Two-stations queueing networks with moving servers, blocking, and customer loss

Winfried K. Grassmann
grassman@cs.usask.ca

Javad Tavakoli

Follow this and additional works at: http://repository.uwyo.edu/ela

Recommended Citation
DOI: https://doi.org/10.13001/1081-3810.1151

This Article is brought to you for free and open access by Wyoming Scholars Repository. It has been accepted for inclusion in Electronic Journal of Linear Algebra by an authorized editor of Wyoming Scholars Repository. For more information, please contact scholcom@uwyo.edu.
TWO-STATION QUEUEING NETWORKS WITH MOVING SERVERS, BLOCKING, AND CUSTOMER LOSS

WINFRIED K. GRASSMANN† AND JAVAD TAVAKOLI‡

Abstract. This paper considers a rather general model involving two exponential servers, each having its own line. The first line is unlimited, whereas the second line can only accommodate a finite number of customers. Arrivals are Poisson, and they can join either line, and once finished, they can either leave the system, or they can join the other line. Since the space for the second line is limited, some rules are needed to decide what happens if line 2 is full. Two possibilities are considered here: either the customer leaves prematurely, or he blocks the first server. The model also has moving servers, that is, the server at either station, while idle, can move to help the server of the other station. This model will be solved by an eigenvalue method. These eigenvalue methods may also prove valuable in other contexts.

Key words. Tandem queues, Generalized eigenvalues, Markov chains, Blocking.

AMS subject classifications. 60K25, 90B22, 15A22, 65F15.

1. Introduction. In this paper, we show how to find the equilibrium probabilities of a two station Markovian queueing network with rather general routing and with movable servers. The waiting space of the second station is limited to some finite number $N$. To solve this problem, we use eigenvalues. These eigenvalues, in turn, are obtained by using homogeneous difference equations.

In a way, movable servers are very natural. Indeed, if servers are human, they most likely would co-operate if one of the two servers is idle. Indeed, there are industrial applications where servers do move [1], [4].

The assumption that the second queue is finite introduces an asymmetry which some people may consider unnatural. However, one of the objectives of this paper is to investigate the effect of blocking, and there is no blocking in infinite queues. Moreover, one can always truncate one of the two queues to finite length. If done properly, this truncation will not normally lead to significant errors.

We published two related papers earlier, one dealing with blocking [9], the other one with a movable server and loss instead of blocking [10]. In both papers, the routing was limited to going from one line to the next one. When discussing our approach with others, the question came up as to how much the methods used in these papers can be generalized, and what new methods are needed to accommodate such generalizations. While doing this, we discovered a number of simplifications which make the problems easier to analyze, and which at the same time help to solve not only the different routings allowed in this paper, but also many other potential generalizations not covered in this paper, including multiple servers.
One problem that has to be addressed here is the issue of the existence of equilibrium probabilities. This issue did not arise in our earlier papers, because in the case of blocking, these results were available [12], and in the case of loss, the problem is relatively simple.

The set of eigenvalues is known as the spectrum, which explains the term spectral analysis for an analysis involving eigenvalues. Spectral analysis has been used by a number of authors to find equilibrium solutions of queueing problems (see [3], [6], [7], [11], [15], [16], [21]).

The spectral methods have to be distinguished from the matrix analytic methods pioneered by Neuts [14], [18], [19], and the question is how these methods compare with ours. There are two parts to this question: what is the computational complexity of the respective methods, and what is their accuracy. To compare the computational complexity, we restrict ourselves to the iterative part of the algorithm, because this part tends to require most of the time. Generally speaking, the approach described here requires $O(N)$ operations to iteratively find all eigenvalues, and matrix analytic methods require matrix multiplications, which leads to $O(N^3)$ operations per iteration to find the matrix in question. Hence, from the point of view of computational complexity, the difference equations approach is better. This, of course, is due to the fact that it can exploit special features which the more general methods cannot exploit.

Regarding the precision of the results, we note all that can be guaranteed for matrix analytic methods is that they satisfy the equilibrium equation with a given precision. Satisfying the equilibrium equations also seems reasonable from a logical point of view, and we therefore adopt it here. As we showed in [7], the eigenvalue approach can guarantee that the equilibrium equations are satisfied within a certain precision. This does not necessarily mean that the eigenvalues are the correct ones: it may very well happen that even when the eigenvalues change within a certain range, the solution is still satisfied at the required precision. In fact, this is exactly what happens if two eigenvalues are close together as the reader may verify. In any case, for our model, we can prove that there are never multiple eigenvalues.

Figure 1.1 provides a picture of the model: there are two stations, each with one exponential server. Each station has its own waiting line, but the line of station 2 is

![Diagram of the system](image-url)
limited to $N - 1$, including the space for service. Arrivals to the network are Poisson, and the arrival rates are $\lambda_i$ for station $i$, $i = 1, 2$. After the service with station 1 is completed, the customer leaves the network with probability $p$, and he moves to station 2 with probability $1 - p$. Similarly, a customer finishing at station 2 leaves the network with probability $u$, and proceeds to station 1 with probability $1 - u$. The service rates are $\mu_i$ for station $i$, $i = 1, 2$. However, if server 1 is idle or blocked, she helps server 2, increasing her service rate to $\mu_4$. Similarly, if server 2 is idle, she helps server 1, bringing her service rate to $\mu_3$.

To properly describe the model, we need to state what happens if all $N - 1$ spaces of line 2 are taken. It seems natural to assume that in this case, no arrivals from the outside the network can join line 2. However, some customers finishing line 1 may not leave, but, since line 2 has no space left, they remain in station 1, thereby blocking this station and preventing any service completions. Since this is undesirable, we assume that when the line reaches $N - 1$, the probability of a customer leaving the network changes from $p$ to $t$, where $t > p$. It is convenient to consider the customer blocking station 1 as belonging to line 2, which means that line 2 can range from 0 to $N$.

The outline of the paper is as follows. We first derive the transition matrix of our model, which is a block-structured matrix. We then formulate the corresponding block-structured equilibrium equations. These equations can be classified as either boundary equations or interior equations. We deal with the interior equations first. They essentially determine whether or not the system is recurrent, a topic addressed in Section 3. We then solve the interior equations in Section 4, and the boundary equations in Section 5. Section 6 provides some numerical considerations. A summary of our procedure is given in Section 7.

2. Mathematical Formulation. The queueing process has two random variables, namely $X_1$ ($X_1 \geq 0$) and $X_2$ ($0 \leq X_2 \leq N$), which are the lengths of lines 1 and 2, respectively. $X_1$ will be called the level, and $X_2$ the phase. If $\pi_{ij} = P\{X_1 = i, X_2 = j\}$ is the steady-state probability, the main objective of this paper is to find $\pi_{ij}$. We assume that the system is recurrent. The condition for the system to be recurrent will be discussed later.

The transition matrix is partitioned according to levels. Consider first the case where $X_1 > 0$. In this case, all rates of the transition matrix increasing the level by 1 are included in the matrix $Q_1$, and all rates decreasing the level by 1 are included in $Q_{-1}$. The matrix $Q_0$ contains the rates that only affect the phases but not the level, and it also includes the diagonal elements, which are determined such that the sums across the rows of the entire transition matrix are equal to 0. If $X_1 = 0$, then the rates leaving the level unchanged are not given by $Q_0$, but by a matrix we call $Q_{00}$. Similarly, the matrix $Q_{01}$ includes all rates that increase the level from 0 to 1, and
this matrix is different from $Q_1$. Hence, the transition matrix becomes

\[
Q = \begin{bmatrix}
Q_{00} & Q_{01} & 0 & 0 & \ldots \\
0 & Q_{0} & Q_{1} & 0 & \ldots \\
0 & 0 & Q_{-1} & Q_{0} & Q_{1} & \ldots \\
0 & 0 & 0 & Q_{-1} & Q_{0} & \ldots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\
\end{bmatrix}.
\]

Markov Processes with transition matrices of the form given by (2.1) are called Quasi-Birth-Death (QBD) processes.

We now discuss how to obtain $Q_1$, $Q_0$, $Q_{-1}$, $Q_{00}$ and $Q_{01}$. $Q_1$ contains all rates which increase $X_1$, provided $X_1 > 0$. These rates are given by two events, namely arrivals from outside and arrivals from the second station. Arrivals from outside occur at a rate of $\lambda_1$, and they leave the phase unchanged. As a consequence, the diagonal of $Q_1$ is $\lambda_1$. Arrivals from station 2 have a rate of $(1-u)\mu_2$ unless server 1 is blocked and helps the second server, causing the rate to increase to $(1-u)\mu_4$. These events decrease $X_2$, which means the subdiagonal of $Q_1$ is $(1-u)\mu_2$, except when $X_2 = N$. It follows that

\[
Q_1 = \begin{bmatrix}
\lambda_1 & 0 & 0 & \ldots & 0 \\
(1-u)\mu_2 & \lambda_1 & 0 & \ldots & 0 \\
0 & (1-u)\mu_2 & \lambda_1 & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & (1-u)\mu_4 & \lambda_1 \\
\end{bmatrix}.
\]

$Q_{01}$ can be obtained from $Q_1$ by replacing all instances of $\mu_2$ by $\mu_4$, indicating that the first server helps the second one.

We look at $Q_{-1}$ next, which is

\[
Q_{-1} = \begin{bmatrix}
p\mu_3 & q\mu_3 & 0 & \ldots & 0 \\
0 & p\mu_4 & q\mu_1 & \ldots & 0 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots \\
0 & 0 & 0 & \ldots & 0 \\
\end{bmatrix},
\]

where $q = 1 - p$. To obtain $Q_{-1}$, note that $X_1$ decreases whenever a service at station 1 is complete, and this happens at a rate of $\mu_3$ if the server of station 2 is idle ($X_2 = 0$), at a rate $\mu_1$ if $0 < X_2 < N$, and at a rate of 0 if server 1 is blocked ($X_2 = N$). After completing service with server 1, customers leave the system with probability $p$ if $X_2 < N - 1$, and the phase does not change. With probability $1 - p$, the customer joins the line of station 2, and the phase increases by 1. In the first case, we get the diagonal elements $p\mu_3$ for $X_2 = 0$, and $p\mu_1$ for $0 < X_2 < N - 1$, and in the second case the superdiagonal elements $q\mu_3$ and $q\mu_1$. For $X_2 = N - 1$, the probability of a customer leaving increases from $p$ to $t$, and the rates have to be
adjusted accordingly. A customer completing service at station 1 who does not leave while all spaces of station 2 are taken blocks station 1, meaning that $X_2 = N$.

$Q_0$ has the following form as the reader may verify

$$Q_0 = \begin{bmatrix}
-(\lambda_1 + \lambda_2 + \mu_3) & \lambda_2 & 0 & \ldots & 0 \\
\mu_2 & -(\lambda_1 + \lambda_2 + \mu_1 + \mu_2) & \lambda_2 & \ddots & \vdots \\
0 & \mu_2 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & -(\lambda_1 + \mu_1 + \mu_2) & 0 \\
0 & \ldots & \ldots & \mu_4 & -(\lambda_1 + \mu_4)
\end{bmatrix}.$$  

Note that when there are $N-1$ customers in line 2, no customers from the outside are admitted. Hence, the rate of going from phase $N-1$ to phase $N$ without changing levels is not $\lambda_2$, but 0.

The matrix $Q_{00}$ reflects the fact that, when the first server is idle, it helps the second server. Consequently, the rate of the second server increases to $\mu_4$. To convert $Q_0$ into $Q_{00}$, one must therefore replace all instances of $\mu_2$ by $\mu_4$. In addition, the diagonal changes. Hence

$$Q_{00} = \begin{bmatrix}
-(\lambda_1 + \lambda_2) & \lambda_2 & 0 & \ldots & 0 \\
\mu_4 & -(\lambda_1 + \mu_4 + \lambda_2) & \lambda_2 & \ddots & \vdots \\
0 & \mu_4 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & -(\lambda_1 + \mu_4) & 0 \\
\vdots & \ddots & \ddots & \mu_4 & -(\lambda_1 + \mu_4)
\end{bmatrix}.$$  

We now have identified all blocks of the matrix $Q$ given in equation (2.1). The row vector of the equilibrium probabilities $\pi$ can now be found in the normal way by solving $\pi Q = 0$, subject to the condition that the sum of all equilibrium probabilities equals 1. If we partition $\pi$ according to levels, with $\pi_i$ corresponding to level $i$, $0 = \pi Q$ expands to

\begin{align*}
(2.2) & \quad 0 = \pi_0 Q_{00} + \pi_1 Q_{-1} \\
(2.3) & \quad 0 = \pi_0 Q_{01} + \pi_1 Q_0 + \pi_2 Q_{-1} \\
(2.4) & \quad 0 = \pi_{n-1} Q_1 + \pi_n Q_0 + \pi_{n+1} Q_{-1}, \quad n > 1.
\end{align*}

Equation (2.4) is a difference equation with matrices as coefficients, and their solutions are similar to the standard difference equations. Specifically, one has (see Bertsimas, [3], Mitrani and Chakka [15] and Morse [16]):

\begin{align*}
(2.5) & \quad \pi_n = g x^{n-1}, n \geq 1,
\end{align*}

where $g = [g_0, g_1, \ldots, g_N]$ must be different from 0. (We use $g x^{n-1}$ rather than $g x^n$ because $\pi_1 = g$ is more conducive for our purposes than $\pi_0 = g$). Substituting (2.5) into (2.4) yields

\begin{align*}
(2.6) & \quad 0 = g x^{n-2} Q_1 + g x^{n-1} Q_0 + g x^n Q_{-1}.
\end{align*}
If we define

\[ Q(x) = Q_1 + Q_0 x + Q_{-1} x^2, \]

then (2.6) implies \( 0 = gQ(x) \). The problem is to find the scalars \( x \) and the corresponding vectors \( g \) which satisfy \( gQ(x) = 0 \). In general, this problem is known as the \textit{generalized eigenvalue problem} (see [5] for details). The matrix \( Q(x) \), or as it is known in literature, \( Q(\lambda) \), is often called a \( \lambda \)-matrix. Generally, any \( x \) satisfying \( gQ(x) = 0 \) is called a generalized eigenvalue, and the corresponding vector \( g \) is a generalized eigenvector. Clearly, a vector \( g \neq 0 \) exists if and only if \( \det Q(x) = 0 \). If this equation has multiple roots, then problems arise which are similar to the ones encountered in standard difference equations with multiple zeros of the characteristic equation. For details, see [5] and [8]. Fortunately, we will show later that if \( \lambda_1 > 0 \), all eigenvalues are distinct, and that they are all positive. Hence, we will assume \( \lambda_1 > 0 \), and that all eigenvalues are distinct. If the process is recurrent, the eigenvalues are also inside the unit circle. To see this, note that according to Neuts [18, Theorem 1.2.1], there is a matrix \( R \) such that

\[ \pi_n = \pi_1 R^{n-1}, \quad n \geq 1. \]

Here, \( R \) is a positive matrix, and its spectral radius is less than 1, which means that all \( N+1 \) eigenvalues of \( R \) are inside the unit circle. According to Naoumov [17], there is a stochastic matrix \( \Gamma \), and an invertible matrix \( Y \) such that

\[ Q(x) = (R - Ix)Y(\Gamma x - I). \]

It follows that all eigenvalues of \( R \) must also be eigenvalues of \( Q(x) \). Therefore, if the process is recurrent, \( Q(x) \) has \( N+1 \) eigenvalues inside the unit circle. This important because we will need \( N+1 \) solutions to satisfy (2.2) and (2.3) as will be shown later, and solutions with \( |x| \geq 1 \) cannot be used because \( \pi_n \) would not converge. We should also note that according to Perron-Frobenius (see e.g. [2]), the largest eigenvalue of \( R \) is simple and positive, and so is the corresponding eigenvector.

### 3. Conditions for Recurrence

Before we investigate how to find the equilibrium probabilities, we have to give conditions for their existence, that is, we have to show that the process is positive recurrent. We do this by considering the drift of the process [14]. The drift is defined as the expected increase, or, if negative, the expected decrease of the process, given the boundaries are ignored. If the drift is negative, that is, the process drifts toward level 0, then the process is positive recurrent. If the drift is 0, it is null-recurrent, and if the drift is away from zero, the process is non-recurrent. To find the drift, we first have to find a vector \( \tilde{\pi} \), which indicates the probabilities of being in the different phases given there are no boundaries. One has

\[ 0 = \tilde{\pi}(Q_{-1} + Q_0 + Q_1). \]

Of course, \( \tilde{\pi} \) is the eigenvector corresponding to the eigenvalue \( x = 1 \). The drift is now \( \tilde{\pi}(Q_1 - Q_{-1})e \), where \( e \) is a vector of appropriate dimension with all its elements equal to 1.
equal to 1. The process is positive recurrent if and only if this expression is negative. This will now be worked out for our model.

We notice that $Q_{-1} + Q_0 + Q_1$ is the incremental generator of a birth-death process, and we find

$$\bar{\pi}_{i+1} = \bar{\pi}_i \beta_i / \delta_{i+1}$$

where the $\beta_i$ are the birth-rates, that is, the rates on the superdiagonal, and $\delta_i$ are the death rates, that is, the rates on the subdiagonal. Here

$$\beta_0 = q \mu_3 + \lambda_2$$
$$\beta_i = q \mu_1 + \lambda_2, \ i = 1, 2, \ldots, N - 2$$
$$\beta_{N-1} = (1 - t) \mu_1$$
$$\delta_i = \mu_2, \ i = 1, 2, \ldots, N - 1$$
$$\delta_N = \mu_4.$$ 

For $i = 1, 2, \ldots, N - 2$, $\beta_i$ is independent of $i$, and we can write

$$\gamma = \frac{\beta_i}{\delta_{i+1}} = \frac{q \mu_1 + \lambda_2}{\mu_2}, \quad 3.1$$

The sign of the drift is obviously independent of any positive factor of $\bar{\pi}$. Hence, if $\bar{\pi}_i$ is the $i$th element of $\bar{\pi}$, we can set $\bar{\pi}_1 = 1$, and we obtain

$$\bar{\pi}_0 = \frac{\mu_2}{\lambda_2 + q \mu_3}, \quad 3.2$$
$$\bar{\pi}_i = \gamma^{i-1}, \ i = 1, 2, \ldots, N - 1$$
$$\bar{\pi}_N = \frac{(1 - t) \mu_1}{\mu_4} \gamma^{N-2}. \quad 3.3$$

As one can easily verify

$$(Q_1 - Q_{-1}) e = [\lambda_1 - \mu_3, (1 - u) \mu_2 + \lambda_1 - \mu_1, \ldots, (1 - u) \mu_2 + \lambda_1 - \mu_1, (1 - u) \mu_4 + \lambda_1]^T.$$

The values of this column vector obviously give the drift for the different states. Multiplying these values with $\bar{\pi}$ as given in equations (3.2) to (3.4) yields

$$\frac{\mu_2 (\lambda_1 - \mu_3)}{\lambda_2 + q \mu_3} + \frac{(1 - u) \mu_2 + \lambda_1 - \mu_1}{1 - \gamma} + \frac{(1 - u) \mu_4 + \lambda_1}{1 - (1 - t) \mu_1} \gamma^{N-2}, \quad \gamma \neq 1$$

$$\frac{\mu_2 (\lambda_1 - \mu_3)}{\lambda_2 + q \mu_3} + (N - 1)(1 - u) \mu_2 + \lambda_1 - \mu_1 + \frac{(1 - u) \mu_4 + \lambda_1}{1 - (1 - t) \mu_1}, \quad \gamma = 1$$

This expression must be negative in order for the process to be recurrent.

It is difficult to make statements about this complicated expression, and we therefore consider some special cases. In particular, we consider the case where everyone finishing server 1 joins line 2, and no one else joins line 2. Also, no one returns to
server 1 once finished with server 2. Mathematically, this means \( \lambda_2 = 0, p = t = 0,\)
\( q = 1 \) and \( u = 1. \) Equation (3.5) then implies:

\[
\lambda_1 < \begin{cases} \frac{1-\gamma^n}{(1-\gamma)(\frac{1}{\mu_3} + \frac{1}{\gamma}\gamma^{n-1}) + \frac{1}{\mu_2}(1-\gamma^{n-1})} & \mu_1 \neq \mu_2 \\ \frac{1}{\mu_2} & \mu_1 = \mu_2 = \mu. \end{cases}
\]

If \( \mu_4 = \mu_1 \) and \( \mu_5 = \mu_2, \) that is, if servers do not move, then this result can easily be converted into the one given by [12].

**Remark 1:** If the second server has an insufficient capacity to handle the average flow, it is still possible that the first server can help sufficiently to make the problem recurrent. In fact, in this case, the mathematical analysis does not change if instead of blocking, the first server helps the second one as soon as the length of the second server reaches \( N. \)

**Remark 2:** If server 1 can not help while blocked, then none of the matrices \( Q_{-1}, Q_0, \) or \( Q_1 \) contain \( \mu_4. \) If this is the case, recurrence, or the lack of it, therefore cannot be affected by \( \mu_4. \) This seems somewhat counterintuitive. The explanation of this is that no matter how high \( N \) is, eventually, server 1 is blocked, and at this point in time, the service rate of server 2 determines the potential throughout. Of course, the higher \( \mu_4 \) and \( N, \) the longer it takes the process to become blocked.

4. **Difference Equations for the Eigenvectors.**

4.1. **Direct Solution of the Difference Equations.** The equation

\[ gQ(x) = g(Q_1 + xQ_0 + x^2Q_{-1}) = 0 \]

can be expanded as follows (see [9] and [10]):

\[
\begin{align*}
(4.1) & \quad 0 = -g_0((\lambda_1 + \lambda_2 + \mu_3)x - \lambda_1 - \mu_3 x^2) + g_1\mu_2(ux + 1 - u) \\
(4.2) & \quad 0 = g_0(q\mu_3 x^2 + \lambda_2 x) - g_1((\lambda_1 + \lambda_2 + \mu_1 + \mu_2)x - \lambda_1 - \mu_1 x^2) \\
& \quad + g_2\mu_2(ux + 1 - u) \\
(4.3) & \quad 0 = g_{i-1}(q\mu_1 x^2 + \lambda_2 x) - g_i((\lambda_1 + \lambda_2 + \mu_1 + \mu_2)x - \lambda_1 - \mu_1 x^2) \\
& \quad + g_{i+1}\mu_2(ux + 1 - u), \quad i = 2, \ldots, N - 2 \\
(4.4) & \quad 0 = g_{N-2}(q\mu_1 x^2 + \lambda_2 x) - g_{N-1}((\lambda_1 + \mu_1 + \mu_2)x - \lambda_1 - \mu_1 x^2) \\
& \quad + g_{N}\mu_4(ux + 1 - u) \\
(4.5) & \quad 0 = g_{N-1}(\mu_1(1 - t)x^2) - g_N((\lambda_1 + \mu_4)x - \lambda_1).
\]

These equations must be solved for \( x \) and for \( g = [g_0, g_1, \ldots, g_N], \) up to a factor. The idea is to fix \( x, \) and solve all equations except the last one, which has a residual. The aim is then to reduce the residual to zero. To do this, we set \( g_0 = 1, \) and solve for \( g_1,
$g_2, \ldots$. This leads to the following set of equations, where $\beta = u + (1-u)/x$:

\begin{align}
(4.6) \quad g_1 &= \frac{\lambda_1 + \lambda_2 + \mu_3 - \lambda_1/x - \mu_3 x}{\mu_2 \beta} \\
(4.7) \quad g_2 &= g_1 \frac{\lambda_1 + \lambda_2 + \mu_1 + \mu_2 - \lambda_1/x - \mu_1 x}{\mu_2 \beta} - \frac{q \mu_3 x + \lambda_2}{\mu_2 \beta} \\
(4.8) \quad g_{i+1} &= g_i \frac{\lambda_1 + \lambda_2 + \mu_1 + \mu_2 - \lambda_1/x - \mu_1 x}{\mu_2 \beta} - \frac{q \mu_3 x + \lambda_2}{\mu_2 \beta}, \\
(4.9) \quad g_N &= g_{N-1} \frac{\lambda_1 + \mu_1 + \mu_2 - \lambda_1/x - \mu_1 x}{\mu_4 \beta} - g_{N-2} \frac{q \mu_3 x + \lambda_2}{\mu_4 \beta}.
\end{align}

Equation (4.8) will be called the homogeneous equation, and equations (4.6), (4.7) and (4.9) will be called heterogeneous equations. To express the residual of (4.5) it is convenient to introduce $g_{N+1}$ as follows:

\begin{equation}
(4.10) \quad g_{N+1} = g_N \frac{\lambda_1 + \mu_1 - \lambda_1/x}{\mu_4 \beta} - g_{N-1} \frac{\mu_1 (1-t)x}{\mu_4 \beta}.
\end{equation}

Clearly, $g_{N+1}$ is a function of $x$, and to express this, we will sometimes write $g_{N+1}(x)$. Every eigenvalue $x$ must satisfy $g_{N+1}(x) = 0$, and if $g_{N+1}(x) = 0$, (4.10) reduces to (4.5). We note that if $x = 0$, limits arguments must be used. This causes no problem, however, because we will show that all eigenvalues are positive. If $x > 0$, $g_0$ can never vanish except for the trivial solution, as the reader may verify. This justifies the choice $g_0 = 1$.

For our algorithms, sign variations will be crucial. We say that the sequence $[g_0, g_1, \ldots, g_{N+1}]$ has a sign variation if $g_i g_{i+1} < 0$. If $g_i = 0$, a sign variation occurs if $g_{i-1} g_{i+1} < 0$, $i = 1, 2, \ldots, N$. We denote the number of sign variations for a given value of $x$ by $n(x)$.

**Theorem 4.1.** If $\lambda_1 > 0$, there are exactly $N + 1$ distinct eigenvalues between 0 and 1. Also, if the values $x_0 > x_1 > \cdots > x_N$, then $n(x_i) = i$.

**Proof.** The proof depends on Sturm sequences. For $\{g_i, 0 \leq i \leq N + 1\}$ to be a Sturm sequence, we must have $g_0 > 0$, and if $g_i = 0$, $g_{i-1} g_{i+1} < 0$. It is easily verified that both conditions hold as long as $x > 0$. It is not difficult to show that in any Sturm sequence, $n(x)$ cannot change its value unless $g_{N+1} = g_{N+1}(x)$ has a single zero. Hence, the number of zeros in the interval $(a, b)$ cannot exceed $|n(a) - n(b)|$. If $x_0$ is the Perron-Frobenius eigenvalue of $R$, then $x_0$ is positive, and so is the corresponding eigenvector $g(x_0)$. Hence, $n(x_0) = 0$. On the other hand, if $x = 0 +$, and $\lambda_1 > 0$, then, by equations (4.1) to (4.4) and (4.10), $g_i$ and $g_{i+1}$ have opposite signs for $i = 0, 1, \ldots, N$. This implies $n(0+) = N + 1$. Hence, there must be $N + 1$ eigenvalues between 0 and $x_0 +$, and we can denote these eigenvalues by $x_N, x_{N-1}, \ldots, x_1, x_0$, with $x_N < x_{N-1} < \cdots < x_1 < x_0$. With this convention, $n(x_i) = i$. This completes the proof. A similar proof can be found in [6, pg. 102].

When $\lambda_1 = 0$, then the theorem does not apply. In this case, $x = 0$ is an eigenvalue, and the corresponding eigenvector $[1, 0, 0, \ldots, 0]$ violates the conditions...
of a Sturm sequence. In fact, the case where \( \lambda_1 = 0, u = 0, p = t = 1, \mu_1 = \mu_3 \) and \( \mu_2 = \mu_4 \) has been analysed in [8]. This system is depicted in Figure 4.1. For this system, it turned out that \([N/2]\) (the ceiling of \(N/2\)) eigenvalues are 0, and the remaining ones are in \((0, 1)\). Hence, there are now multiple eigenvalues, an issue dealt with in [8].

Theorem 4.1 implies that there cannot be any zeros in the interval \((a, b)\), \(0 < a < b\), if \(n(a) = n(b)\). This can be used to exclude intervals from further search. In fact, one can start with an appropriate initial interval, and divide the interval into half, and if one subinterval contains no zeros of \(g_{N+1}(x)\), one can concentrate on the other half. This results in either one or two subintervals, which can be searched in a similar fashion. This continues until there is a distinct interval for every zero. The following algorithm to find the eigenvalues \(x_{nx2} = x[nx2]\) implements this idea (see [7]):

```plaintext
procedure getx(x1, nx1, x2, nx2)
if (nx1 = nx2) return
x := (x1+x2)/2
if (x2-x1 ≤ ε)
    then if (nx1 = nx2+1) x[nx2] := x and return
    else report multiple eigenvalues and return
nx := n(x)
getx(x1, nx1, x, nx)
getx(x, nx, x2, nx2)
return
```

The initial call to start the algorithm would be \(\text{getx(a, n(a), b, n(b))}\) if all zeros of interest are in the interval \((a, b)\). In [7], the \(g_n\) were found recursively, starting with \(g_0 = 1\), and using (4.6) to (4.9). Here, we will give a more efficient algorithm.

4.2. Solution of the Difference Equations. It is well known that \(y_i = y^{i-1}\), \(i \geq 1\), is a solution of (4.3). Substituting \(y^{i-1}\) for \(g_i\) in (4.3) yields, after simplifications

\[
0 = y^2 - \frac{\lambda_1 + \lambda_2 + \mu_1 + \mu_2 - \lambda_1/x - p\mu_1 x}{\mu_2 \beta} + \frac{q\mu_1 x + \lambda_2}{\mu_2 \beta}.
\]

\[\text{Fig. 4.1. System with } \lambda_1 = u = 0, p = t = 1.\]
This quadratic equation has two solutions

\[ y_1 = \frac{b(x) - \sqrt{d(x)}}{2}, \quad y_2 = \frac{b(x) + \sqrt{d(x)}}{2}, \]

where

\[ b(x) = \frac{\lambda_1 + \lambda_2 + \mu_1 + \mu_2 - \lambda_1/x - p\mu_1 x}{\mu_2^\beta}, \]

\[ d(x) = b(x)^2 - 4 \left( \frac{q\mu_1 x + \lambda_2}{\mu_2^\beta} \right). \]

If \( d(x) = 0 \), we have \( y_1 = y_2 = y \). Also useful are the equations,

\[ y_1 + y_2 = b(x), \quad y_1 y_2 = \frac{q\mu_1 x + \lambda_2}{\mu_2^\beta}. \]

Because \( x > 0 \) and \( \beta = u + (1 - u)/x > 0 \), \( y_1 y_2 > 0 \). Clearly,

\[ g_n = \begin{cases} 
  d_1 y_1^{n-1} + d_2 y_2^{n-1} & \text{if } d(x) \neq 0 \\
  (d_1 + (n-1)d_2)y^{n-1} & \text{if } d(x) = 0 
\end{cases} \quad n = 1, 2, \ldots, N - 1. \]

To find \( d_1 \) and \( d_2 \), we first obtain \( g_1 \) and \( g_2 \) from (4.6) and (4.7), and we then solve

\[ g_1 = d_1 + d_2, \quad g_2 = d_1 y_1 + d_2 y_2 \quad \text{if } d(x) \neq 0, \]

\[ g_1 = d_1, \quad g_2 = (d_1 + d_2)y \quad \text{if } d(x) = 0. \]

This yields

\[ d_1 = \frac{g_1 y_2 - g_2}{\sqrt{d(x)}}, \quad d_2 = \frac{g_2 - g_1 y_1}{\sqrt{d(x)}} \quad \text{if } d(x) \neq 0, \]

\[ d_1 = g_1, \quad d_2 = \frac{g_2}{y} - g_1 \quad \text{if } d(x) = 0. \]

If \( d(x) \geq 0 \), we can calculate \( g_{N-2} \) and \( g_{N-1} \) using (4.16), and this allows us to obtain \( g_N \) from (4.9) and \( g_{N+1} \) from (4.10). This needs fewer floating point operations than the recursive calculation suggested by (4.3), and since every floating point operation leads to some rounding error, this also tends to increase the accuracy of the results.

When \( d(x) < 0 \), \( y_1 \) and \( y_2 \) are complex conjugate, we use \((d, \alpha)\) and \((r, \phi)\) as the polar coordinates of \( d_1 \) and \( y_1 \), respectively. To fix the sign of the angles, we adhere to the convention

\[ d_1 = d(\cos \alpha - i \sin \alpha) \]

\[ y_1 = r(\cos \phi - i \sin \phi). \]

The expression involving \( y_1 \), together with (4.12) implies that \( \phi > 0 \). On the other hand, \( \alpha \) has the opposite sign as the imaginary part of \( d_1 \). A simple calculation based
on (4.18) shows that the imaginary part of \( d_1 \) is \(-\frac{q_1 b(x)/2-q_2}{\sqrt{-d(x)}}\), and it follows that the sign of \( \alpha \) is equal to the sign of \( g_1 b(x) - 2g_2 \). Moreover

\[
2d = \sqrt{g_1^2 + \frac{(2g_2 - g_1 b(x))^2}{-d(x)}}, \quad \cos \alpha = \frac{g_1}{2d} \tag{4.19}
\]

\[
r = \sqrt{\lambda_2 + \frac{q\mu_1x}{\mu_2\beta}}, \quad \cos \phi = \frac{b(x)}{2r} \tag{4.20}
\]

With the above notations, one obtains, after a minor calculation

\[
g_n = 2dr^{n-1}\cos((n-1)\phi + \alpha). \tag{4.21}
\]

4.3. The Location of the Eigenvalues. To locate the eigenvalues, we need the number of sign variations \( n(x) \). To do this, we will need to make use of the sequence \( \{(-1)^ng_n, n = 0, 1, \ldots, N+1\} \). Since \( g_n \) and \( g_{n+1} \) have different signs whenever \( (-1)^ng_n \) and \( (-1)^{n+1}g_{n+1} \) have the same sign, the number of sign variations of the sequence \( \{g_n, n = 0, 1, \ldots, N+1\} \) equals the number of sign permanences of the sequence \( \{(-1)^ng_n, n = 0, 1, \ldots, N + 1\} \) and vice-versa. We also define \( g(n) \) to be given by (4.16), except that \( n \) is real.

We now concentrate on the number of sign variations of \( g_n \) for \( n = 1, 2, \ldots, N-1 \). If \( d(x) > 0 \), \( y_1 \) and \( y_2 \) have the same sign because of (4.15) and the fact that \( \beta > 0 \). Let us first consider the case where \( y_1 \) and \( y_2 \) are both positive. The equation (4.16) implies that \( g_n \) and \( g_{n+1} \) will have different signs only if \( g(m) = 0 \) for some \( m \) between \( n \) and \( n+1 \). This implies

\[
\frac{d_1}{d_2} = \left(\frac{y_1}{y_2}\right)^m.
\]

This equation has at most one solution, that is, there is at most one sign variation in the sequence \( \{g_n, n = 1, 2, \ldots, N-1\} \). Hence, the number of sign variations can be found as

\[
I(g_1 \leq 0) + I(g_1g_{N-1} < 0) + I(g_Ng_{N+1} < 0) + I(g_{N-1} = 0) + I(g_N = 0). \tag{4.22}
\]

Here, \( I(H) \) is 1 if the condition \( H \) holds, and 0 otherwise. If \( y_1 \) and \( y_2 \) are both negative, we can use the same equation to find the number of sign variations of the sequence \( \{(-1)^ng_n, n = 0, 1, \ldots, N+1\} \), except that all \( g_n \) must be replaced by \( (-1)^ng_n \). If this number is \( \bar{n}(x) \), then \( n(x) = N + 1 - \bar{n}(x) \).

If \( d(x) = 0 \), \( y_1 = y_2 = y = b(x)/2 \), and since \( y_1y_2 \neq 0 \), \( b(x) \neq 0 \). If \( b(x) > 0 \), then \( y^{n-1} = \left(\frac{b(x)}{2}\right)^{n-1} > 0 \). We define \( g(n) \) according to (4.16) and conclude, using also (4.18) that a sign change can only occur if \( g_1 + (m-1)(\frac{2q_2}{\mu_2\beta} - g_1) = 0 \) for some value of \( m \). Again, this expression has at most one zero, which implies that \( n(x) \) can be calculated exactly the same way if \( d(x) > 0 \), as was done above. Similarly, if \( d(x) = 0 \), \( b(x) < 0 \), one obtains essentially the same result as for the case \( d(x) \neq 0 \), \( b(x) < 0 \).
It is clear that if $y_1$ and $y_2$ are both positive, $b(x) > 0$, and if $y_1$ and $y_2$ are both negative, $b(x) < 0$. The converse also holds, that is, if $b(x) > 0$, and $y_1$ and $y_2$ are real, they must both be positive. In this case, (4.22) indicates that $n(x)$ must be less than or equal to $3 + 3 = 6$. Similarly, if $b(x) < 0$, $n(x) \geq N - 2$. It follows that there are at most 3 eigenvalues leading to real values for $y_1$ and $y_2$. All other eigenvalues must correspond to complex values for $y_1$ and $y_2$.

To locate the eigenvalues satisfying $d(x) < 0$, we use the following theorem.

**Theorem 4.2.** The function $d(x)$ given by (4.14) has exactly 2 zeros in $(0, 1)$, say $x^{(l)}$ and $x^{(r)}$. In fact, the eigenvalues corresponding to the complex case, $d(x) < 0$, are in the interval $(x^{(l)}, x^{(r)})$.

**Proof.** We multiply $d(x)$ by $(\beta \mu_2)^2$ to get the following expression

$$
(s - \lambda_1/x - p\mu_1x)^2 - 4\mu_2(\gamma \mu_1x + \lambda_2),
$$

where $s = \lambda_1 + \lambda_2 + \mu_1 + \mu_2$. This expression is a polynomial of degree 4, and it can have at most four zeros. It converges to $\lambda_1^2/x^2$ as $x \to 0$, and it equals $(\lambda_2 + \gamma \mu_1 - \mu_2)^2$ for $x = 1$. Since both values are positive and the function $d(x)$ is continuous on $(0, 1)$, the number of zeros must be even in that interval, that is, there are either 0, 2 or 4 zeros of $d(x)$. However, $d(x) < 0$ if $b(x) = 0$, and $b(x)$ has a zero in $(0, 1)$ and a zero in $[1, \infty)$, which implies that $d(x)$ has exactly two zeros in $(0, 1)$ as claimed. To show that $b(x)$ has one zero in $(0, 1)$ and one zero in $[1, \infty)$, consider

$$
(s - \lambda_1/x - p\mu_1x) = -(p\mu_1x^2 - sx + \lambda_1)/x.
$$

If $x_1$ and $x_2$ are the 2 zeros of $b(x)$, $x_1x_2 = \lambda_1/(p\mu_1) > 0$, that is, $x_1$ and $x_2$ must have the same sign. Moreover, $b(x) < 0$ as $x \to 0$, and $b(1) > 0$. Hence, $b(x)$ has exactly one zero in $(0, 1)$, which means that the other one must be outside the interval. Since $x_1x_2 > 0$, the zero outside $(0, 1)$ must be positive, that is, $0 < x_1 < 1 < x_2$. This completes the proof that $d(x)$ has exactly two zeros in $(0, 1)$, whereas the other two zeros are in $[1, \infty)$. $\blacksquare$

For the case $d(x) < 0$, the number of sign changes is given by

$$
n(x) = I(g_1 < 0) + \frac{(N - 2)\phi + \alpha + 1/2}{\pi} - \frac{\alpha + 1/2}{\pi} + 1
$$

(4.23)

$$
= I(g_{N-1}g_N < 0) + I(g_{N}g_{N+1} < 0) + I(g_N = 0) + I(g_{N+1} = 0).
$$

As before, $I(\cdot)$ is the indicator function, $[x]$ is the floor of $x$, and $\lceil x \rceil$ the ceiling.

To prove (4.23), we have to find the number of sign variations between $g_1$ and $g_{N-1}$. To do this, we define $g(n)$ to be the continuous version of (4.21). Clearly, $g(n)$ is zero whenever $(n - 1)\phi + \alpha$ is equal to $(k + 1/2)\pi$, where $k$ is an integer. Since $n$ ranges from 1 to $N - 1$, this means

$$
\frac{\alpha + 1/2}{\pi} \leq k \leq \frac{\alpha + (N - 2)\phi + 1/2}{\pi}.
$$

Since $k$ is integer, this implies

$$
\left\lfloor \frac{\alpha + 1/2}{\pi} \right\rfloor \leq k \leq \left\lceil \frac{\alpha + (N - 2)\phi + 1/2}{\pi} \right\rceil.
$$
The number of distinct values of $k$ within these bounds equals

$$\left\lfloor \frac{\alpha + (N - 2) \phi + 1/2}{\pi} \right\rfloor - \left\lceil \frac{\alpha + 1/2}{\pi} \right\rceil + 1.$$  

The proof of (4.23) can now be completed by counting the sign variations outside the interval in question. Note that if either $g_1 = 0$ or $g_{N-1} = 0$, this is counted as a sign variation between $g_1$ and $g_{N-1}$. Also note that since $\phi < \pi$, every zero of $g(n)$ corresponds to a sign change.

We now can apply the procedure “getx” as follows. First, we calculate $x(l)$ and $x(r)$, and we find $n(x(l))$ and $n(x(r))$. We also find $g_{N+1}(x(l))$ to see if $x(l)$ is an eigenvalue, and we do the same for $g_{N+1}(x(r))$. After this, the interval from 0 to 1 is divided into three subintervals, namely $(0, x(l))$, $(x(l), x(r))$ and $(x(r), 1)$, and the procedure “getx” is applied for each interval.

5. The Boundary Equations. Due to the fact that $Q_{01} \neq Q_1$, the treatment of the boundary equations in this model is different from the one in [9] or [10]. We first solve (2.2) for $\pi_0$, which yields

$$\pi_0 = \pi_1 Q_{-1} (Q_{00})^{-1}. \quad (5.1)$$

It follows from the theory of Markov chains that if the process is recurrent, $Q_{00}$ has an inverse. We now substitute (5.1) into (2.3) to obtain

$$0 = \pi_1 (Q_0 + Q_{-1} (Q_{00})^{-1} Q_{01}) + \pi_2 Q_{-1} = \pi_1 \bar{Q}_{11} + \pi_2 Q_{-1}, \quad (5.2)$$

where $\bar{Q}_{11} = (Q_0 + Q_{-1} (Q_{00})^{-1} Q_{01})$. We now have to combine the solutions given by (2.5) to satisfy this equation.

For each eigenvalue $x_\nu$, $\nu = 0, 1, \ldots, N$, the eigenvector $g^{(\nu)}$ is given by solving (4.6) to (4.9). Any solution $\pi_n = g^{(\nu)} x_\nu^{n-1}$ solves (2.4), and so does any linear combination of these solutions. In other words, all possible solutions have the form

$$\pi_n = \sum_{\nu=0}^N c_\nu g^{(\nu)} x_\nu^{n-1}, \quad n > 0. \quad (5.3)$$

To write this equation in matrix form, let $\Lambda = \text{diag}(x_\nu)$, and let $G$ be the $(N + 1) \times (N + 1)$ matrix $[g^{(0)}, \ldots, g^{(N)}]^T$, where $T$ denotes the transpose. Therefore, (5.3) becomes:

$$\pi_n = c \Lambda^{n-1} G. \quad (5.4)$$

We need to determine $c = [c_0, c_1, \ldots, c_n]$ in such a way that (5.2) is satisfied. Clearly, $\pi_1 = c G$, $\pi_2 = c \Lambda G$.

Hence (5.2) leads to

$$c (G \bar{Q}_{11} + \Lambda G Q_{-1}) = 0. \quad (5.5)$$
On the other hand, using (5.1) and the fact that the sum of all probabilities must be one, we have

\[ 1 = (\pi_0 + \sum_{n=1}^{\infty} \pi_n) e = c \left( GQ_{-1}(-Q_{00})^{-1} + \sum_{n=1}^{\infty} \Lambda^{-1} G \right) e \]

(5.6)

\[ e \left( GQ_{-1}(-Q_{00})^{-1} + \text{diag}(\frac{1}{1-x_n}) G \right) e. \]

Here, \( e \) is a column vector with all entries equal to 1. Equations (5.5) and (5.6) fully determine \( c \). When solving (5.5), we can express all \( c_i \) as a multiple of \( c_0 \), and use (5.6) to find \( c_0 \). Notice that \( c_0 \) is the coefficient of the dominant eigenvalue \( x_0 \), and it must be non-zero.

6. Numerical Considerations. It has been shown in [7] that given a certain precision \( 2^{-\alpha} \), all eigenvalues can be found by evaluating \( g_{N+1}(x) \) for \((N+1)(\alpha - \log_2(N + 1)) + \log_2(N + 1) \) different values of \( x \). Since the number of operations to evaluate \( g_{N+1}(x) \) is independent of \( N \), the time complexity of finding all eigenvalues is essentially \( O(N \log_2 N) \), which is very satisfactory. One can even improve upon this by using the secant method or other root finding methods once one has \( N+1 \) intervals, each containing exactly one eigenvalue.

The question arises as to what extent rounding errors affect the results. To get a handle on this question, let us compare our method with the method of Neuts [18]. There, \( R \) is determined by using the following matrix equation

(6.1) \[ 0 = Q_1 + RQ_0 + R^2Q_{-1}. \]

We can assume that even if a fast method is used, such as suggested in [13], this equation is only met at a precision of \( \pm \epsilon \), where \( \epsilon \) is some multiple of the machine precision. In our case, we solve

(6.2) \[ 0 = g^{(i)}(Q_1 + x_iQ_0 + x_i^2Q_{-1}), \quad i = 0, 1, \ldots, N. \]

As in the case of (6.1), we concentrate on the residuals. This makes the analysis quite different from the analysis concentrating on the errors of the vectors \( g^{(i)} \). The difference is that when calculating the \( g^{(i)} \) using (4.6) to (4.10), errors in the values of \( g^{(i)}_j \) for low \( j \) often have a catastrophic effect on the \( g^{(i)}_j \) for higher values of \( j \) because they are increased due to subtractive cancellation. The residuals, on the other hand, are independent of what happens earlier in the recursion, as pointed out in [7]. In fact, no matter what values of \( g_{i-1} \) and \( g_i \) is used, \( g_{i+1} \) is always determined such that the residual is zero except for rounding. Hence, earlier errors have no influence on the present residual. This means that except for the last equation, the residuals of (6.2) can be expected to be small. The residuals of the last equation, of course, are essentially given by \( g_{N+1} \), and we make this residual small by choosing the correct value of \( x \). Hence, (6.2) will be satisfied rather accurately.

Once all \( g^{(i)} \) and \( x_i \) are found, one calculates

\[ \pi_n = \sum_{i=0}^{N} c_i g^{(i)} x_i^n. \]
Two-Station Queueing Networks

Table 6.1
The maximum value of $|c_i|$ for selected problems

| $N$ | $\lambda$ | $|c_i|_{\text{max}}$ |
|-----|----------|------------------|
| 5   | 8        | 0.0292           |
| 10  | 8        | 0.0203           |
| 15  | 8        | 0.0090           |
| 20  | 8        | 0.0349           |
| 5   | 9        | 0.0169           |
| 10  | 9        | 0.0127           |
| 15  | 9        | 0.0112           |
| 20  | 9        | 0.0765           |

It is obvious that if all $|c_i| \leq 1$, then the resulting $\pi_n$ is again correct with a high precision. What is possible, however, is that some $c_i$ have high values, and in this case, any errors are multiplied by a large number. Hence, large values of $|c_i|$ are an indication of potential problems. We therefore calculated the $c_i$ for a number of models, and obtained the largest absolute value of $c_i$, which we denote by $|c_i|_{\text{max}}$. The results are given in Table 1. The model is the standard model, with every customer going from line 1 to line 2, no premature departures, and no feedback. The values of $\mu_i$, $i = 1, 2, 3, 4$ are as indicated in Table 1. As one can see, the $c_i$ are rather small, and they increase only slightly as $N$ increases. Hence, the residuals of (6.2) remain small.

We note that our method is already advantageous for $N = 20$ as compared to matrix analytic methods. In matrix analytic methods, matrix multiplications are required in each iteration, and a single matrix multiplication requires $2N^3$ operations, which makes 16000 operations for $N = 20$. This has to be compared with $N \log_2 N$ for our method, which is around 100 for the same value of $N$.

7. Summary. To solve the equation $gQ(x) = 0$, we choose some value for $x$ and determine the vectors $g$ such that only the last entry of the row vector $gQ(x)$ differs from zero. This can be done by the following algorithm:

**Algorithm**
1. Set $g_0 = 1$ and find $g_1$ by (4.6) and $g_2$ by (4.7).
2. Find $y_1$ and $y_2$ using (4.12).
3. Find $d_1$ and $d_2$ by (4.18).
4. To determine $g_{N-2}$ and $g_{N-1}$, use (4.16). If $d(x) < 0$, (4.21) can be used.
5. Finally, get $g_N$ by (4.9) and $g_{N+1}$ by (4.10).

A search procedure is then used to reduce $g_{N+1}$ to zero. By using Sturm sequences, we can locate all eigenvalues within small ranges, which allows for efficient and reliable solution algorithms. Our procedure can be generalized. For instance, there is no problem to extend the model such that it can handle multiple servers, or irregularities.
when the second line reaches its limit. In fact, the method suggested here can be used for any queueing problem leading to a tridiagonal matrix $Q(x)$, such as the bilingual call centre problem studied in [20]. The reason generalizations become easy is that the recursive equations are used directly, and not solved as in [10] and [9].

Notice that the step involving finding all eigenvalues is $O(N)$, and this is the only process that has to be done iteratively. The step involving finding the $g_i$ is $O(N^2)$, and finding $c$ by solving (5.5) is $O(N^3)$. Hence, the main effort is likely to be spent finding $c$, and not finding eigenvectors and eigenvalues.

The equations (4.1) to (4.4) are all satisfied exactly, no matter how accurate or inaccurate the eigenvalue $x$ in question may be. If $x$ is not accurate, of course, (4.5) is not satisfied. If (4.1) to (4.5) are satisfied, so are the equilibrium equations for this particular solution. To satisfy the initial conditions, one has to combine these solutions, and unless there are some huge values $c_i$, this combination of solutions has an accuracy which is comparable to the individual solutions. In fact, one can approximate the accuracy of the final solution by first calculating the residuals of (4.5), say $r_i$ is the residual for eigenvector $g^{(i)}$, and form $\sum_{i=0}^{N} |c_i r_i|$. In all cases we considered, this sum turned out to be small.

We should mention that irregular bounds in QBD processes cause difficulties. For matrix-analytic methods, these difficulties are outlined by Neuts [18, pgs. 24-27]. Though we do not use matrix-analytic methods here, we still would have the same problems as Neuts, and our earlier efforts were considerably hampered because of this difficulty. In this paper, as in [7], we simply bypass this problem by reducing the transition matrix to one that has a regular boundary.

Although in this paper we restricted the second queue to be finite, we believe that this paper may contain a method leading to the solution for two infinite queues. In this case, there is of course no blocking. We are hoping to solve this problem in a not too distant future.

Acknowledgments. This research was supported by the Natural Sciences and Engineering Research Council of Canada. We also thank the referees for their useful comments.

REFERENCES

[7] W. K. Grassmann. The use of eigenvalues for finding equilibrium probabilities of certain Marko-
Two-Station Queueing Networks


