A NEW TYPE AGGREGATION METHOD FOR LARGE MARKOV CHAINS
AND ITS APPLICATION TO QUEUING NETWORKS

Yukio TAKAHASHI
TOHOKU UNIVERSITY
Sendai, Japan

ABSTRACT

In some queueing networks we encounter in traffic models, each node is connected with only a few adjacent nodes. Tandem queueing networks are typical examples of them. In such a network, stochastic behavior of a node might be expected to be approximately independent of that of nonadjacent nodes, though it may depend highly on the behavior of adjacent nodes.

Here, a new aggregation method is proposed to approximately calculate the stationary probabilities of large Markov chains derived from such queueing networks. It exploits the nearly independent feature between nonadjacent nodes, and makes it possible to approximately calculate the stationary marginal probabilities with relatively small computational burden.

1. INTRODUCTION

If a queuing network does not have a product form solution, it is usually very difficult to calculate its stationary probabilities and its performance measures exactly. A usual approach is to form a Markov chain which describes the stochastic behavior of the network and to numerically solve the set of linear equations of the stationary probabilities of the chain. If the network has r nodes and if J, states are required to represent the state of Node k, then we need

\[ J = \prod_{k=1}^{r} J_k \]

states to represent the state of the whole network. J grows very rapidly as r increases. So the naive Markov chain approach fails even for networks with more than a few nodes.

In order to overcome this difficulty, a number of approximation methods have been proposed. For example, [9] and [1] proposed relatively simple approximation methods for open networks with blocking and [10] did so for open networks without blocking.

In most of the approximation methods, the idea of node-by-node decomposition is used. Each node of the network is approximated by a suitable standard queueing model and the performance measures are calculated as if these approximate queues are mutually independent. Such decomposition methods enable us to compute performance measures with relatively small computational burden. However, error assessment is not easy for the results. In some cases they may give a fairly accurate solution but in some cases they may not. We have no means to estimate the accuracy of the results in specific cases. This is a major drawback of the decomposition methods.

In this paper, another approximation method is proposed for networks with a Markov chain representation. This method is based on the aggregation technique for large scale Markov chains. First we choose a set of aggregated states of the chain and form a set of equations for the corresponding stationary probabilities. This is done by adopting the usual aggregation technique. Generally the set of equations so formed contains variables other than the ones first chosen. To make the set of equations solvable, we have to introduce some assumptions on the excess variables. Usually this is done by assuming some statistical independence among nodes. Then we can reform the set of equations so that it contains only the variables first chosen. The resultant equations may not be linear. But a suitable iterative method seems to work well for solving the set of equations.

The merit of this method is in the flexibleness in the choice of the aggregated states and in the independence assumption imposed. If we want a more accurate solution, we may choose more variables first and impose a weaker assumption. If we want a rough result with less computational burden, then we may choose less variables and assume more strict independence on the node. The accuracy of the solution, of course, greatly depends on the closeness of the independence assumption to the real situation. However, if we adopt multiple levels of assumptions and if the discrepancy among the results is small, then we might as well conclude that the results are sufficiently accurate.

In the next section, we investigate the method using an example of the application to a tandem network, and in Section 3, we present some numerical results and give some comments on the applicability of the method.

2. A NEW AGGREGATION METHOD

In order to make the investigation clearer, we shall use an example of a tandem network with 5 nodes.

2.1 A tandem network example

Let us consider the tandem network with 5 nodes designated in Fig.1. Customers arrive at the network through a phase type renewal process. Node k (k=1,2,...,5) has s_k servers for service and n_k buffers for customers waiting for their service. We consider the following simple blocking rule. All servers at Node k are
blocked and stop their service when Node k+1 becomes full. If the service for a customer at Node k+1 is completed and a vacancy appears at the buffer, then the servers at Node k are immediately unblocked. Arriving customers are lost if Node 1 is full.

If the service time distributions are of phase types, then the state of the network can be represented by a quintuple

\[(i_1, i_2, i_3, i_4, i_5),\]

where \(i_k\) represents the state of Node k. If we number the states of the network in lexicographic order, then the infinitesimal generator of the Markov chain which represents the stochastic behavior of the network takes the form

\[
Q = Q_{11} \otimes I_2 \otimes I_3 \otimes I_4 \otimes I_5 \\
+ Q_{12} \otimes I_3 \otimes I_4 \otimes I_5 \\
+ I_1 \otimes Q_{23} \otimes I_4 \otimes I_5 \\
+ I_1 \otimes I_2 \otimes Q_{34} \otimes I_5 \\
+ I_1 \otimes I_2 \otimes I_3 \otimes Q_4 \otimes I_5 \\
+ I_1 \otimes I_2 \otimes I_3 \otimes I_4 \otimes Q_5,
\]

where \(Q_{kj}\) is a \((J_k \times J_j)\) matrix governing the state transitions caused by a service completion at Node k, \(Q_{11}\) is a \(J_1 \times J_1\) matrix governing the state transitions caused by an arrival of a customer, and \(Q_{23}\) is a \(J_2 \times J_3\) matrix governing the state transitions caused by a service completion at Node 5. \(I_k\) is the identity matrix of order \(J_k\), and \(\otimes\) represents the Kronecker product operation of matrices.

Note that

\[
Q_k e_1 = 0, \quad Q_5 e_5 = 0
\]

\[
Q_{k,k+1} (e_k \otimes e_{k+1}) = 0, \quad k=0,\ldots,5.
\]

The stationary probability vector \(x\) satisfies the equations

\[
xQ = 0
\]

\[
x e = 1
\]

where \(e\) is a column vector with all entries equal to 1.

2.2 Some notations

Now we prepare some notations for stationary probabilities.

We denote the entry corresponding to the state \((i_1, i_2, i_3, i_4, i_5)\) of the stationary probability vector \(x\) by \(x(i_1, i_2, i_3, i_4, i_5)\) and their marginals by

\[
x^k(i_k) = \sum_{\ell \notin k} x(i_1, i_2, i_3, i_4, i_5)
\]

\[
\sum_{\ell \notin k} x^k(i_k) = \sum_{\ell \notin k} x(i_1, i_2, i_3, i_4, i_5)
\]

\[
x^{k,j}(i_k, i_j) = \sum_{\ell \notin k,j} x(i_1, i_2, i_3, i_4, i_5)
\]

Row vectors having these marginal probabilities as entries are denoted by \(x_k, x_{kj}, x_{kj,m}\) respectively. If we put

\[
z_k = e_1 \otimes \ldots \otimes e_{k-1} \otimes I_k \otimes e_{k+1} \otimes \ldots \otimes e_5
\]

\[
z_{kj} = e_1 \otimes \ldots \otimes e_{k-1} \otimes I_k \otimes e_{k+1} \otimes \ldots \otimes e_{j-1} \otimes I_j \otimes e_{j+1} \otimes \ldots \otimes e_5
\]

\[
z_{kj,m} = e_1 \otimes \ldots \otimes e_{k-1} \otimes I_k \otimes e_{k+1} \otimes \ldots \otimes e_{j-1} \otimes I_j \otimes e_{j+1} \otimes \ldots \otimes e_{m-1} \otimes I_m \otimes e_{m+1} \otimes \ldots \otimes e_5,
\]

where \(e_j\) is the column vector of order \(J_j\) with all entries equal to 1, then we have

\[
x_k = x_k
\]

\[
x_{kj} = x_{kj}
\]

\[
x_{kj,m} = x_{kj,m}
\]

We also denote conditional probabilities as follows.

\[
x^k_{i_k} = \frac{x^{k,j}(i_k, i_j)}{x^k(i_k)}
\]

\[
x^{m,j}_{i_k} = \frac{x^{k,j,m}(i_k, i_j)}{x^{k,j}(i_k, i_j)}
\]

2.3 The first stage approximation

For the first stage approximation, we will take \(\{i_k\}, \ i_k=1,\ldots,J_k, \ k=1,\ldots,5\), as aggregated states. Namely, we take marginal probabilities \(x^k(i_k)\) as variables to be determined.
In order to derive equations for the variables \( x^k(i_k) \), we will post multiply \( z_k \) to the first equation of (3). Then from (1) and (6), we have

\[
0 = xQz_1 = x^1Q_1 + x^{12}Q_{12}(I_1 \otimes e_2)
\]

\[
0 = xQz_2 = x^{12}Q_{12}(e_1 \otimes I_2) + x^{23}Q_{23}(I_2 \otimes e_3)
\]

\[
0 = xQz_3 = x^{23}Q_{23}(e_2 \otimes I_3) + x^{3k}Q_{3k}(I_3 \otimes e_4)
\]

\[
0 = xQz_4 = x^{3k}Q_{3k}(e_3 \otimes I_4) + x^{45}Q_{45}(I_4 \otimes e_5)
\]

\[
0 = xQz_5 = x^{45}Q_{45}(e_4 \otimes I_5) + x^5Q_{23}.
\]

These equations contains vectors \( x^k,k+1,s \) other than \( x^k,s \). So we have to make some assumption which relates \( x^k,k+1,s \) to \( x^k,s \). The most natural one is the following.

**Assumption A1**

We assume that for any possible choice of \( k, i_{k-1}, i_k \)

\[
(9a) \quad x^k|k-1(i_{k-1}i_k) = x^k(i_k)
\]

or equivalently,

\[
(9b) \quad x^k,k+1 = x^k \otimes x^{k+1}.
\]

This assumption asserts that the stochastic behavior of successive nodes is mutually independent. However it does not mean that

\[
(10) \quad x(i_{1,1},i_{2,1},i_{3,1},i_{4,1},i_{5,1}) = \prod_k x^k(i_k).
\]

Hence, the independence assumption (9) is not so strict one.

Inserting (9) to (8), we have the desired set of equations for \( x^k,s \).

\[
(11a) \quad 0 = x^1Q_1 + (x^1 \otimes x^2)Q_{12}(I_1 \otimes e_2)
\]

\[
(11b) \quad 0 = (x^2 \otimes x^2)Q_{12}(e_1 \otimes I_2)
\]

\[
(11c) \quad 0 = (x^3 \otimes x^3)Q_{23}(e_2 \otimes I_3)
\]

\[
(11d) \quad 0 = (x^4 \otimes x^4)Q_{34}(e_3 \otimes I_4)
\]

\[
(11e) \quad 0 = (x^5 \otimes x^5)Q_{45}(e_4 \otimes I_5) + x^5Q_{23}.
\]

Together with the normalization constraints

\[
(11f) \quad x^ke_k = 1, \text{ for } k=1,\ldots,5,
\]

these equations uniquely determine the marginal probabilities \( x^k(i_k) \).

Most of the equations in (11) are nonlinear equations. However, if we regard, for example, \( x^1 \) and \( x^2 \) in (11b) being known, then (11b) is a set of linear equations for variables \( x^2(i_1) \)’s.

So, the set of equations (11) can be solved numerically using a suitable iterative method as usually done in the aggregation/disaggregation method.

### 2.4 The second stage approximation

For the second stage approximation, we will take pairs \( (i_k,i_{k+1}) \) of states of successive nodes. So, we will take \( x(i_{k+1}) \)’s and \( x^k|k-1(i_{k-1}i_k) \)’s as variables to be determined.

Here we don’t take \( x^k|k-1(i_{k-1}i_k) \)’s as variables because some of them are related as

\[
(12) \quad x^k|k-1(i_{k-1}i_k \otimes I_k) = x^k \otimes x^{k+1}(I_k \otimes e_k).
\]

As we have done in the first stage approximation, we can derive a set of equations for these variables by post multiplying \( z_k \) to the first equation of (3) as follows.

\[
0 = xQz_1 = x^1Q_1 + x^{12}Q_{12}(I_1 \otimes e_2)
\]

\[
0 = xQz_{12} = (x^{12}Q_{12}(e_1 \otimes I_2) + x^{23}Q_{23}(I_2 \otimes e_3)
\]

\[
0 = xQz_{3} = x^{23}Q_{23}(e_2 \otimes I_3) + x^{3k}Q_{3k}(I_3 \otimes e_4)
\]

\[
0 = xQz_{4} = x^{3k}Q_{3k}(e_3 \otimes I_4) + x^{45}Q_{45}(I_4 \otimes e_5)
\]

\[
0 = xQz_{5} = x^{45}Q_{45}(e_4 \otimes I_5) + x^5Q_{23}.
\]

In this case, the most natural assumption will be the one given below.

**Assumption A2**

We assume that for any possible choice of \( k, i_{k-1}, i_k, i_{k+1} \)

\[
(14) \quad x^{k+1}|k-1,i_k(i_{k-1}i_ki_{k+1}) = x^{k+1}(i_{k+1}i_k).
\]

This assumption A2 asserts that the stochastic behavior of Node \( k \) and Node \( k+1 \) is independent under the condition of the state of Node \( k \) being \( i_k \). Strictly speaking, the assumptions A1 (which was imposed in the first stage approximation) and A2 here cannot be compared directly. A1 does not automatically imply A2. However, intuitively, we may consider that A2 is weaker than A1 and that the resultant solution is closer to the real situation.

Using the relation (14), we can reform the set (13) of equations to the one for variables \( x^k(i_k) \)’s and \( x^k|k-1(i_{k-1}i_k) \)’s. The resultant set of equations is slightly more complicated than (11) to write down in a vector and matrix form, but is not difficult to write a computer program for implementation of an iterative calculation.
Table 1. Numerical results of the application to a tandem network example

<table>
<thead>
<tr>
<th>Case 3</th>
<th>1st stage approximation</th>
<th>2nd stage approximation</th>
<th>exact</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda = 1$</td>
<td>$\mu_1 = \mu_3 = \mu_5 = .5$</td>
<td>$\mu_2 = \mu_4 = 1$</td>
<td></td>
</tr>
<tr>
<td>loss prob.</td>
<td>$L_1$</td>
<td>$L_2$</td>
<td>$L_3$</td>
</tr>
<tr>
<td>.2425</td>
<td>2.309</td>
<td>1.184</td>
<td>2.283</td>
</tr>
<tr>
<td>.2640</td>
<td>2.404</td>
<td>1.479</td>
<td>2.353</td>
</tr>
<tr>
<td>.2661</td>
<td>2.407</td>
<td>1.465</td>
<td>2.347</td>
</tr>
<tr>
<td>Case 2</td>
<td>1st stage approximation</td>
<td>2nd stage approximation</td>
<td>exact</td>
</tr>
<tr>
<td>$\lambda = 1$</td>
<td>$\mu_1 = \mu_3 = \mu_5 = .5$</td>
<td>$\mu_2 = \mu_4 = .5$</td>
<td></td>
</tr>
<tr>
<td>loss prob.</td>
<td>$L_1$</td>
<td>$L_2$</td>
<td>$L_3$</td>
</tr>
<tr>
<td>.3589</td>
<td>2.739</td>
<td>2.426</td>
<td>2.209</td>
</tr>
<tr>
<td>.3612</td>
<td>2.753</td>
<td>2.434</td>
<td>2.204</td>
</tr>
<tr>
<td>.3580</td>
<td>2.746</td>
<td>2.429</td>
<td>2.204</td>
</tr>
<tr>
<td>Case 3</td>
<td>1st stage approximation</td>
<td>2nd stage approximation</td>
<td>exact</td>
</tr>
<tr>
<td>$\lambda = 1$</td>
<td>$\mu_1 = \mu_3 = \mu_5 = .5$</td>
<td>$\mu_2 = \mu_4 = .5$</td>
<td></td>
</tr>
<tr>
<td>loss prob.</td>
<td>$L_1$</td>
<td>$L_2$</td>
<td>$L_3$</td>
</tr>
<tr>
<td>.4248</td>
<td>2.944</td>
<td>2.777</td>
<td>2.071</td>
</tr>
<tr>
<td>.4263</td>
<td>2.952</td>
<td>2.782</td>
<td>2.098</td>
</tr>
<tr>
<td>.4273</td>
<td>2.950</td>
<td>2.784</td>
<td>2.098</td>
</tr>
</tbody>
</table>

Model: A tandem network with 5 nodes. Every node has two servers and a buffer of size 2. Customers arrive through a Poisson process with rate $\lambda$, and are served at Node $k$ subjecting to the exponential distribution with rate $\mu_k$. If Node $k+1$ is full, then the servers at Node $k$ are blocked and stop their service. Arriving customers are lost if Node 1 is full. $\bar{n}_k$ is the mean number of customers at Node $k$ in the steady state.

3. NUMERICAL EXAMPLES AND SOME COMMENTS

3.1 Numerical examples

This approximation method was tested with the tandem network used in the preceding section to investigate the method. The model tested is as follows. Each node of the network has two servers and a buffer of size 2. Namely, $s_k = m_k = 2$ for all $k$ from 1 through 5. Customers arrive at the network through a Poisson process with rate $\lambda$, and are served at Node $k$ subjecting to the exponential distribution with rate $\mu_k$.

Table 1 summarizes the results of three cases with different parameter sets. For each case, the values of the loss probability and of the mean number $\bar{n}_k$ of customers at Node $k$, $k = 1, \ldots, 5$, are calculated from the results of the first and the second stage approximations. They are close with each other. This seems to indicate that those values are fairly accurate. In fact, they are very close to the exact values calculated by the aggregation/disaggregation method. Especially, the values by the 2nd stage approximation are very accurate. The maximum relative error is less than 1.5%.

An intuitive reasoning of the accuracy of the results by the 2nd stage approximation is that the stochastic behavior of nonadjacent nodes is nearly independent though that of adjacent nodes is highly dependent. This must be a great hint for applications of this method to the analyses of queueing networks.

The number of equations to be solved in the 1st stage approximation is 25, and that in the 2nd stage approximation is 125, while a system of 3125 equations have to be solved in the exact calculation by the aggregation/disaggregation method.

3.2 Some comments on the applicability

The aggregation/disaggregation method makes it possible for us to calculate exact solutions of fairly large Markov chains (see, e.g., [6], [8], [4] and [5]). However, it still has a dimensionality problem that the computation is at least linear order of the number of the states. So, even using the aggregation/disaggregation method, practically we cannot treat Markov chains having more than 100000 states.

Another type of aggregation method was developed by Courtois in [2]. A nice application of it was given in [3]. It saves a lot of computing time, if it can be applied. However, the required condition is not general enough. The applicability is restricted.

The approximated method proposed in this paper, on the other hand, has much flexibility. We can choose several levels of approximations, and by comparing the results we can estimate the accuracy of the solutions. Hence, this method seems to have a wide range of applicability.

However, at present we have only a little experience of application. We have to have more experience for evaluating the applicability correctly.
REFERENCES


