

THROUGHPUT ANALYSIS OF TWO-STAGE MANUFACTURING SYSTEMS WITH MERGE AND BLOCKING[†]

YANG WOO SHIN* AND DUG HEE MOON

ABSTRACT. Parallel lines are often used to increase production rate in many manufacturing systems where the main line splits into several lines in parallel, and after some operations, they merge into a main line again. Queueing networks with finite buffers have been widely used for modeling and analyzing manufacturing systems. This paper provides an approximation technique for multi-server two-stage networks with merge configuration and blocking which will be a building block for analysis of general manufacturing systems with parallel lines and merge configuration. The main idea of the method is to decompose the original system into subsystems that have two service stations with multiple servers, two buffers and external arrivals to the second stage are allowed. The subsystems are modeled by level dependent quasi-birth-and-death (LDQBD) process.

AMS Mathematics Subject Classification : 60K25, 68M20.

Key words and phrases : parallel line, queueing networks, finite buffers, multi-server, merge configuration, blocking.

1. Introduction

A complex manufacturing system usually consists of one or more complex operations, such as assembly, disassembly, split, merge, rework, parallel, feed-forward and scrap, etc. [10]. Parallel lines are often used to increase production rate in many manufacturing systems. In parallel systems, the main line splits into several lines in parallel, and after some operations, they merge into a main line again. For example, three types of lens module installed in smart phone are assembled in their own assembly line. After finishing assembly, all lens modules are sent to the resolution inspection machines which can inspect all types of lens modules automatically.

Received July 3, 2014. Revised October 12, 2014. Accepted October 17, 2014. *Corresponding author. [†]A part of this paper was presented in ICAMM2014, PSG College of Technology, Coimbatore, India. *This research was supported by Changwon National University in 2013–2014.

Queueing networks with finite buffers have been widely used for modeling and analyzing many manufacturing systems. However, exact analytical results for queueing networks with finite buffers are limited only to some special systems with one or two queues [4, 6, 13, 16, 19]. So many approximation methods are presented for analysis of queueing networks with finite buffers. Most of the approximation methods for serial lines or assembly systems are either based on decomposition or on aggregation approaches [3]. The principal procedure of decomposition methods is to decompose the long line into subsystems that are mathematically tractable, and derives a set of equations that determine the unknown parameters of each subsystem, and finally develops an algorithm to solve these equations. The basic idea of aggregation method is to replace a two-node system by a single equivalent node and reduce the number of nodes in the network. For a detailed list references for approximations and their application to manufacturing system, see review papers [3, 11, 14] and monographs [1, 2, 7, 12, 15]. The literature on the analysis of queueing networks with finite buffer and multiple servers is more scarce than the case of single server system, e.g. see [5, 18] which are focused on the approximation of serial lines with reliable servers. Almost all decomposition method use subsystems with two nodes and single buffer between them. Recently, approximate analysis for multi-server tandem queues with exponential service times and finite buffers is presented in [17] where the subsystems with two nodes and two buffers are used to reflect the dependence between consecutive stages and improves the quality of approximation.

The literature on the approximation for the system with parallel lines with multiple servers and finite buffer is very limited. The purpose of this paper is to provide an approximation method for multi-server two-stage networks with merge configuration and blocking which will be a building block for analysis of multi-server-finite-buffer queueing networks with parallel lines and merge configuration. The main idea of the method is to decompose the original system into subsystems that have two service stations with multiple servers, two buffers and external arrivals to the second stage are allowed. The subsystems are modeled by level dependent quasi-birth-and-death (LDQBD) process.

This paper is organized as follows. In Section 2, the model is described in detail. Approximation procedure is presented in Section 3. Numerical results and concluding remarks are presented in Sections 4 and 5, respectively.

2. Model description

We model the manufacturing with parallel lines by the two-stage open queueing network in which the first stage consists of J multi-server queues with finite buffers and the second stage is one queue with multiple servers and finite buffer. Two stages are linked in merge configuration as depicted in Fig. 1. The queues (merging queues) at the first stage will be denoted by node i , $i = 1, 2, \dots, J$, and the queue (merged queue) at the second stage is denoted by node 0. The node i consists of a service station M_i with m_i parallel identical servers and a

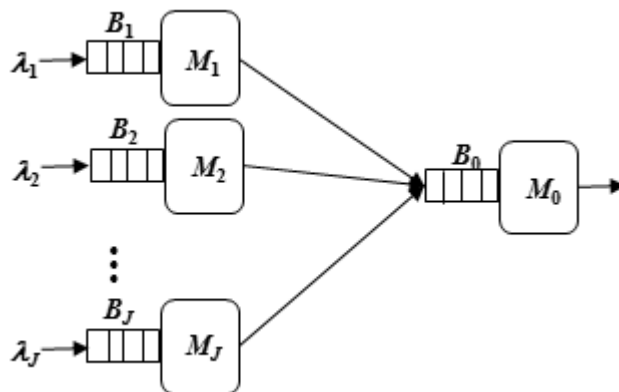


FIGURE 1. A two-stage queueing network with merge configuration

buffer B_i of size b_i and let $l_i = m_i + b_i$, $i = 0, 1, 2, \dots, J$. The index i can be viewed as the sequential number of the nodes at the first stage runs from 1 to J unless otherwise stated. The service time of each server at M_i is exponentially distributed with rate μ_i , $i = 0, 1, 2, \dots, J$. Customers arrive to node i from outside according to a Poisson process with rate λ_i , $i = 1, 2, \dots, J$. The arrivals to any node at the first stage are dispatched to some of its servers if there is one free, and if all servers are occupied the request waits in the corresponding buffer. When an arrival finds no free position in the buffer (buffer is full), it is lost. After completion of the service at node of the first stage the request proceeds to node 0, and is taken immediately into service if one of the servers is free (no queue at node 0), otherwise is placed into the buffer B_0 and waits there until a server is available. If buffer B_0 is full at the instant of service completion at node i , the customer is forced to stay at node i until a space becomes available at node 0. During this period, the server just completed its service is blocked. This type of blocking mechanism is called blocking-after-service (BAS). Since there are more than one node directly linked to node 0, the servers in more than one upstream node can be blocked simultaneously. In this case, we assume that the corresponding blocked customers enter node 0 on a random dispatching rule. Let $\alpha_j(y_1, \dots, y_J)$ be the probability that a blocked customers at node j joins the node 0 upon a space in B_0 is available when there are y_i servers blocked at node i , $i = 1, 2, \dots, J$. A customer who has completed service at node 0 leaves the system.

Let X_i be the number of customers at node i which includes the customers being served at M_i and waiting requests at B_i but does not include the customers blocked to enter node 0 and Y_i the number of blocked servers at M_i , $i = 1, 2, \dots, J$ in stationary state. By X_0 denote the number of customers at node 0 that include the customers blocked at node i , $i = 1, 2, \dots, J$. It

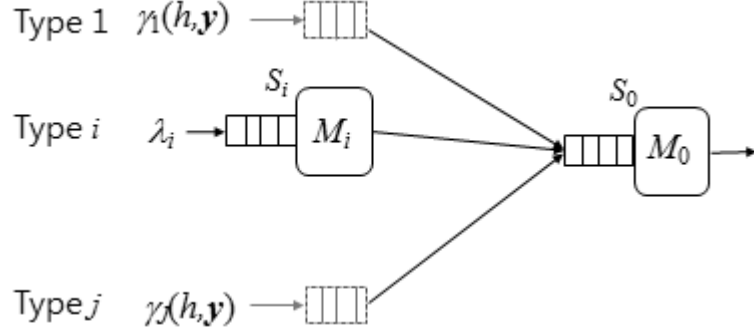


FIGURE 2. Two-node tandem queue L_i with virtual buffers and external arrivals to the second node

can be seen that $0 \leq X_0 \leq H$, where $H = l_0 + m$ with $m = \sum_{i=1}^J m_i$. Let $\mathbf{X} = (X_1, \dots, X_J)$ and $\mathbf{Y} = (Y_1, \dots, Y_J)$. Then the state space of $(X_0, \mathbf{X}, \mathbf{Y})$ is $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1$, where $\mathcal{S}_0 = \cup_{h=0}^{l_0} \{(h, \mathbf{x}, \mathbf{0}) : 0 \leq x_i \leq l_i, i = 1, 2, \dots, J\}$ and $\mathcal{S}_1 = \cup_{k=1}^m \{(l_0 + k, \mathbf{x}, \mathbf{y}) : \mathbf{y}\mathbf{e} = k, \max(k - m + m_i, 0) \leq y_i \leq \min(k, m_i), 0 \leq x_i + y_i \leq l_i, i = 1, 2, \dots, J\}$ where $\mathbf{x} = (x_1, \dots, x_J)$, $\mathbf{y} = (y_1, \dots, y_J)$ and \mathbf{e} is the column vector of appropriate size whose components are all 1. The state space \mathcal{S} is so complex and its size increase drastically as the number J of nodes and/or m_i, b_i increase.

Main goal of this analysis is to provide an approximation method for the mean departure rate from node 0 that is the throughput of the system.

3. Approximation

In this section, we derive an algorithm for approximate analysis of throughput of the system. First we fix a node i . For an investigation of the behavior of node i and node 0, we consider the two-node queue denoted by L_i where the arrivals from outside to the second stage are allowed as depicted in Fig.2. We denote the first node and the second node by \hat{S}_i and \hat{S}_0 , respectively. The node \hat{S}_i is the same as node i of the original system and has a service station, that is, there are m_i exponential servers with rate μ_i in parallel and a finite buffer of size b_i . The BAS blocking rule is assumed in the system. The node \hat{S}_0 consists of a service station M_0 with m_0 exponential servers with rate μ_0 in parallel and a buffer \hat{B}_0 of size b_0 . In order to reflect the effects of node $j, j \neq i$, we assume that \hat{S}_0 has $J - 1$ virtual buffers \hat{B}_j whose size is of $m_j, j = 1, 2, \dots, J, j \neq i$ and type j customers arrive from outside to queue \hat{S}_0 . If a type j customer finds that the queue \hat{S}_0 is full upon arrival, the customer is blocked and stayed at the virtual buffer \hat{B}_j until the queue \hat{S}_0 can accommodate the customer. We assume that the distribution of inter-arrival time of type $j, j \neq i$ customer is exponential with

$(X_i^*, Y_i^*) = (n, \mathbf{y})$. The components of $C_i^{(h)}$ are given as follows: for $1 \leq h \leq l_0$,

$$C_i^{(h)}[(n, \mathbf{y}), (n', \mathbf{y}')] = \begin{cases} \min(h, m_0)\mu_0, & (n', \mathbf{y}') = (n, \mathbf{y}) \\ 0, & \text{otherwise,} \end{cases}$$

and for $l_0 + 1 \leq h \leq H$,

$$C_i^{(h)}[(n, \mathbf{y}), (n', \mathbf{y}')] = \begin{cases} m_0\mu_0\alpha_k(\mathbf{y}), & (n', \mathbf{y}') = (n, \mathbf{y} - \mathbf{e}_k), 1 \leq k \leq J \\ 0, & \text{otherwise,} \end{cases}$$

Let X_0^* , X_i^* and \mathbf{Y}^* be the stationary version of $X_0^*(t)$, $X_i^*(t)$ and $\mathbf{Y}^*(t)$, respectively and

$$\begin{aligned} \pi_i(h, n, \mathbf{y}) &= P((X_0^*, X_i^*, \mathbf{Y}^*) = (h, n, \mathbf{y})), (h, n, \mathbf{y}) \in \mathcal{S}_i \\ \boldsymbol{\pi}_i(h) &= (\pi_i(h, n, \mathbf{y}), (h, n, \mathbf{y}) \in \mathbf{h}), h = 0, 1, \dots, H. \end{aligned}$$

The stationary distribution $\boldsymbol{\pi}_i = (\boldsymbol{\pi}_i(0), \boldsymbol{\pi}_i(1), \dots, \boldsymbol{\pi}_i(H))$ satisfies the following equations

$$\boldsymbol{\pi}_i(n) = \boldsymbol{\pi}_i(n-1)R_i^{(n)}, n = 1, 2, \dots, H,$$

where $R_i^{(n)}$ is obtained recursively as follows

$$\begin{aligned} R_i^{(H)} &= A_i^{(H-1)}(-B_i^{(H)})^{-1} \\ R_i^{(n)} &= A_i^{(n-1)}[-(B_i^{(n)} + R_i^{(n+1)}C_i^{(n+1)})]^{-1}, n = H-1, \dots, 2, 1. \end{aligned}$$

The vector $\boldsymbol{\pi}_i(0)$ is the unique solution of the equations

$$\begin{aligned} \boldsymbol{\pi}_i(0)(B_i^{(0)} + R_i^{(1)}C_i^{(1)}) &= 0 \\ \boldsymbol{\pi}_i(0) \left[\mathbf{e} + \sum_{1 \leq n \leq H} R_i^{(1)} \cdots R_i^{(n)} \mathbf{e} \right] &= 1. \end{aligned}$$

Noting that the number Ψ_i of busy servers at \hat{M}_i is $\Psi_i = \min(X_i^*, m_i - Y_i^*)$, once $\boldsymbol{\pi}_i$ is given, we can calculate the approximation formulae for arrival rate from node i to node 0 given $X_0^* = h$, $\mathbf{Y}^* = \mathbf{y} = (y_1, \dots, y_J)$ as follows :

$$\begin{aligned} \gamma_i(h, \mathbf{y}) &= \mu_i E[\Psi_i | X_0^* = h, \mathbf{Y}^* = \mathbf{y}] \\ &= \mu_i \sum_{n=0}^{l_i} E[\Psi_i | \mathbf{Z}^* = (h, n, \mathbf{y})] P(X_i^* = n | X_0^* = h, \mathbf{Y}^* = \mathbf{y}) \\ &= \sum_{\{n : (h, n, \mathbf{y}) \in \mathbf{h}\}} \mu_i(n, y_i) \pi_i(h, n, \mathbf{y}) / P_i(h, \mathbf{y}), \end{aligned} \quad (1)$$

where $P_i(h, \mathbf{y}) = \sum_{\{n : (h, n, \mathbf{y}) \in \mathbf{h}\}} \pi_i(h, n, \mathbf{y})$ and the arrival rate from node i to node 0 given $X_0^* = h$

$$\begin{aligned} \gamma_i(h) &= \mu_i E[\Psi_i | X_0^* = h] \\ &= \sum_{\{\mathbf{y} \in \mathcal{S}_i(h, \mathbf{y})\}} \gamma_i(h, \mathbf{y}) P_i(h, \mathbf{y}) / P(X_0^* = h) \end{aligned} \quad (2)$$

where $\mathcal{S}_i(h, \mathbf{y}) = \{\mathbf{y} : \mathbf{y}\mathbf{e} = \max(h - l_0, 0)\}$ and $P(X_0^* = h) = \sum_{\{\mathbf{y} \in \mathcal{S}_i(h, \mathbf{y})\}} P_i(h, \mathbf{y})$.

We use the iterative algorithm to determine the parameters and calculate the throughput. Let $L_i^{(k)}$, $k = 0, 1, 2, \dots$ be the system L_i in the k th iteration and $\gamma_j^{(k)}(h, \mathbf{y})$ be the arrival rate to $L_i^{(k)}$.

Algorithm

Step 0: Initial Step: Initially we assume that no servers at node j , $j = 2, \dots, J$ are blocked and consider the node j as ordinary $M/M/m_j/l_j$ queue with arrival rate λ_j and service rate μ_j . Set

$$\gamma_j^{(0)}(h, \mathbf{y}) = \sum_{n=0}^{N_j} p_j(n) \mu_j(n, y_j).$$

where $p_j(n)$ is the stationary distribution of $M/M/m_j/l_j$ queue.

Step 1: For $i = 1, 2, \dots, J$, calculate the stationary distribution $\pi_i^{(k)}$ of $L_i^{(k)}$ with parameters $\gamma_j^{(k)}(h, \mathbf{y})$, $j < i$ and $\gamma_j^{(k-1)}(h, \mathbf{y})$, $j > i$. Update $\gamma_i^{(k)}(h, \mathbf{y})$ using (1).

Step2: Check the stopping criterion

$$TOL = \max_{i, h, \mathbf{y}} |\gamma_i^{(k)}(h, \mathbf{y}) - \gamma_i^{(k-1)}(h, \mathbf{y})| < \epsilon, \quad (3)$$

where $\epsilon > 0$ is a given tolerance. If the stopping condition is satisfied, then stop the iteration. Otherwise, go to Step 1.

Once $\gamma_i(h, \mathbf{y})$ are obtained, calculate the stationary distribution π_i and $\gamma_i(h)$ using (2), $i = 1, 2, \dots, J$. Modeling node 0 as $M/M/m_0/H$ queue with state dependent arrival rate $\lambda_0(h) = \sum_{i=1}^J \gamma_i(h)$, $h = 0, 1, \dots, H$ and service rate μ_0 and calculate throughput (TP)

$$TP = \sum_{h=0}^H \pi_0(h) \min(h, m_0) \mu_0. \quad (4)$$

No proof of convergence or accuracy can be given for this algorithm, as for many similar algorithms for serial line decomposition. However extensive numerical experiments show the convergence of the algorithm.

4. Numerical results

In this section, some numerical results are presented for accuracy of approximations for the performance measures such as throughput, mean number $E[X_0]$ of customers at node 0 and mean number $E[Y] = \sum_{i=1}^J E[Y_i]$ of blocked customers. The selection probability is used as

$$\alpha_j(y_1, \dots, y_J) = \begin{cases} 1/(\sum_{i=1}^J I(y_i = y^*)), & y_i = y^* \\ 0, & \text{otherwise,} \end{cases}$$

where $y^* = \max_{1 \leq i \leq J} y_i$. Note that $\sum_{i=1}^J I(y_i = y^*)$ is the number of nodes in which the number of blocked customers is biggest among other nodes.

TABLE 1. Approximation results for the system with $J = 2$,
 $\mathbf{m} = (4, 3, 2)$, $\mathbf{m}\boldsymbol{\mu} = (2.0, 1.0, 1.0)$

λ	\mathbf{b}	Throughput		$E[X_0]$		$E[Y]$	
		Exact	App(Err(%))	Exact	App(Err(%))	Exact	App(Err(%))
0.5	(0,1,2)	0.9404	0.9403 (0.0)	1.5371	1.5372 (0.0)	0.1265	0.1267 (0.1)
	(2,3,4)	0.9866	0.9866 (0.0)	1.6775	1.6775 (0.0)	0.0359	0.0359 (0.0)
	(5,4,3)	0.9856	0.9856 (0.0)	1.6874	1.6875 (0.0)	0.0036	0.0036 (0.3)
1.0	(0,1,2)	1.4170	1.4164 (0.0)	2.6520	2.6517 (0.0)	0.5265	0.5270 (0.1)
	(2,3,4)	1.5948	1.5945 (0.0)	3.6196	3.6193 (0.0)	0.4151	0.4154 (0.1)
	(5,4,3)	1.6366	1.6364 (0.0)	4.3000	4.3013 (0.0)	0.1947	0.1953 (0.3)
3.0	(0,1,2)	1.6131	1.6131 (0.0)	3.2896	3.2896 (0.0)	0.8699	0.8699 (0.0)
	(2,3,4)	1.7273	1.7273 (0.0)	4.3236	4.3236 (0.0)	0.6566	0.6566 (0.0)
	(5,4,3)	1.8055	1.8055 (0.0)	5.7982	5.7982 (0.0)	0.4636	0.4636 (0.0)

In the following tables, the vectors $\mathbf{m} = (m_0, \dots, m_J)$ and $\mathbf{b} = (b_0, \dots, b_J)$ denote the server allocation and buffer size allocation, respectively. The total service rate allocation is denoted by $\mathbf{m}\boldsymbol{\mu} = (m_0\mu_0, \dots, m_J\mu_J)$. For example, the vector (2, 3, 4) in the column corresponding to \mathbf{b} means that $b_0 = 2$, $b_1 = 3$, $b_2 = 4$ and the vector $\mathbf{m}\boldsymbol{\mu} = (2.0, 1.0, 1.0)$ means $\mu_0 = 2.0/m_0$, $\mu_1 = 1.0/m_1$, $\mu_2 = 1.0/m_2$ in Table 1. The arrival rates from outside to node i are assumed to be the same and denote it by $\lambda_i = \lambda$, $i = 1, \dots, J$ and the tolerance for the stopping criterion (3) is chosen as $\epsilon = 10^{-5}$.

Since the state space of the Markov chain for the system with $J = 2$ is small, we can have the exact results by solving the balance equations. In Table 1, approximation results (App) for the system with two merging nodes ($J = 2$) are compared with exact one (Exact). The relative percentage error is calculated by $\text{Err}(\%) = |\text{App} - \text{Exact}| \times 100/\text{Exact}$. The errors in Table 1 seem to be due to the approximation of arrival rates $\gamma_i(h, \mathbf{y})$ by decomposition. In Tables 2-4, approximation results for the throughput, $E[X_0]$ and $E[Y]$ in the system with $J = 3$, $\mathbf{m} = (5, 4, 3, 2)$, $\mathbf{m}\boldsymbol{\mu} = (2.5, 1.0, 1.0, 1.0)$ are compared with those of simulation (Sim). Simulation models are developed with ARENA [9]. Simulation run time is set to 220,000 unit times including 20,000 unit times of warm-up period. Ten replications are conducted for each case and the average value and the half length of 95% confidence interval (c.i.) are obtained. The deviation (Dev) between simulation and approximation in Tables 2-4 is calculated by $\text{Dev} = \text{App} - \text{Sim}$. The number of iterations of the algorithm to obtain each result in the tables is less than or equal to 4.

Numerical results show the approximation works well. Almost all the approximations for throughput are contained in the 95% confidence intervals and the relative percentage errors are all less than 1%. The approximations for $E[X_0]$ and $E[Y]$ also provide satisfactory results.

TABLE 2. The throughput for the system with $J = 3$, $\mathbf{m} = (5, 4, 3, 2)$, $\mathbf{m}\boldsymbol{\mu} = (2.5, 1.0, 1.0, 1.0)$

\mathbf{b}	$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 3.0$
	Sim(c.i.)	Sim(c.i.)	Sim(c.i.)
	App(Dev)	App(Dev)	App(Dev)
(0,0,0,0)	1.2772(± 0.0011)	1.8415(± 0.0010)	2.2005(± 0.0019)
	1.2769(-0.0003)	1.8403(-0.0012)	2.1991(-0.0014)
(0,1,2,3)	1.4440(± 0.0015)	2.1388(± 0.0013)	2.2968(± 0.0012)
	1.4445($+0.0006$)	2.1373(-0.0015)	2.2975($+0.0007$)
(2,3,4,5)	1.4872(± 0.0012)	2.2997(± 0.0011)	2.3787(± 0.0019)
	1.4872($+0.0000$)	2.2997($+0.0000$)	2.3785(-0.0007)
(3,3,3,3)	1.4752(± 0.0020)	2.2902(± 0.0012)	2.4027(± 0.0026)
	1.4755($+0.0003$)	2.2896(-0.0005)	2.4021(-0.0005)
(1,5,5,5)	1.4939(± 0.0018)	2.3073(± 0.0023)	2.3617(± 0.0029)
	1.4930(-0.0010)	2.2943(-0.0129)	2.3471(-0.0146)
(5,5,5,5)	1.4948(± 0.0016)	2.3822(± 0.0014)	2.4379(± 0.0017)
	1.4940(-0.0007)	2.3814(-0.0008)	2.4363(-0.0015)

TABLE 3. $E[X_0]$ for the system with $J = 3$, $\mathbf{m} = (5, 4, 3, 2)$, $\mathbf{m}\boldsymbol{\mu} = (2.5, 1.0, 1.0, 1.0)$

\mathbf{b}	$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 3.0$
	Sim(c.i.)	Sim(c.i.)	Sim(c.i.)
	App(Dev)	App(Dev)	App(Dev)
(0,0,0,0)	2.6425(± 0.0057)	4.2146(± 0.0040)	5.8163(± 0.0062)
	2.6437($+0.0012$)	4.2180($+0.0034$)	5.8156(-0.0007)
(0,1,2,3)	3.0885(± 0.0071)	5.5429(± 0.0071)	6.5134(± 0.0066)
	3.0930($+0.0045$)	5.5457($+0.0028$)	6.5079(-0.0055)
(2,3,4,5)	3.2626(± 0.0057)	7.2068(± 0.0123)	7.9973(± 0.0129)
	3.2660($+0.0035$)	7.1998(-0.0071)	7.9947(-0.0027)
(3,3,3,3)	3.2318(± 0.0060)	7.4237(± 0.0202)	8.7362(± 0.0136)
	3.2362($+0.0044$)	7.4201(-0.0036)	8.7410($+0.0048$)
(1,5,5,5)	3.2298(± 0.0096)	6.7939(± 0.0108)	7.2470(± 0.0129)
	3.2675($+0.0377$)	6.8351($+0.0412$)	7.2686($+0.0216$)
(5,5,5,5)	3.3223(± 0.0097)	9.3021(± 0.0221)	10.3287(± 0.0165)
	3.3182(-0.0040)	9.3048($+0.0027$)	10.3324($+0.0037$)

5. Conclusion

In this paper, we developed an approximation method for a multi-server two-stage queueing network with finite buffers and merge configuration. We decompose the system into subsystem. The subsystem is approximated by the tandem queue with two service stations and two buffer spaces and external arrivals at second stage. Approximate system is modeled with LDQBD process. An iterative algorithm is presented for calculating the parameters. The approximation

TABLE 4. $E[Y]$ for the system with $J = 3$, $\mathbf{m} = (5, 4, 3, 2)$,
 $\mathbf{m}\boldsymbol{\mu} = (2.5, 1.0, 1.0, 1.0)$

\mathbf{b}	$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 3.0$
	Sim(c.i.)	Sim(c.i.)	Sim(c.i.)
	App(Dev)	App(Dev)	App(Dev)
(0,0,0,0)	0.0883(± 0.0016)	0.5330(± 0.0022)	1.4169(± 0.0039)
	0.0899(+0.0016)	0.5375(+0.0045)	1.4173(+0.0004)
(0,1,2,3)	0.2023(± 0.0018)	1.2682(± 0.0041)	1.9165(± 0.0052)
	0.2039(+0.0017)	1.2712(-0.0071)	1.9129(-0.0036)
(2,3,4,5)	0.0818(± 0.0016)	1.3235(± 0.0062)	1.7481(± 0.0071)
	0.0826(+0.0008)	1.3196(-0.0039)	1.7461(-0.0020)
(3,3,3,3)	0.0421(± 0.0015)	1.0658(± 0.0088)	1.6713(± 0.0072)
	0.0430(+0.0009)	1.0654(-0.0004)	1.6738(+0.0025)
(1,5,5,5)	0.1232(± 0.0025)	1.5185(± 0.0068)	1.7953(± 0.0083)
	0.1480(+0.0248)	1.5727(+0.0542)	1.8375(+0.0422)
(5,5,5,5)	0.0183(± 0.0007)	1.1848(± 0.0080)	1.5845(± 0.0089)
	0.0177(-0.0006)	1.1864(+0.0016)	1.5861(+0.0016)

method proposed can be used to analyze the more complex queueing networks with merge configuration.

REFERENCES

1. T. Altiok, *Performance Analysis of Manufacturing Systems*, New York: Springer, 1997.
2. J.A. Buzacott and J.G. Shantikumar, *Stochastic Models of Manufacturing Systems*, Englewood Cliffs, NJ: Prentice-Hall, 1993.
3. Y. Dallery and S.B. Gershwin, Manufacturing flow line systems: A review of models and analytical results, *Queueing Systems* **12** (1992), 3 – 94.
4. A.C. Diamantidis, C.T. Papadopoulos, Exact analysis of a two-workstation one-buffer flow line with parallel unreliable machines, *European Journal of Operational Research* **197** (2009), 592 – 580.
5. A.C. Diamantidis, C.T. Papadopoulos, C. Heavey, Approximate analysis of serial flow lines with multiple parallel-machine stations, *IIE Transactions* **39** (2007), 361 – 375.
6. M.M. Fadiloglu and S. Yeralan, Models of production lines as quasi-birth-death processes, *Mathematics and Computer Modeling* **35** (2002), 913-930.
7. S.B. Gershwin, *Manufacturing Systems Engineering*, Englewood Cliffs, NJ: Prentice-Hall, 1994.
8. S. Jain and J. MacGregor Smith, Open finite queueing networks with $M/M/C/K$ parallel servers, *Computers & Operations Research* **21** (1994), 297 – 317.
9. W. D. Kelton, R. P. Sadowski and D. A. Sadowski, *Simulation with ARENA, 2nd Ed.*, McGraw-Hill, New York, 1998.
10. J. Li, Overlapping decomposition: A system-theoretic method for modeling and analysis of complex manufacturing, *IEEE Transactions on Automation Science and Engineering* **2** (2005), 40 – 53.
11. J. Li, D.E. Blumenfeld, N. Huang and J.M. Alden, Throughput analysis of production systems: recent advances and future topics, *International Journal of Production Research* **47** (2009), 3823 – 3851.

12. J. Li and S.M. Meerkov, *Production Systems Engineering*, Springer, 2009.
13. J. Liu, S. Yang, A. Wu, S.J. Hu, Multi-stage throughput analysis of a two-stage manufacturing system with parallel unreliable machines and a finite buffer, *European Journal of Operational Research* **219** (2012), 296 – 304.
14. H.T. Papadopoulos and C. Heavey, Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines, *European Journal of Operational Research* **92** (1996), 1 – 27.
15. H.G. Perros, *Queueing Networks with Blocking*, Oxford University Press, 1994.
16. Y.W. Shin, Algorithmic solution for $M/M/c$ retrial queue with PH_2 -retrial times, *J. Appl. Math. & Informatics* **29** (2011), 803-811.
17. Y.W. Shin and D.H. Moon, Approximation of throughput in tandem queues with multiple servers and blocking, *Applied Mathematical Modelling* **38** (2014), 6122-6132.
18. M. van Vuuren, I.J.B.F. Adan, S.A.E. Resing-Sassen, Performance analysis of multi-server tandem queues with finite buffers and blocking, *OR Spectrum* **26** (2005), 315 – 338.
19. M.I. Vidalis and H.T. Papadopoulos, A redursive algorithm for generating the transition matrices of multistation multiserver exponential reliable queueing networks, *Computers & Operations Research* **28** (2001), 853 – 883.

Yang Woo Shin received B.S. from Kyungpook National University, and M.Sc. and Ph.D in Mathematics at KAIST. He is currently a professor at Changwon National University since 1991. His research interests include queueing theory and its applications.

Department of Statistics, Changwon National University, Changwon, Gyeongnam 641-773, Korea.

ywshin@changwon.ac.kr

Dug Hee Moon received B.Sc. from Hanyang University, and M.Sc. and Ph.D in Industrial Engineering at KAIST. He is currently a professor at Changwon National University since 1990. His research interests include simulation, manufacturing system design, queueing theory and its applications.

Department of Industrial and Systems Engineering, Changwon National University, Changwon, Gyeongnam 641-773, Korea.

dhmoon@changwon.ac.kr