OR Spectrum (2005) 27: 315–338
DOI: 10.1007/s00291-004-0189-z

OR Spectrum

Performance analysis of multi-server tandem queues with finite buffers and blocking

Marcel van Vuuren[1], Ivo J.B.F. Adan[1], and Simone A.E. Resing-Sassen[2]

[1] Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands (e-mail: m.v.vuuren@tue.nl, i.j.b.f.adan@tue.nl)
[2] CQM BV, P.O. Box 414, 5600 AK Eindhoven, The Netherlands (e-mail: resing@cqm.nl)

Abstract. In this paper we study multi-server tandem queues with finite buffers and blocking after service. The service times are generally distributed. We develop an efficient approximation method to determine performance characteristics such as the throughput and mean sojourn times. The method is based on decomposition into two-station subsystems, the parameters of which are determined by iteration. For the analysis of the subsystems we developed a spectral expansion method. Comparison with simulation shows that the approximation method produces accurate results. So it is useful for the design and analysis of production lines.

Keywords: Approximation – Blocking – Decomposition – Finite buffers – Multi-server tandem queues – Production lines – Spectral expansion

1 Introduction

Queueing networks with finite buffers have been studied extensively in the literature; see, e.g., Dallery and Gershwin [6], Perros [17,18], and Perros and Altiok [19], and the references therein. Most studies, however, consider *single*-server models. The few references dealing with *multi*-server models typically assume exponential service times. In this paper we focus on multi-server tandem queues with general service times, finite buffers and Blocking After Service (BAS).

Models with finite buffers and phase-type service times can be represented by finite state Markov chains. Hence, in theory, they can be analyzed exactly. However, the number of states of the Markov chain can be very large, which makes numerical solutions intractable. In practice, only small systems with one or two queues can be solved exactly; for exact methods we refer to Perros [18].

We develop an efficient method to approximate performance characteristics such as the throughput and the mean sojourn time. The method only needs the first two moments of the service time and it decomposes the tandem queue into subsystems with one buffer. fitted on Each multi-server subsystem is approximated by a single (super) server system with state dependent arrival and departure rates, the queue length distribution of which can be efficiently computed by a spectral expansion method. The parameters of the inter-arrival and service times of each subsystem are determined by an iterative algorithm. Numerical results show that this method produces accurate estimates for important performance characteristics as the throughput and the mean sojourn time.

Decomposition techniques have also been used by, e.g., Buzacott [2], Dallery et al. [5], Perros [18], and Kerbache and MacGregor Smith [11]. These papers deal with single-server queueing networks. Methods for multi-server queueing networks with finite buffers are presented by Tahilramani et al. [21], Jain and MacGregor Smith [9], and Cruz et al. [3,4]. These methods, however, do not assume general service times. An excellent survey on the analysis of manufacturing flow lines with finite buffers is presented by Dallery and Gershwin [6].

In the analysis of queueing networks with blocking three basic approaches can be distinguished. The first approach decomposes the network into subsystems and the parameters of the inter-arrival and service times of the subsystems are determined iteratively. This is the most common approach. It involves three steps:

1. Characterize the subsystems;
2. Derive a set of equations that determine the unknown parameters of each subsystem;
3. Develop an iterative algorithm to solve these equations.

This approach is treated in Perros' book [18] and in the survey of Dallery and Gershwin [6]. The approach in this paper also involves the three steps mentioned above, as we will explain in Section 5. There are also decomposition methods available for finite buffer models with some special features, such as assembly/disassembly systems (see Gershin and Burman [7]) and systems with multiple failure modes (see Tolio et al. [23]).

The second approach is also based on decomposition of the network, but instead of iteratively determining the parameters of the inter-arrival and service times of the subsystems, holding nodes are added to represent blocking. This so-called expansion method has been introduced by Kerbache and Smith [11]. The expansion method has been successfully used to model tandem queues with the following kinds of nodes: $M/G/1/K$ [20], $M/M/C/K$ [9] and $M/G/C/C$ [3,4].

The expansion method consist of the following three stages:

1. Network reconfiguration;
2. Parameter estimation;
3. Feedback elimination.

This method is very efficient; it produces accurate results when the buffers are large.

The third approach has been introduced by Kouvatsos and Xenios [12]. They developed a method based on the maximum entropy method (MEM) to analyze single-server networks. Here, holding nodes are also used and the characteristics of the queues are determined iteratively. For each subsystem in the network the queue-length distribution is determined by using a maximum entropy method. This algorithm is a linear program where the entropy of the queue-length distribution is maximized subject to a number of constraints. For more information we refer the reader to [12]. This method has been implemented in QNAT by Tahilramani et al. [21]; they also extended the method to multi-server networks. This method works well; the average error in the throughput is typically around 5%.

There are also several methods available for optimizing tandem queues with finite buffers. For example, Hillier and So [8] give some insight into the general form of the optimal design of tandem queues with the expected service times, the queue capacities and the number of servers at each station as the decision variables. Li et al. [13] have developed a method for optimization of tandem queues using techniques and concepts like simulation, critical path and perturbation analysis.

The paper is organized as follows. In Section 2 we introduce the tandem queue and its decomposition. In the section thereafter we elaborate on the arrivals at and departures from the subsystems. The spectral expansion method for analyzing the subsystems is discussed in Section 4. Section 5 describes the iterative algorithm. Numerical results are presented in Section 6. The results of the approximation method are compared with simulation and with QNAT. Finally, Section 7 contains some concluding remarks.

2 Model and decomposition

We consider a tandem queue (L) with M server-groups and $M-1$ buffers B_i, $i = 1, \dots, M-1$, of size b_i in between. The server-groups are labelled M_i, $i = 0, \dots, M-1$; server-group M_i has m_i parallel identical servers. The random variable S_i denotes the service time of a server in group M_i; S_i is generally distributed with rate $\mu_{p,i}$ (and thus with mean $1/\mu_{p,i}$) and coefficient of variation $c_{p,i}$. Each server can serve one customer at a time and the customers are served in order of arrival. The servers of M_0 are never starved and we consider the BAS blocking protocol. Figure 1 shows a tandem queue with four server groups.

The tandem queue L is decomposed into $M-1$ subsystems $L_1, L_2, \dots, L_{M-1}$. Subsystem L_i consists of a finite buffer of size b_i, m_{i-1} so-called arrival servers in front of the buffer, and m_i so-called departure servers after the buffer. The arrival and departure servers are virtual servers who describe the arrivals to a buffer and the departures from a buffer. The decomposition of L is shown in Figure 1.

The random variable A_i denotes the service time of an arrival-server in subsystem L_i, $i = 1, \dots, M-1$. This random variable represents the service time of a server in server-group M_{i-1} *including possible starvation* of this server. The random variable D_i denotes the service time of a departure-server in subsystem L_i; it represents the service time of a server in server-group M_i *including possible blocking* of this server. Let us indicate the rates of A_i and D_i by $\mu_{a,i}$ and $\mu_{d,i}$ and their coefficients of variation by $c_{a,i}$ and $c_{d,i}$, respectively. If these characteristics are known, we are able to approximate the queue-length distribution of each subsystem. Then, by using the queue-length distribution we can also approximate characteristics of the complete tandem queue, such as the throughput and mean sojourn time.

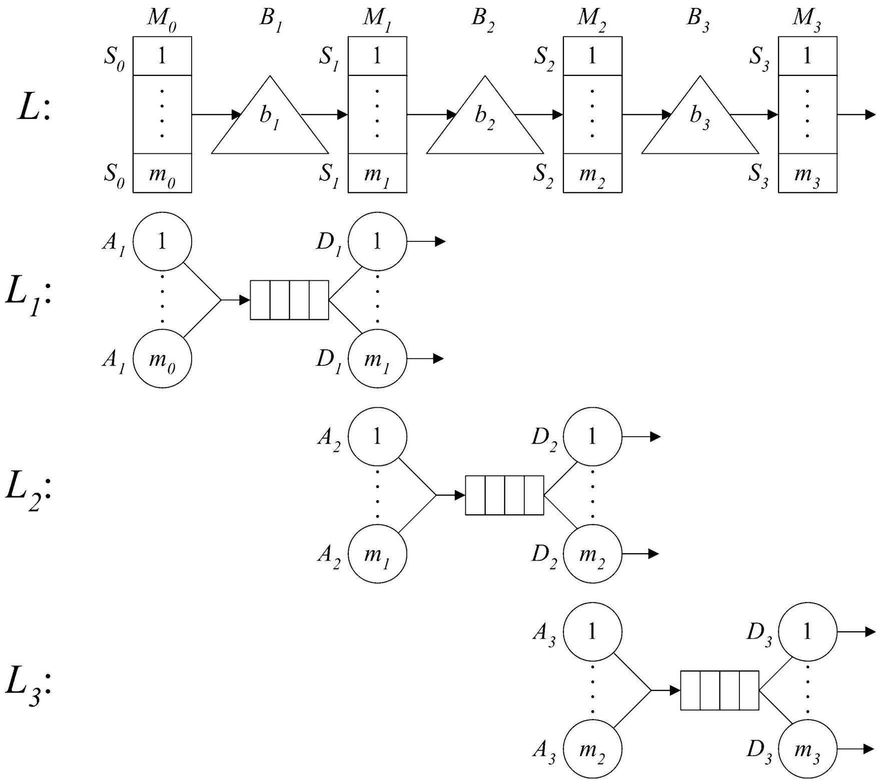

Fig. 1. The tandem queue L and its decomposition into three subsystems L_1, L_2 and L_3

3 Service times of arrival and departure servers

In this section we describe how the service times of the arrival and departure servers in subsystem L_i are modelled.

The service-time D_i of a departure-server in subsystem L_i is approximated as follows. We define $b_{i,j}$ as the probability that just after service completion of a server in server-group M_i, exactly j servers of server-group M_i are blocked. This means that, with probability $b_{i,j}$, a server in server-group M_i has to wait for one *residual* inter-departure time and $j-1$ full inter-departure times of the *next server-group* M_{i+1} before the customer can leave the server. The inter-departure times of server-group M_{i+1} are assumed to be independent and distributed as the inter-departure times of the *superposition* of m_{i+1} independent service processes, each with service times D_{i+1}; the residual inter-departure time is approximated by the equilibrium residual inter-departure time of the superposition of these service processes. Let the random variable SD_{i+1} denote the inter-departure time of server-group M_{i+1} and RSD_{i+1} the residual inter-departure time. Figure 2 displays a representation of the service time of a departure-server of subsystem L_i.

In the appendix it is explained how the rates and coefficients of variation of SD_{i+1} and RSD_{i+1} can be determined. If also the blocking probabilities $b_{i,j}$

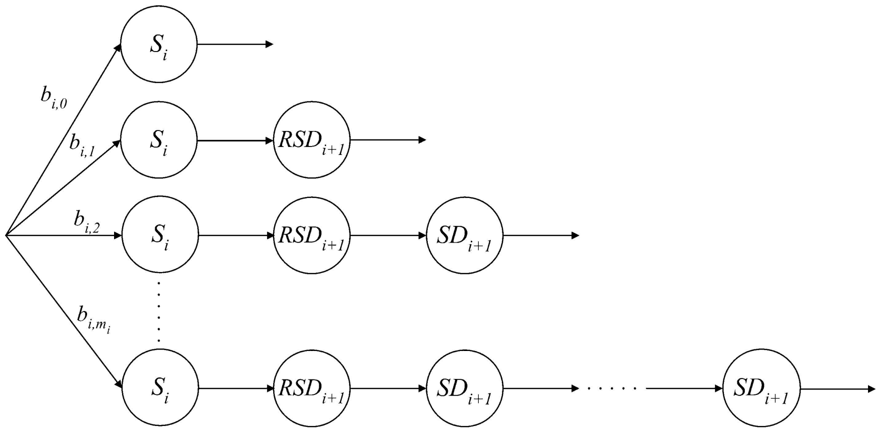

Fig. 2. Representation of the service time D_i of a departure-server of subsystem L_i

are known, then we can determine the rate $\mu_{d,i}$ and coefficient of variation $c_{d,i}$ of the service time D_i of a departure-server of subsystem L_i. The distribution of D_i is approximated by fitting an Erlang$_{k-1,k}$ or Coxian$_2$ distribution on $\mu_{d,i}$ and $c_{d,i}$, depending on whether $c_{d,i}^2$ is less or greater than $1/2$. More specifically, if $c_{d,i}^2 > 1/2$, then the rate and coefficient of variation of the Coxian$_2$ distribution with density

$$f(t) = (1-q)\mu_1 e^{-\mu_1 t} + q\frac{\mu_1\mu_2}{\mu_1-\mu_2}\left(e^{-\mu_2 t} - e^{-\mu_1 t}\right), \qquad t \geq 0,$$

matches with $\mu_{d,i}$ and $c_{d,i}$, provided the parameters μ_1, μ_2 and q are chosen as (cf. Marie [14]):

$$\mu_1 = 2\mu_{d,i}, \quad q = \frac{1}{2c_{d,i}^2}, \quad \mu_2 = \mu_1 q. \tag{1}$$

If $1/k \leq c_{d,i}^2 \leq 1/(k-1)$ for some $k > 2$, then the rate and coefficient of variation of the Erlang$_{k-1,k}$ with density

$$f(t) = p\mu^{k-1}\frac{t^{k-2}}{(k-2)!}e^{-\mu t} + (1-p)\mu^k\frac{t^{k-1}}{(k-1)!}e^{-\mu t}, \qquad t \geq 0,$$

matches with $\mu_{d,i}$ and $c_{d,i}$ if the parameters μ and p are chosen as (cf. Tijms [22]):

$$p = \frac{kc_{d,i}^2 - \sqrt{k(1+c_{d,i}^2) - k^2c_{d,i}^2}}{1+c_{d,i}^2}, \quad \mu = (k-p)\mu_{d,i}. \tag{2}$$

Of course, also other phase-type distributions may be fitted on the rate and coefficient of variation of D_i, but numerical experiments suggest that other distributions only have a minor effect on the results, as shown in [10].

The service times A_i of the arrival-servers in subsystem L_i are modelled similarly. Instead of $b_{i,j}$ we now use $s_{i,j}$ defined as the probability that just after service completion of a server in server-group M_i, exactly j servers of M_i are starved. This means that, with probability $s_{i,j}$, a server in server-group M_i has to wait one

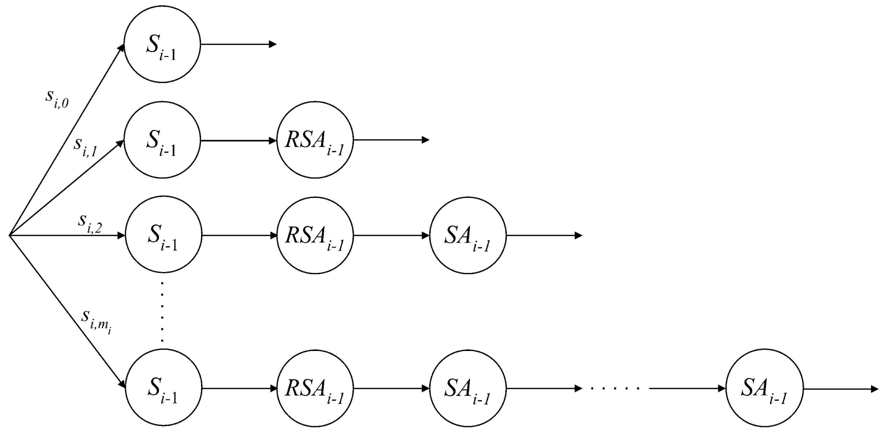

Fig. 3. Representation of the service time A_i of an arrival-server of subsystem L_i

residual inter-departure time and $j-1$ full inter-departure times from the *preceding server-group* M_{i-1}. Figure 3 displays a representation of the service time of an arrival-server of subsystem L_i.

4 Spectral analysis of a subsystem

By fitting Coxian or Erlang distributions on the service times A_i and D_i, subsystem L_i can be modelled as a finite state Markov process; below we describe this Markov process in more detail for a subsystem with m_a arrival servers, m_d departure servers and a buffer of size b.

To reduce the state space we replace the arrival and departure servers by *super servers* with *state-dependent* service times. The service time of the super arrival server is the inter-departure time of the service processes of the non-blocked arrival servers. If the buffer is not full, all arrival servers are working. In this case, the inter-departure time (or super service time) is assumed to be Coxian$_l$ distributed, where phase j ($j = 1, \ldots, l$) has parameter λ_j and p_j is the probability to proceed to the next phase (note that Erlang distributions are a special case of Coxian distributions). If the buffer is full, one or more arrival servers may be blocked. Then the super service time is Coxian distributed, the parameters of which depend on the number of active servers (and follow from the inter-departure time distribution of the active service processes). The service time of the super departure server is defined similarly. In particular, if none of the departure servers is starved, the super service time is the inter-departure time of the service processes of all m_d departure servers. This inter-departure time is assumed to be Coxian$_n$ distributed with parameters μ_j and q_j ($j = 1, \ldots, n$). So, the time spend in phase j is exponentially distributed with parameter μ_j and the probability to proceed to the next phase is q_j.

Now the subsystem can be described by a Markov process with states (i, j, k). The state variable i denotes the total number of customers in the subsystem. Clearly, i is at most equal to $m_d + b + m_a$. Note that, if $i > m_d + b$, then $i - m_d - b$ actually indicates the number of blocked arrival servers. The state variable j (k) indicates the phase of the service time of the super arrival (departure) server. If $i \leq m_d + b$, then the service time of the super arrival server consists of l phases; the number of phases depends on i for $i > m_d + b$. Similarly, the number of phases of the service time of the super departure server is n for $i \geq m_d$, and it depends on i for $i < m_d$.

The steady-state distribution of this Markov process can be determined efficiently by using the *spectral expansion method*, see e.g. Mitrani [16]. Using the spectral expansion method, Bertsimas [1] analysed a multi-server system with an infinite buffer; we will adapt this method for finite buffer systems. The advantage of the spectral expansion method is that the time to solve a subsystem is *independent* of the size of the buffer.

Below we formulate the equilibrium equations for the equilibrium probabilities $P(i,j,k)$. Only the equations in the states (i,j,k) with $m_d < i < m_d + b$ are presented; the form of the equations in the other states appears to be of minor importance to the analysis.

So, for $m_d < i < m_d + b$ we have:

$$P(i,1,1)(\lambda_1+\mu_1) = \sum_{j=1}^{l}(1-p_j)\lambda_j P(i-1,j,1) + \sum_{k=1}^{n}(1-q_k)\mu_k P(i+1,1,k) \tag{3}$$

$$P(i,j,1)(\lambda_j+\mu_1) = p_{j-1}\lambda_{j-1}P(i,j-1,1) + \sum_{k=1}^{n}(1-q_k)\mu_k P(i+1,j,k), \quad j = 2,\ldots,l \tag{4}$$

$$P(i,1,k)(\lambda_1+\mu_k) = q_{k-1}\mu_{k-1}P(i,1,k-1) + \sum_{j=1}^{l}(1-p_j)\lambda_j P(i-1,j,k), \quad k = 2,\ldots,n \tag{5}$$

$$P(i,j,k)(\lambda_j+\mu_k) = p_{j-1}\lambda_{j-1}P(i,j-1,k) + q_{k-1}\mu_{k-1}P(i,j,k-1), \quad j = 2,\ldots,l, \quad k = 2,\ldots,n. \tag{6}$$

We are going to use the separation of variables technique presented in Mickens [15], by assuming that the equilibrium probabilities $P(i,j,k)$ are of the form

$$P(i,j,k) = D_j R_k w^i, \qquad m_d \leq i \leq m_d + b, \quad 2 \leq j \leq l, \quad 2 \leq k \leq n. \tag{7}$$

Substituting (7) in the equilibrium equations (3)–(6) and dividing by common powers of w yields:

$$D_1R_1(\lambda_1+\mu_1) = \frac{1}{w}\sum_{j=1}^{l}(1-p_j)\lambda_j D_j R_1 + w\sum_{k=1}^{n}(1-q_k)\mu_k D_1 R_k \tag{8}$$

$$D_jR_1(\lambda_j+\mu_1) = p_{j-1}\lambda_{j-1}D_{j-1}R_1 + w\sum_{k=1}^{n}(1-q_k)\mu_k D_j R_k, \quad 2\leq j\leq l \tag{9}$$

$$D_1R_k(\lambda_1+\mu_k) = \frac{1}{w}\sum_{j=1}^{l}(1-p_j)\lambda_j D_j R_k + q_{k-1}\mu_{k-1}D_1R_{k-1}, \quad 2\leq k\leq n \tag{10}$$

$$D_jR_k(\lambda_j+\mu_k) = p_{j-1}\lambda_{j-1}D_{j-1}R_k + q_{k-1}\mu_{k-1}D_jR_{k-1} \quad 2 \leq j \leq l,\ 2 \leq k \leq n \tag{11}$$

We can rewrite (11) as:

$$\frac{\lambda_j D_j - p_{j-1}\lambda_{j-1}D_{j-1}}{D_j} = \frac{-\mu_k R_k + q_{k-1}\mu_{k-1}R_{k-1}}{R_k}, \quad 2\le j\le l, \quad 2\le k\le n. \tag{12}$$

Since (12) holds for each combination of j and k, the left-hand side of (12) is independent of k and the right-hand side of (12) is independent of j. Hence, there exists a constant x, depending on w, such that

$$-xD_j = \lambda_j D_j - p_{j-1}\lambda_{j-1}D_{j-1}, \qquad 2 \le j \le l, \tag{13}$$

$$-xR_k = -\mu_k R_k + q_{k-1}\mu_{k-1}R_{k-1}, \qquad 2 \le k \le n. \tag{14}$$

Solving equation (13) gives

$$D_j = D_1 \prod_{r=1}^{l-1} \frac{p_r\lambda_r}{x+\lambda_{r+1}} \tag{15}$$

Substituting (15) in (10) and using equation (14) we find the following relationship between x and w,

$$w = \sum_{j=1}^{l} \frac{(1-p_j)\lambda_j}{x+\lambda_j} \prod_{r=1}^{j-1} \frac{p_r\lambda_r}{x+\lambda_r}. \tag{16}$$

Note that w is equal to the Laplace Stieltjes transform $f_A(s)$ of the service time of the super arrival server, evaluated at $s = x$. Now we do the same for (9) yielding another relationship between x and w,

$$\frac{1}{w} = \sum_{k=1}^{n} \frac{(1-q_k)\mu_k}{-x+\mu_k} \prod_{r=1}^{k-1} \frac{q_r\mu_r}{-x+\mu_r}. \tag{17}$$

Clearly, $1/w$ is equal to the Laplace Stieltjes transform $f_D(s)$ of the service time of the super departure server, evaluated at $s = -x$. Substituting (16) and (17) in (8) and using (13) and (14) we find that

$$1 = f_A(x) f_D(-x).$$

This is a polynomial equation of degree $l+n$; the roots are labeled $x_t, t = 1, \ldots, l+n$, and they are assumed to be distinct. Note that these roots may be complex-valued. Using equation (17) we can find the corresponding $l+n$ values for w_t for $t = 1, \ldots, l+n$. Summarizing, for each t, we obtain the following solution of (3)–(6),

$$P(i,j,k) = B_t \left(\prod_{r=1}^{j-1} \frac{p_r\lambda_r}{x_t+\lambda_{r+1}}\right) \left(\prod_{r=1}^{k-1} \frac{q_r\mu_r}{-x_t+\mu_{r+1}}\right) w_t^i,$$
$$m_b \le i \le m_d + b, \quad 1 \le j \le l, \quad 1 \le k \le n,$$

where $B_t = D_{1,t}R_{1,t}$ is some constant. Since the equilibrium equations are linear, any linear combination of the above solutions satisfies (3)–(6). Hence, the general solution of (3)–(6) is given by

$$P(i,j,k) = \sum_{t=1}^{l+n} B_t \left(\prod_{r=1}^{j-1} \frac{p_r\lambda_r}{x(w_t)+\lambda_{r+1}}\right) \left(\prod_{r=1}^{k-1} \frac{q_r\mu_r}{-x(w_t)+\mu_{r+1}}\right) w_t^i,$$
$$m_b \le i \le m_d + b, \quad 1 \le j \le l, \quad 1 \le k \le n.$$

Finally, the unknown coefficients B_t and the unknown equilibrium probabilities $P(i,j,k)$ for $i < m_d$ and $i > m_d + b$ can be determined from the equilibrium equations for $i \leq m_d$ and $i \geq m_d + b$ and the normalization equation.

5 Iterative algorithm

We now describe the iterative algorithm for approximating the performance characteristics of tandem queue L. The algorithm is based on the decomposition of L in $M-1$ subsystems $L_1, L_2, \ldots, L_{M-1}$. Before going into detail in Section 5.2, we present the outline of the algorithm in Section 5.1.

5.1 Outline of the algorithm

- Step 0: Determine initial characteristics of the service times D_i of the departure servers of subsystem L_i, $i = M-1, \ldots, 1$.
- Step 1: For subsystem L_i, $i = 1, \ldots, M-1$:
    1. Determine the first two moments of the service time A_i of the arrival servers, given the queue-length distribution and throughput of subsystem L_{i-1}.
    2. Determine the queue-length distribution of subsystem L_i.
    3. Determine the throughput T_i of subsystem L_i.
- Step 2: Determine the new characteristics of the service times D_i of the departure servers of subsystem L_i, $i = M-1, \ldots, 1$.
- Repeat Step 1 and 2 until the service time characteristics of the departure servers have converged.

5.2 Details of the algorithm

Step 0: Initialization: The first step of the algorithm is to set $b_{i,j} = 0$ for all i and j. This means that we initially assume that there is no blocking. This also means that the random variables D_i are initially the same as the service times S_i.

Step 1: Evaluation of subsystems: We now know the service time characteristics of the departure servers of L_i, but we also need to know the characteristics of the service times of its arrival servers, before we are able to determine the queue-length distribution of L_i.

(a) Service times of arrival servers

For the first subsystem L_1, the characteristics of A_1 are the same as those of S_0, because the servers of M_0 cannot be starved.

For the other subsystems we proceed as follows. By application of Little's law to the arrival servers, it follows that the throughput of the arrival servers multiplied with the service time of an arrival server is equal to mean number of active (i.e.

non-blocked) arrival servers. The service time of an arrival server of subsystem i is equal to $1/\mu_{a,i}$ and the mean number of active servers is equal to

$$\left(1 - \sum_{j=1}^{m_{i-1}} p_{i,m_i+b_i+j}\right) m_{i-1} + \sum_{j=1}^{m_{i-1}} p_{i,m_i+b_i+j}(m_{i-1} - j).$$

So, we have for the throughput T_i of subsystem L_i,

$$T_i = \left(1 - \sum_{j=1}^{m_{i-1}} p_{i,m_i+b_i+j}\right) m_{i-1}\mu_{a,i} + \sum_{j=1}^{m_{i-1}} p_{i,m_i+b_i+j}(m_{i-1} - j)\mu_{a,i}, \quad (18)$$

where $p_{i,j}$ denotes the probability of j customers in subsystem L_i. By substituting the estimate $T_{i-1}^{(n)}$ for T_i and $p_{i,n_i+j}^{(n-1)}$ for p_{i,n_i+j} we get as new estimate for the service rate $\mu_{a,i}$,

$$\mu_{a,i}^{(n)} = \frac{T_{i-1}^{(n)}}{(1 - \sum_{j=1}^{m_{i-1}} p_{i,m_i+b_i+j}^{(n-1)})m_{i-1} + \sum_{j=1}^{m_{i-1}} p_{i,m_i+b_i+j}^{(n-1)}(m_{i-1} - j)},$$

where the super scripts indicate in which iteration the quantities have been calculated.

To approximate the coefficient of variation $c_{a,i}$ of A_i we use the representation for A_i as described in Section 3 (which is based on $s_{i-1,j}$, S_{i-1}, RSA_{i-1} and SA_{i-1}).

(b) Analysis of subsystem L_i

Based on the (new) characteristics of the service times of both arrival and departure servers we can determine the steady-state queue-length distribution of subsystem L_i. To do so we first fit Coxian$_2$ or Erlang$_{k-1,k}$ distributions on the first two moments of the service times of the arrival-servers and departure-servers as described in Section 3. Then we calculate the equilibrium probabilities $p_{i,j}$ by using the spectral expansion method as described in Section 4.

(c) Throughput of subsystem L_i

Once the steady-state queue length distribution is known, we can determine the new throughput $T_i^{(n)}$ according to (cf. (18))

$$T_i^{(n)} = \left(1 - \sum_{j=0}^{m_i-1} p_{i,j}^{(n)}\right) m_i\mu_{d,i}^{(n-1)} + \sum_{j=1}^{m_i-1} p_{i,j}^{(n)} j\mu_{d,i}^{(n-1)}. \quad (19)$$

We also determine new estimates for the probabilities $b_{i-1,j}$ that j servers of server-group M_{i-1} are blocked after service completion of a server in server-group M_{i-1} and the probabilities $s_{i,j}$ that j servers of server-group M_i are starved after service completion of a server in server-group M_i.

We perform Step 1 for every subsystem from L_1 up to L_{M-1}.

Step 2: Service times of departure servers: Now we have new information about the departure processes of the subsystems. So we can again calculate the first two moments of the service times of the departure-servers, starting from D_{M-2} down to D_1. Note that D_{M-1} is always the same as S_{M-1}, because the servers in server-group M_{M-1} can never be blocked.

A new estimate for the rate $\mu_{d,i}$ of D_i is determined from (cf. (18))

$$\mu_{d,i}^{(n)} = \frac{T_{i+1}^{(n)}}{(1 - \sum_{j=0}^{m_i-1} p_{i,j}^{(n)})m_i + \sum_{j=1}^{m_i-1} p_{i,j}^{(n)} j} \tag{20}$$

The calculation of a new estimate for the coefficient of variation $c_{d,i}$ of D_i is similar to the one of A_i.

Convergence criterion: After Step 1 and 2 we check whether the iterative algorithm has converged by comparing the departure rates in the $(n-1)$-th and k-th iteration. We decide to stop when the sum of the absolute values of the differences between these rates is less than ε; otherwise we repeat Step 1 and 2. So the convergence criterion is

$$\sum_{i=1}^{M-1} \left| \mu_{d,i}^{(n)} - \mu_{d,i}^{(n-1)} \right| < \varepsilon.$$

Of course, we may use other stop-criteria as well; for example, we may consider the throughput instead of the departure rates. The bottom line is that we go on until all parameters do not change anymore.

Remark. Equality of throughputs.

It is easily seen that, after convergence, the throughputs in all subsystems are equal. Let us assume that the iterative algorithm has converged, so $\mu_{d,i}^{(n)} = \mu_{d,i}^{(n-1)}$ for all $i = 1, \ldots, M-1$. From equations (19) and (20) we find the following:

$$\begin{aligned} T_i^{(n)} &= \left(1 - \sum_{j=0}^{m_i-1} p_{i,j}^{(n)}\right) m_i \mu_{d,i}^{(n-1)} + \sum_{j=1}^{m_i-1} p_{i,j}^{(n)} j \mu_{d,i}^{(n-1)} \\ &= \left(1 - \sum_{j=0}^{m_i-1} p_{i,j}^{(n)}\right) m_i \mu_{d,i}^{(n)} + \sum_{j=1}^{m_i-1} p_{i,j}^{(n)} j \mu_{d,i}^{(n)} \\ &= T_{i+1}^{(n)}. \end{aligned}$$

Hence we can conclude that the throughputs in all subsystems are the same after convergence.

Complexity analysis: The complexity of this method is as follows. Within the iterative algorithm, solving a subsystem consumes most of the time. In one iteration a subsystem is solved M times. The number of iterations needed is difficult to predict, but in practice this number is about three to seven iterations.

The time consuming part of solving a subsystem is solving the boundary equations. This can be done in $O((m_a + m_d)(k_a k_d)^3)$ time, where k_a is the number of phases of the distribution of one arrival process and k_d is the number of phases of the distribution of one departure process. Then, the time complexity of one iteration becomes $O(M \max_i((m_i + m_{i-1})(k_i k_{i-1})^3))$. This means that the time complexity is polynomial and it doesn't depend on the sizes of the buffers.

6 Numerical results

In this section we present some numerical results. To investigate the quality of our method we compare it with discrete event simulation. After that, we compare our method with the method developed by Tahilramani et al. [21], which is implemented in QNAT [25].

6.1 Comparison with simulation

In order to investigate the quality of our method we compare the throughput and the mean sojourn time with the ones produced by discrete event simulation. We are especially interested in investigating for which set of input-parameters our method gives satisfying results. Each simulation run is sufficiently long such that the widths of the 95% confidence intervals of the throughput and the mean sojourn time are smaller than 1%.

In order to test the quality of the method we use a broad set of parameters. We test two different lengths M of tandem queues, namely with 4 and 8 server-groups. For each tandem queue we vary the number of servers m_i in the server-groups; we use tandems with 1 server per server-group, 5 servers per server-group and with the sequence $(4, 1, 2, 8)$. We also vary the level of balance in the tandem queue; every server-group has a maximum total rate of 1 and the group right after the middle can have a total rate of 1, 1.1, 1.2, 1.5 and 2. The coefficient of variation of the service times varies between 0.1, 0.2, 0.5, 1, 1.5 and 2. Finally we vary the buffer sizes between 0, 2, 5 and 10. This leads to a total of 720 test-cases. The results for each category are summarized in Table 1 up to 5. Each table lists the average error in the throughput and the mean sojourn time compared with the simulation results. Each table also gives for 4 error-ranges the percentage of the cases which fall in that range. The results for a selection of 54 cases can be found in Tables 6 and 7.

Table 1. Overall results for tandem queues with different buffer sizes

Buffer	Error in throughput					Error in mean sojourn time				
sizes (b_i)	Avg.	0–5%	5–10%	10–15%	>15%	Avg.	0–5%	5–10%	10–15%	>15%
0	5.7%	55.0%	35.0%	4.4%	5.6%	6.8%	42.8%	35.0%	14.4%	7.8%
2	3.2%	76.1%	22.8%	1.1%	0.0%	4.7%	57.2%	35.0%	7.2%	0.6%
5	2.1%	90.6%	9.4%	0.0%	0.0%	4.5%	60.6%	32.2%	7.2%	0.0%
10	1.4%	95.6%	4.4%	0.0%	0.0%	5.1%	53.3%	34.4%	12.2%	0.0%

Table 2. Overall results for tandem queues with different balancing rates

Rates	Error in throughput					Error in mean sojourn time				
unbalanced server-group ($m_i \mu_{p,i}$)	Avg.	0–5%	5–10%	10–15%	>15%	Avg.	0–5%	5–10%	10–15%	>15%
1.0	3.3%	76.4%	20.8%	1.4%	1.4%	3.4%	74.3%	22.2%	2.1%	1.4%
1.1	3.1%	78.5%	18.1%	2.1%	1.4%	4.0%	68.1%	27.1%	3.5%	1.4%
1.2	3.0%	79.2%	18.8%	0.7%	1.4%	4.6%	59.7%	34.7%	4.2%	1.4%
1.5	3.0%	81.3%	16.0%	1.4%	1.4%	6.5%	38.2%	43.1%	16.7%	2.1%
2.0	3.1%	81.3%	16.0%	1.4%	1.4%	7.9%	27.1%	43.8%	25.0%	4.2%

Table 3. Overall results for tandem queues with different coefficients of variation of the service times

Coefficients	Error in throughput					Error in mean sojourn time				
of variation ($c^2_{p,i}$)	Avg.	0–5%	5–10%	10–15%	>15%	Avg.	0–5%	5–10%	10–15%	>15%
0.1	4.4%	54.2%	44.2%	1.7%	0.0%	3.1%	77.5%	21.7%	0.8%	0.0%
0.2	2.6%	88.3%	11.7%	0.0%	0.0%	3.4%	75.8%	22.5%	1.7%	0.0%
0.5	2.2%	90.8%	9.2%	0.0%	0.0%	4.5%	60.8%	32.5%	6.7%	0.0%
1.0	1.5%	93.3%	2.5%	4.2%	0.0%	4.1%	64.2%	30.0%	5.0%	0.8%
1.5	3.0%	82.5%	13.3%	0.0%	4.2%	7.5%	25.8%	54.2%	15.0%	5.0%
2.0	4.8%	66.7%	26.7%	2.5%	4.2%	9.1%	16.7%	44.2%	32.5%	6.7%

Table 4. Overall results for tandem queues with a different number of servers per server-group

Number of	Error in throughput					Error in mean sojourn time				
servers (m_i)	Avg.	0-5%	5–10%	10–15%	>15%	Avg.	0–5%	5–10%	10–15%	>15%
All 1	2.9%	83.8%	9.2%	2.9%	4.2%	5.9%	46.3%	39.2%	10.0%	4.6%
All 5	3.8%	68.3%	30.8%	0.8%	0.0%	4.6%	60.0%	29.2%	10.8%	0.0%
Mixed	2.6%	85.8%	13.8%	0.4%	0.0%	5.3%	54.2%	34.2%	10.0%	1.7%

We may conclude the following from the above results. First, we see in Table 1 that the performance of the approximation becomes better when the buffer sizes increase. This may be due to less dependencies between the servers-groups when the buffers are large.

We also notice that the performance is better for balanced lines (Table 2); for unbalanced lines, especially the estimate for the mean sojourn time is not as good as for balanced lines. If we look at the coefficients of variation of the service times (Table 3), we get the best approximations for the throughput when the coefficients

Table 5. Overall results for tandem queues with 4 and 8 server-groups

Number of server-groups (M)	Error in throughput					Error in mean sojourn time				
	Avg.	0–5%	5–10%	10–15%	>15%	Avg.	0–5%	5–10%	10–15%	>15%
4	2.3%	87.2%	12.2%	0.6%	0.0%	4.7%	57.5%	32.8%	9.7%	0.0%
8	3.9%	71.4%	23.6%	2.2%	2.8%	5.8%	49.4%	35.6%	10.8%	4.2%

Table 6. Detailed results for balanced tandem queues

m_i	M	$c^2_{p,i}$	Buffers	T App.	T Sim.	Diff.	S App.	S Sim.	Diff.
1	4	0.1	0	0.735	0.771	−4.7%	4.70	4.63	1.5%
	8		2	0.906	0.926	−2.2%	16.14	15.99	0.9%
	4		10	0.981	0.985	−0.4%	19.22	19.03	1.0%
	8	1.0	0	0.488	0.443	10.2%	11.73	13.43	−12.7%
	4		2	0.703	0.700	0.4%	9.09	9.25	−1.7%
	8		10	0.855	0.855	0.0%	49.52	49.81	−0.6%
	4	1.5	0	0.504	0.473	6.6%	5.82	6.27	−7.2%
	8		2	0.607	0.581	4.5%	21.94	23.52	−6.7%
	4		10	0.834	0.835	−0.1%	22.38	22.31	0.3%
5	4	0.1	0	0.789	0.856	−7.8%	22.48	21.78	3.2%
	8		2	0.827	0.926	−10.7%	52.35	49.71	5.3%
	4		10	0.927	0.983	−5.7%	36.88	35.24	4.7%
	8	1.0	0	0.693	0.697	−0.6%	49.20	49.14	0.1%
	4		2	0.797	0.808	−1.4%	26.37	26.17	0.8%
	8		10	0.867	0.882	−1.7%	83.09	83.96	−1.0%
	4	1.5	0	0.742	0.724	2.5%	22.99	23.90	−3.8%
	8		2	0.759	0.737	3.0%	54.63	57.27	−4.6%
	4		10	0.867	0.874	−0.8%	37.97	38.86	−2.3%
Mixed	4	0.1	0	0.746	0.793	−5.9%	16.19	16.28	−0.6%
	8		2	0.845	0.921	−8.3%	39.90	38.96	2.4%
	4		10	0.956	0.984	−2.8%	31.61	30.05	5.2%
	8	1.0	0	0.619	0.604	2.5%	37.90	38.55	−1.7%
	4		2	0.756	0.757	−0.1%	20.15	20.14	0.0%
	8		10	0.863	0.871	−0.9%	71.67	71.74	−0.1%
	4	1.5	0	0.633	0.619	2.3%	16.78	18.01	−6.8%
	8		2	0.705	0.678	4.0%	43.38	46.32	−6.3%
	4		10	0.850	0.856	−0.7%	31.43	32.37	−2.9%

Table 7. Detailed results for unbalanced tandem queues

m_i	M	$c^2_{p,i}$	Buffers	T App.	T Sim.	Diff.	S App.	S Sim.	Diff.
1	8	0.1	0	0.718	0.751	−4.4%	8.90	9.27	−4.0%
	4		2	0.960	0.958	0.2%	6.18	6.41	−3.6%
	8		10	0.980	0.983	−0.3%	38.45	43.22	−11.0%
	4	1.0	0	0.594	0.561	5.9%	4.84	5.28	−8.3%
	8		2	0.690	0.670	3.0%	18.81	20.31	−7.4%
	4		10	0.918	0.912	0.7%	16.20	17.41	−7.0%
	8	1.5	0	0.482	0.409	17.8%	11.26	13.79	−18.3%
	4		2	0.714	0.691	3.3%	8.03	8.60	−6.6%
	8		10	0.830	0.819	1.3%	46.75	50.16	−6.8%
5	8	0.1	0	0.781	0.851	−8.2%	43.03	42.65	0.9%
	4		2	0.902	0.958	−5.8%	21.63	21.50	0.6%
	8		10	0.922	0.983	−6.2%	71.89	73.95	−2.8%
	4	1.0	0	0.801	0.794	0.9%	20.79	21.13	−1.6%
	8		2	0.789	0.787	0.3%	51.52	53.49	−3.7%
	4		10	0.927	0.929	−0.2%	30.37	32.61	−6.9%
	8	1.5	0	0.730	0.692	5.5%	44.43	47.95	−7.3%
	4		2	0.850	0.828	2.7%	21.95	23.70	−7.4%
	8		10	0.864	0.862	0.2%	74.69	81.01	−7.8%
Mixed	8	0.1	0	0.744	0.790	−5.8%	30.96	32.41	−4.5%
	4	0.1	2	0.920	0.953	−3.5%	16.72	17.14	−2.5%
	8	0.1	10	0.945	0.983	−3.9%	61.00	62.54	−2.5%
	4	1.0	0	0.714	0.702	1.7%	16.22	16.43	−1.3%
	8	1.0	2	0.750	0.742	1.1%	39.64	42.20	−6.1%
	4	1.0	10	0.926	0.919	0.8%	25.99	27.60	−5.8%
	8	1.5	0	0.628	0.588	6.8%	32.68	37.66	−13.2%
	4	1.5	2	0.787	0.773	1.8%	17.52	18.93	−7.4%
	8	1.5	10	0.844	0.843	0.1%	61.82	69.32	−10.8%

of variation are 1, and also the estimate for the mean sojourn time is better for small coefficients of variation.

The quality of the results seems to be rather insensitive to the number of servers per server-group (Table 4), in spite of the super-server approximation used for multi-server models. Finally we may conclude from Table 5 that the results are better for shorter tandem queues.

Most crucial to the quality of the approximation of the throughput appears to be the buffer-size. For the sojourn time this appears to be the coefficient of variation of the service time. In Figures 4 and 5 we present a scatter-plot of simulation results versus approximation results for the throughput and mean sojourn times; the plotted cases are the same as in Tables 6 and 7. The results of the throughput are split-up

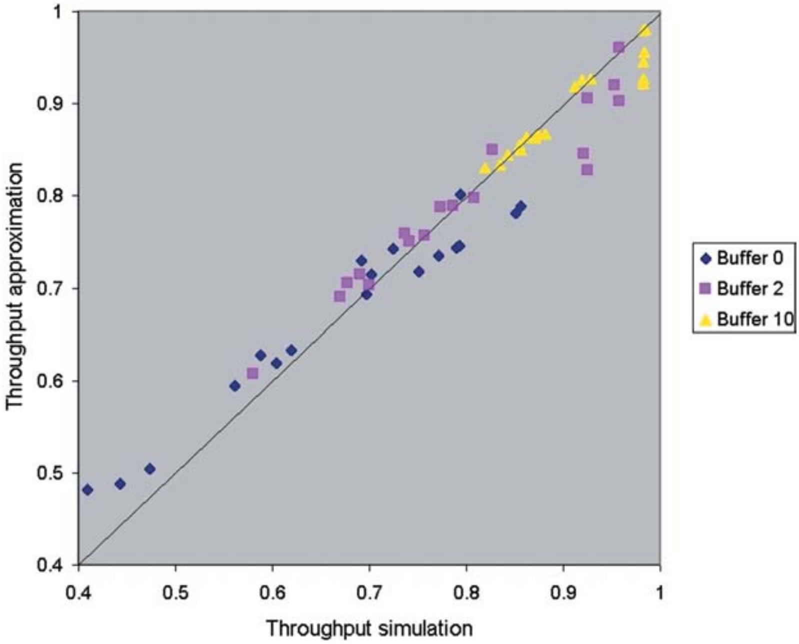

Fig. 4. Scatter-plot of the throughput of 54 cases split up by buffer-size

according to the buffer-size; the one for the sojourn times are split-up according to the squared coefficient of variation of the service times.

Overall we can say that the approximation produces accurate results in most cases. In the majority of the cases the error of the throughput is within 5% of the simulation and the error of the mean sojourn time is within 10% of the simulation (see also Tables 6 and 7). The worst performance is obtained for unbalanced lines with zero buffers and high coefficients of variation of the service times. But these cases are unlikely (and undesired) to occur in practice.

The computation times are very short. On a modern computer the computation times are much less than a second in most cases, only in cases with service times with low coefficients of variation and 1 server per server-group the computation times increase to a few seconds. Therefore, for the design of production lines, this is a very useful approximation method.

6.2 Comparison with QNAT

We also compare the present method with QNAT, a method developed by Tahilramani et al. [21]. We use a tandem queue with four server-groups. It was only possible to test cases where the first server-group consists of a single exponential server. The reason is that the two methods assume a different arrival process to the system. Both processes, however, coincide for the special case of a single exponential server at the beginning of the line. We varied the number of servers per server-group and the size of buffers. Table 8 shows the results.

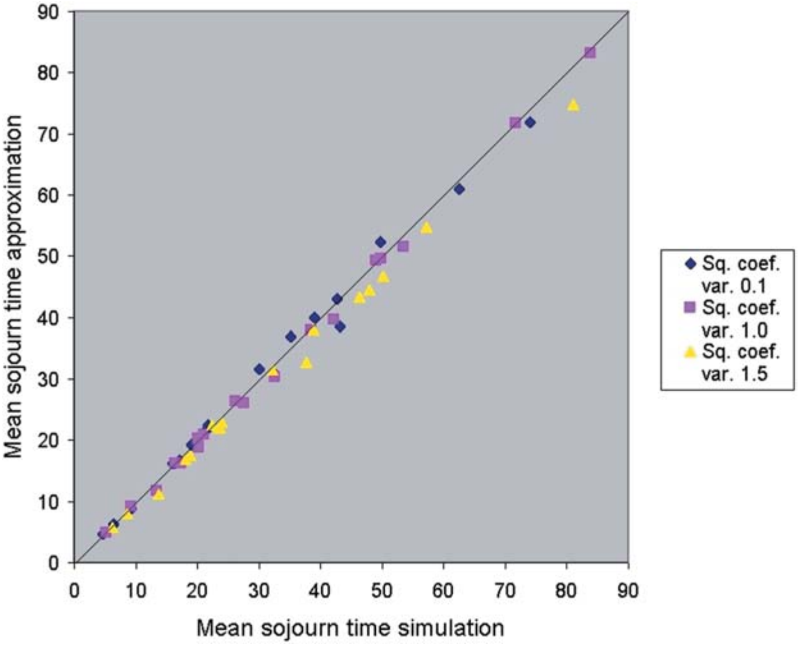

Fig. 5. Scatter-plot of the mean sojourn time of 54 cases split up by coefficient of variation

Table 8. Comparison of our method with QNAT

m_i	b_i	TP Sim.	TP App.	Our error	TP QNAT	QNAT Error	Soj. Sim.	Soj. App.	Our error	Soj. QNAT	QNAT error
(1,1,1,1)	0	0.515	0.537	−4.3%	0.500	2.9%	5.95	5.61	5.7%	–	–
(1,1,1,1)	2	0.702	0.703	−0.1%	0.750	−6.8%	9.25	9.10	1.7%	8.17	11.7%
(1,1,1,1)	10	0.879	0.876	0.3%	0.917	−4.3%	21.43	21.41	0.1%	18.55	13.5%
(1,5,5,5)	0	0.711	0.717	−0.8%	0.167	76.5%	17.87	17.67	1.1%	–	–
(1,5,5,5)	2	0.791	0.788	0.3%	0.800	−1.1%	20.53	20.45	0.4%	–	–
(1,5,5,5)	10	0.898	0.884	1.6%	0.895	0.3%	32.27	32.59	−1.0%	22.88	29.1%
(1,4,2,8)	0	0.677	0.692	−2.3%	0.200	70.5%	16.59	16.28	1.9%	–	–
(1,4,2,8)	2	0.775	0.774	0.1%	0.800	−3.2%	19.29	19.15	0.7%	–	–
(1,4,2,8)	10	0.893	0.886	0.8%	0.902	−1.0%	31.03	30.86	0.6%	23.04	25.7 %

We see that the present approximation method is much more stable than QNAT and gives in almost all cases better results. Especially the approximation of the mean sojourn time is much better; in a number of cases QNAT is not able to produce an approximation of the mean sojourn time. Of course, one should be careful with drawing conclusions from this limited set of cases. Table 8 only gives an indication of how the two methods perform.

6.3 Industrial case

To give an indication of the performance of our method in practice, we present the results of an industrial case. The case involves a production line for the production of light bulbs. The production line consists of 5 production stages with buffers in between. Each stage has a different number of machines varying between 2 and 8. The machines have deterministic service times, but they do suffer from breakdowns. In the queueing model we included the breakdowns into the coefficient of variation of the service times, yielding effective service times with coefficients of variation larger than 0. In Table 9 the parameters of the production line are shown.

Table 9. Parameters for the production line for the production of bulbs

Stage	m_i	$\mu_{p,i}$	$c^2_{p,i}$	b_i
1	2	5.73	0.96	–
2	8	1.53	0.09	21
3	4	3.43	0.80	11
4	1	32.18	0.57	34
5	4	16.12	0.96	19

We only have data of the throughput and not of the mean sojourn time of the line, so we can only test the approximation for the throughput. The output of the production line based on the measured data is 11.34 products per time unit. If we simulate this production line, we obtain a throughput of 11.41 products per time unit. The throughput given by our approximation method is 11.26, so in this case the approximation is a good prediction for the actual throughput.

7 Concluding remarks

In this paper we described a method for the approximate analysis of a multi-server tandem queue with finite buffers and general service times. We decomposed the tandem queue in subsystems. We used an iterative algorithm to approximate the arrivals and departures at the subsystems and to approximate some performance characteristics of the tandem queue. Each multi-server subsystem is approximated by a single (super) server queue with state-dependent inter-arrival and service times, the steady-state queue length distribution of which is determined by a spectral expansion method.

This method is robust and efficient; it provides a good and fast alternative to simulation methods. In most cases the errors for performance characteristics as the throughput and mean sojourn time are within 5% of the simulation results. Numerical results also give an indication of the performance of the method compared with QNAT. The method can be extended in several directions. One may think of more

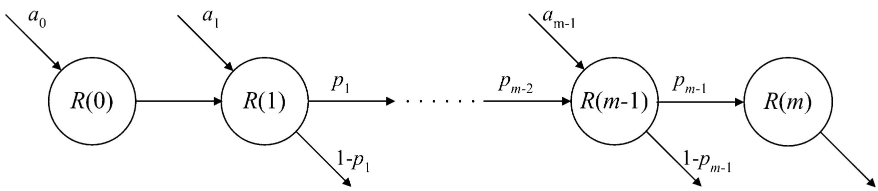

Fig. 6. Phase diagram of an arbitrary inter-departure time

general configurations, like splitting and merging of streams or the possibility of feedback. Other possibilities for extension are for example unreliable machines and assembly/disassembly (see [24]). Possibilities for improving the quality of the approximation are, for example, using a more detailed description of the arrival to and departures from the subsystems (e.g. including correlations between consecutive arrivals and departures) or improving the subsystem analysis by using a description of the service process that is more detailed than the super-server approach.

Appendix: Superposition of service processes

Let us consider m independent service processes, each of them continuously servicing customers one at a time. The service times are assumed to be independent and identically distributed. We are interested in the first two moments of an arbitrary inter-departure time of the superposition of m service processes. Below we distinguish between Coxian$_2$ service times and Erlang$_{k-1,k}$ service times.

A.1 Coxian$_2$ service times

We assume that the service times of each service process are Coxian$_2$ distributed with the same parameters. The rate of the first phase is μ_1, the rate of the second phase is μ_2 and the probability that the second phase is needed is q. The distribution of an arbitrary inter-departure time of the superposition of m service processes can be described by a phase-type distribution with $m+1$ phases, numbered $0, 1, \ldots, m$. In phase i exactly i service processes are in the second phase of the service time and $m - i$ service processes are in the first phase. A phase diagram of the phase-type distribution of an arbitrary inter-departure time is shown in Figure 6. The probability to start in phase i is denoted by a_i, $i = 0, \ldots, m - 1$. The sojourn time in phase i is exponentially distributed with rate $R(i)$, and p_i is the probability to continue with phase $i + 1$ after completion of phase i. Now we explain how to compute the parameters a_i, $R(i)$ and p_i.

The probability a_i can be interpreted as follows. It is the probability that i service processes are in phase 2 just after a departure (i.e., service completion). There is at least one process in phase 1, namely the one that generated the departure. Since the service processes are mutually independent, the number of service processes in phase 2 is binomially distributed with $m - 1$ trials and success probability p.

The success probability is equal to the fraction of time a single service process is in phase 2, so

$$p = \frac{q\mu_1}{q\mu_+\mu_2}.$$

Hence, for the initial probability a_i we get

$$a_i = \binom{m-1}{i} \left(\frac{q\mu_1}{q\mu_1 + \mu_2}\right)^i \left(\frac{\mu_2}{q\mu_1 + \mu_2}\right)^{m-1-i} \tag{21}$$

To determine the rate $R(i)$, note that in state i there are i processes in phase 2 and $m - i$ in phase 1, so the total rate at which one of the service processes completes a service phase is equal to

$$R(i) = (m - i)\mu_1 + i\mu_2 \tag{22}$$

It remains to find p_i, the probability that there is no departure after phase i. In phase i three things may happen:

- Case (i): A service process completes phase 1 and immediately continues with phase 2;
- Case (ii): A service process completes phase 1 and generates a departure;
- Case (iii): A service process completes phase 2 (and thus always generates a departure).

Clearly, p_i is the probability that case (i) happens, so

$$p_i = \frac{q(m-i)\mu_i}{R(i)} \tag{23}$$

Now the parameters of the phase-type distribution are known, we can determine its first two moments. Let X_i denote the total sojourn time, given that we start in phase i, $i = 0, 1, \ldots, m$. Starting with

$$EX_m = \frac{1}{R(m)}, \qquad EX_m^2 = \frac{2}{R(m)^2},$$

the first two moments of X_i can be calculated from $i = m - 1$ down to $i = 0$ by using

$$EX_i = \frac{1}{R(i)} + p_i EX_i, \tag{24}$$

$$EX_i^2 = \frac{2}{R(i)^2} + p_i \left(\frac{2EX_{i+1}}{R(i)} + EX_{i+1}^2\right). \tag{25}$$

Then the rate μ_s and coefficient of variation c_s of an arbitrary inter-departure time of the superposition of m service processes follow from

$$\mu_s^{-1} = \sum_{i=0}^{m} a_i EX_i \;=\; \frac{1}{m}\left(\frac{1}{\mu_1} + \frac{q}{\mu_2}\right), \tag{26}$$

$$c_s^2 = \mu_s^2 \left(\sum_{i=0}^{m} a_i EX_i^2\right) - 1 \tag{27}$$

A.2 Erlang$_{k-1,k}$ service times

Now the service times of each service process are assumed to be Erlang$_{k-1,k}$ distributed, i.e., with probability p (respectively $1-p$) a service time consists of $k-1$ (respectively k) exponential phases with parameter μ. Clearly, the time that elapses until one of the m service processes completes a service phase is exponential with parameter $m\mu$. The number of service phases completions before one of the service processes generates a departure ranges from 1 up to $m(k-1)+1$. So the distribution of an arbitrary inter-departure time of the superposition of m service processes is a mixture of Erlang distributions; with probability p_i it consists of i exponential phases with parameter $m\mu$, $i = 1, \ldots, m(k-1)+1$. Figure 7 depicts the phase diagram. Below we show how to determine the probabilities p_i.

An arbitrary inter-departure time of the superposition of m service processes is the minimum of $m-1$ equilibrium residual service times and one full service time. Both residual and full service time have a (different) mixed Erlang distribution. In particular, the residual service consists with probability r_i of i phases with parameter μ, where

$$r_i = \begin{cases} 1/(k-p), & i = 1, 2, \ldots, k-1; \\ (1-p)/(k-p), & i = k. \end{cases}$$

The minimum of two mixed Erlang service times has again a mixed Erlang distribution; below we indicate how the parameters of the distribution of the minimum can be determined. Then repeated application of this procedure yields the minimum of m mixed Erlang service times.

Let X_1 and X_2 be two independent random variables with mixed Erlang distributions, i.e., with probability $q_{k,i}$ the random variable X_k $(k = 1, 2)$ consists of i exponential phases with parameter μ_k, $i = 1, \ldots, n_k$. Then the minimum of X_1

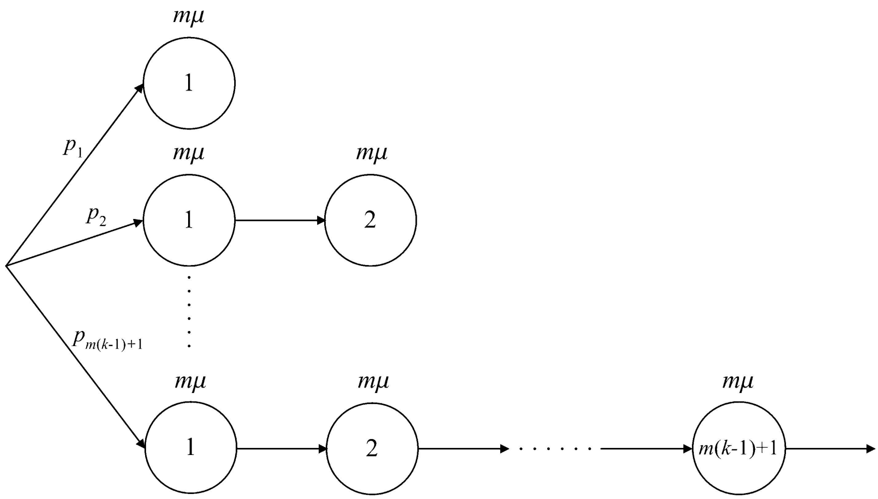

Fig. 7. Phase diagram of an arbitrary inder-departure time

and X_2 consists of at most n_1+n_2-1 exponential phases with parameter $\mu_1+\mu_2$. To find the probability q_i that the minimum consists of i phases, we proceed as follows. Define $q_i(j)$ as the probability that the minimum of X_1 and X_2 consists of i phases transitions, where $j(\le i)$ transitions are due to X_1 and $i-j$ transitions are due to X_2. Obviously we have

$$q_i = \sum_{j=\max(0,i-n_2)}^{\min(i,n_1)} q_i(j), \qquad i = 1,2,\ldots,n_1+n_2-1.$$

To determine $q_i(j)$ note that the ith phase transition of the minimum can be due to either X_1 or X_2. If X_1 makes the last transition, then X_1 clearly consists of exactly j phases and X_2 of at least $i-j+1$ phases; the probability that X_2 makes $i-j$ transitions before the jth transition of X_1 is negative-binomially distributed with parameters j and $\mu_1/(\mu_1+\mu_2)$. The result is similar if X_2 instead of X_1 makes the last transition. Hence, we obtain

$$\begin{aligned} q_i(j) = {} & \binom{i-1}{j-1}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^j\left(\frac{\mu_2}{\mu_1+\mu_2}\right)^{i-j} q_{1,j}\left(\sum_{k=i-j+1}^{n_2} q_{2,k}\right) \\ & +\binom{i-1}{j}\left(\frac{\mu_1}{\mu_1+\mu_2}\right)^j\left(\frac{\mu_2}{\mu_1+\mu_2}\right)^{i-j}\left(\sum_{k=j+1}^{n_1} q_{1,k}\right) q_{2,i-j}, \\ & 1 \le i \le n_1+n_2-1, \quad 0 \le j \le i, \end{aligned}$$

where by convention, $q_{1,0}=q_{2,0}=0$.

By repeated application of the above procedure we can find the probability p_i that the distribution of an arbitrary inter-departure time of the superposition of m Erlang$_{k-1,k}$ service processes consists of exactly i service phases with parameter $m\mu$, $i = 1,2,\ldots,m(k-1)+1$. It is now easy to determine the rate μ_s and coefficient of variation c_s of an arbitrary inter-departure time, yielding

$$\mu_s^{-1} = \frac{1}{m}\left(\frac{p(k-1)}{\mu}+\frac{(1-p)k}{\mu}\right) = \frac{k-p}{m\mu},$$

and, by using that the second moment of an E_k distribution with scale parameter μ is $k(k+1)/\mu^2$,

$$c_s^2 = \mu_s^2 \sum_{i=1}^{m(k-1)+1} p_i \frac{i(i+1)}{(m\mu)^2} - 1 = -1 + \frac{1}{(k-p)^2}\sum_{i=1}^{m(k-1)+1} p_i i(i+1).$$

A.3 Equilibrium residual inter-departure time

To determine the first two moments of the equilibrium residual inter-departure time of the superposition of m independent service processes we adopt the following simple approach.

Let the random variable D denote an arbitrary inter-departure time and let R denote the equilibrium residual inter-departure time. It is well known that

$$E(R) = \frac{E(D^2)}{2E(D)}, \qquad E(R^2) = \frac{E(D^3)}{3E(D)}.$$

In the previous sections we have shown how the first two moments of D can be determined in case of Coxian$_2$ and Erlang$_{k-1,k}$ service times. Its third moment is approximated by the third moment of the distribution fitted on the first two moments of D, according to the recipe in Section 3.

References

1. Bertsimas D (1990) An analytic approach to a general class of G/G/s queueing systems. Operations Rearch earch 1: 139–155
2. Buzacott JA (1967) Automatic transfer lines with buffer stock. International Journal of Production Research 5: 183–200
3. Cruz FRB, MacGregor Smith J (2004) Algorithm for analysis of generalized $M/G/C/C$ state dependent queueing networks. http://www.compscipreprints.com/comp/Preprint/fcruzfcruz/20040105/1
4. Cruz FRB, MacGregor Smith J, Queiroz DC (2004) Service and capacity allocation in $M/G/C/C$ state dependent queueing networks. Computers & Operations Research (to appear)
5. Dallery Y, David R, Xie X (1989) Approximate analysis of transfer lines with unreliable machines and finite buffers. IEEE Transactiona on Automatic Control 34(9): 943–953
6. Dallery Y, Gershwin B (1992) Manufacturing flow line systems: a review of models and analytical results. Queueing Systems 12: 3–94
7. Gershwin SB, Burman MH (2000) A decomposition method for analyzing inhomogeneous assembly/disassembly systems. Annals of Operation Research 93: 91–115
8. Hillier FS, So KC (1995) On the optimal design of tandem queueing systems with finite buffers. Queueing Systems Theory Application 21: 245–266
9. Jain S, MacGregor Smith J (1994) Open finite queueing networks with $M/M/C/K$ parallel servers. Computers Operations Research 21(3): 297–317
10. Johnson MA (1993) An empirical study of queueing approximations based on phase-type distributions. Communication Statistic-Stochastic Models 9(4): 531–561
11. Kerbache L, MacGregor Smith J (1987) The generalized expansion method for open finite queueing networks. The European Journal of Operations Research 32: 448–461
12. Kouvatsos D, Xenios NP (1989) MEM for arbitrary queueing networks with multiple general servers and repetative-service blocking. Performance Evaluation 10: 169–195
13. Li Y, Cai X, Tu F, Shao X (2004) Optimization of tandem queue systems with finite buffers. Computers & Operations Research 31: 963–984
14. Marie RA (1980) Calculating equilibrium probabilities for $\lambda(n)/C_k/1/N$ queue. Proceedings Performance '80, Toronto, pp 117–125
15. Mickens R (1987) Difference equations. Van Nostrand-Reinhold, New York
16. Mitrani I, Mitra D (1992) A spectral expansion method for random walks on semi-infinite strips. In: Beauwens R, de Groen P (eds) Iterative methods in linear algebra, pp 141–149. North-Holland, Amsterdam
17. Perros HG (1989) A bibliography of papers on queueing networks with finite capacity queues. Perf Eval 10: 255–260
18. Perros HG (1994) Queueing networks with blocking. Oxford University Press, Oxford
19. Perros HG, Altiok T (1989) Queueing networks with blocking. North-Holland, Amsterdam
20. MacGregor Smith J, Cruz FRB (2000) The buffer allocation problem for general finite buffer queueing networks. http://citeseer.nj.nec.com/smith00buffer.html
21. Tahilramani H, Manjunath D, Bose SK (1999) Approximate analysis of open network of $GE/GE/m/N$ queues with transfer blocking. MASCOTS'99, pp 164–172

22. Tijms HC (1994) Stochastic models: an algorithmic approach. Wiley, Chichester
23. Tolio T, Matta A, Gershwin SB (2002) Analysis of two-machine lines with multiple failure modes. IIE Transactions 34: 51–62
24. van Vuuren M (2003) Performance analysis of multi-server tandem queues with finite buffers. Master's Thesis, University of Technology Eindhoven, The Netherlands
25. http://poisson.ecse.rpi.edu/ hema/qnat/