Journal of the Operations Research
Society of Japan
Vol. 34, No. 3, September 1991

AGGREGATE APPROXIMATION FOR TANDEM QUEUEING SYSTEMS WITH PRODUCTION BLOCKING

Yu Song Yukio Takahashi
Tokyo Institute of Technology

(Received May 29, 1990)

Abstract In this paper, we study problems arising in applications of the cross aggregation method to tandem queueing systems with production blocking, and propose two types of applications with different state descriptions.

The cross aggregation method provides a nested family of approximations of stationary state probabilities of the model by imposing several different levels of assumptions on independence among nodes. Namely, in Level 1 we derive an approximate model by looking at one node at a time, in Level 2 by looking at two adjacent node at a time, in Level 3 by looking at three adjacent nodes at a time, and so on.

The method, however, cannot be applied in a naive form to tandem queueing systems with production blocking since the state space of the system is not a product space of individual state spaces of nodes. We propose two ways of state description to derive a Markov chain. Using one of them, the method can be applied in Levels 2, 3 and higher, but not in Level 1. Using the other, the method can be applied in any levels of approximation after modifying the Markov chain to have a product state space, but transition rates of the modified chain become complicated.

A comprehensive numerical test shows that in most cases the method provides very good approximations in Level 3 and sufficiently accurate approximations even in Level 2 for practical purposes.

1. Introduction

Tandem queueing models with blocking are often used to study telecommunication systems, production systems and other stochastic systems consisting of a series of subsystems. The concept of these tandem queueing models is not sophisticated but analysis of them is rather difficult. It is practically impossible to analyze them with traditional analytical methods since dependencies among nodes are not easy to deal with, and it is also limited to analyze by direct calculation of stationary state probabilities since the number of states needed for describing a model increases explosively as the size of the model grows larger. To overcome this difficulty, various approximation methods have been proposed to approximately evaluate performance measures of such models [1-5,7-11,13,14,16,18]. The cross aggregation method to be discussed here is one of them.

The cross aggregation method was first proposed in [16] to approximately analyze tandem queueing models with communication blocking. It provides a nested family of approximate models to get different levels of approximations of stationary state probabilities of nodes. Namely, it derives Level-1 approximate model by looking at one node at a time, Level-2 approximate model by looking at two nodes at a time, Level-3 approximate model by looking at three nodes at a time, and so on. It was applied in [17] to acyclic queueing networks with communication blocking. Brandwajn and Jow proposed an approximation method for tandem queueing models with production blocking in [3]. Their model is equivalent to our Level-2 approximate model. Numerical tests have shown that the computational burden of the cross aggregation method increases almost in linear order of the number of nodes in any

level of approximation.

For models with communication blocking, the stochastic behavior of a system is represented by a Markov chain on a product space of individual state spaces of nodes, and this makes us possible to apply the cross aggregation method in a simple way. However, for models with production blocking, the state space does not become a product space of individual state spaces of nodes. The purpose of this paper is to study the applicability of the cross aggregation method to tandem queueing systems with production blocking and to check the accuracy of the approximations through a comprehensive numerical test.

Our results are as follows. We deal with two state descriptions. One is a generalization of the state description used in [3, 13], for which Brandwajn and Jow [3] showed the feasibility of the cross aggregation method in Level 2. Here we show the method can be applied in Levels 2, 3 and higher for the state description but not in Level 1. The second one is a new state description in which a blocking customer is counted in the number of customers in the next node. For this state description, we can modify the Markov chain so that it is ergodic on the whole product space of individual state spaces of nodes, and then apply the cross aggregation method in any level of approximation. The numbers of states in the approximate models for this state description are less than those for the first, but the transition rate matrices are more complex. A numerical test shows that in most cases the method provides very good approximate values in Level 3, and even in Level 2 it provides sufficiently accurate ones for practical purposes. The computational burden is roughly in linear order of the number of nodes in any level of approximation.

Our paper is organized as follows.

In the next section, we introduce a simple tandem queueing model having a Poisson input, multiple exponential servers and a finite buffer at each node. A production type blocking rule is adopted. We use this model through the paper to discuss problems arising in applications of the cross aggregation method. However, the results of this paper can be applied to more general tandem queueing models with phase-type servers and a phase-type input.

For reader's convenience, in Section 3, we summarize fundamental concepts of the cross aggregation method. Precise description of approximation schemes for Levels 1, 2 and 3 are given in Appendix A.

In Section 4, we introduce two State descriptions A and B for the system. For both state descriptions we show that the state space is not a product space of individual state spaces of nodes. For State description A, however, by scrutinizing the structure of the state space of the Markov chain, we show how Level-2 and higher approximations can be applied but not Level-1. With State description B, we propose modifying the Markov chain so that the state space becomes a product space of individual state spaces of nodes. Then it becomes possible to apply the cross aggregation method in any levels of approximations in a naive way, though the transition rate matrix becomes complicated.

In Section 5, we present numerical results of a test for 50 cases.

2. Tandem Queueing System with Production Blocking

We consider a K-node tandem queueing system as shown in Fig. 1. Node $k, k = 1, 2, \ldots, K$, consists of s_k servers and a finite buffer of size b_k. Customers arrive at the system through a Poisson process with rate λ, and are served at Node k subjecting to an exponential distribution with rate μ_k. When the buffer of Node 1 is full, arriving customers are lost.

Blocking may occur due to finite buffers. When the service of a customer at Node k completes, he proceeds to Node $k+1$ if the buffer of Node $k+1$ is not full, and the server

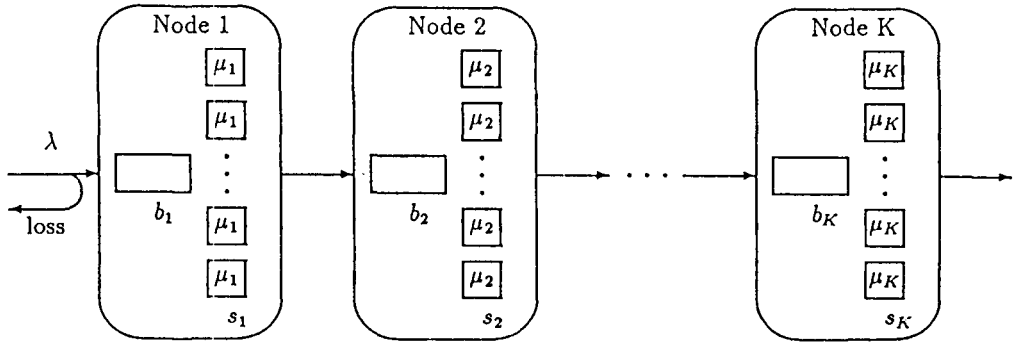


Figure 1: The tandem queueing system with production blocking

who has just completed the service is released for next service. Contrary, if the buffer of Node $k+1$ is full at that time, the customer blocks the server until a vacancy appears in the buffer of Node $k+1$. When a service completes at Node $k+1$ and a vacancy appears in the buffer of the node, a blocking customer at Node k, if there is any, proceeds to Node $k+1$ and releases (deblocks) the blocked server for new service. This rule is referred as *production blocking.*

A conspicuous feature of the tandem queueing system with production blocking is a simultaneous deblocking, in which several customers move simultaneously in consecutive nodes. For instance, if a server at Node k is blocked and the buffer in Node k is full, a server at Node $k-1$ may be also blocked when he completes a service. Then upon a service completion at Node $k+1$, the blocked servers are deblocked consecutively, i.e. the blocking customer at Node k advances to Node $k+1$, as well as the one at Node $k-1$ advances to Node k. Such a simultaneous deblocking does not occur in a system with communication blocking, and the existence of simultaneous deblockings makes the analysis of the system with production blocking more difficult.

To analyze such a system, it is convenient to describe the stochastic behavior of the system with a time-continuous vector-valued Markov chain $(\boldsymbol{X}_1(t), \boldsymbol{X}_2(t), \ldots, \boldsymbol{X}_K(t))$, where $\boldsymbol{X}_k(t)$ represents the state of Node k at time t. In Section 4, we introduce two kinds of state descriptions of nodes and have two Markov chains to describe the stochastic behavior of the system.

3. Approximation by the Cross Aggregation Method

In this section, for reader's convenience, we briefly summarize variables and assumptions treated in the cross aggregation method.

Let $S_k = \{0, 1, \ldots, N_k - 1\}$ be the set of local states of Node k, where N_k is the number of local states of the node. We consider a time-continuous Markov chain $(\boldsymbol{X}_1(t), \boldsymbol{X}_2(t), \ldots, \boldsymbol{X}_K(t))$ on the product state space $S_{1,2,\ldots,K} = S_1 \times S_2 \times \cdots \times S_K$, and at this moment we assume the chain is ergodic and in steady-state. We denote by $x(n_1, n_2, \ldots, n_K)$ the stationary probability of the chain in state $(n_1, n_2, \ldots, n_K)$, and by $(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_K)$ a random vector subjecting to the stationary distribution.

Level-1 Approximation. In Level 1, we look at one node at a time (Fig. 2a), and take marginal probabilities

$$x_k(n_k) = \Pr\{\boldsymbol{X}_k = n_k\} = \sum_{j \neq k} \sum_{n_j} x(n_1, n_2, \ldots, n_k, \ldots, n_K), \quad n_k = 0, 1, \ldots, N_k - 1,$$

of Node k ($k = 1, 2, \ldots, K$) as variables to be treated.

Using the aggregation technique for Markov chains [6,12,15], for each fixed k, we derive a system of linear equations for $x_k(n_k)$, $n_k = 0, 1, \ldots, N_k - 1$ (see Appendix A). The coefficients in the equations are given by weighted sums of transition rates with weights

$$(1) \quad x_{1,2,\ldots,k,\ldots,K|k}(m_1, m_2, \ldots, m_k, \ldots, m_K \mid m_k) = x(m_1, m_2, \ldots, m_k, \ldots, m_K)/x_k(m_k).$$

Hence to evaluate the weights we have to approximate $x(n_1, n_2, \ldots, n_k, \ldots, n_K)$'s from marginal probabilities $x_k(n_k)$'s. Here we simply assume that the nodes are statistically independent as

Assumption 1:

$$(2) \quad x(n_1, n_2, \ldots, n_K) = \prod_k x_k(n_k).$$

Using this assumption we get a system of N_k linear equations for $x_k(n_k)$, $n_k = 0, 1, \ldots, N_k - 1$, containing other $x_j(n_j)$'s in the coefficients. If we derive such a system of equations for each k, $k = 1, 2, \ldots, K$, totally we have $\sum_{k=1}^{K} N_k$ equations for $\sum_{k=1}^{K} N_k$ variables $x_k(n_k)$, $n_k = 0, 1, \ldots, N_k - 1$, $k = 1, 2, \ldots, K$. Solving this combined system of equations numerically, we get Level-1 approximate values of $x_k(n_k)$'s. Level-1 approximate values for other performance measures are calculated from these approximate values of $x_k(n_k)$'s.

It is important to point out that though the coefficients of the linear equations are given in terms of conditional probabilities $x_{1,2,\ldots,K|k}(n_1, \ldots, n_K \mid n_k)$ (see (15) in Appendix A), we need not calculate all of them and hence not all of the conditional probabilities because most of the coefficients are 0's or constants (see Appendix A). Therefore the calculating effort for the approximation is much less than that for the exact solution. This notice is also valid for Levels-2 and -3 approximations stated below.

Level-2 Approximation. In Level 2, we look at two adjacent nodes at a time (Fig. 2b) and take

$$\begin{aligned} x_{k,k+1}(n_k, n_{k+1}) &= \Pr\{\boldsymbol{X}_k = n_k, \boldsymbol{X}_{k+1} = n_{k+1}\} \\ &= \sum_{j \neq k, k+1} \sum_{n_j} x(n_1, n_2, \ldots, n_k, n_{k+1}, \ldots, n_K), \\ &\qquad n_k = 0, 1, \ldots, N_k - 1, \quad n_{k+1} = 0, 1, \ldots, N_{k+1} - 1, \\ &\qquad k = 1, 2, \ldots, K - 1, \end{aligned}$$

as variables. The number of variables here is $\sum_{k=1}^{K-1} N_k \times N_{k+1}$. Using the aggregation technique and Assumption 2 below, we can derive a system of $\sum_{k=1}^{K-1} N_k \times N_{k+1}$ equations for these variables as in Level 1.

Assumption 2:

$$(3) \quad x(n_1, n_2, \ldots, n_K) = x_{1,2}(n_1, n_2) x_{3|2}(n_3 \mid n_2) \ldots x_{K|K-1}(n_K \mid n_{K-1}),$$

where

$$\begin{aligned} x_{k|k-1}(n_k \mid n_{k-1}) &= \Pr\{\boldsymbol{X}_k = n_k \mid \boldsymbol{X}_{k-1} = n_{k-1}\} \\ &= x_{k-1,k}(n_{k-1}, n_k) / \sum_{m_k} x_{k-1,k}(n_{k-1}, m_k). \end{aligned}$$

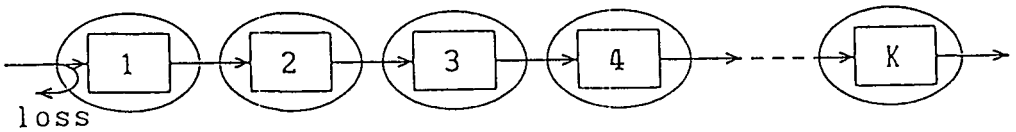


Figure 2a: Level-1 approximation

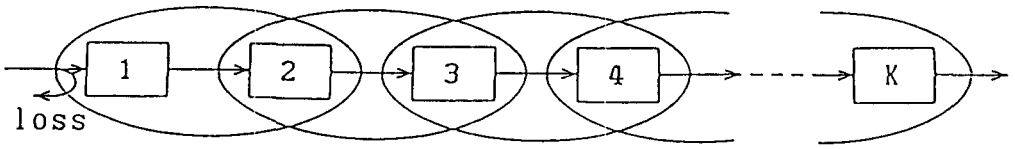


Figure 2b: Level-2 approximation

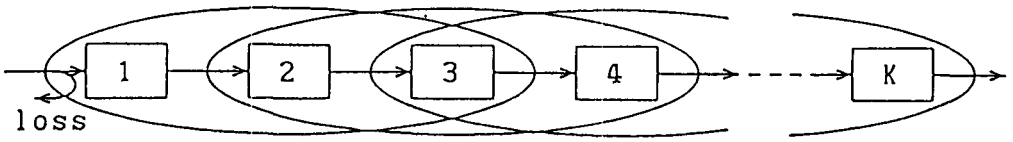


Figure 2c: Level-3 approximation

By solving the system of equations numerically, we obtain a set of approximate values of $x_{k,k+1}(n_k, n_{k+1})$'s. Level-2 approximate values for other performance measures are calculated from these Level-2 approximate values of $x_{k,k+1}(n_k, n_{k+1})$'s.

Level-1 variables $x_k(n_k)$'s are marginals of Level-2 variables $x_{k,k+1}(n_k, n_{k+1})$'s. In this sense, Level-2 approximation is finer than Level-1 approximation. So it is reasonable to expect that Level-2 approximate values of performance measures are closer to the exact values than Level-1 approximate values. Furthermore, if approximate values of the two levels are close enough for most of the principal performance measures, we may expect that they are close to the exact values, too, at least in a practical sense. Then Level-2 approximation is enough for our model analysis. On the contrary, if approximate values of the two levels are not close enough for some principal performance measures, then we suspect that at least Level-1 approximate values are not close to the exact values, and we should proceed to Level-3 approximation.

Level-3 and Higher Approximations. In Level 3, we look at three adjacent nodes at a time (Fig. 2c) and take

$$\begin{aligned} x_{k,k+1,k+2}(n_k, n_{k+1}, n_{k+2}) &= \Pr\{X_k = n_k, X_{k+1} = n_{k+1}, X_{k+2} = n_{k+2}\} \\ &= \sum_{j \neq k,k+1,k+2} \sum_{n_j} x(n_1, n_2, \ldots, n_k, n_{k+1}, n_{k+2}, \ldots, n_K), \\ & \qquad n_k = 0, 1, \ldots, N_k - 1, \quad n_{k+1} = 0, 1, \ldots, N_{k+1} - 1, \\ & \qquad n_{k+2} = 0, 1, \ldots, N_{k+2} - 1, \quad k = 1, 2, \ldots, K - 2, \end{aligned}$$

as variables. The number of variables here is $\sum_{k=1}^{K-2} N_k \times N_{k+1} \times N_{k+2}$. Using the aggregation technique and Assumption 3 below, we can derive a system of equations for these variables as previously.

Assumption 3:

$$(4) \qquad \begin{aligned} & x(n_1, n_2, \ldots, n_K) \\ = \; & x_{1,2,3}(n_1, n_2, n_3) x_{4|2,3}(n_4 \mid n_2, n_3) \ldots x_{K|K-2,K-1}(n_K \mid n_{K-1}, n_{K-2}), \end{aligned}$$

where

$$\begin{aligned} & x_{k|k-2,k-1}(n_k \mid n_{k-2}, n_{k-1}) \\ = \; & \Pr\{\boldsymbol{X}_k = n_k \mid \boldsymbol{X}_{k-2} = n_{k-2}, \boldsymbol{X}_{k-1} = n_{k-1}\} \\ = \; & x_{k-2,k-1,k}(n_{k-2}, n_{k-1}, n_k) / \sum_{m_k} x_{k-2,k-1,k}(n_{k-2}, n_{k-1}, m_k). \end{aligned}$$

By solving the system of $\sum_{k=1}^{K-2} N_k \times N_{k+1} \times N_{k+2}$ equations numerically, we obtain Level-3 approximate values of $x_{k,k+1,k+2}(n_k, n_{k+1}, n_{k+2})$'s. Level-3 approximate values for other performance measures are calculated from them.

If approximate values of Levels 2 and 3 are close enough, we may expect they are close to the exact values, too. If they are not close enough, we should proceed higher levels of approximations. In this way, we can evaluate the errors in approximate values to some extent in the process of the cross aggregation approximation. As will be shown in Section 5, the approximate values of Levels 2 and 3 are accurate enough for most examples tested. So the authors think that the cross aggregation by Level 3 is enough in most practical situations.

In the case we need higher level approximations, we look at i $(3 < i < K)$ successive nodes at a time in Level i. Variables and assumptions are selected similarly as those in Levels 1, 2 and 3.

4. State Description and Modification of the Markov Chain

It is easily seen from assumptions in the preceding section that the cross aggregation method is applied in a natural way if the state space of the underlying vector-valued Markov chain is a product space of individual state spaces of nodes. However, the Markov chain derived in the analysis of a tandem queueing system with production blocking does not have this desirable property. In this section, we introduce two kinds of state descriptions of such a system, and discuss fitness of the structure of the state space to the approximation assumptions of the cross aggregation method.

4.1 State Descriptions

There are several possible ways of state descriptions for a tandem queueing system with production blocking. Here we discuss two of them. The first is a generalization of the one used in [3, 13, 14] and others, and the second is a new one in which a blocking customer is counted in the number of customers in the next node.

State description A. Let σ_k be the number of customers in Node k and β_k the number of servers blocked at Node k. Then the state of the system can be represented by a row vector $(\sigma_1, \beta_1; \sigma_2, \beta_2; \ldots; \sigma_K, \beta_K)$ with $\beta_K = 0$. If we denote by S_k the set of all possible states of Node k, then

$$S_k = \begin{cases} \{(\sigma_k, \beta_k) \mid \sigma_k = 0, 1, \ldots, s_k + b_k,\ \beta_k = 0, 1, \ldots, s_k,\ \sigma_k \geq \beta_k\}, & k \neq K, \\ \{(\sigma_K, 0) \mid \sigma_K = 0, 1, \ldots, s_K + b_K\}, & k = K. \end{cases}$$

Remind that s_k is the number of servers and b_k is the size of the buffer at Node k.

State description B. Let σ_k and β_k be as above and

$$\tau_k = \sigma_k - \beta_k + \beta_{k-1}, \quad k = 1, 2, \ldots, K,$$

with $\beta_0 = 0$. Then the state of the system can be expressed by a row vector $(\tau_1, \tau_2, \ldots, \tau_K)$. In this case, we regard τ_k as the state of Node k, though it may depend on the original state of downstream nodes. Then the set S_k of possible states of Node k is given by

$$S_k = \{\tau_k \mid \tau_k = 0, 1, \ldots, s_{k-1} + s_k + b_k\}, \quad k = 1, 2, \ldots, K$$

with $s_0 = 0$.

State description A is natural and easy to imagine the situation of the system. However, under a simultaneous deblocking, more than two consecutive nodes may change their states. For example, consider a 4-node system with $s_k = b_k = 2$ for $k = 1, 2, 3, 4$. Suppose the numbers of customers at consecutive 4 nodes are 3,4,4 and 3 respectively, and a server at Node 1 and a server at Node 2 are blocked. Then this state is represented as (3,1;4,1;4,0;3,0). If one of the customers in node 3 completes its service, then the blocked servers at Nodes 2 and 1 are deblocked consecutively, and the state changes to (2,0;4,0;4,0;4,0). Thus 3 out of 4 nodes change their states.

On the other hand, in State description B, state changes occur at most two nodes simultaneously. For example, in the above situation, the state changes from (2,4,5,3) to (2,4,4,4).

4.2 Shape of the State Space

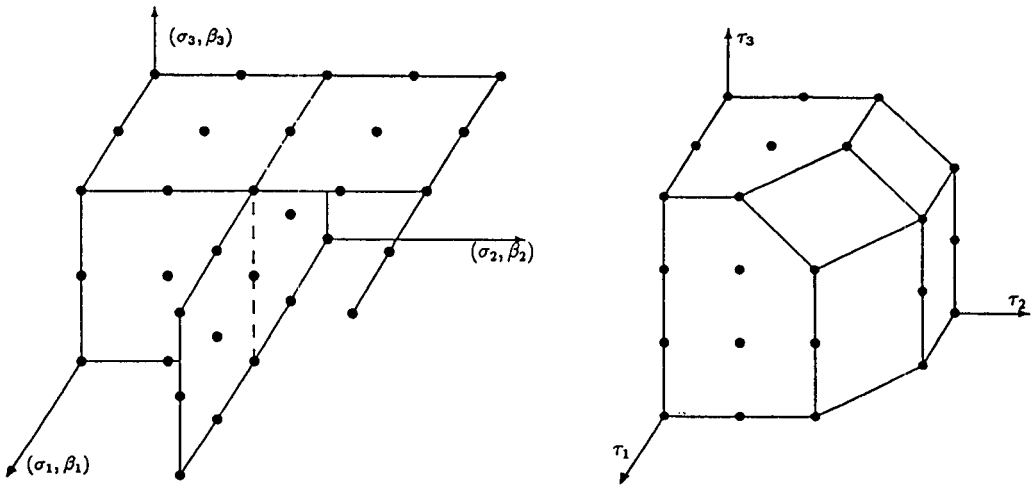
Here we investigate the shape of the state space for both State descriptions A and B. We denote by $\tilde{S}_{1,2,\ldots,K}$ the proper state space of the system, i.e. the set of all possible states of the system. Then in both State descriptions A and B, $\tilde{S}_{1,2,\ldots,K}$ is clearly a subset of the product space $S_{1,2,\ldots,K} = S_1 \times S_2 \times \cdots \times S_K$ of state spaces of individual nodes, and indeed it is a proper subset.

To see this, let's consider a case where $K = 3$ and $s_k = 1$, $b_k = 1$ for $k = 1, 2, 3$. Figs. 3a and 3b show the shape of $\tilde{S}_{1,2,\ldots,K}$ in State descriptions A and B, respectively. The shape of $\tilde{S}_{1,2,\ldots,K}$ for State description A is complicated, but that for State description B is rather simple. The latter is a polyhedron formed from a rectangular parallelepiped by cutting along two edges and a corner with three planes.

If we define a Markov chain $(\boldsymbol{X}_1(t), \boldsymbol{X}_2(t), \ldots, \boldsymbol{X}_K(t))$ on the product space $S_{1,2,\ldots,K}$ to describe the behavior of the system, then the chain never visit states outside of $\tilde{S}_{1,2,\ldots,K}$, and the corresponding stationary state probabilities are equal to zero. In Assumptions 1, 2 and 3, we impose positive approximate values on all stationary probabilities $x(n_1, n_2, \ldots, n_K)$ for $(n_1, n_2, \ldots, n_K) \in S_{1,2,\ldots,K}$. In order for the cross aggregation approximation to work well, the approximate value should be at least positive if $(n_1, n_2, \ldots, n_K) \in \tilde{S}_{1,2,\ldots,K}$, and zero if not. In the next subsection, we discuss whether this property is satisfied or not.

4.3 Applicability of the Cross Aggregation Method

First, we shall introduce some notations. For $j \leq k$, let $\tilde{S}_{j,j+1,\ldots,k}$ be the set of possible combinations of states of consecutive $k - j + 1$ nodes $j, j+1, \ldots, k$ in the system, and



a: State description A b: State description B

Figure 3: State space of a 3-node model with $s_k = b_k = 1$ for $k = 1, 2, 3$

$S_{j,j+1,\ldots,k}$ be the product space $S_j \times S_{j+1} \times \cdots \times S_k$. Clearly $\tilde{S}_{j,j+1,\ldots,k} \subset S_{j,j+1,\ldots,k}$. When $j = k$, we put $\tilde{S}_j = S_j$. We define a set operation $\tilde{\times}$ between two $\tilde{S}$'s by

$$\tilde{S}_{i,\ldots,j,\ldots,k} \tilde{\times} \tilde{S}_{j,\ldots,k,\ldots,\ell} = \{(n_i, \ldots, n_j, \ldots, n_k, \ldots, n_\ell) \mid (n_i, \ldots, n_j, \ldots, n_k) \in \tilde{S}_{i,\ldots,j,\ldots,k},\ (n_j, \ldots, n_k, \ldots, n_\ell) \in \tilde{S}_{j,\ldots,k,\ldots,\ell}\},$$

for $i \le j \le k \le \ell$, and

$$\tilde{S}_{i,\ldots,k} \tilde{\times} \tilde{S}_{j,\ldots,\ell} = \tilde{S}_{i,\ldots,k} \times \tilde{S}_{j,\ldots,\ell} = \{(n_i, \ldots, n_k; n_j, \ldots n_\ell) \mid (n_i, \ldots, n_k) \in \tilde{S}_{i,\ldots,k},\ (n_j, \ldots, n_\ell) \in \tilde{S}_{j,\ldots,\ell}\},$$

for $i \le k < j \le \ell$.

If we denote by $\chi_{i,\ldots,k}(n_i, \ldots, n_k)$ the indicator function of the set $\tilde{S}_{i,\ldots,k}$ on the product space $S_{i,\ldots,k}$, then $\tilde{S}_{i,\ldots,j,\ldots,k} \tilde{\times} \tilde{S}_{j,\ldots,k,\ldots,\ell}$ is interpreted as a set having an indicator function $\chi_{i,\ldots,j,\ldots,k}(n_i, \ldots, n_j, \ldots, n_k) \times \chi_{j,\ldots,k,\ldots,\ell}(n_j, \ldots, n_k, \ldots, n_\ell)$ on the product space $S_{i,\ldots,j,\ldots,k,\ldots,\ell}$.

Note that in Level-3 approximation $\tilde{S}_{1,2,3} \tilde{\times} \tilde{S}_{2,3,4} \tilde{\times} \cdots \tilde{\times} \tilde{S}_{K-2,K-1,K}$ is the set of $(n_1, \ldots, n_K)$'s on which the approximate value of $x(n_1, \ldots, n_K)$ in (4) is positive. Namely

$$\tilde{S}_{1,2,3} \tilde{\times} \tilde{S}_{2,3,4} \tilde{\times} \cdots \tilde{\times} \tilde{S}_{K-2,K-1,K} = \{(n_1, \ldots, n_K) \mid x(n_1, \ldots, n_K) > 0 \text{ under Assumption 3}\}.$$

Similarly

$$\tilde{S}_{1,2} \tilde{\times} \tilde{S}_{2,3} \tilde{\times} \cdots \tilde{\times} \tilde{S}_{K-1,K} = \{(n_1, \ldots, n_K) \mid x(n_1, \ldots, n_K) > 0 \text{ under Assumption 2}\},$$

and

$$\tilde{S}_1 \tilde{\times} \tilde{S}_2 \tilde{\times} \cdots \tilde{\times} \tilde{S}_K = \{(n_1, \ldots, n_K) \mid x(n_1, \ldots, n_K) > 0 \text{ under Assumption 1}\}.$$

Proposition 1. In State description A

(i) $\tilde{S}_{1,2,\ldots,k,k+1}$ is a proper subset of $\tilde{S}_{1,2,\ldots,k} \tilde{\times} \tilde{S}_{k+1}$ for $k = 1, 2, \ldots, K-1$.

(ii) $\tilde{S}_{1,\ldots,i,\ldots,j} \tilde{\times} \tilde{S}_{i,\ldots,j,\ldots,k}$ coincides with $\tilde{S}_{1,\ldots,i,\ldots,j,\ldots,k}$ for $1 < i \leq j < k$.

Proof: A state $(\sigma_\ell, \beta_\ell; \sigma_{\ell+1}, \beta_{\ell+1}; \ldots; \sigma_{\ell'}, \beta_{\ell'})$, $\ell < \ell'$, belongs to $\tilde{S}_{\ell,\ell+1,\ldots,\ell'}$ if and only if

$\langle a \rangle$ $\quad (\sigma_h, \beta_h) \in \tilde{S}_h$, or equivalently $0 \leq \beta_h \leq s_h$ and $\beta_h \leq \sigma_h \leq s_h + b_h$,

for $h = \ell, \ell+1, \ldots, \ell'$, and

$\langle b \rangle$ $\quad \beta_h = 0 \quad$ if $\quad \sigma_{h+1} < s_{h+1} + b_{h+1}$,

for $h = \ell, \ell+1, \ldots, \ell'-1$. Condition $\langle b \rangle$ reflects the fact that Node h is not blocked unless the buffer at Node $h+1$ is full.

To prove (i) we note that

$$\tilde{S}_{1,\ldots,k,k+1} = \{(\sigma_1, \beta_1; \ldots; \sigma_k, \beta_k; \sigma_{k+1}, \beta_{k+1}) \mid \langle a \rangle \text{ for } h = 1, \ldots, k, k+1, \text{ and } \langle b \rangle \text{ for } h = 1, 2, \ldots, k\}$$

and that

$$\tilde{S}_{1,2,\ldots,k} \tilde{\times} \tilde{S}_{k+1} = \{(\sigma_1, \beta_1; \ldots; \sigma_k, \beta_k; \sigma_{k+1}, \beta_{k+1}) \mid \langle a \rangle \text{ for } h = 1, \ldots, k, k+1, \text{ and } \langle b \rangle \text{ for } h = 1, 2, \ldots, k-1\}.$$

Since condition $\langle b \rangle$ for $h = k$ is added in the condition of $\tilde{S}_{1,2,\ldots,k+1}$ to that of $\tilde{S}_{1,2,\ldots,k} \tilde{\times} \tilde{S}_{k+1}$, state $(0, 0; \ldots; 0, 0; 1, 1; 0, 0)$ is in $\tilde{S}_{1,2,\ldots,k} \tilde{\times} \tilde{S}_{k+1}$ but not in $\tilde{S}_{1,2,\ldots,k+1}$.

To see (ii), we note that

$$\tilde{S}_{1,\ldots,i,\ldots,j,\ldots,k} = \{(\sigma_1, \beta_1; \ldots; \sigma_i, \beta_i; \ldots; \sigma_j, \beta_j; \ldots; \sigma_k, \beta_k) \mid \langle a \rangle \text{ for } h = 1, \ldots, i, \ldots, j, \ldots, k, \text{ and } \langle b \rangle \text{ for } h = 1, \ldots, i, \ldots, j, \ldots, k-1\},$$

and

$$\tilde{S}_{1,\ldots,i,\ldots,j} \tilde{\times} \tilde{S}_{i,\ldots,j,\ldots,k} = \{(\sigma_1, \beta_1; \ldots; \sigma_i, \beta_i; \ldots; \sigma_j, \beta_j; \ldots; \sigma_k, \beta_k) \mid \langle a \rangle \text{ for } h = 1, \ldots, i, \ldots, j, \ \langle b \rangle \text{ for } h = 1, \ldots, i, \ldots, j-1, \ \langle a \rangle \text{ for } h = i, \ldots, j, \ldots, k, \text{ and } \langle b \rangle \text{ for } h = i, \ldots, j, \ldots, k-1\}.$$

It is clear that, if $1 < i \leq j < k$, the condition in the right hand side reduces to

$$\langle a \rangle \text{ for } h = 1, \ldots, i, \ldots, j, \ldots, k, \text{ and } \langle b \rangle \text{ for } h = 1, \ldots, i, \ldots, j, \ldots, k-1.$$

Hence $\tilde{S}_{1,\ldots,i,\ldots,j,\ldots,k} = \tilde{S}_{1,\ldots,i,\ldots,j} \tilde{\times} \tilde{S}_{i,\ldots,j,\ldots,k}$. □

Corollary: In State description A

$$\tilde{S}_1 \tilde{\times} \tilde{S}_2 \tilde{\times} \cdots \tilde{\times} \tilde{S}_K = S_{1,2,\ldots,K} \neq \tilde{S}_{1,2,\ldots,K}, \tag{5}$$

$$\tilde{S}_{1,2} \tilde{\times} \tilde{S}_{2,3} \tilde{\times} \cdots \tilde{\times} \tilde{S}_{K-1,K} = \tilde{S}_{1,2,\ldots,K}, \tag{6}$$

$$\tilde{S}_{1,2,3} \tilde{\times} \tilde{S}_{2,3,4} \tilde{\times} \cdots \tilde{\times} \tilde{S}_{K-2,K-1,K} = \tilde{S}_{1,2,\ldots,K}. \tag{7}$$

This corollary is easily proved from Proposition 1 using mathematical induction.

Assumption 1 in Section 3 provides a positive approximate value of $x(n_1, \ldots, n_K)$ for any $(n_1, \ldots, n_K)$ in $S_{1,2,\ldots,K}$. However, $x(n_1, \ldots, n_K) = 0$ outside of $\tilde{S}_{1,2,\ldots,K}$. Hence, in Level 1 $x(n_1, \ldots, n_K) = 0$ is approximated with a positive value for $(n_1, \ldots, n_K)$ in $S_{1,2,\ldots,K} - \tilde{S}_{1,2,\ldots,K}$. This indicates that *Level-1 approximation of the cross aggregation approximation will not work well with State description A.*

In approximations of Levels 2 and 3, the approximate values for $x(n_1, \ldots, n_K)$ by Assumptions 2 and 3 are positive if and only if $(n_1, \ldots, n_K) \in \tilde{S}_{1,2,\ldots,K}$. Hence approximations of Levels 2 and 3 might work well. Brandwajn and Jow proposed this Level-2 approximation in [3] and reported it would provide good approximations. We will show the results of our test of these approximations of Levels 2 and 3 for two tandem queueing systems in Section 5.

For State description B, as easily guessed from Fig. 3b

$$\tilde{S}_1 \tilde{\times} \tilde{S}_2 \tilde{\times} \cdots \tilde{\times} \tilde{S}_K = S_{1,2,\ldots,K} \neq \tilde{S}_{1,2,\ldots,K}, \tag{8}$$

$$\tilde{S}_{1,2} \tilde{\times} \tilde{S}_{2,3} \tilde{\times} \cdots \tilde{\times} \tilde{S}_{K-1,K} \neq \tilde{S}_{1,2,\ldots,K}, \tag{9}$$

$$\tilde{S}_{1,2,3} \tilde{\times} \tilde{S}_{2,3,4} \tilde{\times} \cdots \tilde{\times} \tilde{S}_{K-2,K-1,K} \neq \tilde{S}_{1,2,\ldots,K}. \tag{10}$$

To see this, we note that state $(\tau_k, \tau_{k+1}, \ldots, \tau_\ell)$ belongs to $\tilde{S}_{k,k+1,\ldots,\ell}$ for $1 \leq k \leq \ell \leq K$ if and only if

$$\tau_i + \cdots + \tau_j \leq s_{i-1} + s_i + b_i + \cdots + s_j + b_j$$

for all i, j such that $k \leq i \leq j \leq \ell$, with $s_0 = 0$. The left-hand side gives the sum of numbers of customers from Node i to Node j, and the right-hand side is its upper bound. Then we can easily check that $(0, \ldots, 0, s_{K-2} + s_{K-1} + b_{K-1}, s_{K-1} + s_K + b_K)$ belongs to $\tilde{S}_1 \tilde{\times} \tilde{S}_2 \tilde{\times} \cdots \tilde{\times} \tilde{S}_K$ but not to $\tilde{S}_{1,2,\ldots,K}$. Hence $\tilde{S}_{1,2,\ldots,K}$ is a proper subset of $\tilde{S}_1 \tilde{\times} \tilde{S}_2 \tilde{\times} \cdots \tilde{\times} \tilde{S}_K$. Equations (9) and (10) are checked in the similar manner.

Therefore the cross aggregation method will not work well with State description B. However the shape of the state space for State description B is rather simple as shown in Fig. 3b. So if we can modify the Markov chain so that it is ergodic on the whole product space $S_{1,2,\ldots,K}$, then the cross aggregation method will work for this modified Markov chain since the above equations hold with equalities for S's instead of $\tilde{S}$'s. In the next subsection, we discuss the underlying idea of the modification.

4.4 Modification of Markov Chain for State Description B

For brevity, let us consider a simple 2-node system with $s_1 = s_2 = b_1 = b_2 = 2$. In this system, $S_1 = \{0, 1, 2, 3, 4\}$, $S_2 = \{0, 1, \cdots, 6\}$ and $\tilde{S}_{1,2} = \{(\tau_1, \tau_2) \mid \tau_1 \in S_1, \tau_2 \in S_2$, and $\tau_1 + \tau_2 \leq 8\}$. State $(4, 5) \in S_{1,2} = S_1 \times S_2$ is not in $\tilde{S}_{1,2}$, because $\tau_2 = 5$ means that Node 2 is full and one server at Node 1 is blocked. In such a case, the blocked server is treated as a buffer of Node 2, and τ_1 can be 3 at most. Besides state (4,5), states (3,6) and (4,6) are also not in $\tilde{S}_{1,2}$.

Table 1 lists the conditional probabilities $x_{1|2}(\tau_1 \mid \tau_2) = \Pr\{\boldsymbol{X}_1 = \tau_1 \mid \boldsymbol{X}_2 = \tau_2\}$. In order for Assumption 1 to provide good approximate values, it is desirable that the rows of Table 1 are close with each other. So we shall modify the Markov chain so that rows 5 and

Table 1: M/M/2/4→M/2/4 : $\lambda = 4,\ \mu_1 = 2,\ \mu_2 = 2$

	$\Pr\{X_1 = \tau_1 \mid X_2 = \tau_2\}$				
τ_2	$\tau_1 = 0$	$\tau_1 = 1$	$\tau_1 = 2$	$\tau_1 = 3$	$\tau_1 = 4$
0	.0709	.1473	.1576	.1917	.4324
1	.0829	.1754	.1962	.2641	.2814
2	.0980	.2120	.2447	.2547	.1905
3	.1135	.2456	.2561	.2261	.1588
4	.1228	.2526	.2654	.2394	.1197
5	.1130	.2311	.3298	.3261	—
6	.1303	.2510	.6187	—	—

6 become closer to other rows.

We see that $x_{1|2}(2 \mid 6)$ is relatively large in column $\tau_1 = 2$ and so is $x_{1|2}(3 \mid 5)$ in column $\tau_1 = 3$. Hence it is natural to modify the Markov chain so that the stationary probability $x(2,6)$ is shared with states (2,6), (3,6) and (4,6) in the modified chain and $x(3,5)$ is shared with states (3,5) and (4,5). A concrete algorithm to achieve such a modification for more general systems is presented in Appendix B. Adopting this algorithm to our 2-node system, the following 7 transitions are added as shown in Fig. 4b in Appendix B:

from (2,6) to (3,6) with rate λ
from (3,6) to (4,6) with rate λ
from (3,5) to (4,5) with rate λ
from (4,5) to (3,6) with rate μ_1
from (3,6) to (5,2) with rate $2\mu_2$
from (4,6) to (5,2) with rate $2\mu_2$
from (4,5) to (3,4) with rate $2\mu_2$.

Then, $x_{1|2}(2 \mid 6) = .6187$ is split into $x_{1|2}(2 \mid 6) = .2358$, $x_{1|2}(3 \mid 6) = .1914$ and $x_{1|2}(4 \mid 6) = .1914$, and $x_{1|2}(3 \mid 5) = .3261$ is split into $x_{1|2}(3 \mid 5) = .1957$ and $x_{1|2}(4 \mid 5) = .1304$.

By this modification, the Markov chain becomes ergodic on the product space $S_{1,2}$. For a more general K-node system, we can modify the Markov chain in a similar manner. Then the state space becomes the product space of state spaces of individual nodes, and the cross aggregation method can be applied in a naive form in any level of approximation.

In the next section, we will numerically test these approximations for two tandem queueing systems with production blocking.

5. Numerical Results

The approximate procedures proposed in the preceding sections are tested for more than 50 cases. In this section, we show some results among them and discuss accuracy of the cross aggregation method.

Tables 3 through 8 are numerical results for a 4-node model with $s_k = b_k = 2$ for $k = 1,2,3,4$. The approximate values are compared with exact values. In the tables, row "Exact" indicates exact values, and rows "Level 1B", "Level 2B" and "Level 3B" indicate approximate values in Levels 1, 2 and 3 with State Description B, respectively. Similarly, rows "Level 2A" and "Level 3A" indicate approximate values in Levels 2 and 3 with State Description A. Column "Max" indicates the maximum of relative errors in percentage in $\Pr\{X_k = n_k\}$'s over $n_k = 0,1,\ldots,N_k - 1,\ k = 1,2,3,4$ in each level.

Table 9 shows results of a 5-node model with $s_k = b_k = 1$ for $k = 1,2,3,4,5$ with a set of parameters used in [10] and [3].

With State description B, we can start approximations from Level 1. With State description A, we cannot apply the cross aggregation method in Level 1, but programming is easier than with State description B, because the transition rate matrix of the modified Markov chain is rather complex (see Appendix C).

The number of variables treated is different between State Descriptions A and B even in the same level. It is listed in Table 2.

Table 2: Number of variables in the approximate models

	The 4-node model		The 5-node model	
Original MC	1367		531	
Level 1B	26	(=5+7+7+7)	19	(=3+4+4+4+4)
Level 2B	133	(=35+49+49)	60	(=12+16+16+16)
Level 2A	194	(=81+81+32)	68	(=19+19+19+11)
Level 3B	588	(=245+343)	176	(=48+64+64)
Level 3A	740	(=531+209)	183	(=71+71+41)

As stated in Section 3, most entries of the aggregated rate matrix are equal to 0 and some others are constants. For example, in Level-2B of the 4-node model, there are $49^2 = 2401$ entries in $\boldsymbol{Q}_{2,3}$. Among them, 2206 entries are 0's and 24 entries are non-zero constants. Only 171 entries must be calculated in each iteration. In Level-2A approximation of the 4-node model, among $81^2 = 6561$ entries of $\boldsymbol{Q}_{2,3}$, 6220 of them are 0's, 42 of them are constants, and we have only 299 entries to be calculated in each iteration.

Roughly speaking, the computing time to get approximate values is almost proportional to the number of variables treated, though it varies with system parameters. For instance, the computing time for Level 3A is generally longer than for Level 3B. This indicates that, if N_k is constant, the computational burden for any level of approximation is expected to be in linear order of K as K increases.

The results of Tables 3 through 9, and other numerical results we got, show that Level-1B approximation is rather rough because sometimes relative errors exceed 20%. As we have expected, Level-2B approximation is better. The relative errors in average number of customers do not exceed 5% in most cases though some relative errors in marginal probabilities exceed 20%. Level-2A approximation is even better. The relative errors in marginal probabilities and average number of customers do not exceed 10% and 5% respectively except only a few cases.

As for Level-3A and Level-3B, in most cases, the relative errors are below 2% and 3% respectively in both marginal probabilities and average number of customers. But in a few exceptional cases, relative errors become rather large. The case shown in Table 7 is one of such cases. The relative errors of Level-3 approximate values of $\Pr\{\boldsymbol{X}_k = 0\}$, k=1,2,3 and 4, exceed 10%. In this case, with probability 0.6130, servers at Nodes 1, 2 and 3 are blocked simultaneously. In consequence, the stochastic behavior of Node 1 is largely affected by that of Node 4, and this violates our Assumption 3.

In our cross aggregation method, we intuitively expect the following two properties:

1) The higher the level of approximation, the more accurate the approximate values are, since higher approximate models are finer than lower ones.

Table 3: M/M/2/4→M/2/4→M/2/4→M/2/4 : $\lambda = 6,\ \mu_1 = 3,\ \mu_2 = 3,\ \mu_3 = 3,\ \mu_4 = 3$

	$\Pr(X_i = 4)$				Average number of customers				
	Node 1	Node 2	Node 3	Node 4	Node 1	Node 2	Node 3	Node 4	
Exact	.2932	.3844	.3272	.2300	2.5524	2.5977	2.3915	2.0230	
Level 1B	.2640	.3432	.2935	.2181	2.4238	2.4599	2.2756	1.9583	
Level 2B	.2828	.3799	.3297	.2310	2.5314	2.5863	2.3882	2.0207	
Level 2A	.2990	.3850	.3242	.2266	2.5632	2.5890	2.3681	2.0022	
Level 3B	.2916	.3813	.3265	.2315	2.5526	2.2937	2.3894	2.0298	
Level 3A	.2937	.3842	.3272	.2299	2.5533	2.5963	2.3902	2.0215	
Error in %									Max
Level 1B	-9.96	-10.72	-10.31	-5.16	-5.04	-5.31	-4.85	-3.20	20.04
Level 2B	-3.55	-1.15	0.76	0.43	-0.82	-0.44	-0.14	-0.11	3.55
Level 2A	1.99	0.17	-0.94	-1.48	0.42	-0.34	-0.98	-1.03	4.21
Level 3B	-0.53	-0.79	-0.23	0.68	0.01	-0.15	-0.09	0.33	1.60
Level 3A	0.18	-0.04	-0.00	-0.02	0.03	-0.06	-0.05	-0.07	0.33

Table 4: M/M/2/4→M/2/4→M/2/4→M/2/4 : $\lambda = 1,\ \mu_1 = 3,\ \mu_2 = 3,\ \mu_3 = 3,\ \mu_4 = 3$

	$\Pr(X_i = 4)$				Average number of customers				
	Node 1	Node 2	Node 3	Node 4	Node 1	Node 2	Node 3	Node 4	
Exact	.0011	.0012	.0013	.0013	0.3420	0.3422	0.3423	0.3421	
Level 1B	.0011	.0013	.0013	.0013	0.3420	0.3424	0.3424	0.3422	
Level 2B	.0011	.0012	.0013	.0013	0.3420	0.3423	0.3424	0.3422	
Level 2A	.0011	.0012	.0013	.0013	0.3420	0.3423	0.3424	0.3422	
Level 3B	.0011	.0012	.0013	.0013	0.3420	0.3422	0.3423	0.3421	
Level 3A	.0011	.0012	.0013	.0013	0.3420	0.3422	0.3423	0.3421	
Error in %									Max
Level 1B	-0.01	5.16	3.04	2.29	0.01	0.05	0.04	0.03	5.16
Level 2B	0.00	0.04	1.74	1.94	0.00	0.00	0.02	0.02	1.94
Level 2A	0.00	0.07	1.83	2.01	0.00	0.00	0.02	0.02	2.01
Level 3B	0.00	-0.00	0.02	0.51	0.00	0.00	0.00	0.00	0.51
Level 3A	0.00	0.00	0.02	0.52	0.00	0.00	0.00	0.00	0.52

Table 5: M/M/2/4→M/2/4→M/2/4→M/2/4 : $\lambda = 2,\ \mu_1 = 1,\ \mu_2 = 3,\ \mu_3 = 1,\ \mu_4 = 3$

	$\Pr(X_i = 4)$				Average number of customers				
	Node 1	Node 2	Node 3	Node 4	Node 1	Node 2	Node 3	Node 4	
Eaxct	.2345	.0814	.3344	.0056	2.2834	1.0572	2.3457	0.5404	
Level 1B	.2220	.0238	.3341	.0068	2.2255	0.8999	2.3779	0.5510	
Level 2B	.2395	.0928	.3602	.0059	2.3148	1.1318	2.4004	0.5448	
Level 2A	.2352	.0877	.3501	.0059	2.2871	1.0994	2.3701	0.5408	
Level 3B	.2349	.0793	.3334	.0056	2.2856	1.0514	2.3431	0.5402	
Level 3A	.2345	.0814	.3344	.0057	2.2834	1.0572	2.3457	0.5405	
Error in %									Max
Level 1B	-5.34	-70.76	-0.10	21.96	-2.53	-14.88	-0.10	21.96	70.76
Level 2B	2.11	13.97	7.72	5.68	1.37	7.05	2.32	0.81	14.67
Level 2A	0.27	7.70	4.70	4.73	0.16	3.98	1.04	0.08	8.18
Level 3B	0.15	-2.58	-0.31	0.51	0.10	-0.56	-0.11	-0.03	2.58
Level 3A	0.00	0.01	-0.00	0.83	0.00	-0.00	0.00	0.03	0.83

Table 6: M/M/2/4→M/2/4→M/2/4→M/2/4 : $\lambda = 2,\ \mu_1 = 3,\ \mu_2 = 1,\ \mu_3 = 3,\ \mu_4 = 3$

	$\Pr(X_i = 4)$				Average number of customers				
	Node 1	Node 2	Node 3	Node 4	Node 1	Node 2	Node 3	Node 4	
Exact	.1315	.5343	.0093	.0093	1.5680	2.9451	0.6267	0.6237	
Level 1B	.0752	.6045	.0127	.0127	1.5193	3.1858	0.6692	0.6644	
Level 2B	.1314	.5340	.0092	.0100	1.5676	2.9447	0.6270	0.6253	
Level 2A	.1317	.5342	.0094	.0100	1.5688	2.9446	0.6272	0.6253	
Level 3B	.1315	.5342	.0093	.0093	1.5680	2.9451	0.6266	0.6236	
Level 3A	.1315	.5343	.0093	.0094	1.5680	2.9452	0.6267	0.6238	
Error in %									Max
Level 1B	-42.77	13.15	37.04	35.84	-3.10	8.17	6.79	6.53	46.25
Level 2B	-0.05	-0.05	-0.83	6.85	-0.02	-0.02	0.05	0.25	6.85
Level 2A	0.14	-0.00	1.16	7.42	0.05	-0.02	0.09	0.25	7.42
Level 3B	0.00	-0.01	0.00	-0.43	0.00	-0.00	-0.01	-0.01	0.43
Level 3A	0.00	0.00	0.21	0.27	0.00	0.00	0.01	0.01	0.27

Table 7: M/M/2/4→M/2/4→M/2/4→M/2/4 : $\lambda = 2,\ \mu_1 = 3,\ \mu_2 = 2,\ \mu_3 = 1,\ \mu_4 = .5$

	$\Pr(X_i = 4)$				Average number of customers				
	Node 1	Node 2	Node 3	Node 4	Node 1	Node 2	Node 3	Node 4	
Exact	.5080	.9686	.9812	.8461	3.1627	3.9493	3.9763	3.7452	
Level 1B	.6115	.9686	.9677	.8458	3.2951	3.9591	3.9584	3.7445	
Level 2B	.5265	.9679	.9834	.8463	3.1263	3.9563	3.9797	3.7456	
Level 2A	.5080	.9550	.9742	.8458	3.1198	3.9257	3.9662	3.7447	
Level 3B	.5033	.9713	.9839	.8462	3.1031	3.9558	3.9798	3.7454	
Level 3A	.5080	.9634	.9808	.8461	3.1431	3.9404	3.9757	3.7452	
Error in %									Max
Level 1B	20.39	0.00	-1.38	-0.04	4.19	0.25	-0.45	-0.02	96.22
Level 2B	3.64	-0.07	0.22	0.02	-1.15	0.18	0.09	0.01	72.13
Level 2A	0.00	-1.40	-0.72	-0.04	-1.35	-0.60	-0.26	-0.01	75.92
Level 3B	-0.92	0.27	0.28	0.01	-1.88	0.16	0.09	0.00	26.72
Level 3A	0.00	-0.54	-0.04	-0.00	-0.62	-0.23	-0.02	-0.00	19.71

Table 8: M/M/2/4→M/2/4→M/2/4→M/2/4 : $\lambda = 2,\ \mu_1 = .5,\ \mu_2 = 1,\ \mu_3 = 2,\ \mu_4 = 3$

	$\Pr(X_i = 4)$				Average number of customers				
	Node 1	Node 2	Node 3	Node 4	Node 1	Node 2	Node 3	Node 4	
Exact	.5281	.0521	.0039	.0008	3.2299	1.1268	0.4951	0.3215	
Level 1B	.5179	.0584	.0048	.0010	3.2020	1.1329	0.4945	0.2301	
Level 2B	.5281	.0521	.0041	.0009	3.2299	1.1269	0.4957	0.3219	
Level 2A	.5281	.0521	.0041	.0009	3.2299	1.1270	0.4957	0.3219	
Level 3B	.5281	.0521	.0039	.0008	3.2299	1.1268	0.4951	0.3215	
Level 3A	.5281	.0521	.0039	.0008	3.2299	1.1268	0.4951	0.3215	
Error in %									Max
Level 1B	-1.94	12.91	22.91	24.82	-0.86	0.54	-0.11	-0.41	27.82
Level 2B	-0.00	0.04	5.11	14.25	-0.00	0.01	0.12	0.14	14.25
Level 2A	0.00	0.10	6.01	14.87	0.00	0.02	0.14	0.14	14.87
Level 3B	-0.00	-0.00	0.00	1.34	-0.00	0.00	0.00	0.01	1.34
Level 3A	-0.00	-0.00	0.01	1.51	0.00	0.00	0.00	0.01	1.51

2) If differences between corresponding approximate values of two consecutive levels are small enough, the values are also close to the exact ones.

Our numerical results seem to support these properties. The authors think that these properties hold in most cases of tandem queueing systems with production blocking, though they have never been proved. Based on these properties we can perform our approximation process roughly estimating the magnitude of errors.

Table 9: M/M/1/2→M/1/2→M/1/2→M/1/2→M/1/2

$\lambda = 3,\ \mu_1 = 2,\ \mu_2 = 3,\ \mu_3 = 4,\ \mu_4 = 3,\ \mu_5 = 2$

	$\Pr\{X_i = 2\}$				
	Node 1	Node 2	Node 3	Node 4	Node 5
EXACT	.5190	.3246	.3261	.4652	.4673
Level 1B	.5142	.2701	.2360	.4080	.4670
Level 2B	.5289	.3270	.3305	.4708	.4709
Level 2A	.5231	.3276	.3245	.4555	.4604
Level 3B	.5286	.3250	.3309	.4753	.4710
Level 3A	.5194	.3238	.3265	.4680	.4681
Error in %					
Level 1B	-0.92	-16.80	-27.64	-12.30	-0.06
Level 2B	1.90	0.72	1.35	1.20	0.77
Level 2A	0.78	0.91	-0.50	-2.09	-1.48
Level 3B	1.85	0.12	1.46	2.15	0.80
Level 3A	0.08	-0.26	0.11	0.60	0.18

	Average number of customers					
	Node 1	Node 2	Node 3	Node 4	Node 5	
EXACT	1.3593	0.9415	0.9298	1.1790	1.1872	
Level 1B	1.3376	0.8256	0.7720	1.1170	1.1917	
Level 2B	1.3699	0.9427	0.9225	1.1879	1.1936	
Level 2A	1.3642	0.9450	0.9144	1.1598	1.1757	
Level 3B	1.3704	0.9413	0.9298	1.1937	1.1938	
Level 3A	1.3593	0.9410	0.9217	1.1828	1.1890	
Error in %						Max
Level 1B	-1.59	-12.31	-16.08	-5.26	0.25	27.64
Level 2B	0.78	0.13	0.29	0.75	0.41	2.85
Level 2A	0.36	0.37	-0.59	-1.63	-1.10	3.31
Level 3B	0.82	-0.03	1.09	1.25	0.42	2.51
Level 3A	0.01	-0.06	0.21	0.33	0.02	0.70

Appendices

A. Formulas and Algorithm

Here we derive a system of equations to be solved in each of Levels 1, 2 and 3 in the cross aggregation method.

A.1 Level-1 Approximation

Let $q(m_1,\ldots,m_K;n_1,\ldots,n_K)$ be the transition rate of the Markov chain $\{(\boldsymbol{X}_1(t),\ldots,\boldsymbol{X}_K(t))\}$ from $(m_1,\ldots,m_K)$ to $(n_1,\ldots,n_K)$, and $\boldsymbol{Q}=\{q(m_1,\ldots,m_K;n_1,\ldots,n_K)\}$ be the transition rate matrix. The stationary probabilities $x(m_1,\ldots,m_K)$ satisfy the equilibrium equations

$$\sum_{(m_1,\ldots,m_K)} x(m_1,\ldots,m_K)\, q(m_1,\ldots,m_K;n_1,\ldots,n_K)=0, \qquad (n_1,\ldots,n_K)\in S_{1,2,\ldots,K}, \tag{11}$$

and

$$\sum_{(m_1,\ldots,m_K)} x(m_1,\ldots,m_K)=1. \tag{12}$$

As stated in Section 3, we look at one node at a time in Level 1, and take $x_k(n_k)$'s as variables. For each fixed k, $k=1,2,\ldots,K$, to derive equations for $x_k(n_k)$, $n_k=0,1,\ldots,N_k-1$, we take summations $\sum_{(m_1,\ldots,m_K)}$ in (11) and (12) in two steps as $\sum_{m_k}(\sum_{j\neq k}\sum_{m_j})$. Then we have

$$\sum_{m_k} x_k(m_k)q_k(m_k,n_k)=0, \quad n_k=0,1,\ldots,N_k-1 \tag{13}$$

and

$$\sum_{m_k} x_k(m_k)=1, \tag{14}$$

where

$$q_k(m_k,n_k)=\sum_{i\neq k}\sum_{m_i} x_{1,2,\ldots,K|k}(m_1,\ldots,m_K\mid m_k) \times\sum_{j\neq k}\sum_{n_j} q(m_1,\ldots,m_K;n_1,\ldots,n_K), \tag{15}$$

and

$$x_{1,2,\ldots,K|k}(m_1,\ldots,m_K\mid m_k)=x(m_1,\ldots,m_K)/x_k(m_k). \tag{16}$$

In terminology of the aggregation theory, n_k is called an *aggregate state* if we regard it as a set of states $(n_1,n_2,\ldots,n_K)$ with n_k in the k-th place in common, and $x_k(n_k)$ is called an *aggregate variable*. Equations (13) and (14) are called *aggregate equations*.

Note that $q_k(m_k,n_k)$ in (15) is a weighted sum of transition rates $q(m_1,\ldots,m_K;n_k)=\sum_{j\neq k}\sum_{n_j} q(m_1,\ldots,m_K;n_1,\ldots,n_K)$ from state $(m_1,\ldots,m_K)$ to aggregate state n_k with weights $x_{1,2,\ldots,K|k}(m_1,\ldots,m_K\mid m_k)$. The weight $x_{1,2,\ldots,K|k}(m_1,\ldots,m_K\mid m_k)$ can be interpreted as conditional probability $\Pr\{\boldsymbol{X}_1=m_1,\boldsymbol{X}_2=m_2,\ldots,\boldsymbol{X}_K=m_K\mid \boldsymbol{X}_k=m_k\}$.

From (13) and (14) we can see that $x_k(n_k)$'s, $n_k=1,2,\ldots,N_k$, satisfy the equilibrium equations for a Markov chain with *aggregate transition rate matrix* $\boldsymbol{Q}_k=\{q_k(m_k,n_k)\}$. So

if we know the values of weights $x_{1,2,\ldots,K|k}(m_1,\ldots,m_K \mid m_k)$, then we can get the values of $x_k(n_k)$'s by solving equations (13) and (14) in a usual way done in Markov chain analyses.

Unfortunately, however, we do not know the values of $x_{1,2,\ldots,K|k}(m_1,\ldots,m_K \mid m_k)$'s, so we have to approximate them. Using Assumption 1 in Section 3, we may approximate as

$$x_{1,2,\ldots,K|k}(m_1,\ldots,m_K \mid m_k) = \prod_{j\neq k} x_j(m_j). \tag{17}$$

Then Equations (13) and (14) with (15) and (16) constitute a set of nonlinear equations for variables $x_k(n_k)$, $n_k = 0,1,\ldots,N_k-1$, $k = 1,2,\ldots,K$.

The following algorithm solves the set of equations numerically.

ALGORITHM

1. Initialization

   Set an appropriate initial value of $x_k(n_k)$ for each k and n_k.

2. Loop 1

   (a) Loop 2

   A) Set $k = 1$

   B) Loop 3

   i. Use the latest values of $x_j(n_j)$'s ($j \neq k$) to calculate the aggregate transition rate matrix $\boldsymbol{Q}_k$ from (15) and (16).

   ii. Solve (13) for $x_k(n_k)$, $n_k = 0,1,\ldots,N_k-1$, with a usual numerical method for a system of linear equations such as the Gauss-Seidel's method.

   iii. Normalize them using (14).

   Loop 3 is over.

   C) If $k = K$ then Loop 2 is over, else increase k by 1, go to B).

   (b) Compare values at the latest two iterations. If a certain convergence criterion is satisfied, Loop 1 is over. Otherwise go to (a).

3. Calculation of performance measures

   From the approximate values of $x_k(n_k), n_k = 0,1,\ldots,N_k-1$, $k = 1,2,\ldots,K$, calculate approximate performance measures of the original model such as the loss probability and the mean numbers of customers, etc.

The convergence of the algorithm has never been proved because of nonlinearity of the aggregate equations. But in all the cases the authors tested, the algorithm does converge.

In the first step of Loop 3, we do not need to calculate all of the entries $q_k(m_k,n_k)$ of $\boldsymbol{Q}_k$ from the definition (15), because most of them are equal to 0 or constants. In Appendix C, we discuss which transition rates must be calculated using approximation assumptions, for the case of Level 2. The case of Level 1 can be easily derived from the discussion.

A.2 Level-2 Approximation

The aggregate equations corresponding to (13) and (14) are

$$\sum_{m_k}\sum_{m_{k+1}} x_{k,k+1}(m_k, m_{k+1})\; q_{k,k+1}(m_k, m_{k+1}; n_k, n_{k+1}) = 0, \tag{18}$$
$$n_k = 0, 1, \ldots, N_k - 1;\; n_{k+1} = 0, 1, \ldots, N_{k+1} - 1,$$

and

$$\sum_{m_k}\sum_{m_{k+1}} x_{k,k+1}(m_k, m_{k+1}) = 1, \tag{19}$$

where $k = 1, 2, \ldots, K-1$.

In Level-2 approximation, we look at two adjacent nodes at a time and form an aggregate Markov chain with transition rate matrix $\boldsymbol{Q}_{k,k+1}$ with entries

$$\begin{aligned} &q_{k,k+1}(m_k, m_{k+1}; n_k, n_{k+1}) \\ &\quad = \sum_{j\neq k,k+1}\sum_{m_j} x_{1,2,\ldots,K|k,k+1}(m_1, \ldots, m_K \mid m_k, m_{k+1}) \\ &\qquad\qquad \times \sum_{i\neq k,k+1}\sum_{n_i} q(m_1, \ldots, m_K; n_1, \ldots, n_K), \end{aligned} \tag{20}$$

where

$$\begin{aligned} &x_{1,2,\ldots,K|k,k+1}(m_1, \ldots, m_K \mid m_k, m_{k+1}) \\ &\quad = \Pr\{\boldsymbol{X}_1 = m_1, \boldsymbol{X}_2 = m_2, \ldots, \boldsymbol{X}_K = m_K \mid \boldsymbol{X}_k = m_k, \boldsymbol{X}_{k+1} = m_{k+1}\} \\ &\quad = x(m_1, \ldots, m_K)/x_{k,k+1}(m_k, m_{k+1}). \end{aligned} \tag{21}$$

Assumption 2 makes us possible to approximate $x_{1,2,\ldots,K|k,k+1}(m_1, \ldots, m_K \mid m_k, m_{k+1})$ from aggregate variables $x_{k,k+1}(n_k, n_{k+1})$.

A.3 Level-3 Approximation

In Level-3, we look at 3 adjacent nodes at a time, and form an aggregate Markov chain with transition rate matrix $\boldsymbol{Q}_{k,k+1,k+2}$ with entries

$$\begin{aligned} &q_{k,k+1,k+2}(m_k, m_{k+1}, m_{k+2}; n_k, n_{k+1}, n_{k+2}) \\ &\quad = \sum_{j\neq k,k+1,k+2}\sum_{m_j} x_{1,2,\ldots,K|k,k+1}(m_1, \ldots, m_K \mid m_k, m_{k+1}, m_{k+2}) \\ &\qquad\qquad \times \sum_{i\neq k,k+1,k+2}\sum_{n_i} q(m_1, \ldots, m_K; n_1, \ldots, n_K), \end{aligned} \tag{22}$$

where

$$\begin{aligned} &x_{1,2,\ldots,K|k,k+1,k+2}(m_1, \ldots, m_K \mid m_k, m_{k+1}, m_{k+2}) \\ &\quad = Pr\{\boldsymbol{X}_1 = m_1, \ldots, \boldsymbol{X}_K = m_K \mid \boldsymbol{X}_k = m_k, \boldsymbol{X}_{k+1} = m_{k+1}, \boldsymbol{X}_{k+2} = m_{k+2}\} \\ &\quad = x(m_1, \ldots, m_K)/x_{k,k+1,k+2}(m_k, m_{k+1}, m_{k+2}). \end{aligned} \tag{23}$$

With Assumption 3, we take the dependency among three adjacent nodes into account, and assume that their stochastic behaviors are affected by other nodes only through a chain of connecting nodes.

Using a similar scheme to Levels 1 and 2, we can get Level-3 approximation.

B. Modification of Markov Chain for State Description B

B.1 Scheme of the Modification

Here we propose a scheme of modification of the Markov chain derived from Description B to enlarge the state space from $\tilde{S}_{1,2,\ldots,K}$ to $S_{1,2,\ldots,K} = S_1 \times S_2 \times \cdots \times S_K$.

We classify states in $S_{1,2,\ldots,K}$ into three categories, inner states, boundary states and dummy states. We call state $(\tau_1, \tau_2, \ldots, \tau_K)$ in $S_{1,2,\ldots,K} - \tilde{S}_{1,2,\ldots,K}$ as *dummy state*. For state $(\tau_1, \tau_2, \ldots, \tau_K)$ in $\tilde{S}_{1,2,\ldots,K}$, we call it a *boundary state* if $(\tau_1, \tau_2, \ldots, \tau_{k-1}, \tau_k + 1, \tau_{k+1}, \ldots, \tau_K)$ is a dummy state for some k, $(k = 1, 2, \ldots, K)$, and we call it an *inner state* if not.

For illustration, we consider a 2-node model with $s_1 = s_2 = b_1 = b_2 = 2$. In this case, $S_1 = \{0, 1, 2, 3, 4\}$, $S_2 = \{0, 1, \ldots, 6\}$ and $\tilde{S}_{1,2} = \{(\tau_1, \tau_2) \mid \tau_1 \in S_1,\ \tau_2 \in S_2,\ \tau_1 + \tau_2 \leq 8\}$. Here (3,6), (4,5) and (4,6) are dummy states, and (2,6), (3,5) and (4,4) are boundary states.

For each dummy state, later we choose a boundary state called the *root* of the dummy state. The dummy state is called a *descendant* of the root. A boundary state $(\tau_1, \tau_2, \ldots, \tau_K)$ and its descendants form a *group* $G(\tau_1, \tau_2, \ldots, \tau_K)$. If state $(\tau_1, \tau_2, \ldots, \tau_K)$ is an inner state or a boundary state without any descendants, it forms a group $G(\tau_1, \tau_2, \ldots, \tau_K)$ by itself. Then $S_{1,2,\ldots,K}$ is divided into groups of states, and $G : (\tau_1, \tau_2, \ldots, \tau_K) \to G(\tau_1, \tau_2, \ldots, \tau_K)$ is a bijection from $\tilde{S}_{1,2,\ldots,K}$ to the set of all groups $\{G(\tau_1, \tau_2, \ldots, \tau_K)\}$. Our goal is to derive an ergodic Markov chain on $S_{1,2,\ldots,K}$ whose stationary probability of $G(\tau_1, \tau_2, \ldots, \tau_K)$ coincides with that of $(\tau_1, \tau_2, \ldots, \tau_K)$ in the original chain.

For the purpose, we modify the Markov chain by imposing dummy transitions from/into dummy states. The rates of transitions imposed are, in principle, parallel to neighborhood. However, we have to satisfy the following

> *Condition C:* The rate of a transition from a dummy state in $G(\tau_1', \tau_2', \ldots, \tau_K')$ to another group $G(\tau_1'', \tau_2'', \ldots, \tau_K'')$ is set equal to that from $(\tau_1', \tau_2', \ldots, \tau_K')$ to $(\tau_1'', \tau_2'', \ldots, \tau_K'')$.

Then clearly the modified Markov chain has the desired property.

In practice, we proceed the modification in two steps. In the first step, we modify the original chain by assigning dummy transitions from/into dummy states in the following manner.

For a boundary or dummy state $(\tau_1, \tau_2, \ldots, \tau_K)$, we suppose a dummy arrival changes the state of the chain from $(\tau_1, \tau_2, \ldots, \tau_K)$ to $(\tau_1 + 1, \tau_2, \ldots, \tau_K)$ with transition rate λ if $(\tau_1 + 1, \tau_2, \ldots, \tau_K)$ is a dummy state. Similarly we suppose a dummy service completion at Node k $(k < K)$ changes the state from $(\tau_1, \tau_2, \ldots, \tau_K)$ to $(\tau_1, \ldots, \tau_{k-1}, \tau_k - 1, \tau_{k+1} + 1, \tau_{k+2}, \ldots, \tau_K)$ with transition rate $\min\{s_k, \tau_k\}\mu_k$ if $(\tau_1, \ldots, \tau_{k-1}, \tau_k - 1, \tau_{k+1} + 1, \tau_{k+2}, \ldots, \tau_K)$ is a dummy state. A dummy service completion at Node K changes the state from $(\tau_1, \tau_2, \ldots, \tau_K)$ to $(\tau_1, \ldots, \tau_{K-1}, \tau_K - 1)$ with transition rate $\min\{s_K, \tau_K\}\mu_K$ if $(\tau_1, \ldots, \tau_{K-1}, \tau_K - 1)$ is a dummy state. For example, in the 2-node model, as shown in Fig. 4a, a dummy arrival changes (2,6) to (3,6), and a dummy service completion at Node 1 changes (4,5) to (3,6).

The original chain (we will refer it as MC-B) is thus modified to an ergodic chain on the state space $S_{1,2,\ldots,K}$ (we will refer it as MC-B_1). But the stationary state probabilities of MC-B_1 may different from those of MC-B even for inner states. In the second step, we further modify MC-B_1 so that Condition C is satisfied and hence the stationary state probabilities coincide with those of MC-B.

First we must determine which dummy states belong to the group $G(\tau_1, \tau_2, \ldots, \tau_K)$ of a boundary state $(\tau_1, \tau_2, \ldots, \tau_K)$. Because MC-$B_1$ is ergodic, the state can be reached from any boundary state sooner or later. We can prove that there is a unique boundary state from

which the dummy state can be reached in less number of steps than any other boundary states. We define this boundary state as the *root* of the dummy state, and refer the dummy state as a descendent of the root boundary state. This root-descendent relation divides $S_{1,2,\ldots,K}$ into groups $\{G(\tau_1, \tau_2, \ldots, \tau_K)\}$. In the 2-node model, (4,5) can be reached from (3,5) in one step, and from other boundary states in more steps, so we decide (3,5) is the root of (4,5) and let (4.5) belongs to $G(3,5)$. Similarly, (2,6) is the root of (3,6) and (4,6), and $G(2,6)$ consists of these three states.

In Appendix B.2, we present an algorithm to find the root of a dummy state.

Next we modify MC-B_1 so that Condition C is satisfied. The basic idea is as follows.

i) *Modification of the rates of dummy transitions.* If a transition from a dummy state due to a dummy service completion at Node k, the transition rate must be set equal to that due to a service completion at Node k when MC-B_1 is in the root of the dummy state. In the 2-node model, the transition rate from (4,5) to (3,6) should be changed to μ_1 because the transition rate from (3,5) to (2,6) in MC-B_1 is μ_1. The transition rate from (3,6) to (3,5) remains $2\mu_2$ because the transition rate from (2,6) to (2,5) in MC-B_1 is $2\mu_2$. Transition rates due to dummy arrivals need not to be modified. In the 2-node model, the transition rate from (3,6) to (4,6) is the case.

ii) *Modification of the destinations of dummy transitions.* For some dummy transitions, we have to change their destinations. For a transition from a dummy state, if the destination of the corresponding parallel transition from the root is an inner state, then its destination should be changed to the same inner state. For example, in the 2-node model, MC-B_1 in (3,5) transits to inner state (3,4) due to a service completion at Node 2, hence the destination of a parallel dummy transition from (4,5) to (4,4) should be changed to (3,4). Similarly, the destination of transitions from (3,6) to (3,5) and from (4,6) to (4,5) are changed to (2,5). See Fig. 4b for the modification.

In this manner, MC-B_1 is modified to, say, MC-B_2. Clearly MC-B_2 satisfies Condition C, and the sum of the stationary probabilities of states in $G(\tau_1, \tau_2, \ldots, \tau_K)$ in MC-B_2 coincides with the stationary probabilities of $(\tau_1, \tau_2, \ldots, \tau_K)$ in MC-B.

B.2 Algorithm for the Roots of Dummy States

For $1 \leq i \leq j \leq K$, state $(\tau_1, \tau_2, \ldots, \tau_K)$ is not a dummy state if and only if

$$\tau_i + \cdots + \tau_j \leq s_{i-1} + s_i + b_i + \cdots + s_j + b_j \tag{24}$$

with $s_0 = 0$. For notational simplicity, we use the following symbols:

$$\nu_i = \begin{cases} \sum_{k=i}^{K} \tau_k, & i = 1, 2, \ldots, K \\ 0, & i = K+1 \end{cases} \tag{A}$$

$$t_{ij} = \sum_{k=i}^{j} b_k + \sum_{k=i-1}^{j} s_k. \tag{B}$$

Then (24) can be written as

$$\nu_i - \nu_{j+1} \leq t_{ij}. \tag{25}$$

Applying the following algorithm to any state $(\tau_1, \tau_2, \ldots, \tau_K)$, we know whether it is a dummy state or not. And if it is a dummy state, we find its root.

ALGORITHM

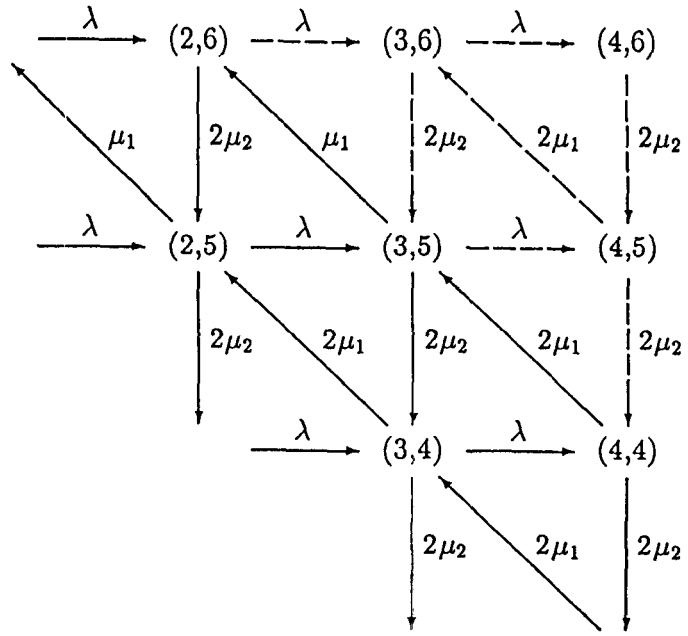


Figure 4a: MC-B_1 for the 2-node model

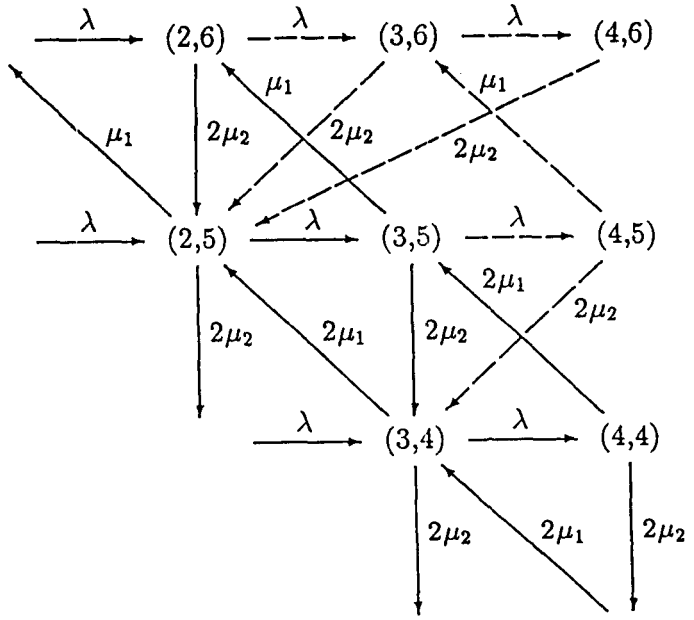


Figure 4b: MC-B_2 for the 2-node model

1. Step 1

   Calculate each ν_i for $(\tau_1, \tau_2, \ldots, \tau_K)$.

2. Step 2

   (a) Let $i = 1$ and $j = 1$.

   (b) If inequality (25) is satisfied then go to (d) else go to (c).

   (c) Reduce ν_i until equality holds in inequality (25), then go to (d).

   (d) If $j < K$, increase j by 1 and return to (b).
   If $j = K$ and $i < K$, increase i by 1, set $j = 1$, and return to (b).
   If $j = K$ and $i = K$ then go to step 3.

3. Step 3

   1) If the original vector $(\tau_1, \tau_2, \ldots, \tau_K)$ satisfies inequality (25) for all i and j, it is not a dummy state.

   2) If $(\tau_1, \tau_2, \ldots, \tau_K)$ is a dummy state, calculate new n_k from new $\{\nu_k\}$ using Equation (A). The acquired vector is the root of the dummy state.

C. Aggregate Transition Rates for State Description B

Here we show how to calculate the aggregate transition rate matrices using assumptions in the case of State description B, specifically for Level-2 approximation. For State description A, readers may refer Appendix 2 in [3].

To avoid cumbersome discussion on the number of active servers in the system with multi-server nodes, we first consider the case of single-server nodes. Even such a case the transition rates become complex because dummy transitions at Node k' $(k' > k)$ may change state of Node k. To make discussion clearer, similar to states in $S_{1,2,\ldots,K}$, we classify states in $S_{k,k+1,\ldots,k'}$ into three categories: inner states, dummy states and boundary states. We call state $(\tau_k, \tau_{k+1}, \ldots, \tau_{k'})$ in $S_{k,k+1,\ldots,k'} - \tilde{S}_{k,k+1,\ldots,k'}$ as a dummy state. For state $(\tau_k, \tau_{k+1}, \ldots, \tau_{k'})$ in $\tilde{S}_{k,k+1,\ldots,k'}$, we call it a boundary state if $(\tau_k, \tau_{k+1}, \ldots, \tau_{j-1}, \tau_j + 1, \tau_{j+1}, \ldots, \tau'_k)$ is a dummy state for some j, $(j = k, k+1, \ldots, k')$, and we call it an inner state if not. Applying the algorithm proposed in Appendix B to a vector $(\tau_k, \tau_{k+1}, \ldots, \tau_{k'})$ with $\tau_{k'} > 0$, we can see whether it is a dummy state or not. If it is a dummy state, we can also determine its destination $(\tau'_k, \tau'_{k+1}, \ldots, \tau'_{k'})$ and the rate of transition which contributes to $q_{k,k+1}(\tau_k, \tau_{k+1}; \tau'_k, \tau'_{k+1})$.

The aggregate transition rate matrix $\boldsymbol{Q}_{k,k+1}$ $(k < K)$ for the single-server case is given in the following 4 steps.

1) Service completions at Node $k-1$ $(1 < k < K)$ yield arrivals to node pair $(k, k+1)$, and they occur in the case that $\tau_{k-1} > 0$ and $\tau_k < N_k - 1$. Hence

$$
\begin{aligned}
&q_{k,k+1}(\tau_k, \tau_{k+1}; \tau_k + 1, \tau_{k+1}) \\
&\quad = \sum_{\tau_{k-1}=1}^{N_{k-1}-1} \mu_{k-1} \Pr\{\boldsymbol{X}_{k-1} = \tau_{k-1} \mid \boldsymbol{X}_k = \tau_k, \boldsymbol{X}_{k+1} = \tau_{k+1}\} \\
&\quad \approx \sum_{\tau_{k-1}=1}^{N_{k-1}-1} \mu_{k-1}\, x_{k-1|k}(\tau_{k-1} \mid \tau_k), \\
&\qquad\qquad \tau_k < N_k - 1,\ \tau_{k+1} = 0, 1, \ldots, N_{k+1} - 1,
\end{aligned}
$$

where

$$\begin{aligned} x_{k-1|k}(\tau_{k-1} \mid \tau_k) &= \Pr\{\boldsymbol{X}_{k-1} = \tau_{k-1} \mid \boldsymbol{X}_k = \tau_k\} \\ &= x_{k-1,k}(\tau_{k-1}, \tau_k) / \sum_{\tau'_{k-1}} x_{k-1,k}(\tau'_{k-1}, \tau_k). \end{aligned}$$

2) A service completion at Node k ($k < K$) transfers a customer from Node k to Node $k+1$, and the rate associated with it does not depend on states of other nodes. Hence

$$q_{k,k+1}(\tau_k, \tau_{k+1}, \tau_k - 1, \tau_{k+1} + 1) = \mu_k, \quad \tau_k > 0,\ \tau_{k+1} < N_{k+1} - 1.$$

3) Outputs from node pair $(k, k+1)$ ($k < K-1$) are due to service completions at Node $k+1$. They occur in the case that $\tau_{k+1} > 0$ and $\tau_{k+2} < N_{k+2} - 1$. Their destinations are rather complicated because some of them are from dummy states to inner states. So we have to apply the algorithm proposed before to find out such destinations. Suppose the destination is $(\tau'_k, \tau_{k+1} - 1)$, then the corresponding aggregate transition rate is

$$\begin{aligned} & q_{k,k+1}(\tau_k, \tau_{k+1}; \tau'_k, \tau_{k+1} - 1) \\ &= \sum_{\tau_{k+2}=0}^{N_{k+2}-2} \mu_{k+1} \Pr\{\boldsymbol{X}_{k+2} = \tau_{k+2} \mid \boldsymbol{X}_k = \tau_k, \boldsymbol{X}_{k+1} = \tau_{k+1}\} \\ &\approx \sum_{\tau_{k+2}=0}^{N_{k+2}-2} \mu_{k+1}\ x_{k+2|k+1}(\tau_{k+2} \mid \tau_{k+1}) \\ &= \sum_{\tau_{k+2}=0}^{N_{k+2}-2} \mu_{k+1}\ x_{k+1,k+2}(\tau_{k+1}, \tau_{k+2}) / \sum_{n_{k+2}} x_{k+1,k+2}(\tau_{k+1}, n_{k+2}), \\ & \qquad \tau_k = 0, 1, \ldots, N_k - 1,\ \tau_{k+1} > 0. \end{aligned}$$

4) We further need some revision because dummy transitions at Node k' ($k' > K+1$) may also change state of Node k. To a vector $(\tau_k, \tau_{k+1}, \ldots, \tau_{k'})$ with $\tau_{k'} > 0$, applying the algorithm proposed before, we know whether it is a dummy state or not. If it is a dummy state, we can determine its destination $(\tau'_k, \tau'_{k+1}, \ldots, \tau'_{k'})$ and the rate of transition which contributes to $q_{k,k+1}(\tau_k, \tau_{k+1}; \tau'_k, \tau'_{k+1})$. The rate of the transition is given by

$$\begin{aligned} & \sum_{\tau_{k'+1}=0}^{N_{k'+1}-2} \mu_{k'} \Pr\{\boldsymbol{X}_{k+2} = \tau_{k+2}, \ldots, \boldsymbol{X}_{k'} = \tau_{k'}, \boldsymbol{X}_{k'+1} = \tau_{k'+1} \mid \boldsymbol{X}_k = \tau_k, \boldsymbol{X}_{k+1} = \tau_{k+1}\} \\ & \qquad \approx \sum_{\tau_{k'+1}=0}^{N_{k'+1}-2} \mu_{k'}\ x_{k+2|k+1}(\tau_{k+2} \mid \tau_{k+1}) \cdots x_{k'+1|k'}(\tau_{k'+1} \mid \tau_{k'}). \end{aligned}$$

We sum up all rates of such transitions from (τ_k, τ_{k+1}) to (τ'_k, τ'_{k+1}). If (τ'_k, τ'_{k+1}) is one of the vectors we used in 1), 2) or 3), the sum should be added to the transition rate we have gotten. Otherwise, the sum becomes the aggregate transition rate $q_{k,k+1}(\tau_k, \tau_{k+1}; \tau'_k, \tau'_{k+1})$. In both cases, the diagonal entry $q_{k,k+1}(\tau_k, \tau_{k+1}; \tau_k, \tau_{k+1})$ must be revised correspondingly, too.

For a system with multi-server nodes, the aggregate transition rate matrix becomes more complicated, because we can not determine the number of active servers in Node k only from

the states of Nodes k and $k+1$ in many cases.

In the case of 1) above, if $\tau_k \leq b_k$, we only need to insert a coefficient $\min\{\tau_{k-1}, s_{k-1}\}$ before μ_{k-1} in the second and third lines of the equation. If $\tau_k > b_k$ and $\tau_{k+1} \leq b_{k+1}$, the coefficient becomes $\min\{\tau_{k-1}, s_{k-1}, N_k - \tau_k\}$. Otherwise, the coefficient is not a constant, and it is given as a mixture of the number of active servers which depends on the states of Nodes $k+2, k+3, \ldots, K$, with the corresponding probability approximated from Assumption 2.

In the case of 2), if $\tau_{k+1} \leq b_{k+1}$, we only need to put a coefficient $\min\{\tau_k, s_k\}$ before μ_k in the equation. Otherwise, the coefficient is given by a mixture as above.

As for 3) and 4), the rate becomes more complex and has to be determined individually.

References

[1] Altiok, T.: Approximate Analysis of Exponential Tandem Queues with Blocking, *European Journal of Operational Research*, Vol. 11 (1982), 390–398.

[2] Brandwajn, A.: Equivalence and Decomposition in Queueing Systems — A Unified Approach, *Performance Evaluation*, Vol. 5 (1985), 175–186.

[3] Brandwajn, A. and Jow, Y.L.: Tandem Exponential Queues with Finite Buffers, in *Computers Networking and Performance Evaluation*, Hasegawa, T., Takagi, H. and Takahashi, Y. (Eds.) Elsevier, (North Holland) (1986), 245–257.

[4] Brandwajn, A. and Jow, Y.L.: An Approximation Method for Tandem Queues with Blocking, *Operations Research*, Vol. 36 (1988), 73–83.

[5] Choong, Y.F. and Gershwin, S.B.: A Decomposition Method for the Approximate Evaluation of Capacitated Transfer Lines with Unreliable Machines and Random Processing, *IIE Transaction*, Vol. 19 (1987), 150–159.

[6] Courtois, P.J.: *Decomposability — Queueing and Computer System Approximations*, Academic Press, New York, (1977).

[7] Gershwin, S.B.: An Efficient Decomposition Method for the Approximate Evaluation of Tandem Queues with Finite Storage Space and Blocking, *Operations Research*, Vol. 35 (1987), 291-305.

[8] Jun, K.P. and Perros, H.G.: *An Approximate Analysis of Open Tandem Queueing Networks with Blocking and General Service Times*, Center for Communications and Signal Processing Operations Research Program, North Carolina State University, Raleigh, (1986).

[9] Liu, X.G. and Buzacott, J.A.: A Zero-buffer Equivalence Technique for Decomposing Queueing Network with Blocking, in *Queueing Networks with Blocking*, Perros, H.G. and Altiok, T. (Eds), North-Holland (1989), 87–104.

[10] Perros, H.G. and Altiok, T.: Approximate Analysis of Open Networks of Queues with Blocking: Tandem Configurations, *IEEE Transactions on Software Engineering*, SE-12 (1986), 450–461.

[11] Pollock, S.M., Birge, J.R., and Alden, J.M.: *Approximation Analysis for Open Tandem Queues with Blocking*, Univ. of Michigan, Dept. of Industrial and Operations Engineering, Ann Arbor, Michigan. Technical Report 85-30 (1985).

[12] Schweitzer, P.J.: Aggregation Methods for Large Markov Chains, in *Mathematical Computer Performance and Reliability*, Iazeolla, G., Courtois, P.J. and Hordijk, A. (Eds.), Elsevier (North Holland) (1984), 275–286.

[13] Schweitzer. P.J. and Altiok, T.: Aggregate Modeling of Tandem Queues with Blocking, in *Computer Performance and Reliability*, Iazeolla, G., Courtois, P.J. and Boxma, O.J. (Eds.), Elsevier (North Holland) (1988), 135–149.

[14] Schweitzer. P.J. and Altiok, T.: Aggregate Modeling of Tandem Queues without Intermediate Buffers, in *Queueing Networks with Blocking*, Perros, H.G. and Altiok T.

(Eds.), North-Holland (1989), 47–72.

[15] Takahashi, Yukio: *A Lumping Method for Numerical Calculation of Stationary Distributions of Markov Chains*, Research Reports on Information Sciences B-18, Tokyo Institute Technology, Tokyo, (1975).

[16] Takahashi, Yukio: A New Type Aggregation Method for Large Markov Chains and its Application to Queueing networks, in *Proceedings of ITC 11*, (Kyoto, Japan, Sept. 1985) Akiyama, M. (EdS.), Elsevier (North Holland), IAC 1985, pp. 1,4 A.1.1–4.

[17] Takahashi, Yukio: Aggregate Approximation for Acyclic Queueing Networks with Communication Blocking, in *Queueing Networks with Blocking*, Perros, H.G. and Altiok T. (Eds.), North-Holland (1989), 33–46.

[18] Takahashi, Yutaka, Miyahara, H. and Hasegawa, T.: An Approximation Method for Open Restricted Queueing Networks, *Operations Research*, Vol. 28 (1980), 597–602.

Yu SONG and Yukio TAKAHASHI:
Department of Information Science
Tokyo Institute of Technology
Oh-okayama, Meguro-ku
Tokyo 152, Japan