<table>
<thead>
<tr>
<th>Title</th>
<th>An algorithm for computing the exact steady state distribution of a cyclically connected queueing system (Combinatorial Structure in Mathematical Models)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Jimbo, Masakazu; Mori, Tatsuji</td>
</tr>
<tr>
<td>Citation</td>
<td>数理解析研究所講究録 (1993), 853: 1-19</td>
</tr>
<tr>
<td>Issue Date</td>
<td>1993-11</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/2433/83746">http://hdl.handle.net/2433/83746</a></td>
</tr>
<tr>
<td>Type</td>
<td>Departmental Bulletin Paper</td>
</tr>
<tr>
<td>Textversion</td>
<td>publisher</td>
</tr>
</tbody>
</table>

Kyoto University
An algorithm for computing the exact steady state
distribution of a cyclically connected queueing system
巡回型待ち行列系の定常分布を求めるためのアルゴリズム
について

Masakazu Jimbo, Gifu Univ.)
Tatsuji Mori, Suzuki Co.)

In this paper, a kind of queueing network is treated, which is referred to a "cyclically
connected queueing system." In this queueing network, servers are settled on nodes which
are connected cyclically by edges and a customer receives a service at the arrival node or
at one of the neighbour nodes. Here, for a cyclically connected queueing system, a fast
algorithm to obtain the exact steady state distribution as functions of the traffic intensity
is proposed.

Queueing networks have been studied by many authors (see, for example, Hunt (1956),
Chapter 6)). Queueing networks can be categorized as: open networks, closed networks,
restricted networks, unrestricted networks and so on. In these queueing networks, servers are settled
on nodes and customers move along edges, that is, after completion of a service at one node a customer either goes to another node, or turns away from the system.

In this paper, we consider a different kind of queueing network. In our queueing network, a server is settled on each node and these nodes are connected by edges similarly to the above queueing network. But a customer receives service only once at the arrival node or at one of the neighbour nodes and he/she turns away after receiving a service. Let $V$ be a set of nodes and $E$ be a set of edges connecting these nodes. Then the queueing system is represented by a graph $G = (V, E)$. For nodes $u$ and $v$, if there is an edge which connects the nodes $u$ and $v$, then $u$ and $v$ are said to be adjacent. For a node $v$, the number of nodes which are adjacent to $v$ is called the degree of $v$, denoted by $\text{deg}(v)$. If the number of service nodes are equal to the number of edges and $\text{deg}(v) = 2$ for any $v \in V$ then the queueing system is said to
be *cyclically connected*. That is, in a cyclically connected queueing system, the nodes are connected like a loop. For example, there are 11 queueing systems with four nodes as is shown in Figure 1. Among these, $G_9$ is cyclically connected.

**Figure 1:** Network queueing systems with four nodes

In our queueing systems, we assume the following conditions:

(A1) Customer arrivals occur at each service node.

(A2) Customers arrive at each service node with mean arrival rate $\lambda$, and the arrivals are independent each other. Customers receive service with mean service rate $\mu$ at any server. That is, we assume Poisson arrivals and exponential services.
(A3) There is no waiting line (queue) for service, that is, the number of the customers in each service node is 0 or 1.

(A4) A customer receives a service either at the arriving service node or the adjacent service node. When a customer arrives at a service node, he receives a service as follows:

(a) If the arrival node is not occupied, then the customer receives a service at the arrival node.

(b) If the arrival node is occupied and if there are \(m(>0)\) adjacent nodes which are not occupied then the customer receives a service at one of the \(m\) vacant nodes with probability \(1/m\).

(c) If the arrival node and all of the adjacent nodes are occupied, the customer turns away without receiving a service.

(A5) In the system, at most one transaction (arrival or served) can occur in a very short period, say \(\Delta t\).

In the following sections, we will consider a cyclically connected queueing system and will give a fast algorithm to solve the “steady state equations.” In Section 1, the steady state equation for a cyclically connected queueing system is constituted. In Section 2, the number of variables and equations are reduced by using symmetry of graphs. In Section 3, we consider a fast algorithm to solve the reduced steady state equations. In Section 4, the probability of loss is computed using the result of Section 3.

1. THE STATE SPACE AND STEADY STATE EQUATIONS

Let \(V = \{1,2,\cdots,n\}\) be a set of service nodes. If there is a customer on a service node, the state of the node is 1, otherwise 0. Let \(R = \{0,1\}\), and \(R^n = \{(x_1,x_2,\cdots,x_n)|x_i \in R\}\). We call \(R^n\) the state space and each element of \(R^n\) a state of the system (or simply a state). For a state \((x_1,\cdots,x_n) \in R^n\), \(x_i = 0\) implies that the \(i\)-th node is vacant and \(x_i = 1\) implies that the node is occupied. For example, the state \((1,0,0,0)\) means that there is one customer on service node 1 and there are no customers on service nodes 2, 3 and 4.

For any two states \(x = (x_1,\cdots,x_n)\) and \(y = (y_1,\cdots,y_n)\), the number of \(i\)'s such that \(x_i \neq y_i\) is called the Hamming distance between \(x\) and \(y\),
denoted by $d(x, y)$. The Hamming distance between $x$ and $0$ is called the *Hamming weight* of $x$, denoted by $w(x)$.

Hereafter we focus our attention to a cyclically connected queueing system. In order to constitute steady state equations we need to consider state transitions and transition probabilities. For cyclically connected queueing systems, a state transition can occur between two states whose Hamming distance is 1 or 0, since at most one transition can occur in a short period $\Delta t$.

**Example 1.** In the case of $n = 4$, consider the two states $(1,0,0,0)$ and $(1,1,0,0)$. The transition $(1,1,0,0) \rightarrow (1,0,0,0)$ occurs when (i) the customer at node 2 completes his service, (ii) the service does not finish at node 1 and (iii) no customer arrives within a short period $\Delta t$. Then the transition probability is $\mu \Delta t (1 - \mu \Delta t)(1 - \lambda \Delta t)^4 = \mu \Delta t$. On the other hand, the transition $(1,0,0,0) \rightarrow (1,1,0,0)$ occurs when (i) one customer arrives at the occupied service node 1 and choose the adjacent vacant node 2 with probability $1/2$, or (ii) one customer arrives at the vacant node 2. This is shown in Figure 2. By assumption (a) and (b) of (A4), the transition probability is

$$\frac{1}{2}\lambda \Delta t (1 - \lambda \Delta t)^3 (1 - \mu \Delta t) + \lambda \Delta t (1 - \lambda \Delta t)^3 (1 - \mu \Delta t) = \frac{3}{2} \lambda \Delta t.$$

![Figure 2: Arrival pattern of transition $(1,0,0,0) \rightarrow (1,1,0,0)$](image)
In general, for a cyclically connected queueing system, we call the transition \( x \to y \) **descending transition** if \( w(x) > w(y) \), and **ascending transition** if \( w(x) < w(y) \). The probability that a descending transition \( x \to y \) occurs is

\[
\mu \Delta t (1 - \mu \Delta t)^{w(y)} (1 - \lambda \Delta t)^{n} = \mu \Delta t
\]

for \( x \neq 0 \). On the other hand, the probability of an ascending transition depends on the states of the neighbours. When the state is on \( x = (x_1, \cdots, x_n) \) at time \( t \), if an ascending transition occurs at the node \( i \), then of course \( x_i \) must be 0. Let \( y = (y_1, \cdots, y_n) \) be the state after the transition occurs, then \( y_j = x_j \) holds for any \( j \neq i \) except \( y_i = 1 \). And the probability of transition \( x \to y \) depends on the value of \( x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2} \). In Table I, we list the transition probabilities for \( n \geq 5 \). Note that the reverse of the pattern \( x_{i+2}, x_{i+1}, x_i, x_{i-1}, x_{i-2} \) have the same transition probability with \( x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2} \).

### Table I

<table>
<thead>
<tr>
<th>( x_{i-2} )</th>
<th>( x_{i-1} )</th>
<th>( x_i )</th>
<th>( x_{i+1} )</th>
<th>( x_{i+2} )</th>
<th>transition probabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>( \lambda \Delta t )</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>( \lambda \Delta t )</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>( (3/2) \lambda \Delta t )</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>( 2 \lambda \Delta t )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>( (3/2) \lambda \Delta t )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>( \lambda \Delta t )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>( 2 \lambda \Delta t )</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>( (5/2) \lambda \Delta t )</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>( 2 \lambda \Delta t )</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>( 3 \lambda \Delta t )</td>
</tr>
</tbody>
</table>

Furthermore, we need to take into account of a transition \( x \to x \). A customer who arrives at a node such that the arrival node and both of the adjacent nodes are occupied turns away without receiving a service. Let \( l(x) \) be the number of 3-consecutive 1's in a state \( x \in R^n \). For example, there are three 3-consecutive 1's in \((1,1,0,1,1,1)\) since these six nodes are arranged in a loop. And there are four 3-consecutive 1's in \((1,1,1,1)\). Then the probability of a transition \( x \to x \) is

\[
(1 - \lambda \Delta t)^{n - l(x)} (1 - \mu \Delta t)^{w(x)} \triangleq 1 - (n - l(x)) \lambda \Delta t - w(x) \mu \Delta t.
\]
Let $P_x(t)$ be the probability that the state $x$ occurs at time $t$. Then in the above way we can construct the transition equations from time $t$ to time $t + \Delta t$. For example, in the case of $n = 4$, we obtain the following $2^4$ transition equations.

\[
\begin{align*}
P_{0000}(t + \Delta t) &= (1 - \lambda \Delta t)^4 P_{0000}(t) + (1 - \lambda \Delta t)^4 \mu \Delta t P_{1000}(t) + (1 - \lambda \Delta t)^4 \mu \Delta t P_{0100}(t) + (1 - \lambda \Delta t)^4 \mu \Delta t P_{0010}(t) + (1 - \lambda \Delta t)^4 \mu \Delta t P_{0001}(t), \\
P_{1000}(t + \Delta t) &= (1 - \lambda \Delta t)^4 (1 - \mu \Delta t) P_{1000}(t) + (1 - \lambda \Delta t)^4 \mu \Delta t P_{1100}(t) + (1 - \lambda \Delta t)^4 \mu \Delta t P_{1010}(t) + (1 - \lambda \Delta t)^4 \mu \Delta t P_{1001}(t) + \lambda \Delta t P_{0000}(t), \\
P_{0100}(t + \Delta t) &= (1 - \lambda \Delta t)^4 (1 - \mu \Delta t) P_{0100}(t) + (1 - \lambda \Delta t)^4 \mu \Delta t P_{0110}(t) + (1 - \lambda \Delta t)^4 \mu \Delta t P_{0101}(t) + (1 - \lambda \Delta t)^4 \mu \Delta t P_{0101}(t) + \lambda \Delta t P_{0000}(t), \\
P_{1100}(t + \Delta t) &= (1 - \lambda \Delta t)^4 (1 - \mu \Delta t)^2 P_{1100}(t) + (1 - \lambda \Delta t)^3 \mu \Delta t P_{1110}(t) + (1 - \lambda \Delta t)^3 \mu \Delta t P_{1101}(t) + \frac{3}{2} \lambda \Delta t (1 - \mu \Delta t) P_{1000}(t) + \frac{3}{2} \lambda \Delta t (1 - \mu \Delta t) P_{0100}(t), \\
&\vdots \\
P_{1111}(t + \Delta t) &= (1 - \mu \Delta t)^4 P_{1111}(t) + 3 \lambda \Delta t (1 - \mu \Delta t)^3 P_{1110}(t) + 3 \lambda \Delta t (1 - \mu \Delta t)^3 P_{1101}(t) + 3 \lambda \Delta t (1 - \mu \Delta t)^3 P_{1011}(t) + 3 \lambda \Delta t (1 - \mu \Delta t)^3 P_{0111}(t).
\end{align*}
\]

Now, let $p_x = \lim_{t \to \infty} P_x(t)$. Then the steady state equations for this case is as follows:

\[
\begin{align*}
-4 \lambda p_{0000} + \mu p_{1000} + \mu p_{0100} + \mu p_{0010} + \mu p_{0001} &= 0, \\
(-4 \lambda - \mu) p_{1000} + \mu p_{1100} + \mu p_{1010} + \mu p_{1001} + \lambda p_{0000} &= 0, \\
(-4 \lambda - \mu) p_{0100} + \mu p_{1100} + \mu p_{0110} + \mu p_{0101} + \lambda p_{0000} &= 0, \\
&\vdots \\
(-4 \lambda - 2 \mu) p_{1100} + \mu p_{1110} + \mu p_{1101} + \frac{3}{2} \lambda p_{1000} + \frac{3}{2} \lambda p_{0100} &= 0, \\
&\vdots \\
-4 \mu p_{1111} + 3 \lambda p_{1110} + 3 \lambda p_{1101} + 3 \lambda p_{1011} + 3 \lambda p_{0111} &= 0.
\end{align*}
\]

There are $2^n$ equations, in general, and as the number of node increases, the number of equations grows exponentially. In the next section, we reduce the number of variables and equations by using the symmetry of the cyclically connected queue.
2. A REDUCTION OF STEADY STATE EQUATIONS

For \((x_1, \cdots, x_n) \in \mathbb{R}^n\), let
\[
\sigma : (x_1, \cdots, x_n) \mapsto (x_2, \cdots, x_n, x_1),
\tau : (x_1, \cdots, x_n) \mapsto (x_n, \cdots, x_1).
\]
That is, \(\sigma\) acts on \(\mathbb{R}^n\) as a function of rotation, while \(\tau\) acts as a function of reverse. Let \(G\) be a permutation group which is generated by \(\sigma\) and \(\tau\). Then \(G\) is called the dihedral group which has \(2n\) elements. Two states \(x, y \in \mathbb{R}^n\) are said to be equivalent if \(x^g = y\) for some \(g \in G\). Then \(\mathbb{R}^n\) is divided into equivalence classes (orbits) by \(G\). When \(\mathbb{R}^n\) is divided into \(m\) equivalence classes, let \(S = \{S_0, S_1, \cdots, S_{m-1}\}\) be the collection of equivalence classes. Of course, \(w(x) = w(y)\) holds for any \(x, y \in S_i\). For an equivalence class \(S_i\) and \(x \in S_i\), we define the weight of \(S_i\) by \(w(S_i) = w(x)\). We assume that \(S_i\) is numbered in the ascending order of \(w(S_i)\).

Example 2. In the case of \(\mathbb{R}^4 = \{0,1\}^4\), for example, \((1,0,0,0)^\sigma = (0,1,0,0)\) and \((1,0,0,0)^\tau = (0,0,0,1)\) hold. The equivalence classes are as follows:
\[
\begin{align*}
&S_0 = \{(0,0,0,0)\}, \\
&S_1 = \{(1,0,0,0), (0,1,0,0), (0,0,1,0), (0,0,0,1)\}, \\
&S_2 = \{(1,1,0,0), (0,1,1,0), (0,0,1,1), (1,0,0,1)\}, \\
&S_3 = \{(1,0,1,0), (0,1,0,1)\}, \\
&S_4 = \{(1,1,1,0), (0,1,1,1), (1,0,1,1), (1,1,0,1)\}, \\
&S_5 = \{(1,1,1,1)\}.
\end{align*}
\]

The well-known Burnside’s Lemma and Pólya’s theorem are useful to enumerate the number of equivalence classes \(m\) (see, for example, Liu (1968), Bender and Williamson (1991)). In Table II, we show the number of equivalence classes.

<table>
<thead>
<tr>
<th>The Number of Equivalence Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of service nodes (n)</td>
</tr>
<tr>
<td>The number of equivalence classes (m)</td>
</tr>
</tbody>
</table>

It is obvious that if \(x, y \in S_i\), \(px = py\) holds. Hence, we can define \(p_{S_i} = px\) for a equivalence class \(S_i\) and for \(x \in S_i\).
Example 3. By Lemma 1, in the case of $n = 4$, we have

\[
\begin{aligned}
  p_{S_0} &= p_{0000}, \\
  p_{S_1} &= p_{1000} = p_{0100} = p_{0010} = p_{0001}, \\
  p_{S_2} &= p_{1100} = p_{0110} = p_{0011} = p_{1001}, \\
  p_{S_3} &= p_{1010} = p_{0101}, \\
  p_{S_4} &= p_{1110} = p_{0111} = p_{1011} = p_{1101}, \\
  p_{S_5} &= p_{1111}.
\end{aligned}
\]  

(3)

Therefore, by (1) and (3), the steady state equation is reduced as follows:

\[
\begin{aligned}
  -4\lambda p_{S_0} + 4\mu p_{S_1} &= 0, \\
  (-16\lambda - 4\mu)p_{S_1} + 8\mu p_{S_2} + 4\mu p_{S_3} + 4\lambda p_{S_0} &= 0, \\
  (-16\lambda - 8\mu)p_{S_2} + 8\mu p_{S_4} + 12\lambda p_{S_1} &= 0, \\
  (-8\lambda - 4\mu)p_{S_3} + 4\mu p_{S_5} + 4\lambda p_{S_4} &= 0, \\
  (-12\lambda - 12\mu)p_{S_4} + 4\mu p_{S_5} + 16\lambda p_{S_2} + 8\lambda p_{S_3} &= 0, \\
  -4\mu p_{S_5} + 12\lambda p_{S_4} &= 0.
\end{aligned}
\]  

(4)

In general, for a cyclically connected queueing system with $n$ nodes, the steady state equations are written in a matrix form as

\[
\begin{bmatrix}
  -n\rho & n & 0 & 0 & \cdots & 0 \\
  n\rho & -n^2\rho - n & 2n & 2n & \cdots & 0 \\
  0 & \frac{3}{2}cn\rho & -n^2\rho - 2n & 0 & \cdots & 0 \\
  \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & 0 & \cdots & -3n\rho - n(n-1) & n \\
  0 & 0 & 0 & \cdots & 3n\rho & -n
\end{bmatrix}
\begin{bmatrix}
  p_{S_0} \\
  p_{S_1} \\
  p_{S_2} \\
  \vdots \\
  p_{S_{m-2}} \\
  p_{S_{m-1}}
\end{bmatrix} = 0,
\]  

(5)

where $c$ is a constant determined by Table I and $\rho = \lambda/\mu$, which is called a traffic intensity. It is easy to see that (5) is linearly dependent since the sum of each column elements is 0. On the other hand, since the total sum of probabilities is 1, we have

\[
\sum_{i=0}^{m-1} |S_i|p_{S_i} = p_{S_0} + np_{S_1} + \cdots + np_{S_{m-2}} + p_{S_{m-1}} = 1,
\]  

(6)

where $|S_i|$ is the number of elements (states) in $S_i$. We want to obtain
$p_{S_0}, \ldots, p_{S_{m-1}}$ by solving (5) and (6), which is the main subject in this paper. We tried to solve the equations (5) and (6) by using the function "Linear-Solve" which is implemented in the software "Mathmatica 2.1." In Table III, the computing time for Sun4/2GS workstation is listed.

### Table III

<table>
<thead>
<tr>
<th>number of service nodes $n$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>computing time (second)</td>
<td>0.133</td>
<td>0.35</td>
<td>6.95</td>
<td>1579.35</td>
<td>Out of memory</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

By this method we can not solve the linear equations (5) and (6) even when $n = 6$, because a large amount of computing time and memory are necessary. Thus in the following, we derive a faster algorithm to solve the linear equation (5) and (6).

### 3. A FAST ALGORITHM

In (5), we can eliminate $p_{S_0}$ and $p_{S_{m-1}}$ by substituting $\rho p_{S_0} = p_{S_1}$ and $3\rho p_{S_{m-2}} = p_{S_{m-1}}$. For the matrix in the left hand side of (5), let $A$ be the $(m-2) \times (m-2)$ matrix obtained by adding the first row of the matrix to the second row, by adding the last row to the $(m-1)$-th row and by deleting the first and the last rows and columns. Then $A$ is written as

$$A = \begin{bmatrix}
-n^2 \rho & 2n & 2n & \cdots & 0 \\
\frac{3}{2}c n \rho & -n^2 \rho - 2n & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & \cdots & -n(n-1)
\end{bmatrix}, \quad (7)
$$

By examining the matrix $A$ carefully, we can see that $A$ is divided into sub-matrices as follows since a transition $x \rightarrow y$ can occur only when $d(x, y) \leq 1$. 
where $A_k$ is an $m_{k+1} \times m_{k+1}$ diagonal matrix, $C_{n-k}$ is an $m_{k-1} \times m_{k-1}$ diagonal matrix, $B_k$ is an $m_{k+1} \times m_k$ matrix, $D_{n-k}$ is an $m_{k-1} \times m_k$ matrix and $m_k$ is the number of equivalence classes $S_i$ with $w(S_i) = k$. Note that $A_k$, $B_k$, $C_k$ and $D_k$ do not contain the variable $\rho$. Here, we can show the following theorem.

**Theorem.** For a cyclically connected queueing models with $n$ nodes, let $m$ be the number of equivalence class $S_i$'s. Then the limit probability distribution is represented by

$$p_{S_i} = \frac{\rho^{w(S_i)} A_{S_i}(\rho)}{B(\rho)}$$
for \( i = 1, 2, \ldots, m - 2 \), where \( A_{S_i}(\rho) \)'s are polynomials of \( \rho \) with degree \( m - n - 1 \) and \( B(\rho) \) is a polynomial of \( \rho \) with degree \( m - 1 \). Furthermore, we have

\[
p_{S_0} = p_0 = \frac{A_{S_1}(\rho)}{B(\rho)} \quad \text{and} \quad p_{S_{m-1}} = \frac{3\rho^n A_{S_{m-2}}(\rho)}{B(\rho)}.
\]

**Proof.** By (5), we have

\[
\begin{cases}
p_{S_0} = \frac{1}{\rho} p_{S_1}, \\
p_{S_{m-1}} = 3\rho p_{S_{m-2}}.
\end{cases}
\]

Thus by (9), we obtain

\[
\sum_{i=0}^{n} |S_i| p_{S_i} = (1/\rho + n)p_{S_1} + np_{S_2} + \cdots + (n + 3\rho)p_{S_{m-2}} = 1,
\]

hence

\[
(n\rho + 1)p_{S_1} + np_{S_2} + \cdots + (n\rho + 3\rho^2)p_{S_{m-2}} = \rho. \quad (10)
\]

By (5), (7) and (10), we have

\[
\begin{bmatrix}
A \\
n\rho + 1, n\rho, \cdots (n\rho + 3\rho^2)
\end{bmatrix}
\begin{bmatrix}
p_{S_1} \\
\vdots \\
p_{S_{m-1}}
\end{bmatrix} =
\begin{bmatrix}
0 \\
\vdots \\
0 \\
\rho
\end{bmatrix},
\]

where \( A \) is a matrix given in (8). As was stated in the previous section, the rows of \( A \) are not independent and \( \text{rank}(A) = m - 3 \). Let \( \bar{A} \) be the \((m - 2) \times (m - 2)\) matrix which is obtained by deleting the \((m - 2)\)-th row of the \((m - 1) \times (m - 2)\) matrix of the left hand side of (11). Then (11) can be written as

\[
\bar{A} \times \begin{bmatrix}
p_{S_1} \\
\vdots \\
p_{S_{m-1}}
\end{bmatrix} = \begin{bmatrix}
0 \\
\vdots \\
0 \\
\rho
\end{bmatrix}.
\]

Since each element of \( \bar{A} \) is a polynomial of \( \rho \), each element of \( \bar{A}^{-1} \) is represented by a rational formula of \( \rho \). Hence, we have

\[
p_{S_i} = \frac{\rho}{|A|} \tilde{a}_{i,m-2},
\]
where $\tilde{a}_{ij}$ is the $(i, j)$-th cofactor of $\tilde{A}$. Further, we have $\deg(|\tilde{A}|) = m - 3 + 2 = m - 1$ and $\deg(\tilde{a}_{i,m-2}) = m - 4 + 1 = m - 3$, since the $(m - 2, m - 2)$-th element of $\tilde{A}$ is a polynomial of $\rho$ of degree 2 and the other elements of $\tilde{A}$ are degree 1 or 0. Therefore, $p_{S_{i}}$ is written by

$$p_{S_{i}} = \frac{\rho f_{S_{i}}(\rho)}{B(\rho)},$$

(12)

where $f_{S_{i}}(\rho)$ and $B(\rho)$ are polynomials with $\deg(f_{S_{2}}(\rho)) \leq n - 3$ and $\deg(B(\rho)) \leq m - 1$. Let $Q_{k}$ be the $m_{k}$-dimensional vector of polynomials $f_{S_{i}}(\rho)$ with $w(S_{i}) = k$. Then by examining (8) carefully, we have

$$\begin{align*}
-\rho C_{n-2}Q_{1} + D_{n-2}Q_{2} &= 0, \\
\rho B_{1}Q_{1} + (-C_{n-3}\rho - A_{1})Q_{2} + D_{n-3}Q_{3} &= 0, \\
\vdots \\
\rho B_{n-3}Q_{n-3} + (-C_{1}\rho - A_{n-3})Q_{n-2} + D_{1}Q_{n-1} &= 0, \\
\rho B_{n-2}Q_{n-2} + (-A_{n-2})Q_{n-1} &= 0.
\end{align*}$$

(13)

Let $(H_{k})$ be the following claim:

$(H_{k})$ The degree of the lowest term of polynomials of $\rho$ in $Q_{j}$ is at least $k$ for any $j \geq k + 1$.

It is obvious that $(H_{0})$ holds. Assume that $(H_{k})$ holds for a given $k \leq n - 2$. Then by the last equation, the degree of the lowest term of polynomials of $\rho$ in $Q_{n-1}$ is at least $k + 1$, since that of $\rho B_{n-2}Q_{n-2}$ is at least $k + 1$ and $A_{n-2}$ is a diagonal matrix whose diagonal elements are nonzero. By the second equation from the last of (13), the degree of the lowest term of $\rho$ in $Q_{n-2}$ is at least $k + 1$. Continuing this process, we can show that $(H_{k+1})$ holds. Hence, $(H_{k})$ holds for any $k = 0, 1, \ldots, n - 2$, which implies that $f_{S_{i}}(\rho)$ is represented by $f_{S_{i}}(\rho) = \rho^{w(S_{i})-1}A_{S_{i}}(\rho)$, where $\deg(A_{S_{i}}(\rho)) \leq m - w(S_{i}) - 2$.

Now let $Q'_{k} = Q_{k}/\rho^{k-1}$, then $Q'_{k}$ is a vector of polynomials with degree
$m - k - 2$. Then (13) is rewritten as follows:

\[
\begin{cases}
-C_{n-2}Q_1' + D_{n-2}Q_2' = 0, \\
(B_1Q_1' - A_1Q_2') - \rho(C_{n-3}Q_2' - D_{n-3}Q_3') = 0, \\
\vdots \\
(B_{k-1}Q_{k-1}' - A_{k-1}Q_k') - \rho(C_{n-k-1}Q_k' - D_{n-k-1}Q_{k+1}') = 0, \\
\vdots \\
(B_{n-3}Q_{n-3}' - A_{n-3}Q_{n-2}') - \rho(C_1Q_{n-2}' - D_1Q_{n-1}') = 0, \\
B_{n-2}Q_{n-2}' + (-A_{n-2})Q_{n-1}' = 0.
\end{cases}
\]

(14)

Similarly, consider the following claim:

\((H'_{k})\) \(\deg(Q_j') \leq m - k - 2\) holds for any \(j \leq k\).

It is obvious that \((H'_{1})\) holds. Assume that \((H'_{k})\) holds. In equation (14), \(\deg(B_{k-1}Q_{k-1}' - A_{k-1}Q_{k}') \leq m - k - 3\) holds by \((H'_{k})\) and we have \(\deg(Q'_{k+1}) \leq m - k - 3\) by the definition of \(Q'_{k+1}\). Hence \(\deg(Q_k') \leq m - k - 2\) holds, since \(C_k's\) are diagonal matrix with nonzero diagonal elements. Similarly, continuing this process, we can show that \((H'_{k+1})\) holds. Hence by induction, it is shown that \((H'_{n-1})\) holds, which proves the theorem.

By this theorem, \(A_{S_i}(\rho)\) can be written as follows:

\[A_{S_i}(\rho) = a_{i0} + a_{i1}\rho + a_{i2}\rho^2 + \cdots + a_{im-n-2}\rho^{n-m-2} + a_{im-n-1}\rho^{m-n-1}.\]

Now, let \(q_{ks}\) be the vector of the coefficients of \(\rho^s\) in the polynomials in \(Q_k'\) for \(s = 0, 1, \cdots, m - n - 1\). Then by (14), we have

\[q_{k+1,0} = A_k^{-1}B_kq_{k,0}\]

(15)

for \(k = 0, 1, 2, \cdots, n - 2\), where \(q_{10} = (a_{10})\). Further,

\[q_{k+1,s} = A_k^{-1}\left[ B_kq_{ks} - C_{n-2-k}q_{k+1,s-1} + D_{n-2-k}q_{k+2,s-1} \right]\]

(16)

holds for \(k = 1, 2, \cdots, n - 2\) and for \(s = 1, 2, \cdots, m - n - 2\), where we assume \(C_0 = o, D_0 = o\) for convenience. Thus by (15) and (16), each coefficient \(a_{ij}\) can be obtained recursively as a linear combination of \(a_{10}, a_{11}, \cdots, a_{1m-n-2}\) as follows:

\[a_{ij} = \sum_{s=0}^{m-n-2} b_{ij} a_{1s}\]

(17)
for $i = 2, \cdots, m - 1; \ j = 0, \cdots, m - n - 1$. (Note that we do not need to solve linear equations for computing $q_{ks}$ by (15) and (16), since the matrices $A_k$'s are diagonal.) Substituting (17) to (14), we obtain a linear equation with respect to $a_{10}, a_{11}, \cdots, a_{1m-n-1}$. By solving this equation, we can determine the coefficients $a_{11}, \cdots, a_{1m-n-1}$ as a scalar multiple of $a_{10}$. Thus the coefficients of $A_{S_i}(\rho)$ are determined as a scalar multiple of $a_{10}$. Finally, by substituting $A_{S_i}(\rho)$ to

$$B(\rho) = A_{S_1}(\rho) + 3\rho^n A_{S_{m-2}}(\rho) + \sum_{i=1}^{m-2} |S_i|\rho^{w(S_i)} A_{S_i}(\rho),$$

(18)

$B(\rho)$ is computed. Thus, $p_{S_i}$ can be obtained by (12). Note that the value of $a_{10}$ does not matter to determine $p_{S_i}$.

**Example 4.** We determine the exact steady state distribution of cyclically connected queueing system with four service nodes. In this case, the steady state equation (4) is written by

$$\begin{bmatrix}
-16\rho & 8 & 4 & 0 \\
12\rho & -16\rho & -8 & 0 \\
4\rho & 0 & -8\rho & -4 \\
0 & 16\rho & 8\rho & -12
\end{bmatrix}
\begin{bmatrix}
\rho A_{S_1}(\rho) \\
\rho^2 A_{S_2}(\rho) \\
\rho^2 A_{S_3}(\rho) \\
\rho^3 A_{S_4}(\rho)
\end{bmatrix}
= 0,$$

where $\deg(A_{S_i}(\rho)) = 6 - 4 - 1 = 1$, that is $A_{S_i}(\rho) = a_{i0} + xa_{i1}$. In this example, we have

$$A_1 = \begin{pmatrix} 8 & 0 \\ 0 & 4 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 12 \\ 4 \end{pmatrix}, \quad B_1 = \begin{pmatrix} 12 \\ 4 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 16 & 8 \end{pmatrix},$$

$$C_1 = \begin{pmatrix} 16 & 0 \\ 0 & 8 \end{pmatrix}, \quad C_2 = \begin{pmatrix} 16 \\ 4 \end{pmatrix}, \quad D_1 = \begin{pmatrix} 8 \\ 4 \end{pmatrix}, \quad D_2 = \begin{pmatrix} 8 & 4 \end{pmatrix},$$

and

$$q_{1s} = \begin{pmatrix} a_{1s} \\ \end{pmatrix}, q_{2s} = \begin{pmatrix} a_{2s} \\ a_{3s} \end{pmatrix}, q_{3s} = \begin{pmatrix} a_{4s} \end{pmatrix}$$

for $s = 0, 1$. Then by the algorithm,

$$\begin{pmatrix} a_{20} \\ a_{30} \end{pmatrix} = q_{30} = A_1^{-1}B_1q_{10} = \begin{pmatrix} 3/2 \\ 1 \end{pmatrix} a_{10},$$
\[ a_{40} = q_{30} = A_{2}^{-1}B_{2}q_{20} = \frac{8}{3}a_{10}, \]
\[
\begin{pmatrix}
  a_{21} \\
  a_{31}
\end{pmatrix} = q_{21} = A_{1}^{-1}(B_{1}q_{11} - C_{1}q_{20} + D_{1}q_{30}) = \begin{pmatrix}
  3/2 \\
  1
\end{pmatrix} a_{11} + \begin{pmatrix}
  -1/3 \\
  2/3
\end{pmatrix} a_{10}
\]
and
\[ a_{41} = q_{31} = A_{2}^{-1}B_{2}q_{21} = \frac{8}{3}a_{11} \]
can be obtained. Substituting these to (14), we obtain \( a_{11} = 2a_{10} \). Thus
\[ a_{21} = \frac{3}{2}a_{10}, \quad a_{31} = \frac{8}{3}a_{10}, \quad a_{41} = \frac{16}{3}a_{10} \]
hold. Hence, we have \( A_{S_{1}}(\rho) = (1 + 2\rho)a_{10}, A_{S_{2}}(\rho) = (\frac{3}{2} + \frac{8}{3}\rho)a_{10}, A_{S_{3}}(\rho) = (1 + \frac{8}{3}\rho)a_{10} \) and \( A_{S_{4}}(\rho) = (\frac{8}{3} + \frac{16}{3}\rho)a_{10} \). Therefore,
\[
B(\rho) = A_{S_{1}} + 4\rho A_{S_{1}} + 4\rho^{2}A_{S_{2}} + 2\rho^{2}A_{S_{3}} + 4\rho^{3}A_{S_{4}} + 3\rho^{4}A_{S_{5}}
\]
\[ = (1 + 2\rho)(1 + 4\rho + 6\rho^{2} + \frac{32}{3}\rho^{3} + 8\rho^{4})a_{10} \]
and the steady state distribution can be written as follows:
\[
p_{S_{0}} = \frac{3}{3 + 12\rho + 24\rho^{2} + 32\rho^{3} + 24\rho^{4}},
\]
\[
p_{S_{1}} = \frac{3\rho}{3 + 12\rho + 24\rho^{2} + 32\rho^{3} + 24\rho^{4}},
\]
\[
p_{S_{2}} = \frac{\rho^{2}(9 + 16\rho)}{2(1 + 2\rho)(3 + 12\rho + 24\rho^{2} + 32\rho^{3} + 24\rho^{4})},
\]
\[
p_{S_{3}} = \frac{\rho^{2}(3 + 8\rho)}{2(1 + 2\rho)(3 + 12\rho + 24\rho^{2} + 32\rho^{3} + 24\rho^{4})},
\]
\[
p_{S_{4}} = \frac{8\rho^{3}}{3 + 12\rho + 24\rho^{2} + 32\rho^{3} + 24\rho^{4}},
\]
\[
p_{S_{5}} = \frac{24\rho^{4}}{3 + 12\rho + 24\rho^{2} + 32\rho^{3} + 24\rho^{4}}.
\]
Our algorithm to obtain the exact steady state distribution is summarized as follows:

**Step 1.** Constitute the equivalence classes.

**Step 2.** Constitute the matrices $A_k$, $B_k$, $C_k$ and $D_k$ from the steady state equation.

**Step 3.** Using the recurrence equations (15) and (16), represent $a_{ij}$ by $a_{10}$, $a_{11}$, $\cdots$, $a_{1 m-n-1}$.

**Step 4.** Substitute $a_{ij}$ to (14) and make a linear equation with respect to $a_{10}$, $a_{11}$, $\cdots$, $a_{1 m-n-1}$.

**Step 5.** Represent $a_{11}$, $\cdots$, $a_{1 m-n-1}$ as a scalar multiple of $a_{10}$ by solving the equation.

**Step 6.** Compute $B(\rho)$ by (18) and obtain $p_{S_i}$.

In Table IV, the computing times for our algorithm and those for LinearSolve in Mathematica 2.1 are listed. Our algorithm was also programmed by Mathematica 2.1. It is observed that our algorithm is faster than LinearSolve.

<table>
<thead>
<tr>
<th>number of servers</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>computing time</td>
<td>5.8</td>
<td>12.27</td>
<td>31.2</td>
<td>104.0</td>
<td>712.52</td>
<td>5203.4</td>
<td>58946.2</td>
</tr>
<tr>
<td>for our algorithm</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>computing time</td>
<td>6.9</td>
<td>1579.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>for LinearSolve</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table IV**

**Comparison of computing time for the both algorithms**

4. **THE PROBABILITY OF LOSS**

Here, we consider the probability of a loss of cyclically connected queueing system. A customer turns away without receiving service if all of the three service nodes (the arrival node and the two adjacent service nodes) are busy. Let $l_i$ be the number of 3-consecutive 1's in a state vector $x \in S_i$. Then the probability of loss for the cyclically connected queueing system is given by

$$p_{\text{loss}} = \frac{1}{n} \sum_{i=1}^{m} |S_i| l_i p_{S_i}.$$
Example 5. We consider the probability of loss for \( n = 4 \). In this case, note that there are 6 equivalence classes \( S_0, \ldots, S_5 \), which are listed in (2). And we have \( l_0 = l_1 = l_2 = l_3 = 0, l_4 = 1 \) and \( l_5 = 4 \). Therefore the probability of loss on this system is

\[
p_{\text{loss}} = \frac{1}{4} (4 \cdot 1 \cdot p_{S_4} + 1 \cdot 4 \cdot p_{S_5}) = p_{S_4} + p_{S_5}.
\]

In Table V, the probability of loss for cyclically connected queueing models (abriviated as c.c. queue) are listed together with those for M/M/c loss in the case when the traffic intensity is \( \rho = \lambda/\mu = 1 \). Note that M/M/c loss system can be considered as a network queueing system in which every two nodes are connected by an edge.

<table>
<thead>
<tr>
<th>number of servers</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>c.c. queue</td>
<td>0.336842</td>
<td>0.333994</td>
<td>0.333346</td>
<td>0.333177</td>
<td>0.333135</td>
</tr>
<tr>
<td>M/M/c loss</td>
<td>0.31068</td>
<td>0.284868</td>
<td>0.264922</td>
<td>0.24887</td>
<td>0.23557</td>
</tr>
<tr>
<td></td>
<td>0.333125</td>
<td>0.333122</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.2243</td>
<td>0.214582</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As is expected, the probability of loss is decreasing. Furthermore, we may observe that the probability of loss is converging around 0.3331.

5. A COMPARISON OF PROBABILITY OF LOSS FOR QUEUING NETWORK SYSTEMS WITH 4 NODES

Finally, as an example, we consider 11 models \( G_1, \ldots, G_{11} \) in Figure 1 with four service nodes. We computed the probability of loss by Mathmatica 2.1. In the case when traffic intensity is \( \rho = \lambda/\mu = 1 \). The result is listed in Table VI.
It can be observed that in proportion as the number of edges increase, the probability of loss is decreasing in general. $G_{11}$ is nothing but a M/M/c loss system, and this probability of loss is the smallest among the all models in Figure 1. If we restrict ourselves to the models with at most 4 edges, $G_9$ has the smallest probability of loss, where $G_9$ is a cyclically connected queueing system.

Now we would like to conclude this paper by the following conjecture.

**Conjecture.** A cyclically connected queueing system with $n$ nodes has the smallest probability of loss among the network queueing systems with $n$ nodes and with at most $n$ edges.

### REFERENCES


