

PERFORMANCE ANALYSIS OF MULTI-SERVER TANDEM QUEUES WITH FINITE BUFFERS

Marcel van Vuuren^{a,b}, Ivo J.B.F. Adan^a and Simone A. E. Resing-Sassen^b

^aEindhoven 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

^bCQM 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. Comparison with simulation shows that the method produces very accurate results, and therefore it is very useful for the design and analysis of production lines.

Key words: multi-server tandem queues, approximation, decomposition, finite buffers, blocking, production lines.

1 Introduction

Queueing networks with finite buffers have been studied extensively in the literature; see, e.g., [1] and [5]. These models have many applications in manufacturing, communication and computer systems. Most studies, however, consider *single-server* models. In this paper we propose a method for the approximative analysis of *multi-server* tandem queues with general service times, finite buffers and blocking after service (BAS). We are interested in the queue-length distribution at each buffer; these distributions may be used to determine performance characteristics, such as the throughput and mean sojourn time.

The model we analyze in this paper is as follows. 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 servers. The random variable P_i denotes the service time of a server in group M_i ; P_i is generally distributed with rate $\mu_{p,i}$ and coefficient of variation $c_{p,i}$. Each server can serve one customer at a time. 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.

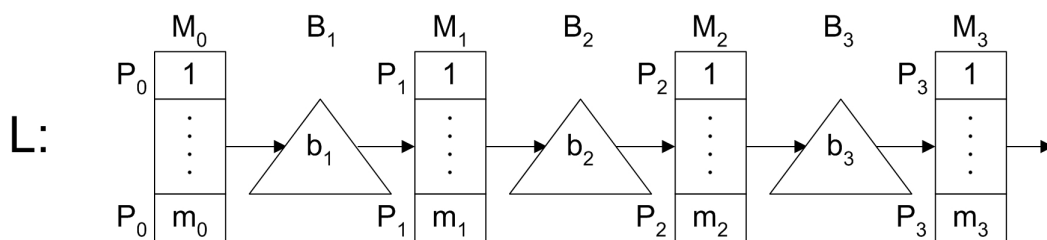


Figure 1: A tandem queue with four server-groups.

Our method to approximate the queue-length distribution of the buffers is based on decomposition of the tandem queue in subsystems and on the first two moments of the service times. Each buffer is considered in isolation with its own arrival and departure processes. By means of an iterative algorithm the parameters of the processes are tuned. To approximate the arrival and departure processes of the subsystems, Erlang $_{k-1,k}$ or Coxian $_2$ distributions are fitted on the first two moments of the inter-arrival

and inter-departure times. Further, each multi-server subsystem is approximated by a single (super) server model.

Decomposition techniques have also been used by a.o. Perros [6] and Kerbache and MacGregor Smith [3]. Their methods deal with single-server queueing networks. To the best of our knowledge, the only methods for multi-server queueing networks with finite buffers available in the literature are presented in Tahilramani et al. [7] and Jain and MacGregor Smith [2]. These methods however, do not cover general service times. An excellent survey on manufacturing flow lines with finite buffers is presented by Dallery and Gershwin [1].

The paper is organized as follows. In Section 2 we explain the decomposition of the tandem queue in subsystems. In the section thereafter we analyze the subsystems. Section 4 describes the iterative algorithm. Numerical results are presented in Section 5 and they are compared with simulation and an existing method. Finally, Section 6 contains some concluding remarks.

2 Decomposition

We decompose the original tandem queue L 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. In Figure 2 we see the decomposition of line L of Figure 1.

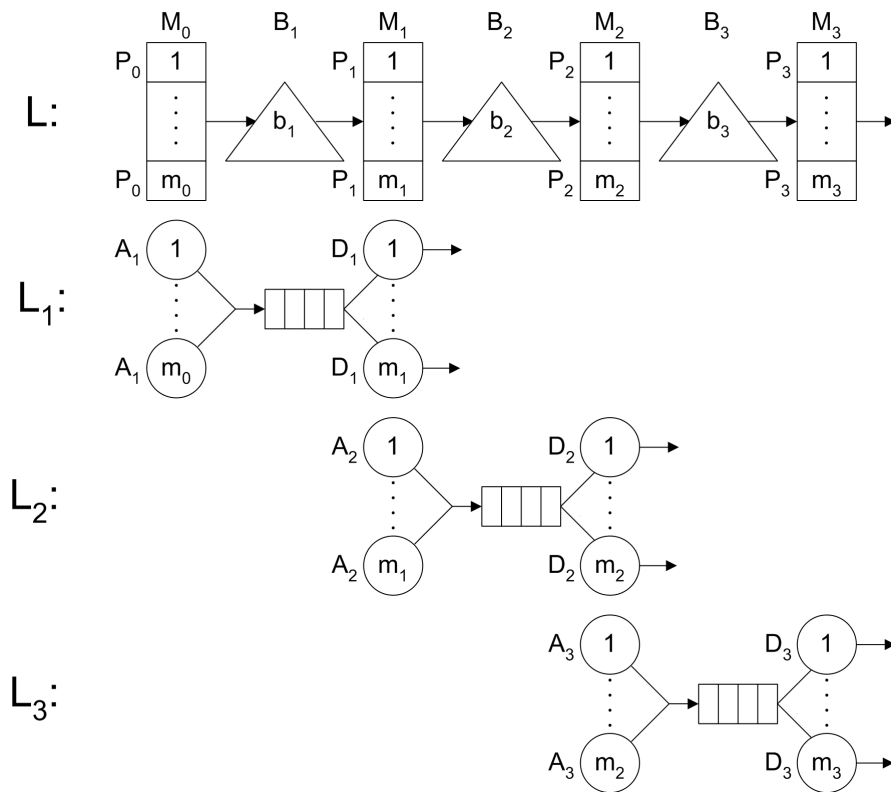


Figure 2: Decomposition of the tandem queue of Figure 1.

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 an original 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, also the characteristics of the complete tandem queue, such as the throughput and mean sojourn time, can be approximated.

3 Subsystems

In this section we briefly describe the analysis of a subsystem.

3.1 Approximating the arrival and departure times

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 1 *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 approximated by the inter-departure time of the superposition of m_{i+1} independent (departure-)service processes with service times D_{i+1} . Figure 3 displays a phase-representation of the service time of a departure-server of subsystem L_i , where SD_{i+1} and RSD_{i+1} indicate the inter-departure time and residual inter-departure time of server-group M_{i+1} , respectively.

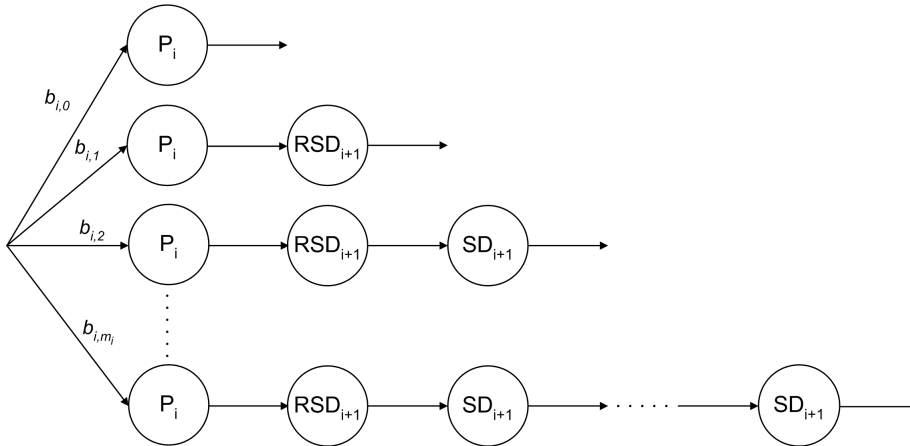


Figure 3: The (approximate) service time D_i of a departure-server of subsystem L_i .

The random variable SD_i , the inter-departure time of server-group M_i , has rate $\mu_{sd,i}$ and coefficient of variation $c_{sd,i}$. The random variable RSD_i , the residual inter-departure time of server-group M_i , has rate $\mu_{rsd,i}$ and coefficient of variation $c_{rsd,i}$. The values of the rates, the coefficients of variation as well as the blocking probabilities, will become known during the execution of the iterative algorithm.

The service times A_i of the arrival-servers 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 1 residual inter-departure time and $j - 1$ full inter-departure times from the preceding server-group M_{i-1} .

3.2 Two moment fit

We will model the distribution of a random variable with rate μ and coefficient of variation c as a Coxian₂ distribution if $c^2 \geq 0.5$, and otherwise, as a mixed Erlang _{$k-1,k$} distribution. For fitting a Coxian₂ distribution with parameters μ_1, μ_2 and p we use the set suggested by [4]:

$$\mu_1 = 2\mu, \quad p = \frac{1}{2c^2}, \quad \mu_2 = \mu_1 p.$$

For fitting an Erlang _{$k-1,k$} with parameters ν and p we use the set suggested by [8]:

$$p = \frac{kc^2 - \sqrt{k(1+c^2) - k^2c^2}}{1+c^2}, \quad \nu = (k-p)\mu.$$

where $k(> 1)$ is chosen such that

$$\frac{1}{k} \leq c^2 \leq \frac{1}{k-1}.$$

There exist also other parameter choices for fitting these distributions, and other distributions for fitting, like the hyper-exponential distribution. Using other distributions or parameters does not affect the quality of our model.

3.3 Analyzing a subsystem

By fitting Coxian or Erlang distributions on the service times A_i and D_i , subsystem L_i can be described by a finite state Markov process; we will further simplify this system by replacing the arrival- and departure-servers of L_i by *super servers* with *state-dependent* service times. Then the number of states of this Markov process reduces to the maximal number of customers in subsystem L_i (which is $m_i + b_i = n_i$) plus the number of servers in server-group M_{i-1} (which is m_{i-1}). Below we describe the states, and the flow in and out of each state.

In state m_i, \dots, n_i both arrival- and departure-servers are working at full power, because there is no blocking or starvation. So the arrival process is the superposition of the service processes of m_{i-1} arrival-servers (with service times A_i), and the departure process is the superposition of the service processes of m_i departure-servers (with service times D_i). In state $j = 0, \dots, m_i - 1$ there are $m_i - j$ departure-servers starved; the arrival-servers are working at full power. Hence the arrival process is the superposition of the m_{i-1} service processes of the arrival-servers and the departure process is the superposition of the j service processes of the (non-starved) departure-servers. Finally, in state $j = n_i + 1, \dots, n_i + m_{i-1}$ there are $j - n_i$ arrival-servers blocked and the departure-servers are all working. So the arrival process is the superposition of the $n_i + m_{i-1} - j$ service processes of the (non-blocked) arrival servers and the departure process is the superposition of m_i service processes of the departure-servers.

Now we replace both arrival-servers and departure-servers by a single super server. The service times of the two super servers are the inter-arrival and inter-departure times of the arrival and departure processes of subsystem L_i , and we fit again Coxian or Erlang distributions on the service times of the super servers (as described in Section 3.1).

The steady-state queue-length distribution of this system can be determined efficiently by using the spectral expansion method; see [9]. This gives us the probabilities $p_{i,j}$, $j = 0, \dots, n_i + m_{i-1}$, where $p_{i,j}$ is the probability that subsystem L_i is in state j . From these queue-length distributions we can easily derive other performance measures.

4 An iterative algorithm

We will now describe the iterative algorithm for approximating the characteristics of tandem queue L . The algorithm is based on the decomposition of L in $M - 1$ subsystems L_1, L_2, \dots, L_{M-1} . Before going into detail in Section 4.2, we present the outline of the algorithm in Section 4.1.

4.1 Outline of the algorithm

- Step 0: Determine initial characteristics of the departure processes for all subsystems L_1, \dots, L_{M-1} .
- Step 1: For subsystem $L_i = L_1, \dots, L_{M-1}$:
 - (a) 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} .
 - (b) Determine the queue-length distribution of subsystem L_i .
 - (c) Determine the throughput T_i of subsystem L_i .
- Step 2: Determine the new characteristics of the departure processes for all subsystems L_{M-1}, \dots, L_1 .
- Repeat Step 1 and 2 until the characteristics of the departure processes have converged.

4.2 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 P_i .

Step 1: Evaluation of subsystems

We know what the departure processes of L_i look like, but we also need to know the characteristics of its arrival processes, before we are able to determine the queue-length distribution of L_i .

(a) The arrival process

For the first subsystem L_1 , the characteristics of A_1 are the same as those of P_0 , because the servers of M_0 cannot be starved.

For the other subsystems we proceed as follows. By Little's law we have for the throughput T_i of subsystem L_i ,

$$T_i = \left(1 - \sum_{j=1}^{m_{i-1}} p_{i,n_i+j} \right) m_{i-1} \mu_{a,i} + \sum_{j=1}^{m_{i-1}-1} p_{i,n_i+j} (m_{i-1} - j) \mu_{a,i}.$$

By substituting the estimate $T_{i-1}^{(k)}$ for T_i and $p_{i,n_i+j}^{(k-1)}$ for p_{i,n_i+j} we get as new estimate for the service rate $\mu_{a,i}$,

$$\mu_{a,i}^{(k)} = \frac{T_{i-1}^{(k)}}{\left(1 - \sum_{j=1}^{m_{i-1}} p_{i,n_i+j}^{(k-1)} \right) m_{i-1} + \sum_{j=1}^{m_{i-1}-1} p_{i,n_i+j}^{(k-1)} (m_{i-1} - j)},$$

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

The coefficient of variation of A_i cannot be determined in this way; to approximate the coefficient of variation we use the model for A_i described in section 3.1. We know the characteristics of P_{i-1} and we can also determine the characteristics of RSA_{i-1} and SA_{i-1} ; see [9] for details.

(b) Analysis of subsystem L_i

Based on the (new) characteristics of both arrival and departure processes we can determine the steady-state distribution of subsystem L_i . To do so we first need to fit Coxian₂ 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 3.2. Then we calculate the equilibrium probabilities $p_{i,j}$ as described in Section 3.3.

(c) Determining the throughput of L_i

Once the steady-state distribution is known, we can determine the new throughput $T_i^{(k)}$ according to

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

We also determine new estimates for the probabilities $b_{i,j}$ that j servers of server-group M_{i-1} are blocked after service completion and the probabilities $s_{i,j}$ that j servers of server-group M_i are starved after service completion; for more details the reader is referred to [9].

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

Step 2: The departure process

Now we have new information about the departure processes of the subsystems. So we can recalculate 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 P_{M-1} , because the servers in server-group M_{M-1} can never be blocked.

The new rate of D_i is determined from

$$\mu_{d,i}^{(k)} = \frac{T_{i+1}^{(k)}}{\left(1 - \sum_{j=0}^{m_i-1} p_{i,j}^{(k)} \right) m_i + \sum_{j=1}^{m_i-1} p_{i,j}^{(k)} j}$$

The calculation of the new coefficient of variation of D_i is similar to the one of A_i ; see [9].

Convergence

After Step 1 and 2 we can check whether the iterative algorithm has converged or not. We check this by comparing the departure rates in the $(k-1)$ -th and k -th iteration. When the sum of the absolute values of the differences between these rates is less than ε we stop; otherwise we repeat Step 1 and 2. So the stop-criterion is

$$\sum_{i=1}^{M-1} \left| \mu_i^{(k)} - \mu_i^{(k-1)} \right| < \varepsilon.$$

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 'nothing' changes anymore.

5 Numerical Results

In this section we will present some results. To verify the accuracy of our method we compare it with discrete event simulation. After that, we compare our method with the method developed by Tahilramani et al. [7], which is implemented in QNAT [11].

5.1 Comparison with simulation

In order to verify the quality of our method we compared the throughput and the mean sojourn time with the ones produced by discrete event simulation. 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%.

We tested two different lengths M of tandem queues, namely with 4 and 8 server-groups. For each tandem queue we varied the number of servers m_i at the server-groups; we used tandems with 1 server per server-group, 5 servers per server-group and with the sequence (4, 1, 2, 8). We also varied 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 each machine varies between 0.1, 0.2, 0.5, 1, 1.5 and 2. Finally we varied 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 gives 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 Table 6.

Buffer sizes (b_i)	Error in throughput					Error in mean sojourn time				
	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 1: Overall results for tandem queues with different buffer sizes.

Rates unbalanced server-group ($m_i\mu_{p,i}$)	Error in throughput					Error in mean sojourn time				
	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 2: Overall results for tandem queues with different balancing rates.

Coefficients of variation ($c_{p,i}^2$)	Error in throughput					Error in mean sojourn time				
	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 3: Overall results for tandem queues with different coefficients of variation.

Number of servers (m_i)	Error in throughput					Error in mean sojourn time				
	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 %

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

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

m_i	$m_i \mu_{p,i}$	$c_{p,i}^2$	Buffers	T App.	T Sim.	Diff.	S App.	S Sim.	Diff.
(1,1,1,1)	(1,1,1,1)	0.1	0	0.735	0.771	-4.7 %	4.70	4.63	1.5 %
(1,1,1,1)	(1,1,1,1)	0.1	10	0.981	0.985	-0.4 %	19.22	19.03	1.0 %
(1,1,1,1)	(1,1,1,1)	1.0	2	0.703	0.700	0.4 %	9.09	9.25	-1.7 %
(1,1,1,1)	(1,1,1,1)	1.5	0	0.504	0.473	6.6 %	5.82	6.27	-7.2 %
(1,1,1,1)	(1,1,1,1)	1.5	10	0.834	0.835	-0.1 %	22.38	22.31	0.3 %
(1,1,1,1)	(1,1,1,5,1)	0.1	2	0.960	0.958	0.2 %	6.18	6.41	-3.6 %
(1,1,1,1)	(1,1,1,5,1)	1.0	0	0.594	0.561	5.9 %	4.84	5.28	-8.3 %
(1,1,1,1)	(1,1,1,5,1)	1.0	10	0.918	0.912	0.7 %	16.20	17.41	-7.0 %
(1,1,1,1)	(1,1,1,5,1)	1.5	2	0.714	0.691	3.3 %	8.03	8.60	-6.6 %
(5,5,5,5)	(1,1,1,1)	0.1	0	0.789	0.856	-7.8 %	22.48	21.78	3.2 %
(5,5,5,5)	(1,1,1,1)	0.1	10	0.927	0.983	-5.7 %	36.88	35.24	4.7 %
(5,5,5,5)	(1,1,1,1)	1.0	2	0.797	0.808	-1.4 %	26.37	26.17	0.8 %
(5,5,5,5)	(1,1,1,1)	1.5	0	0.742	0.724	2.5 %	22.99	23.90	-3.8 %
(5,5,5,5)	(1,1,1,1)	1.5	10	0.867	0.874	-0.8 %	37.97	38.86	-2.3 %
(5,5,5,5)	(1,1,1,5,1)	0.1	2	0.902	0.958	-5.8 %	21.63	21.50	0.6 %
(5,5,5,5)	(1,1,1,5,1)	1.0	0	0.801	0.794	0.9 %	20.79	21.13	-1.6 %
(5,5,5,5)	(1,1,1,5,1)	1.0	10	0.927	0.929	-0.2 %	30.37	32.61	-6.9 %
(5,5,5,5)	(1,1,1,5,1)	1.5	2	0.850	0.828	2.7 %	21.95	23.70	-7.4 %
(4,1,2,8)	(1,1,1,1)	0.1	0	0.746	0.793	-5.9 %	16.19	16.28	-0.6 %
(4,1,2,8)	(1,1,1,1)	0.1	10	0.956	0.984	-2.8 %	31.61	30.05	5.2 %
(4,1,2,8)	(1,1,1,1)	1.0	2	0.756	0.757	-0.1 %	20.15	20.14	0.0 %
(4,1,2,8)	(1,1,1,1)	1.5	0	0.633	0.619	2.3 %	16.78	18.01	-6.8 %
(4,1,2,8)	(1,1,1,1)	1.5	10	0.850	0.856	-0.7 %	31.43	32.37	-2.9 %
(4,1,2,8)	(1,1,1,5,1)	0.1	2	0.920	0.953	-3.5 %	16.72	17.14	-2.5 %
(4,1,2,8)	(1,1,1,5,1)	1.0	0	0.714	0.702	1.7 %	16.22	16.43	-1.3 %
(4,1,2,8)	(1,1,1,5,1)	1.0	10	0.926	0.919	0.8 %	25.99	27.60	-5.8 %
(4,1,2,8)	(1,1,1,5,1)	1.5	2	0.787	0.773	1.8 %	17.52	18.93	-7.4 %
(1,1,1,1,1,1,1,1)	(1,1,1,1,1,1,1,1)	0.1	2	0.906	0.926	-2.2 %	16.14	15.99	0.9 %
(1,1,1,1,1,1,1,1)	(1,1,1,1,1,1,1,1)	1.0	0	0.488	0.443	10.2 %	11.73	13.43	-12.7 %
(1,1,1,1,1,1,1,1)	(1,1,1,1,1,1,1,1)	1.0	10	0.855	0.855	0.0 %	49.52	49.81	-0.6 %
(1,1,1,1,1,1,1,1)	(1,1,1,1,1,1,1,1)	1.5	2	0.607	0.581	4.5 %	21.94	23.52	-6.7 %
(1,1,1,1,1,1,1,1)	(1,1,1,1,1,5,1,1,1)	0.1	0	0.718	0.751	-4.4 %	8.90	9.27	-4.0 %
(1,1,1,1,1,1,1,1)	(1,1,1,1,1,5,1,1,1)	0.1	10	0.980	0.983	-0.3 %	38.45	43.22	-11.0 %
(1,1,1,1,1,1,1,1)	(1,1,1,1,1,5,1,1,1)	1.0	2	0.690	0.670	3.0 %	18.81	20.31	-7.4 %
(1,1,1,1,1,1,1,1)	(1,1,1,1,1,5,1,1,1)	1.5	0	0.482	0.409	17.8 %	11.26	13.79	-18.3 %
(1,1,1,1,1,1,1,1)	(1,1,1,1,1,5,1,1,1)	1.5	10	0.830	0.819	1.3 %	46.75	50.16	-6.8 %
(5,5,5,5,5,5,5,5)	(1,1,1,1,1,1,1,1)	0.1	2	0.827	0.926	-10.7 %	52.35	49.71	5.3 %
(5,5,5,5,5,5,5,5)	(1,1,1,1,1,1,1,1)	1.0	0	0.693	0.697	-0.6 %	49.20	49.14	0.1 %
(5,5,5,5,5,5,5,5)	(1,1,1,1,1,1,1,1)	1.0	10	0.867	0.882	-1.7 %	83.09	83.96	-1.0 %
(5,5,5,5,5,5,5,5)	(1,1,1,1,1,1,1,1)	1.5	2	0.759	0.737	3.0 %	54.63	57.27	-4.6 %
(5,5,5,5,5,5,5,5)	(1,1,1,1,1,5,1,1,1)	0.1	0	0.781	0.851	-8.2 %	43.03	42.65	0.9 %
(5,5,5,5,5,5,5,5)	(1,1,1,1,1,5,1,1,1)	0.1	10	0.922	0.983	-6.2 %	71.89	73.95	-2.8 %
(5,5,5,5,5,5,5,5)	(1,1,1,1,1,5,1,1,1)	1.0	2	0.789	0.787	0.3 %	51.52	53.49	-3.7 %
(5,5,5,5,5,5,5,5)	(1,1,1,1,1,5,1,1,1)	1.5	0	0.730	0.692	5.5 %	44.43	47.95	-7.3 %
(5,5,5,5,5,5,5,5)	(1,1,1,1,1,5,1,1,1)	1.5	10	0.864	0.862	0.2 %	74.69	81.01	-7.8 %
(4,1,2,8,4,1,2,8)	(1,1,1,1,1,1,1,1)	0.1	2	0.845	0.921	-8.3 %	39.90	38.96	2.4 %
(4,1,2,8,4,1,2,8)	(1,1,1,1,1,1,1,1)	1.0	0	0.619	0.604	2.5 %	37.90	38.55	-1.7 %
(4,1,2,8,4,1,2,8)	(1,1,1,1,1,1,1,1)	1.0	10	0.863	0.871	-0.9 %	71.67	71.74	-0.1 %
(4,1,2,8,4,1,2,8)	(1,1,1,1,1,1,1,1)	1.5	2	0.705	0.678	4.0 %	43.38	46.32	-6.3 %
(4,1,2,8,4,1,2,8)	(1,1,1,1,1,5,1,1,1)	0.1	0	0.744	0.790	-5.8 %	30.96	32.41	-4.5 %
(4,1,2,8,4,1,2,8)	(1,1,1,1,1,5,1,1,1)	0.1	10	0.945	0.983	-3.9 %	61.00	62.54	-2.5 %
(4,1,2,8,4,1,2,8)	(1,1,1,1,1,5,1,1,1)	1.0	2	0.750	0.742	1.1 %	39.64	42.20	-6.1 %
(4,1,2,8,4,1,2,8)	(1,1,1,1,1,5,1,1,1)	1.5	0	0.628	0.588	6.8 %	32.68	37.66	-13.2 %
(4,1,2,8,4,1,2,8)	(1,1,1,1,1,5,1,1,1)	1.5	10	0.844	0.843	0.1 %	61.82	69.32	-10.8 %

Table 6: Detailed results for tandem queues with 4 and 8 machine-groups.

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 worse. If we look at the coefficients of variation (Table 3), we get the best approximations for the throughput when the coefficients of variation are 1, and 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.

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 Table 6). The worst performance is obtained for lines with buffers of size zero, with servers with high coefficients of variation and very unbalanced lines, but these cases are unlikely (and undesired) to occur in practice.

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

5.2 Comparison with QNAT

We also compared our method with QNAT, a method developed by Tahilramani et al. [7]. We used a tandem queue with four server-groups. It was only possible to test cases with 1 server in the first server-group and exponential service times of that server, because the methods use slightly different models. We varied the number of servers per server-group and the size of buffers. Table 7 shows the results.

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 %

Table 7: Comparison of our method with QNAT.

We see that our 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.

6 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 and used an iterative algorithm to approximate its characteristics. Each multi-server subsystem is approximated by a single (super) server queue with 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 are within 5% of the simulation results. The method can be extended in several directions: one may think of, e.g., more general configurations (splitting, merging, feedback), unreliable machines and assembly/disassembly (see [10]).

References

- [1] Y. Dallery and B. Gershwin (1992) Manufacturing flow line systems: a review of models and analytical results. *Queueing Systems* 12, 3-94.
- [2] S. Jain and J. MacGregor Smith (1994) Open Finite Queueing Networks with $M/M/C/K$ Parallel Servers. *Computers Operations Res.* 21(3), 297-317.
- [3] L. Kerbache and J. MacGregor Smith (1987) The Generalized Expansion Method for Open Finite Queueing Networks. *The European Journal of Operations Research* 32, 448-461.
- [4] R.A. Marie (1980) Calculating equilibrium probabilities for $\lambda(n)/C_k/1/N$ queue. *Proceedings Performance '80, Toronto*, 117-125.
- [5] H.G. Perros (1989) A Bibliography of Papers on Queueing Networks with Finite Capacity Queues. *Perf. Eval.* 10, 255-260.
- [6] H.G. Perros (1994) *Queueing Networks with Blocking*. Oxford University Press.
- [7] H. Tahilramani, D. Manjunath, S.K. Bose (1999) Approximate Analysis of Open Network of $GE/GE/m/N$ Queues with Transfer Blocking. *MASCOTS'99*, 164-172.
- [8] H.C. Tijms (1994) *Stochastic models: an algorithmic approach*. John Wiley & Sons, Chichester.
- [9] M. van Vuuren, I.J.B.F. Adan and S.A.E. Resing-Sassen (2003) Multi-server Tandem Queues with Finite Buffers and Blocking. *SPOR-Report, University of Technology Eindhoven, The Netherlands*.
- [10] M. van Vuuren (2003) Performance Analysis of Multi-Server Tandem Queues with Finite Buffers. *Master's Thesis, University of Technology Eindhoven, The Netherlands*.
- [11] <http://poisson.ecse.rpi.edu/hema/qnat/>