

Table 1: A comparison of mean sojourn times for general service times.

Servers		2			3			5	
Orders ahead	5	10	20	5	10	20	5	10	20
Whitt	18.75	34.38	65.63	11.67	21.39	40.83	11.60	17.10	28.10
% difference	-4.95	6.51	15.65	-20.99	-7.06	5.21	5.23	6.43	8.27
Approximation	19.38	31.88	56.87	14.71	23.20	40.17	10.97	16.24	26.82
% difference	-1.79	-1.24	0.21	-0.39	0.79	3.51	-0.53	1.06	3.34
Simulation	19.73	32.27	56.75	14.77	23.01	38.81	11.02	16.07	25.95
Servers		10			20			30	
Orders ahead	5	10	20	5	10	20	5	10	20
Whitt	8.15	10.78	16.03	6.54	7.82	10.38	6.02	6.86	8.56
% difference	2.25	2.81	3.42	0.81	1.20	1.49	0.68	0.93	1.32
Approximation	8.02	10.79	16.45	6.48	7.81	10.67	5.99	6.84	8.63
% difference	0.62	2.99	6.19	-0.04	1.04	4.26	0.15	0.53	2.19
Simulation	7.96	10.48	15.43	6.46	7.70	10.26	5.97	6.79	8.43
Servers		50			100			200	
Orders ahead	5	10	20	5	10	20	40	60	80
Whitt	5.61	6.11	7.12	5.30	5.55	6.06	6.03	6.53	7.03
% difference	0.61	0.52	0.82	0.21	0.26	0.27	0.16	0.22	0.25
Approximation	5.59	6.09	7.12	5.30	5.55	6.04	6.02	6.51	7.02
% difference	0.39	0.21	0.73	0.19	0.19	0.09	-0.01	-0.01	0.05
Simulation	5.56	5.96	7.06	5.28	5.53	6.03	5.99	6.49	6.99

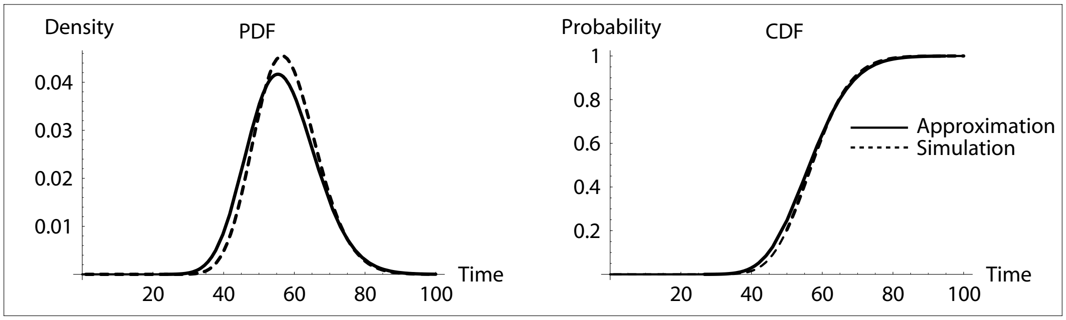

Figure 2: Comparison of the PDF and CDF of the approximation model and the simulation model (2 servers and 20 orders ahead).

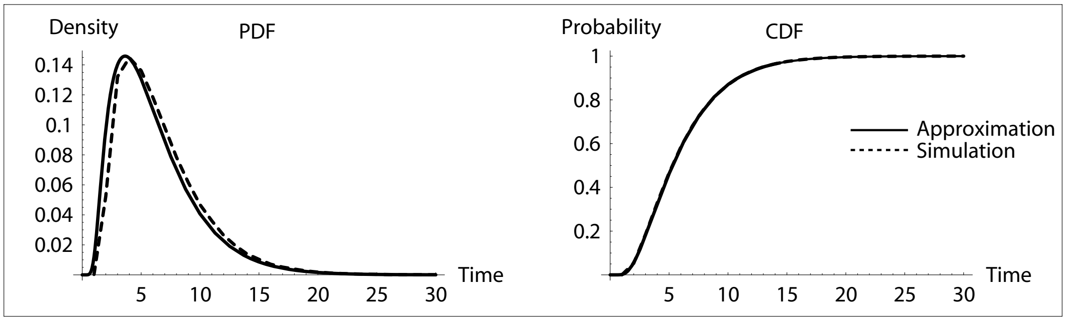

Figure 3: Comparison of the PDF and CDF of the approximation model and the simulation model (50 servers and 10 orders ahead).

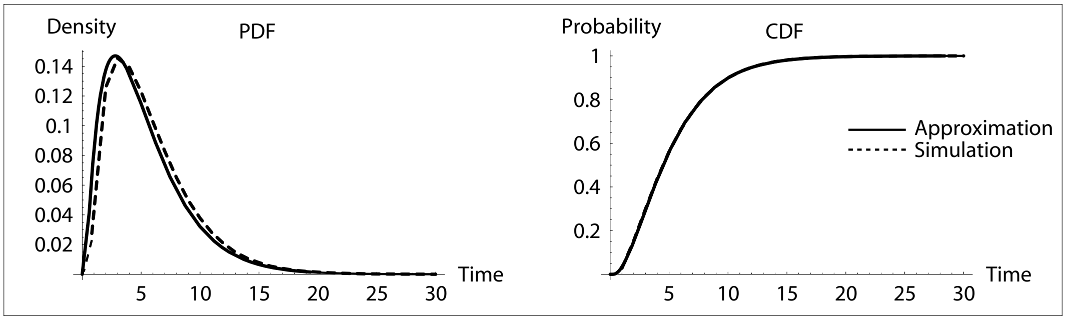

Figure 4: Comparison of the PDF and CDF of the approximation model and the simulation model (100 servers and 5 orders ahead).

Table 2: Anderson-Darling tests for general service time of the single stage queue.

Servers	Orders ahead	A-D	$\alpha = 5\%$	Decision	% difference	
					90^{th}	95^{th}
2	20	1.755		accept	0.92	1.78
50	10	1.828	2.492	accept	1.56	0.51
100	5	2.327		accept	0.22	0.27

Table 3: The percent differences of percentiles between NAM and Simulation for the single stage queue.

Servers		2			20			100	
Orders ahead	5	10	20	5	10	20	5	10	20
90^{th}	-1.69	0.46	1.48	-1.47	-0.53	2.53	-9.23	-1.89	-1.76
95^{th}	-2.73	0.24	2.34	-7.33	-5.84	-2.00	-14.06	-8.16	-7.74

c, where k is the number of orders ahead and there are c servers, which suggests that the model does not perform well under these conditions. The explanation has two parts: First, the Normal approximation will perform well, in general, when (1) successive waiting times have the same distribution, and (2) there are many of them. Second, successive waiting times in our case are *not* identically distributed because of the initial set of remaining times (Section 4.2). However, successive waiting time distributions do converge as memory of the initial remaining times is "lost," which is reflected by successive initial probability vectors $\boldsymbol{\alpha}_k$ converging to the same set of values. Systems with fewer servers converge faster, so, in general, the NAM performs better when $k \gg c$.

5. Multi-Stage Systems

5.1. Approximation model

We extend our work to serial lines and small acyclic queueing networks based on the single stage approximation model. We apply the single-stage model from Section 4 for the stages successively, each time updating the condition of the stage being computed. For example, consider a simple, 3-station serial line, and suppose an order is in queue at the first workstation. The order will experience three sojourn times, one at each queue and workstation, and the total sojourn time is their convolution. After each of the first two sojourn times, the status of the remaining queues changes, so we must estimate how many orders are in these queues when the order of interest arrives. After each sojourn time, the remaining queues change in two ways: orders arrive from the upstream workstation, and some orders are completed and depart. During each stage, then, we must estimate how many orders arrive to and depart from each remaining workstation (Figure 5).

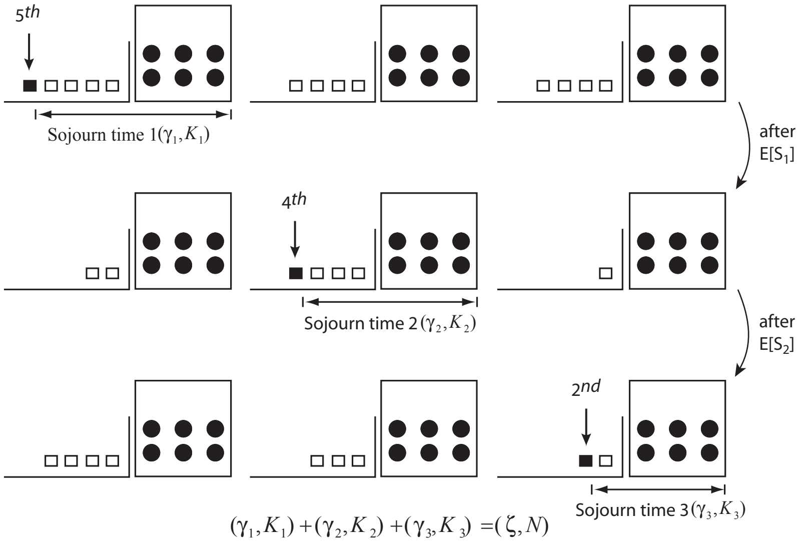

Figure 5: State-dependent queueing network model. Black discs represent workers; squares represent orders.

For the first workstation, we simply apply the single-stage model, giving us a sojourn time distribution and its mean $E[S_1]$. Now, let q_j^i be the estimated queue length of workstation j after the i^{th} sojourn time. Waiting time at the second workstation is based on the starting queue length q_2^0, plus arriving orders, minus departing orders. We assume that during $E[S_1]$, all $q_1^0 + c_1$ orders in front of the order of interest arrive to workstation 2. If all servers were busy during this time, then we would estimate the number processed at workstation 2 by $c_2 \times E[S_1]/E[T_2]$, where $E[T_2]$ is the mean processing time of the second workstation. However, it is possible that some servers could go idle during this time, so we correct for this by using a floor function, $\lfloor c_2 \times E[S_1]/E[T_2] \rfloor$. The expected number of orders ahead upon arrival to the second workstation becomes,

$$q_2^1 = \max(0, q_2^0 + (q_1^0 + c_1) - \lfloor c_2 \times E[S_1]/E[T_2] \rfloor).$$

During each stage of calculation, we must revise the number in queue at remaining workstations. For the third and following workstations, this leads to

$$q_j^i = \max\left(0, q_j^{i-1} + \left\lfloor c_{j-1} \times \frac{E[S_i]}{E[T_{j-1}]} \right\rfloor - \left\lfloor c_j \times \frac{E[S_i]}{E[T_j]} \right\rfloor\right), \text{ for } i = 1, \ldots, j-2.$$

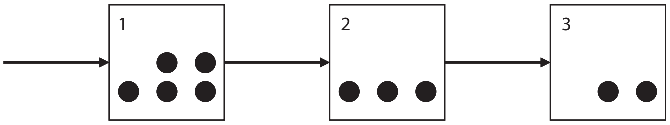

Figure 6: A serial line with 3 stations.

where $E[S_i]$ is the mean sojourn time of workstation i and $E[T_j]$ is the mean processing time of workstation j.

The final revision $(j-1)$ for workstation j is

$$q_j^{j-1} = \max\left(0, q_j^{j-2} + q_{j-1}^{j-2} + c_{j-1} - \left\lfloor c_j \times \frac{E[S_{j-1}]}{E[T_j]} \right\rfloor\right).$$

These approximations assume heavy traffic and are far from exact. However, they are sufficient if the number of stages is not too high, as we are about to show.

For large serial systems, an alternative approach is to modify the NAM by "collapsing" the system into a single-stage queue. If we wish to know the state-dependent sojourn time distribution of an order in front of the first station in a serial line with 3 stations, we assume our system is a single stage queue with c_3 servers, and the number of orders ahead is $q_1+c_1+q_2+c_2+q_3$, assuming all servers are busy. Now we can generate the state-dependent sojourn time distribution using the single stage model directly.

5.2. Numerical results

We apply the approximation model to a serial line with 3 stations (Figure 6) and an acyclic queueing network with 4 stations (Figure 7). The black disc in each station represents an occupied server, and in the case of the acyclic queueing network, an order departing from the first station selects its follow on station with probability p. We test state-dependent sojourn time distributions for both systems when the order is located in front of the first station. We also test the acyclic queueing network when the order is located in front of the second station. The system information is described in Table 4.

We compare the mean sojourn time and the distribution among the approximation, the NAM, and the simulation model under different numbers

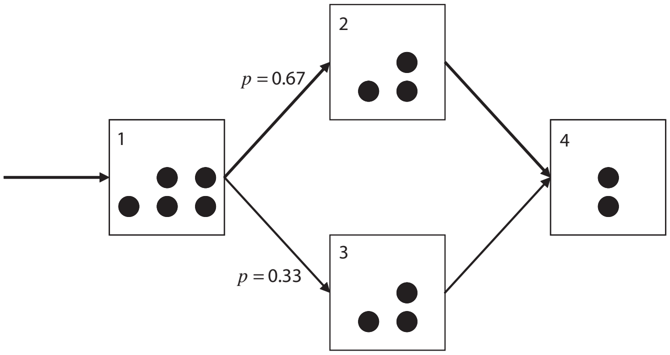

Figure 7: An acyclic queueing network with 4 stations. Black discs represent workers.

Table 4: The system information of the serial line and acyclic queueing network.

Serial line	$E[T]$	C^2	Servers	Utilization(ρ)
Interarrival	1.05	0.7		
Station 1	5	0.8	5	0.95
Station 2	3	0.8	3	0.95
Station 3	2	0.8	2	0.95

Acyclic queueing network	$E[T]$	C^2	Servers	Utilization(ρ)
Interarrival	1.05	0.7		
Station 1	5	0.8	5	0.95
Station 2	4.47	0.8	3	0.95
Station 3	9.09	0.8	3	0.95
Station 4	2	0.8	2	0.95

Table 5: The comparison of the mean sojourn time of the serial line.

Orders in queues	Approximation	% difference	NAM	% difference	Simulation
11-12-13	47.98	0.11	46.75	-2.46	47.93
6-12-13	42.70	-0.12	41.75	-2.34	42.75
16-12-13	53.27	0.50	51.75	-2.37	53.01
11-20-13	55.13	-1.62	54.75	-2.30	56.04
17-12-17	58.33	1.15	56.75	-1.60	57.67
7-20-4	41.90	-2.28	41.75	-2.64	42.88
31-22-29	93.32	0.30	92.75	-0.31	93.04

of orders in each station. We assume service times and interarrival times in the simulation model are Gamma distributed. To estimate a sojourn time distribution in simulation for problems given in Table 5, we collect sojourn times only for orders seeing the required state conditions upon arrival. When an order arrives to the system, we check the number in queue at each workstation. If this vector of values corresponds to the state of interest problem condition, we record the sojourn time; otherwise we do not. This "brute force" method alleviates the problem of estimating upon arrival the remaining times of orders in process. However, because occurrences of a particular state are rare, these simulations can take a very long time to run. For example, the first problem in Table 5 (11-12-13) takes 2 days for 50 runs, with 40 million (simulated) hours in each run.

Table 5 shows the results. In the table, the column "Orders in queues" represents the number of orders in the respective queues. The approximation model is extremely close to the simulation results for nearly every case. The NAM is acceptable, but not quite as good. We believe the NAM underestimates the mean (notice the negative differences) because it does not account for potential starving situations, in which the server has completed a job, but the next job has not yet arrived.

Figures 8 and 9 compare the PDF and CDF of the approximation model with the simulation model under different orders ahead in the queue. Again, the Anderson-Darling (A-D) tests reveal that there is no significant difference between the simulation and approximation results. However distributions generated by the NAM do not exactly fit the simulation (Table 7). Most of the percent differences are greater than 5%, except when the numbers of

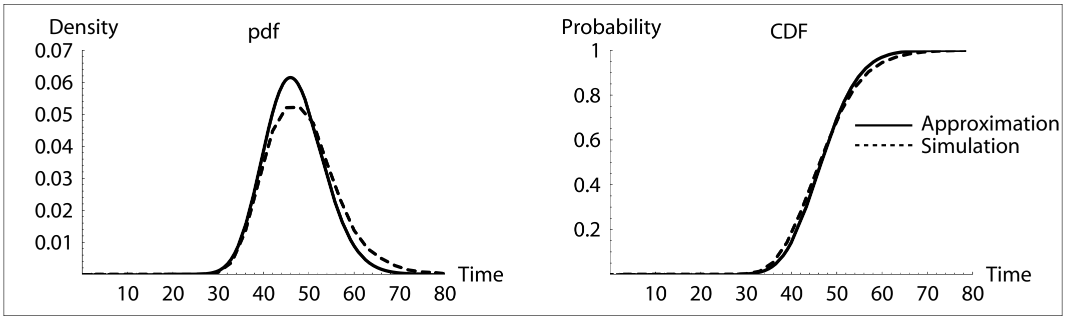

Figure 8: Comparison of the PDF and CDF of the serial line (11-12-13).

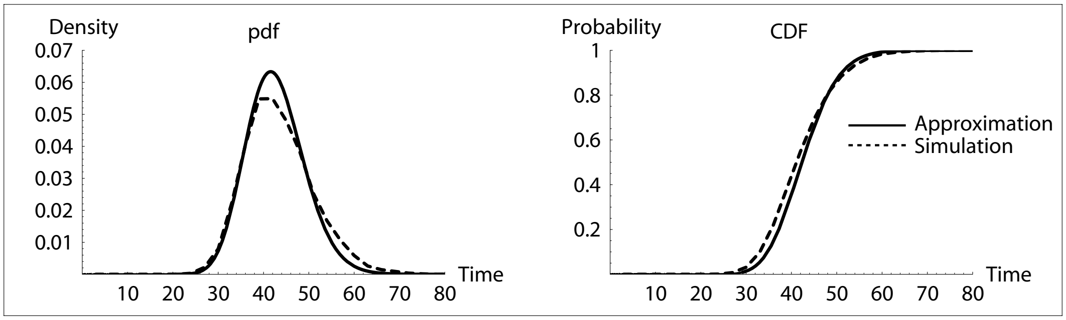

Figure 9: Comparison of the PDF and CDF of the serial line (6-12-13).

orders at each station are 31, 22 and 29. This suggests that the NAM works well only when the total orders ahead is significantly greater than the number of servers at the last station, because of reduced chances of starving.

Table 8 shows the mean sojourn time of the approximation model and simulation for an acyclic queueing network. We test two cases: the order is in front of the first station and in front of the second station. The approximation model appears to work well for both cases.

Figures 10–13 compare the PDF and CDF of the approximation model with the simulation model under different locations of the order of interest. Notice the unusual distribution in Figure 11. This is caused by the mixture of the distributions of the two potential serial line paths (path 1: 1-2-4 and path 2: 1-3-4) because an arriving order can take either path to depart the system. If the difference between the two mean sojourn times is large ($E[S]$ of the two paths are 44.44 and 75.06 hours), the sojourn time distribution has

Table 6: Anderson-Darling tests for the serial line.

Orders in queues	A-D	$\alpha = 5\%$	Decision	% difference	
				90^{th}	95^{th}
11-12-13	2.22	2.492	accept	-3.08	-4.13
6-12-13	1.79		accept	-1.61	-2.52
16-12-13	0.86		accept	-2.29	-2.95
17-12-17	1.85		accept	-0.84	-2.14

Table 7: The percent differences of percentiles between the NAM and simulation for the serial line.

Orders in queues	Total orders ahead	90^{th}	95^{th}
11-12-13	44	-5.24	-7.83
6-12-13	39	-6.64	-9.24
16-12-13	49	-5.28	-7.16
17-12-17	54	-4.59	-6.82
31-22-29	90	-2.43	-3.98

Table 8: The comparison of the mean sojourn time of the acyclic queueing network.

Order location	Orders in queues	Approximation	% difference	Simulation
The first station	11-12-8-13	60.87	1.86%	59.77
	6-4-16-13	55.59	0.76%	55.18
	16-12-15-13	72.26	0.84%	71.66
	11-20-12-13	73.10	0.73%	72.57
	17-12-7-17	70.54	2.78%	68.64
The second station	11-12-8-13	39.56	-1.28%	40.07
	6-8-12-13	33.48	-1.76%	34.08
	16-12-15-13	39.56	-1.81%	40.29
	11-20-12-13	51.72	-0.53%	51.99
	17-12-7-17	43.56	2.25%	42.60

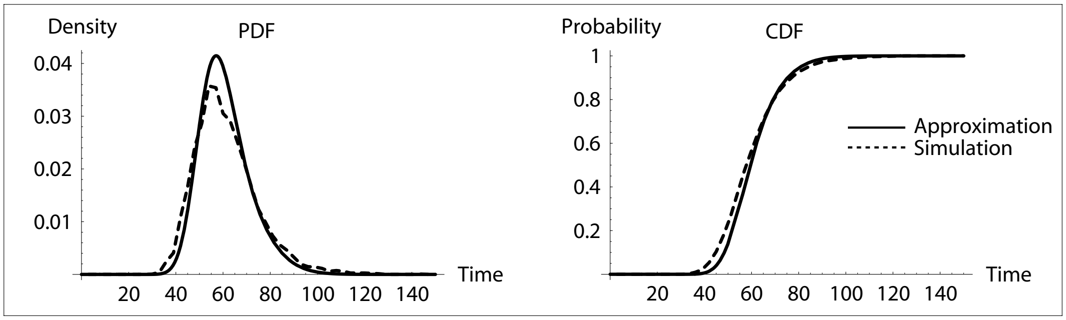

Figure 10: Comparison of the PDF and CDF of the acyclic queueing network (The first station, 11-12-8-13).

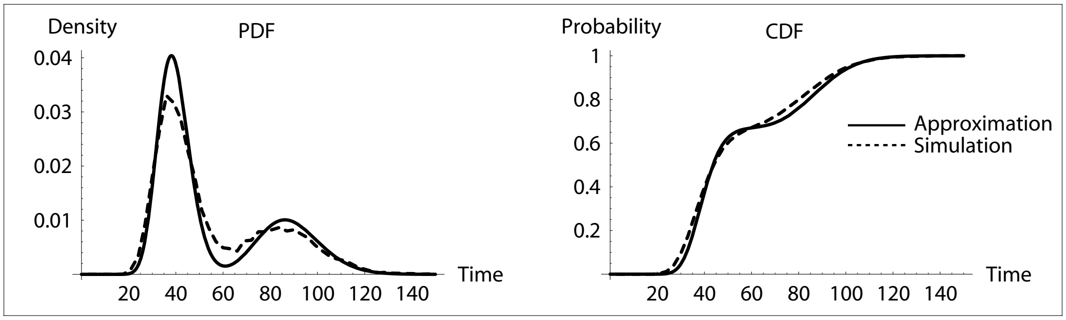

Figure 11: Comparison of the PDF and CDF of the acyclic queueing network (The first station, 6-4-16-13).

two peaks, as in Figure 11. Notice that our model reflects this because we estimate the sojourn time distribution for a random order by approximating the CDF of all possible "serial lines" (paths in the network) and mixing those CDFs according to the probabilities of taking those paths. We use the Anderson-Darling (A-D) test to check the agreement between the two distributions. As shown in Table 9, all test statistics are significant with $\alpha = 5\%$.

6. Conclusions

We have developed an approximation model for state-dependent sojourn time distributions of queueing systems with multiple servers, using the characteristics of phase-type distributions. Our model handles serial and acyclic

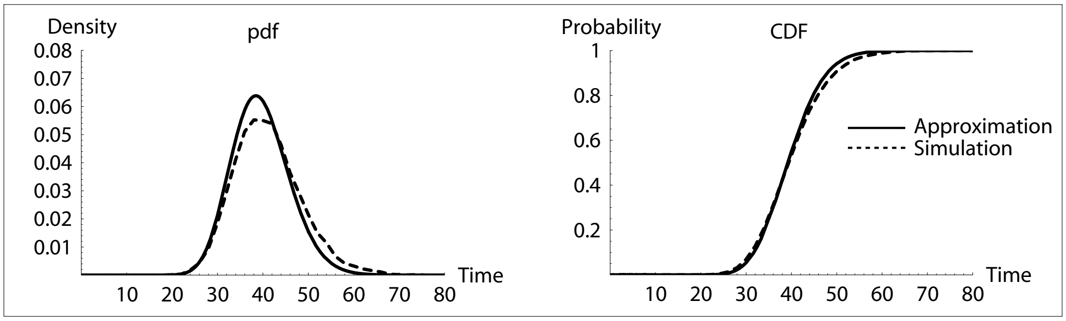

Figure 12: Comparison of the PDF and CDF of the acyclic queueing network (The second station, 11-12-8-13).

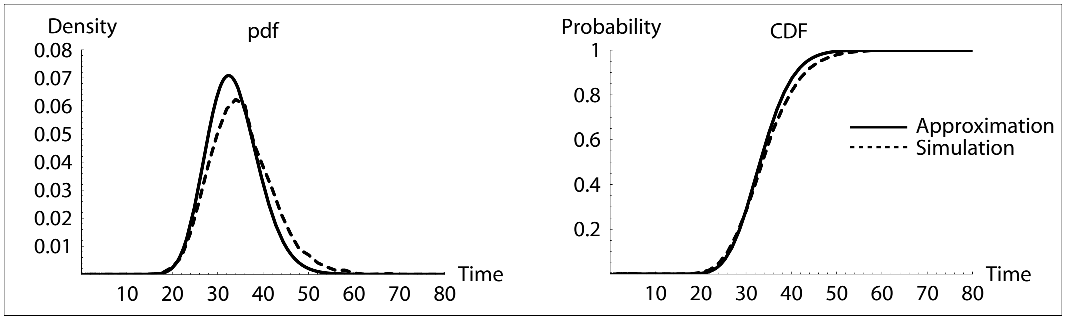

Figure 13: Comparison of the PDF and CDF of the acyclic queueing network (The second station, 6-8-12-13).

Table 9: Anderson-Darling tests for the acyclic queueing network.

Order location	Orders in queues	A-D	$\alpha = 5\%$	Decision	% difference	
					90^{th}	95^{th}
The first station	11-12-8-13	2.01	2.492	accept	-2.53	-4.40
	6-4-16-13	1.65		accept	1.75	-0.39
	16-12-15-13	1.21		accept	-0.91	-1.75
	11-20-12-13	2.31		accept	-3.72	-2.99
	17-12-7-17	1.74		accept	-1.98	-3.63
The second station	11-12-8-13	2.01	2.492	accept	-3.50	-4.15
	6-8-12-13	2.36		accept	-4.85	-6.61
	16-12-15-13	2.46		accept	-3.89	-5.43
	11-20-12-13	1.35		accept	-2.25	-2.65
	17-12-7-17	1.02		accept	-0.81	-0.98

queueing networks and performs well over a wide range of problem sizes. When the number of servers is more than about 200, the model is less effective due to the computation time required for the matrix exponential calculation.

The approximation allows us to estimate the probability that a customer or order will complete its service in less than a specific time, which can be used to offer service promises or to make real-time adjustments to the system in order to effect a better outcome. For example, an internet-based order fulfillment system might make service promises in real time, based on the state of the delivery system. Kim (2009) describes a dynamic worker allocation scheme for warehouses based on the probability that particular orders will finish before a deadline.

Acknowledgements. This research was supported by the Office of Naval Research. The authors also thank an anonymous referee for several helpful insights that improved the paper.

Asmussen, S., Møller, J. R., 2001. Calculation of the Steady State Waiting Time Distribution in GI/PH/c and MAP/PH/c Queues. Queueing Systems 37, 9–29.

Asmussen, S., O'Cinneide, C. A., 1998. Representations for Matrix-geometric and Matrix-exponential Steady-state Distributions with Applications to Many-server Queues. Stochastic Models 14, 369–387.

Gue, K. R., Kim, H. H., 2009. An Approximation Model for Sojourn Time Distributions in Acyclic Multi-server Queueing Networks, working paper.

Jouini, O., Dallery, Y., 2006. Predicting Queueing Delays for Multiclass Call Centers. In: valuetools '06: Proceedings of the 1st international conference on Performance evaluation methodolgies and tools. New York, NY, USA.

Kim, H. H., 2009. Modeling Service Performance and Dynamic Worker Allocation Policies for Order Fulfillment Fystems. Ph.D. dissertation, Auburn University, Auburn Alabama.

Law, A. M., Kelton, W. D., 2000. Simulation Modeling and Analysis, 3rd Edition. McGraw-Hill.

Luh, H., Zheng, Z. X., 2005. PH/PH/1 Queueing Models in Mathematica for Performance Evaluation. International Journal of Operations Research 2 (2), 81–88.

Mandelbaum, A., Massey, W. A., Reiman, M. I., 1998. Strong Approximations for Markovian Service Networks. Queueing Systems 30, 149–201.

Marie, R., 1980. Calculating Equilibrium Probabilities for $\lambda(n)/c_k/1/n$ Queues. In Proceedings of Performance, 117–125.

Nakibly, E., 2002. Predicting Waiting Times in Telephone Service Systems. Ph.d. thesis, The Senate of the Technion, Haifa, Israel.

Neuts, M. F., 1981. Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach. Johns Hopkins University Press, Baltimore, Maryland.

Osogami, T., Harchol-Balter, M., 2003. A Closed-form Solution for Mapping General Distributions to Minimal PH Distributions. Performance Evaluation, Special issue for the selected best papers of TOOLS 2003 63 (6), 524–552.

Sauer, C., Chandy, K., 1975. Approximate Analysis of Central Server Models. IBM Journal of Research and Development 19, 301–313.

Sengupta, B., 1989. Markov Processes Whose Steady State Distribution is Matrix-exponential with an Application to the GI/PH/1 Queue. Advances in Applied Probability 21, 159–180.

Shanthikumar, J., Sumita, O., 1988. Approximations for the Time Spent in a Dynamic Job Shop with Applications to Due Date Assignment. International J. Production Res. 26, 1329–1352.

Tijms, H. C., 1994. Stochastic Models: An Algorithmic Approach. John Wiley & Sons, Chichester.

Whitt, W., June 1999. Predicting Queueing Delays. Management Science 45 (6), 870–888.

You, J. U., Chung, M. Y., Sung, D. K., 2002. Performance Evaluation Using an Approximation Method for Sojourn Time Distributions in an IN/ISDN Signaling Platform. Computer Communications 25, 1283–1296.