Journal of the Operations Reseach
Society of Japan
Vol. 19, No. 2, June, 1976

A NUMERICAL METHOD FOR THE STEADY-STATE PROBABILITIES OF A G1/G/C QUEUEING SYSTEM IN A GENERAL CLASS

YUKIO TAKAHASHI, *Tohoku University*

YOSHINORI TAKAMI, *Tokyo Institute of Technology*

(Received September 22, 1975)

Abstract. A numerical method is proposed for solving the balance equations of the steady-state probabilities of a $GI/G/c$ queueing system in a general class. The method is based on an iterative calculation of conditional probabilities, instead of absolute probabilities, conditioned by the number of customers in the system. By skillfully exploiting a convergence property of the conditional probabilities, it provides a fairly accurate solution of the balance equations with relatively little computational burden.

1. Introduction

In this paper, a numerical method is proposed for solving the balance equations of the steady-state probabilities of a $GI/G/c$ queueing system in a general class. The method is a direct application of the (modified) lumping method introduced in [6] for the stationary distribution of a Markov chain. It is based on an iterative calculation of conditional probabilities of the queueing system conditioned by the number of customers in the system. By using the conditional probabilities, rather than absolute probabilities, the system of linear equations of the steady-state probabilities is divided into a set of smaller systems of linear equations, and it can be solved with less computational burden by exploiting convergence property of the conditional probabilities. Furthermore, errors included in the solution become fairly small. The computational time required for solving the balance equations by our method is nearly independent of the value of the utilization factor ρ . Hence, our method is effective even if ρ is near to 1 .

2. Balance equations of the steady-state probabilities

For many queueing systems, the steady-state probabilities can be expressed as a solution of the balance equations of the form (2.4) below. As an example let us consider the $E_k/E_r/c$ queueing system. In the system, customers arrive at a service facility with c channels in parallel via an Erlang process of order k with mean rate λ/k. If all channles are busy the customers form a single queue and are served in order of arrival. The service times are independent random variables subjecting to the Erlang distribution of order r with mean r/μ.

In order to define states of the system, it is convenient to introduce stages for both the arrival process and the service processes at channels. A service at a channel is considered to consist of r consecutive exponential phases of service and each stage represents a phase of service. The stages for the arrival process are interpreted similarly. Then the state of the system can be represented by an ordered $(r+2)$-tuple of nonnegative integers

$$(2.1) \quad (n, j; m_1, \cdots, m_r),$$

where n denotes the total number of customers in the system, j the stage of the arrival process and m_i the total number of customers in the ith stages of service. Let S_n be the set of all possible states such that the total number of customers in the system is equal to n. Since $m_1 + \cdots + m_r = \min(c, n)$, the number of states in S_n is given by

$$(2.2) \quad s_n = k \times \binom{n' + r - 1}{n'}$$

where $n' = \min(c, n)$. We will number the states in S_n by a suitable rule and refer them by pairs

$$(2.3) \quad (n; i), \qquad i = 1,2,\cdots,s_n, \quad n = 0,1,2,\cdots.$$

Let P_{ni} denote the probability that the state of the system is $(n; i)$ in the steady state, and let α_n be the row vector with entries P_{ni}, $i = 1,2,\cdots,s_n$. Then the balance equations of the system in the steady state are written as

$$(2.4) \quad \begin{aligned} \alpha_0 D_0 &= \alpha_0 B_0 + \alpha_1 A_1 \\ \alpha_n D_n &= \alpha_{n-1} C_{n-1} + \alpha_n B_n + \alpha_{n+1} A_{n+1}, \quad n = 1,2,3,\cdots, \end{aligned}$$

where A_n, B_n and C_n are matrices representing the intensities of the transition probabilities from states in S_n to states in S_{n-1}, S_n and

S_{n+1} respectively, and D_n is the diagonal matrix whose ith diagonal entry is equal to the sum of all entries in ith rows of matirces A_n, B_n and C_n. In other words, D_n is the diagonal matrix satisfying

$$D_0\xi_0 = B_0\xi_0 + C_0\xi_1 \quad , \quad \text{or} \tag{2.5}$$
$$D_n\xi_n = A_n\xi_{n-1} + B_n\xi_n + C_n\xi_{n+1} \qquad \text{for} \quad n \geqq 1 ,$$

where ξ_n is the column vector of order s_n with all entries equal to 1. In this case, all the diagonal entries of D_n are equal to $\lambda + n'\mu$. Further

$$A_n = A_c \ , \ B_n = B_c \ , \ C_n = C_c \ \text{and} \ D_n = D_c \qquad \text{for} \quad n \geqq c . \tag{2.6}$$

If the utilization factor $\rho = r\lambda/ck\mu < 1$, then the steady-state probabilities P_{ni} are uniquely determined by the balance equations (2.4) together with the normalization constraint

$$\sum_{n=0}^{\infty} \sum_{i=1}^{s_n} P_{ni} = 1 . \tag{2.7}$$

We can show that a queueing system with more general interarrival time and service distributions has balance equations of a similar form. Let G_r represent a distribution which can be expressed as the distribution of the absorbing time of a continuous time absorbing Markov chain with r transient states and a singel absorbing state. The transient states of the chain correspond to the stages in the case of the Erlang distribution, and the absorption to the absorbing state represents the completion of, say, a service. A continuous time absorbing Markov chain with transient states labeled $1,2,\cdots,r$ and an absorbing state labeled $r+1$ is characterized by parameters q_{0i} $(i = 1,2,\cdots,r)$, μ_i $(i = 1,2,\cdots,i)$ and q_{ij} $(i = 1,2,\cdots,r \ ; \ j = 1,2,\cdots,r+1)$, where q_{0i} is the probability of starting from state i, $1/\mu_i$ is the mean of an exponentially distributed duration time at state i, and q_{ij} is the conditional transition probability from state i to state j conditioned that a transition from state i occurs. By suitably choosing these parameters, various distributions can be expressed as distributions of absorbing times of such absorbing Markov chains. Clearly, Erlang distributions and mixtures of them are G_r type distributions.

For a queueing system $G_k/G_r/c$, i.e., a queueing system with c channels having G_k type interarrival time distribution and a G_r type service distribution, the state of the system is represented by the $(r+2)$-tuple in (2.1), too. So the balance equations of the system are of the same form as in (2.4), though the matrices A_n, B_n and C_n may have more nonzero entries than in the case of the $E_k/E_r/c$ queueing system.

As will be shown later, when the balance equations are solved numerically, the computation becomes much simpler if the matrices B_n, $n = 0,1,2,\cdots$, are triangular matrices. If the absorbing Markov chains associated with the interarrival time distribution and the service distribution satisfy an acyclic condition that the conditional transition probabilities $q_{ij} = 0$ for $i > j$, then we can number the states in S_n so that B_n becomes triangular. For the purpose we may number the states in the order of

$$(2.8) \qquad m_1 + m_2 c + \cdots + m_r c^{r-1} + j c^r .$$

The Erlang distributions and mixtures of them can be expressed as distributions of the absorbing times of acyclic absorbing Markov chains. So for queueing systems with these distributions as interarrival time and service distributions, the matrices B_n can be made triangular.

3. Equations for conditional probability vectors β_n

Now let us consider a queueing system with the balance equations (2.4). Let $w_n = \alpha_n \xi_n$ and $\beta_n = (b_{ni}) = \frac{1}{w_n}\alpha_n$. Then w_n is the probability that the number of customers in the system is equal to n, and the ith entry b_{ni} of β_n is the conditional probability that the state of the system is $(n\,;\,i)$ given that the number of customers in the system is equal to n.

Here we show that, if the values of the vectors β_{n-1} and β_{n+1} are known, then the vector β_n is obtained by solving a system of linear equations of order $s_n + 2$. The balance equations (2.4) are rewritten as

$$(3.1) \qquad \begin{aligned} \beta_0 D_0 &= \beta_0 B_0 + x_0 \beta_1 A_1 \\ \beta_n D_n &= z_n \beta_{n-1} C_{n-1} + \beta_n B_n + x_n \beta_{n+1} A_{n+1}, \quad n = 1,2,3,\cdots, \end{aligned}$$

where $x_n = w_{n+1}/w_n$ and $z_n = 1/x_{n-1} = w_{n-1}/w_n$. (3.1) provides s_n equations for β_n, but they contain two more unknown variables x_n and z_n. So we need two more equations. One is the normalization constraint

(3.2) $\beta_n \xi_n = 1$.

To derive another one, we note that from (2.4) and (2.5)

$$(3.3)\quad w_{n-1}\beta_{n-1}C_{n-1}\xi_n - w_n\beta_n A_n \xi_{n-1}$$
$$= w_n\beta_n C_n\xi_{n+1} - w_{n+1}\beta_{n+1}A_{n+1}\xi_n$$
$$= w_{n+m}\beta_{n+m}C_{n+m}\xi_{n+m+1} - w_{n+m+1}\beta_{n+m+1}A_{n+m+1}\xi_{n+m}$$

for any $n, m \geqq 1$. Since $w_n \to 0$ as $n \to \infty$, the right side of (3.3) vanishes as $m \to \infty$, and it implies that

$$(3.4)\quad z_n\beta_{n-1}C_{n-1}\xi_n = \beta_n A_n\xi_{n-1}, \quad n = 1,2,3,\cdots .$$

This is the other equation for β_n . If, for $n \geqq 1$, we regard the equations (3.1), (3.2) and (3.4) as $s_n + 2$ equations for $s_n + 2$ variables x_n , z_n and b_{ni} , $i = 1,2,\cdots,s_n$, then they form a system of linearly independent linear equations. So, if the vectors β_{n-1} and β_{n+1} are given, the values of the variables can be obtained by solving the system of equations. Similarly, for $n = 0$, (3.1) and (3.2) form a system of $s_0 + 1$ linearly independent linear equations for $s_0 + 1$ variables x_0 and b_{0i} , $i = 1,2,\cdots,s_0$. Hence β_0 can be obtained from these equations if β_1 is given.

4. Practical algorithm

As was shown in the preceding section, for a queueing system with the balance equations (2.4), the vector β_n is calculated by solving the equations (3.1), (3.2) and (3.4) if the vectors β_{n-1} and β_{n+1} are given. This indicates that the conditional probabilities are calculated by a Gauss-Seidel type block iteration method. Here we will give a practical algorithm of such a method. The algorithm exploits a convergence property of the sequence $\{\beta_n\}$. As will be discussed in the next section, $\{\beta_n\}$ converges to a limit vector β as $n \to \infty$ under a weak condition, and it is expected that the convergence is fast except for the cases with small ρ . So, the exploitment of the convergence property makes the algorithm very efficient.

In the following algorithm $\beta_n^{(h)}$ designates the hth approximation of β_n . At the start of the algorithm, two parameters N and ε must be set.

N is an integer such that β_n is considered to be sufficiently close to the limit vector β if $n \geqq N$, and ε is a positive number such that if all the differences between the corresponding entries of $\beta_n^{(h-1)}$ and $\beta_n^{(h)}$ are less than ε in absolute value then $\beta_n^{(h)}$ is considered to be sufficiently close to β_n.

A practical algorithm

Step 1. *(The first iteration)* Calculate $\beta_0^{(1)}$ according to the procedure stated below using an appropriate initial approximation vector $\beta_1^{(0)}$. Calculate $\beta_n^{(1)}$, $n = 1,2,\cdots,N$, in order of n according to the procedure stated below using $\beta_{n-1}^{(1)}$ and $\beta_{n+1}^{(0)}$, where $\beta_{n+1}^{(0)}$ is an appropriate initial approximation vector, but it will be efficient to use $\beta_{n-1}^{(1)}$ as $\beta_{n+1}^{(0)}$ for $n \geqq c + 1$. Put $h = 2$.

Step 2. *(The h-th iteration)* Calculate $\beta_0^{(h)}$ according to the procedure stated below using $\beta_1^{(h-1)}$. Calculate $\beta_n^{(h)}$, $n = 1,2,\cdots,N$, in order of n according to the procedure stated below using $\beta_{n-1}^{(h)}$ and $\beta_{n+1}^{(h)}$, where $\beta_{N-1}^{(h)}$ is used in place of $\beta_{N+1}^{(h-1)}$.

Step 3. *(Test of convergence)* If all the differences between the corresponding entries of $\beta_n^{(h-1)}$ and $\beta_n^{(h)}$ for $n = 0,1,2,\cdots,N$ are less than ε in absolute value, then go to Step 4. Otherwise increase h by 1 and return to Step 2.

Step 4. *(Calculation of z_n)* Calculate z_n, $n = 1,2,\cdots,N$, from the equation (3.4) using $\beta_{n-1}^{(h)}$ and $\beta_n^{(h)}$.

Step 5. *(Calculation of w_n)* Calculate

$$w_0 = c(1-\rho)\left[c + \sum_{n=1}^{c-1} (c-n) / z_1 \cdots z_n\right]^{-1},$$

and then calculate

$$w_n = w_{n-1} / z_n$$

recursively for $n = 1,2,\cdots,N$.

Step 6. *(Calculation of α_n)* Calculate α_n by

$$\alpha_n = w_n \beta_n^{(h)}$$

for $n = 0,1,2,\cdots,N$.

The determination of w_0 in Step 5 above is based on the relation

$$(4.1) \quad \sum_{n=0}^{c-1} (c - n) w_n = c(1 - \rho)$$

which is satisfied for general queueing systems with c channels. The vector $\beta_n^{(h)}$ in Steps 1 and 2 above can be obtained from $\beta_{n-1}^{(h)}$ and $\beta_{n+1}^{(h-1)}$ by the following procedure.

Procedure for calculating $\beta_n^{(h)}$

(i) Solve the equations $\phi(D_n - B_n) = \beta_{n-1}^{(h)} C_{n-1}$ and $\psi(D_n - B_n) = \beta_n^{(h-1)} A_{n+1}$ for vector valued variables ϕ and ψ respectively.

(ii) Calculate $y = \psi A_n \xi_{n-1} / \phi C_n \xi_{n+1}$.

(iii) Calculate $\eta = y\phi + \psi$.

(iv) Calculate $\beta_n^{(h)}$ by normalizing η as $\beta_n^{(h)} = \frac{1}{\eta \xi_n}\eta$.

For $n = 0$, $\beta_0^{(h)}$ can be obtained only by normalizing the vector ψ defined in (i) as $\beta_0^{(h)} = \frac{1}{\psi \xi_0}\psi$.

We can modify the algorithm so that the parameter N is determined automatically. For the purpose, a test of convergence of the sequence $\{\beta_n\}$ must be added in both Steps 1 and 2. This modification will be effective when the rate of convergence of the sequence $\{\beta_n\}$ is not known.

We conclude this section with a notice about the case of triangular B_n's . Since D_n's are diagonal matrices and diagonal entries of B_n's are equal to zero, if the matrices B_n's are upper triangular matrices, then the entries of the vectors ϕ and ψ in (i) of the above procedure can be obtained in order from the equations

$$(4.2) \quad \phi = \beta_{n-1}^{(h)} C_{n-1} D_n^{-1} + \phi B_n D_n^{-1} \quad \text{and} \quad \psi = \beta_{n+1}^{(h-1)} A_{n+1} D_n^{-1} + \psi B_n D_n^{-1} .$$

In this case the algorithm uses no subtraction operation except for subtractions in testing the convergence in Step 3. Thus we can expect that the solution of the balance equations obtained by this mehtod is very accurate if B_n's are triangular matrices.

5. Convergence property of $\{\beta_n\}$

In the preceding section, we proposed an algorithm for solving the balance equations (2.4) which exploits the convergence property of the sequence $\{\beta_n\}$. In this section we study the convergence property.

Consider the balance equations (2.4) satisfying (2.6). Let $f(\theta) = \sum_{n=c}^{\infty} \alpha_n \theta^n$. $f(\theta)$ is the vector valued generating function of α_n. Multiplying the both sides of (2.4) with θ^n and summing up for $n \geqq c$, then we have

$$(5.1) \quad f(\theta)[D_c - \theta C_c - B_c - \frac{1}{\theta} A_c] = \theta^c \alpha_{c-1} C_{c-1} - \theta^{c-1} \alpha_c A_c .$$

If the matrix in the brackets of the left hand side of (5.1) is nonsingular, then

$$(5.2) \quad f(\theta) = (\theta^c \alpha_{c-1} C_{c-1} - \theta^{c-1} \alpha_c A_c)[D_c - \theta C_c - B_c - \frac{1}{\theta} A_c]^{-1} .$$

Consider the equation for θ

$$(5.3) \quad |D_c - \theta C_c - B_c - \frac{1}{\theta} A_c| = 0 .$$

Let $\theta_1, \theta_2, \cdots$ be the roots of the equation larger than 1 in absolute value, and assume that none of θ_j's is a multiple root and that

$$(5.4) \quad 1 < |\theta_1| < |\theta_2| \leqq |\theta_3| \leqq \cdots .$$

Then from (5.2) $f(\theta)$ must be expressed as

$$(5.5) \quad f(\theta) = \sum_i \gamma_i \frac{(\theta/\theta_i)^n}{1 - \theta/\theta_i} ,$$

and hence

$$(5.6) \quad \alpha_n = \sum_i \frac{1}{\theta_i^{\,n}} \gamma_i , \qquad n \geqq c .$$

Thus the sequence $\{\beta_n\}$ converges to γ_1 and $z_n = w_{n-1}/w_n$ converges to

$\theta_1 > 1$ as $n \to \infty$ under the assumption (5.4). The rate of convergence of the sequence $\{w_n\}$ is governed by $1/\theta_1$ and the rate of convergence of the sequence $\{\beta_n\}$ is governed by $|\theta_1 / \theta_2|$.

Now we shall examine the dependencies of θ_1 and θ_2 to the utilization factor ρ for two simple queueing systems $M/E_2/2$ and $E_2/E_2/2$. The $M/E_2/c$ queueing systems were studied by S. Shapiro [5], and θ_1 and θ_2 can be calculated from an equation derived by him. In the case of $M/E_2/2$, they are given by

$$(5.7) \quad \begin{aligned} \theta_1 &= 8 / \rho\{\rho + 4 + \sqrt{\rho^2 + 8\rho}\} \\ \theta_2 &= (2 + \rho) / \rho \end{aligned}$$

We note that θ_1/θ_2 decreases as ρ increases while $1/\theta_1$ increases with ρ and that $\theta_1/\theta_2 \longrightarrow 1/3$ and $1/\theta_1 \longrightarrow 1$ as $\rho \longrightarrow 1$.

The $E_k/E_r/2$ queueing systems were studied by C. D. Poyntz & R. R. P. Jackson [4], and θ_1 and θ_2 can be obtained by solving an equation derived by them. In the case of $E_2/E_2/2$ the equation is easily solved and

$$(5.8) \quad \begin{aligned} \theta_1 &= 1 / \rho^2 \\ \theta_2 &= (1 + \rho)^2 / \rho^2 \end{aligned} \quad .$$

θ_1 / θ_2 decreases as ρ increases, too, while $1 / \theta_1$ increases with ρ . In this case θ_1 / θ_2 approaches to $1/4$ as ρ tends to 1 .

Thus we might as well conjecture that $|\theta_1 / \theta_2|$ decreases as ρ increases in a general $G_k/G_r/c$ queueing system. In computational experiments by the authors, no case occurred in which the conjecture was violated.

6. Relative merits of the method

In this section we will compare our method with a usual Gauss-Seidel iteration method for a system of linear equations of absolute probabilities. If one wants to use the Gauss-Seidel iteration method for solving the system of balance equations (2.4), he must reduce it to a system of finitely many linear equations by insisting the condition that $\alpha_n = 0$ for $n > N_1$, where N_1 is chosen so that the residual probability $\sum_{n>N_1} w_n$ is negligible. Since the rate of convergence of $\{w_n\}$ is governed by $1 / \theta_1$, N_1 becomes large as ρ approaches to 1 . On the other hand, if one wants to solve the balance equations by our method, he must calculate β_n for $n \leqq N_2$, where N_2 is chosen so that β_n is considered to be sufficiently close to the limit β if $n > N_2$. Since the rate of convergence of $\{\beta_n\}$ is governed by $|\theta_1 / \theta_2|$, we may expect that N_2 decreases as ρ approaches to 1 . Of course one can also exploit the convergence of $\{w_n\}$ in our method. So, the order of the system of equations to be solved is nearly $s_c \times \min(N_1 , N_2)$ in our method, while that is nearly $s_c \times N_1$ in the Gauss-Seidel iteration method. Thus our method is very efficient for large ρ . The values of N_1 and N_2 for the $M/E_5/3$ queueing system are illustrated in Table 1 .

Table 1. N_1 and N_2 for $M/E_5/3$

ρ	0.3	0.6	0.9
N_1	5	10	41
N_2	12	10	9

Allowance limit of errors is 1/1000 .

The second merit of our method is accuracy of the solution. In our method β_n's , $n > N$, are not neglected but are taken into account in calculation of w_n's . So, it is expected that our method provides accurate values not only of α_n's but also of other characteristic quantities of the queueing system such as moments of queue length. (Compare with the case of the Gauss-Seidel iteration method in which α_n's , $n > N$, are set equal to the zero vector.) Furthermore, as was noted in Section 4 , if matrices B_n's are triangular matrices, our method can solve the balance equations without any subtraction operation except for subtractions for testing the convergence of $\beta_n^{(h)}$. So, it is expected that errors arising in the process of computation will be neglibibly small.

The third merit of our method is the fast convergence of $\beta_n^{(h)}$ to β_n . This is due to the exploitment of the convergence property of $\{\beta_n\}$ in the initial setting of $\beta_n^{(0)}$ in Step 1 of the algorithm.

In a word, our method provides an accurate solution of the balance equations with relatively little computational burden.

The authors wrote a FORTRAN program according to our method and tested it on a variety of cases on the FACOM 230-45S at Tokyo Institute of Technology.

In the program an array of size 15,000 was reserved for β_n's , and the authors tested cases with $s_c \leq 500$ by setting $N = 30$ in most trials. By the experiments it seemed that 30 is sufficiently large for N if $s_c < 100$. The computational data of a trial for the $M/E_5/3$ queueing system is shown in Table 2.

Table 2. Computational data of a trial for the $M/E_5/3$ queueing system

s_c	35
ρ	0.3, 0.6, 0.9
N	30
ε	0.00001
Number of iterations	9 for each ρ
Computational time excluding times for compiling and linkage	20 ~ 22 seconds for each ρ

References

[1] Heffer, J. C., Steady-state solution of the $M/E_k/c$ (∞, *FIFO*) queueing system. *INFOR J. Canadian O. R. S.*, vol. 17 (1969), 16-30.

[2] Mayhugh, J. O., & R. E. McCormik, Steady-state solution of the queue $M/E_k/r$. *Management Science*, vol. 14 (1968), 692-712.

[3] Parzen, E., *Stochastic Processes*, Holden-Day, Inc., San Francisco (1962).

[4] Poyntz, C. D., & R. R. P. Jachson, The steady-state solution for the queueing process $E_n/E_m/r$, *Operational Research Quarterly*, vol. 24 (1973), 615-625.

[5] Shapiro, S., The *m*-server queue with Poisson input and gamma-distributed service of order two. *Operations Research*, vol. 14 (1966), 685-694.

[6] Takahashi, Y., A lumping method for numerical calculations of stationary distributions of Markov chains. *Research Reports on Information Sciences*, B - 18 (1975), Department of Information Sciences, Tokyo Institute of Technology.

(Yukio Takahashi, Faculty of Economics, Tohoku University; Kawauchi Sendai 980, Japan.)