GENERALISED INVARIANT SUBSPACE BASED METHOD FOR STEADY STATE ANALYSIS OF QBD-M PROCESSES

Hung T. TRAN and Tien V. DO

Department of Telecommunications
Budapest University of Technology and Economics
H–1117 Budapest, Pázmány Péter Sétány 1/D, Hungary
e-mail: (hung,do)@hit.bme.hu
Phone: +36 1 463 32 79
Fax: +36 1 463 32 63

Abstract

The paper shows that the theory of generalised invariant subspace and matrix sign function can be applied to computing steady state distribution of QBD-M queueing system. This queueing system is an appropriate and useful modeling tool for system analysis and performance evaluation in computer and telecommunications network field. Moreover, this computational method is evaluated in comparison with other well-known and recently developed methods.

Keywords: QBD-M processes, steady state analysis, numerical computation, performance evaluation.

1. Introduction

The search of numerical solutions for queueing models applied in performance analysis of computer systems and communication networks has practical importance and is always a hot research topic. In recent years, the extensive application of two-dimensional queueing systems which state is described by a phase and a level has been widely witnessed, particularly on the performance evaluation of ATM (Asynchronous Transfer Mode) systems [17]. If the level transitions are only possible between adjacent ones, such queueing systems are called QBD (Quasi Birth-Death) processes.

For the steady state solution of this class of two-dimensional Markov chains, several efficient methods have been developed and improved over recent years [2, 4, 6, 10, 11, 14]. However, there are only few works dealing with such two-dimensional queueing systems in which upper-bounded arrival and/or departure batches occur, i.e. multiple jumps in level dimension are possible. The involvement of this kind of queueing systems is quite reasonable and useful during modeling and performance analysis of a variety of problems arising in telecommunication networks and computer systems field. For example, it may be resorted to during analysis of ATM devices having input and output links of different speeds (ATM concentrator, ATM multiplexer [5, 8]), negotiation of processor scheduling scenario [16], modeling SVC based IP-over-ATM scenario [15], analysis of systems of

processors subject to failures and repairs (examined later in this paper) etc. Thus, the analytical solution of this kind of queueing systems bears sufficient importance from the practical point of view which gives credit to the task of developing its numerical methods.

Being motivated by this observation, our current research focuses on developing a numerical method for steady state solution of queueing systems with batch arrivals and batch departures (referred to as QBD-M processes where the letter M stands for the occurrence of batches). In a certain sense, QBD-M processes can be considered as an extension of QBD processes and therefore with some skillful manipulations (e.g. re-blocking), computation methods developed for QBD processes can be applied. From the technical point of view, direct methods, i.e. methods without any additional manipulation such as re-blocking, may have some advantages over the former ones and that is why it is worth devoting research efforts in this direction.

Towards the aim addressed before, this paper proposes a direct computational method for computing steady state distribution of QBD-M processes. The fundamental material of this method is a theory of generalised invariant subspace in linear algebra, which has been first introduced and applied by Nail AKAR et al. (see [1, 3]) to teletraffic problems. The capability of this method will be discussed and demonstrated through a numerical example and will be compared with other existing computational ones, namely with the method of NAOUMOV et al., the spectral expansion method applied after re-blocking and the iterative method proposed in [18].

The paper is organised as follows. Section 2 is devoted to an overview of invariant subspaces and the relation between invariant subspaces and matrix sign function. Section 3 details basic notations and equations of QBD-M processes and gives a brief review on existing computational methods that have been developed for steady state analysis of QBD-M processes. Section 4 discusses the generalised invariant subspace based (GIS) method. Section 5 contains numerical demonstration of the GIS method and a performance comparison between it and other methods. Finally, some conclusions end the paper.

2. Theory of Invariant Subspace and Matrix Sign Function

2.1. Invariant Subspaces

Let us recall briefly some notations and results related with the theory of invariant subspaces, which will play a key role in the computation method described later in Section 4. The main part of this subsection is from [1]. Let $\mathbf{R}^m$ be the real linear space of column vectors of m real numbers, $\mathbf{R}^{m \times n}$ be the linear space of $m \times n$ matrixes with real entries. A *subspace* is a subset of $\mathbf{R}^m$ that is closed under the operations of addition and scalar multiplication. For arbitrary subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$, $\mathcal{S}_1 \subset \mathcal{S}_2$ denotes either inclusion or equality. If $A \in \mathbf{R}^{m \times n}$, then the image of A is

defined as
$$\text{Im } A = \{x \in \mathbf{R}^m \mid x = Ay \text{ for some } y \in \mathbf{R}^n\}.$$

If $\text{Im } A = \text{Im } B$, then there exists a nonsingular matrix U such that $B = AU$. $A\mathcal{S}$ denotes the image of $\mathcal{S}$ under A.

The set of all eigenvalues of a matrix $A \in \mathbf{R}^{m \times m}$ is called the *spectrum* of A and written $\sigma(A)$. A subspace $\mathcal{S}$ of $\mathbf{R}^m$ is said to be A *invariant* where $A \in \mathbf{R}^{m \times m}$, if $A\mathcal{S} \subset \mathcal{S}$, here $A\mathcal{S} = \{x \in \mathbf{R}^m \mid x = Ay \text{ for some } y \in \mathcal{S}\}$. If a k-dimensional subspace $\mathcal{S}$ is A invariant, then $A S = S A_1$ holds for a $k \times k$ matrix A_1 and an $m \times k$ matrix S whose columns form a basis for $\mathcal{S}$, i.e. $\mathcal{S} = \text{Im } S$.

Let $\mathcal{S} + \mathcal{T}$ and $\mathcal{S} \oplus \mathcal{T}$ be the sum and direct sum, respectively, of the subspaces $\mathcal{S}$ and $\mathcal{T}$. Let $\mathcal{S} \oplus \mathcal{T} = \mathbf{R}^m$ and assume that $\mathcal{S}$ and $\mathcal{T}$ are invariant subspaces of a square matrix A of size m.

Then, $\mathcal{S} = \text{Im } S$ and $\mathcal{T} = \text{Im } T$ and U defined by $U = [\ S \quad T \ ]$ satisfy

$$U^{-1}AU = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}.$$

If $\sigma(A_{11})$ $(\sigma(A_{22}))$ lies in the closed right-half (open left-half) plane, then $\mathcal{S}$ ($\mathcal{T}$) is said to be the right (left) invariant subspace of A. When $\sigma(A_{11})$ $(\sigma(A_{22}))$ lies outside (in) the open unit disk, then $\mathcal{S}$ ($\mathcal{T}$) is called the unstable (stable) invariant subspace of A.

Let us assume a regular matrix pencil $\lambda E - A$, which is a polynomial matrix (in the indeterminate λ) of degree one. The *generalised eigenvalue problem* for the matrixes A and E of size m is equivalent to finding the scalar λ for which the equation $Ax = \lambda Ex$ has solutions $x \neq 0$. Such scalars λ are called generalised eigenvalues. A solution $x \neq 0$ corresponding to an eigenvalue λ is called a generalised eigenvector. A generalised eigenvalue satisfies the relation

$$\lambda \in \sigma(E, A) := \{\mu \in C \mid \det(\mu E - A) = 0\},$$

where $\sigma(E, A)$ denotes the generalised spectrum of the matrix pair (E, A) and C is the field of complex numbers.

Any subspace $\mathcal{S}$ satisfying

$$\mathcal{T} = E\mathcal{S} + A\mathcal{S}, \quad \dim(\mathcal{S}) = \dim(\mathcal{T})$$

is called a *generalised invariant subspace (or deflating subspace)* of the pencil $\lambda E - A$. Note that when $E = I$, we indeed have an ordinary invariant subspace[1].

Now, let $\mathcal{S}$ and $\mathcal{S}_c$ be two complementary deflating subspaces of the pencil $\lambda E - A$, i.e. $\mathcal{S} \oplus \mathcal{S}_c = \mathbf{R}^m$. Define $\mathcal{T} = E\mathcal{S} + A\mathcal{S}$ and $\mathcal{T}_c = E\mathcal{S}_c + A\mathcal{S}_c$. These two subspaces are also complementary [1]. Let $\mathcal{S} = \text{Im } S$, $\mathcal{T} = \text{Im } T$, $\mathcal{S}_c = \text{Im } S_c$, $\mathcal{T}_c = \text{Im } T_c$ then there exists a decomposition

$$U^{-1}EV = \begin{bmatrix} E_{11} & 0 \\ 0 & E_{22} \end{bmatrix}, \qquad U^{-1}AV = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix},$$

---

[1]Here and in what follows I denotes the unit matrix with adequate size.

where

$$U = [\ T \quad T_c\ ], \qquad V = [\ S \quad S_c\ ].$$

If $\sigma(E_{11}, A_{11})$ $(\sigma(E_{22}, A_{22}))$ lies in the closed right-half (open left-half) plane, then $\mathcal{S}$ $(\mathcal{S}_c)$ is called the right (left) deflating subspace of the matrix pencil $\lambda E - A$. When $\sigma(E_{11}, A_{11})$ $(\sigma(E_{22}, A_{22})$ lies outside (in) the open unit disk, then $\mathcal{S}$ $(\mathcal{S}_c)$ is called the unstable (stable) deflating subspace of the matrix pencil $\lambda E - A$.

2.2. Computation of Invariant Subspace via Matrix Sign Function

An invariant subspace of a given matrix can be calculated through its matrix sign function. The definition of a matrix sign function is given as follows.

Definition 1 *Let $X \in \mathbf{R}^{m \times m}$ with no pure imaginary eigenvalue. Let X have a Jordan decomposition $X = T(D + Y)T^{-1}$ where $D = \mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ and Y is nilpotent and commutes with D. Then the matrix sign of X is given by*

$$\begin{aligned} Z &= \mathrm{sgn}(X) \\ &:= T.\,\mathrm{diag}\{\mathrm{sgn}(\lambda_1), \mathrm{sgn}(\lambda_2), \ldots, \mathrm{sgn}(\lambda_m)\}.T^{-1}, \end{aligned}$$

where for a complex scalar z with $Re(z) \neq 0$, the sign of z is defined by

$$\mathrm{sgn}(z) = \begin{cases} 1 & \text{if } Re(z) > 0 \\ -1 & \text{if } Re(z) < 0 \end{cases}.$$

From our aspect, the most important property of $Z = \mathrm{sgn}(X)$ is that $\mathrm{Im}(Z - I)$ and $\mathrm{Im}(Z + I)$ yield the left and right invariant subspace of X, respectively. This property means that an orthogonal basis for the left (right) invariant subspace of X which has a dimension r is given by the first r columns of the orthogonal matrix in a rank-revealing QR decomposition of matrix $Z - I$ $(Z + I)$.

The matrix sign function can be computed efficiently in several ways. In this paper a scaling Newton's scheme with Byers scalar shown in *Fig. 1* is adopted. It has been verified that this iterative algorithm converges quadratically for all matrix X for which the matrix sign is well defined

$$\begin{aligned} &k = 0 \\ &Z_0 = X \\ &DO \\ &\qquad t = |\det Z_k|^{-1/m} \\ &\qquad Z_{k+1} = \tfrac{1}{2}(t Z_k + t^{-1} Z_k^{-1}) \\ &\qquad k = k + 1 \\ &WHILE\ (||Z_k - Z_{k-1}||_1 \geq \epsilon ||Z_{k-1}||_1) \end{aligned}$$

Fig. 1. The iterative procedure to obtain matrix sign function

3. QBD-M Processes and Existing Methods for their Steady State Analysis

Consider a discrete-time two dimensional Markov process. Each state of the process is denoted by two integer valued random variables I_n and J_n. I_n is taking a finite set of values $\{0, 1, \ldots N\}$ and J_n is taking an infinite set of values $\{0, 1, \ldots\}$. The states with the same value of J_n ($J_n = j$) define a level (j), whereas the states with the same value of I_n ($I_n = i$) define a phase (i). Multiple jumps in level dimension are possible but are required to be upper-bounded. The possible transition probabilities underlying this Markov process are given by the following matrixes

- A_j: purely phase transitions – From state (i, j) to state (k, j) $(0 \leq i, k \leq N; i \neq k; j = 0, 1, \ldots)$
- $B_{j,s}$: bounded $s-$step upward transitions–From state (i, j) to state $(k, j+s)$ $(0 \leq i, k \leq N; 1 \leq s \leq y_1; y_1 \geq 1; j = 0, 1, \ldots)$
- $C_{j,s}$: bounded $s-$step downward transitions–From state (i, j) to state $(k, j - s)$ $(0 \leq i, k \leq N; s \leq j; 1 \leq s \leq y_2; y_2 \geq 1; j = 0, 1, \ldots)$. If $j < s$ then $C_{j,s} = 0$.

We assume there is a threshold M, $M \geq y_1$, such that:

$$A_j = A, j \geq M; \quad B_{j,s} = B_s, j \geq M - y_1; \quad C_{j,s} = C_s, j \geq M. \tag{1}$$

Recall that the well-known QBD (Quasi Birth Death) processes possess similar properties as the Markovian process described above, except that jumps in level dimension are allowed only between adjacent ones in QBD's case. Thus, from the aspect regarding level changes, the process considered so far is the extension of QBD process. Therefore, in what follows this process is referred to as QBD-M process, where the letter M stands for multiple jumps in level dimension. A QBD-M process has the transition probability matrix of the form shown in *Fig. 2*.

$$
P = \left[
\begin{array}{ccc|ccc|ccc|cc}
A_0 & B_{0,1} & B_{0,2} & B_{0,3} & & & & & & & \\
C_{1,1} & A_1 & B_{1,1} & B_{1,2} & B_{1,3} & & & & & & \\
C_{2,2} & C_{2,1} & A_2 & B_{2,1} & B_{2,2} & B_{2,3} & & & & & \\
\hline
 & C_{3,2} & C_{3,1} & A_3 & B_{3,1} & B_{3,2} & B_3 & & & & \\
 & & C_{3,2} & C_{4,1} & A_4 & B_{4,1} & B_2 & B_3 & & & \\
 & & C_{5,2} & C_{5,1} & A_5 & B_1 & B_2 & B_3 & & & \\
\hline
 & & & C_{6,2} & C_{6,1} & A & B_1 & B_2 & B_3 & & \\
 & & & & C_{7,2} & C_1 & A & B_1 & B_2 & B_3 & \\
 & & & & & C_2 & C_1 & A & B_1 & B_2 & \\
\hline
 & & & & & & C_2 & C_1 & A & B_1 & \\
 & & & & & & & \ddots & \ddots & \ddots & \ddots \\
\end{array}
\right]
$$

Fig. 2. The block structure of the transition probability matrix of a QBD-M process with parameters $y_1 = 3$, $y_2 = 2$, $M = 6$

The main task now is to determine the steady state probability $p_{i,j} = \lim_{n \to \infty} P$ $(I_n = i, J_n = j)$ of the state (i, j) in terms of the known parameters of the process. It is mathematically convenient to introduce the row vectors (called level probability vectors) $\underline{v}_j$ as

$$\underline{v}_j = (p_{0,j}, p_{1,j}, \dots, p_{N,j}) \text{ for } j = 0, 1, \dots \tag{2}$$

For $j = 0, 1, \dots, M - 1$, the balance equations of the system are:

$$\underline{v}_j = \sum_{s=1}^{y_1} \underline{v}_{j-s} B_{j-s,s} + \underline{v}_j A_j + \sum_{s=1}^{y_2} \underline{v}_{j+s} C_{j+s,s}. \tag{3}$$

(It is assumed $\underline{v}_{j-s} = \underline{0}$ if $j < s$). For $j \geq M$, the corresponding j-independent set is:

$$\underline{v}_j = \sum_{s=1}^{y_1} \underline{v}_{j-s} B_s + \underline{v}_j A + \sum_{s=1}^{y_2} \underline{v}_{j+s} C_s. \tag{4}$$

Let $\underline{e}$ be a column vector with all entries equal to 1. In addition, since the sum of all probabilities must be one, we have:

$$\sum_{j=0}^{\infty} \underline{v}_j \underline{e} = 1.0. \tag{5}$$

Eq. (4) can be rewritten as a vector difference equation with constant coefficients, the order of which is $y = y_1 + y_2$:

$$\sum_{k=0}^{y} \underline{v}_{j+k} Q_k = \underline{v}_{j+y_1}, \qquad j \geq M - y_1, \tag{6}$$

where

$$Q_k = \begin{cases} B_{y_1-k} & \text{for } k = 0, 1, \dots, y_1 - 1, \\ A & \text{for } k = y_1, \\ C_{k-y_1} & \text{for } k = y_1 + 1, y_1 + 2, \dots, y_1 + y_2. \end{cases} \tag{7}$$

Available methods for steady state solution of QBD-M processes fall into two categories depending on whether the calculation of level probability vectors takes place in a direct or indirect way. The most well-known indirect solution way is to perform re-blocking (as shown in *Fig.* 2) in the transition probability matrix of the QBD-M process to get a standard QBD process, then solving this transformed QBD process by one of the efficient methods available in literature [2, 4, 6, 11]. Along this line, the method of NAOUMOV et al. [11] and the spectral expansion method [6] are most appealing to use due to their fast and accurate performance. We will refer to these two methods as NA after BL and SE after BL method, respectively. The direct solution way is represented by the spectral expansion method, that is

capable to solve QBD-M processes without re-blocking. However, our numerical experiments do not encourage the use of this direct method, because it is sometimes inclined to fail in ill-condition troubles.

Recently, a new method has been proposed in [18]. This method makes use of re-blocking technique and special relations between level probability vectors to reduce calculation efforts. An iterative procedure has been provided for calculation of the reduced part of the so called rate matrix. This method is referred to as ITE method. Theoretically, it has been proved that regarding each iterative step, the complexity of the ITE method is less than the complexity of matrix geometry based methods (one of them is Naoumov's method itself) applied after re-blocking. Moreover, its gain is also offered by a favourable storage requirement that is definitely less than the case of matrix geometry based methods such as the NA after BL method.

In the next section, we will propose a direct computational method for steady state solution of QBD-M processes. The method is based on the theory of generalised invariant subspaces and matrix sign function background described earlier in Section 2.

4. Generalised Invariant Subspace Based Method

4.1. Formal Description

Let us construct the following matrix polynomials and hypermatrixes

$$Q(\lambda) = Q_0 + Q_1\lambda + Q_2\lambda^2 + \ldots + Q_y\lambda^y, \tag{8}$$

$$D(\lambda) = Q(\lambda) - \lambda^{y_1}.I = D_0 + D_1\lambda + \ldots + D_y\lambda^y, \tag{9}$$

$$G = \begin{bmatrix} 0 & 0 & \cdots & \cdots & -D_0 \\ I & 0 & \cdots & \cdots & -D_1 \\ 0 & I & \cdots & \cdots & -D_2 \\ \vdots & \vdots & \ddots & & \\ 0 & 0 & \cdots & I & -D_{y-1} \end{bmatrix},$$

$$E = \begin{bmatrix} I & & & & \\ & I & & & \\ & & \ddots & & \\ & & & I & \\ & & & & D_y \end{bmatrix}, \quad T = \begin{bmatrix} I \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \tag{10}$$

By introducing the row vectors

$$\underline{w}_k = [\ \underline{v}_{k+(M-y_1)} \quad \cdots \quad \underline{v}_{k+M} \quad \cdots \quad \underline{v}_{k+(M+y_2-1)} \ ] \quad \text{for } k \geq 0 \tag{11}$$

and by algebraic manipulations one can easily show that

$$\underline{w}_{k+1} E = \underline{w}_k G \quad k \geq 0, \tag{12}$$

$$\underline{v}_{k+(M-y_1)} = \underline{w}_k T \quad k \geq 0. \tag{13}$$

Note that G, E have size $y * (N + 1) \times y * (N + 1)$, $\underline{w}_k$ has size $y * (N + 1)$.

The main observation is the singularities of the matrix pencil $\lambda E - G$, i.e. the roots of $\det(\lambda E - G) = 0$ are exactly the roots of the equation $\det(D(\lambda)) = 0$. In [6], an excellent discussion shows that when the system is stable, the equation $\det(D(\lambda)) = 0$ has exactly $y_1 * (N + 1)$ roots in the open unit disk and $y_2 * (N + 1)$ roots outside the unit disk (including the one at $\lambda = 1$). Consequently, if the system is stable then the matrix pencil $\lambda E - G$ has $m_u = y_2 * (N + 1)$ singularities outside the unit disk (including the one at $\lambda = 1$) and $m_s = y_1 * (N + 1)$ singularities in the open unit disk. Let $m = m_u + m_s = y * (N + 1)$. Let $\mathcal{V}_1$ and $\mathcal{V}_2$ be the unstable and stable deflating subspaces of the pencil $\lambda E - G$, respectively. Let $\mathcal{V}_1 = \text{Im } V_1$ and $\mathcal{V}_2 = \text{Im } V_2$ for some matrixes V_1 and V_2 of size $m \times m_u$ and $m \times m_s$, respectively. Also let $\mathcal{U}_1 := E\mathcal{V}_1 + G\mathcal{V}_1 = \text{Im } U_1$ and $\mathcal{U}_2 := E\mathcal{V}_2 + G\mathcal{V}_2 = \text{Im } U_2$ for some matrixes U_1 and U_2 of size $m \times m_u$ and $m \times m_s$, respectively. Define

$$U = [\begin{array}{cc} U_1 & U_2 \end{array}] \text{ and } V = [\begin{array}{cc} V_1 & V_2 \end{array}] \tag{14}$$

then from the theory of generalised invariant subspace presented in Section 2 we have

$$U^{-1} E V = \left[\begin{array}{cc} E_{11} & 0 \\ 0 & E_{22} \end{array}\right] \text{ and } U^{-1} G V = \left[\begin{array}{cc} G_{11} & 0 \\ 0 & G_{22} \end{array}\right] \tag{15}$$

and $\sigma(E_{11}, G_{11})$ and $\sigma(E_{22}, G_{22})$ lie outside and in the open unit disk, respectively. Defining

$$[\begin{array}{cc} \underline{p}_k & \underline{q}_k \end{array}] = \underline{w}_k [\begin{array}{cc} U_1 & U_2 \end{array}]$$

and post-multiplying the model (12) by V, we have two un-couple generalised difference equations for $\underline{p}_k$ and $\underline{q}_k$

$$\underline{p}_{k+1} E_{11} = \underline{p}_k G_{11} \quad k \geq 0, \tag{16}$$

$$\underline{q}_{k+1} E_{22} = \underline{q}_k G_{22} \quad k \geq 0. \tag{17}$$

Since the system is considered under the condition of stability, $\underline{w}_k$ must not be divergent as $k \to \infty$, which is fulfilled only if

$$\underline{p}_0 = \underline{w}_0 U_1 = 0. \tag{18}$$

Moreover, since $\sigma(E_{22}, G_{22})$ lie in the open unit disk, E_{22} is nonsingular and we can write

$$\underline{q}_k = \underline{q}_0 F^k, \tag{19}$$

where the $m_s \times m_s$ matrix F is found as

$$F = G_{22} E_{22}^{-1}. \tag{20}$$

Let us partition U^{-1} as

$$U^{-1} = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix}, \tag{21}$$

where the size of L_1 and L_2 is $m_u \times m$ and $m_s \times m$, respectively, then

$$\underline{v}_{k+(M-y_1)} = \underline{w}_k T = (\underline{p}_k L_1 + \underline{q}_k L_2)T = \underline{w}_0 U_2 F^k L_2 T \quad k \geq 0. \tag{22}$$

By this equation we have derived the geometric form of the probability vector

$$\underline{v}_{k+(M-y_1)} = \underline{g} F^k H \text{ with } k \geq 0 \quad \underline{g} = \underline{w}_0 U_2 \text{ and } H = L_2 T. \tag{23}$$

Eq. (23) reveals the fact that all vectors $\underline{v}_j$ ($j \geq M - y_1$) can be expressed in terms of $\underline{v}_{M-y_1}, \ldots, \underline{v}_M, \ldots, \underline{v}_{M+y_2-1}$. Note that *Eqs.* (3) with $0 \leq j \leq M - 1$ and *Eqs.* (18) then form the set of linear equations in which the number of unknowns is $(M + y_2) * (N + 1)$. The number of equations is the same, but among them there are only $(M + y_2) * (N + 1) - 1$ linearly independent ones. That means we have to replace one equation by the normalized *Eq.* (5) for getting a set of linearly independent equations. Using (23), it can be easily shown that the equivalent form of (5) is

$$\sum_{j=0}^{M-y_1-1} \underline{v}_j \underline{e} + \underline{w}_0 U_2 (I - F)^{-1} L_2 T \underline{e} = 1. \tag{24}$$

Solving this set of equations one gets the probability vectors $\underline{v}_0, \ldots, \underline{v}_{M-y_1}$, $\ldots, \underline{v}_{M+y_2-1}$, based on which $\underline{w}_0$ and through it the rest $\underline{v}_j$ ($j \geq M - y_1$) can be calculated.

4.2. *Computational Algorithm*

Based on the previous section, one can see that the main task is to find bases for the unstable and stable deflating subspaces of the pencil $\lambda E - G$, leading to construction of the matrixes U and V defined in (14). Following the same way in [1], we note that the stable (unstable) subspaces of the matrix pencil $\lambda E - G$ are equal to the left (right) deflating subspaces of the pencil $\lambda X - Y$, where the two matrixes X and Y are defined as

$$X = G + E, \quad Y = G - E.$$

With this transformation, the generalised eigenvalues of the pencil $\lambda E - G$ in (outside) the unit disk are moved to the open left-half (closed right-half) plane. There is one generalised eigenvalue of $\lambda X - Y$ at the origin.

Since $\lambda E - G$ does not have any generalised eigenvalue at $\lambda = -1$, the matrix X is nonsingular and we can define $Z = X^{-1}Y$. The left (right) invariant subspace of Z is equal to the left (right) deflating subspace of the pencil $\lambda X - Y$. One now may make use of the matrix sign function to compute the left and right invariant subspace of Z. However, Z has one eigenvalue on the imaginary axis, at

the origin because of which the matrix sign function cannot be directly applied and therefore further transformation is needed. Let $\underline{\gamma}$, $\underline{\mu}$ be left and right eigenvectors of Z corresponding to the eigenvalues at the origin, i.e.

$$\underline{\gamma} Z = 0, \quad Z \underline{\mu} = 0, \tag{25}$$

then the matrix Z_e defined as

$$Z_e = Z + \frac{\underline{\mu} \cdot \underline{\gamma}}{\underline{\gamma} \cdot \underline{\mu}} \tag{26}$$

is free of imaginary-axis eigenvalues, and the left (right) invariant subspace of Z_e is equal to the left (right) invariant subspace of Z. It is not difficult to show that the vectors $\underline{\gamma}$ and $\underline{\mu}$ defined as

$$\underline{\gamma} = [\ \underline{\pi} \quad \underline{\pi} \quad \dots \quad \underline{\pi}\ ], \quad \underline{\mu} = \begin{bmatrix} \underline{\mu}_0 \\ \underline{\mu}_1 \\ \vdots \\ \underline{\mu}_{y-1} \end{bmatrix}, \tag{27}$$

where $\underline{\pi}$ is the stationary probability vector of $Q(1)$, i.e. $\underline{\pi} Q(1) = \underline{\pi}, \underline{\pi} e = 1$ and

$$\underline{\mu}_0 = D_0 \underline{e}, \quad \underline{\mu}_i = \underline{\mu}_{i-1} - D_i \underline{e} \quad \text{for } 1 \leq i \leq y - 2, \quad \underline{\mu}_{y-1} = \underline{e}$$

satisfy (25).

Using matrix-sign function iterations on Z_e to find bases for the unstable and stable deflating subspaces of the pencil $\lambda E - G$ leads to construction of the matrixes U and V defined as in (14). The algorithm of step-by-step computation can be summarized as follows

1. Define $D(\lambda)$ as in (9).
2. Define the matrixes G, E, T as in (10).
3. Define $Z = (E + G)^{-1}(G - E)$, $\underline{\gamma}$, $\underline{\mu}$ as in (27) and Z_e as in (26). Then find $S = sign(Z_e)$ by the quadratically convergent iteration shown in *Fig. 1*.
4. • Construct matrix Q_r in the rank-revealing QR decomposition $S + I = Q_r R_r \Pi_r$ and define

    $$V_1 = leading\ m_u\ columns\ of\ Q_r$$

   • Construct matrix Q_l in the rank-revealing QR decomposition $S - I = Q_l R_l \Pi_l$ and define

    $$V_2 = leading\ m_s\ columns\ of\ Q_l$$

   • Construct matrix $\widehat{Q}_r$ in the rank-revealing QR decomposition $[\ EV_1 \quad GV_1\ ] = \widehat{Q}_r \widehat{R}_r \widehat{\Pi}_r$ and define

    $$U_1 = leading\ m_u\ columns\ of\ \widehat{Q}_r$$

- Construct matrix $\widehat{Q}_l$ in the rank-revealing QR decomposition
  $[\; E V_2 \quad G V_2 \;] = \widehat{Q}_l \widehat{R}_l \widehat{\Pi}_l$ and define

$$U_2 = \text{leading } m_s \text{ columns of } \widehat{Q}_l$$

5. Define U, V as in (14) and let E_{22} and G_{22} be the lower right $m_s \times m_s$ blocks of $U^{-1} E V$ and $U^{-1} G V$ as in (15), respectively. Then define F as in (20).
6. Solve a set of linearly independent equations for $\underline{v}_j$, $0 \le j \le M - y_1 - 1$ and $\underline{w}_0$.
7. Define L_2 as in (21) and also

$$\underline{g} = \underline{w}_0 U_2 \text{ and } H = L_2 T$$

to obtain the matrix-geometric expression $\underline{v}_{k+(M-y_1)} = \underline{g} F^k H$ for $k \ge 0$.

5. Comparative Performance Study

In order to evaluate the generalised invariant subspace based method delineated above in comparison with existing other ones, the system of homogeneous processors with repairs and breakdowns is chosen to be negotiated. Batch arrivals are in presence by considering MMCPP (Markov Modulated Composed Poisson Process) job arrival process. Batch departures occur due to GE (Generalised Exponential) service time of each processor. The system is modeled by a QBD-M process. The detailed description of this system is found in Appendix A. For determining steady state probabilities, four computational ways were implemented. They are:

- *ITE method*: Using an iterative method developed in [18];
- *NA after BL method*: Using the method of Naoumov et al. [11] after reblocking to get a standard QBD process;
- *SE after BL method*: Using the spectral expansion method [6] after reblocking;
- *GIS method*: Using generalised invariant subspace based method.

All methods have been implemented in C using the Meschach library for matrix operations[2]. All program running and the CPU time measurements have been performed on IBM RISC system/6000 570 station. In all scenarios, the accuracy for stopping the iterative procedures was set to be $1e - 9$. The computation time was measured until the point at which the program is able to compute the level probability vectors. That is in the case of the GIS method, time is stopped after step 5.

Fig. 3 illustrates the effect of system load on computation time. The system load is required not to be greater than 1 in order to keep the infinite system stable. The

---

[2] Meschach library for matrix computation is developed at the School of Mathematical Sciences, Australian National University by David E. Stewart and Zbigniew Leyk and it is available via netlib (ftp.netlib.org/c/meschach).
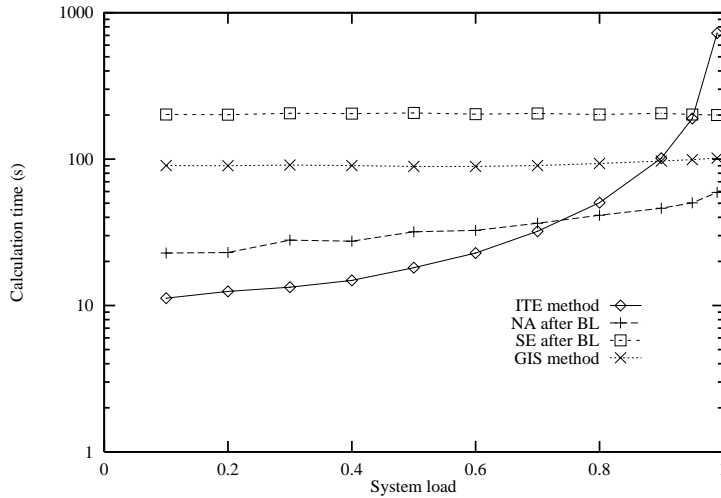
Fig. 3. Computation time versus system load ($N = 30$, $y_1 = 7$, $y_2 = 2$)

curve shows that the GIS method is robust to the traffic load. Its computation time remains constant while varying the traffic load. In contrast, execution time of the ITE method and the NA after BL method exhibit strong and slightly load-dependent nature, respectively. From this point of view, the generalised invariant subspace based method can be ranked among the best methods ranging with the spectral expansion method. However, the GIS method is definitely better to use compared to the SE method due to two aspects. First, the GIS method is faster than the SE after BL method, as shown in *Fig. 3*. Secondly, the GIS method avoids the procedure of computing all relevant eigenvalues and associated eigenvectors, which is compulsory to be done in the case of SE after BL method. Therefore, numerical problems related to ill-conditions such as too close eigenvalues and the computational difficulty deriving from the presence of complex eigenvalues and eigenvectors are no longer imposed.

Fig. 4 shows the computational time as a function of the maximum size of arrival batch (y_1), while the maximum size of departure batches (y_2) is fixed. One can observe from *Fig. 3* and *Fig. 4* that the GIS method has larger computation time compared to the NA after BL method in all reported cases, but in general it is faster than the SE after BL and than the ITE method as the saturation point is approached (i.e. the system load tends to 1).

For a deeper insight into the operation of the GIS method, we take a closer look at the execution time of each step of the step-by-step algorithm presented in Section 4.2. It is expected that the major computation time will be devoted to the iterative procedure in step 3 and four QR rank revealing operations in step 4, plus the time of matrix multiplications and inversion in step 5. In fact, numerical results show that the iterative procedure for matrix sign function converges quite fast after
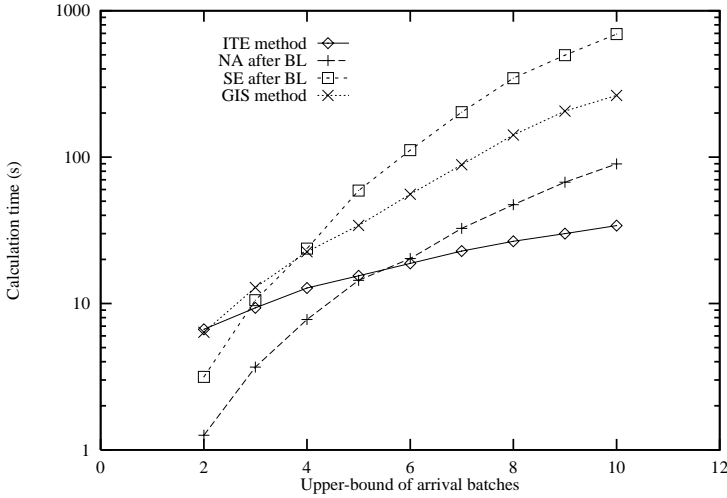
Fig. 4. Computation time versus upper bound of arrival batches ($N = 30$, $y_2 = 2$, $load = 0.6$)

a few iterations. The time required for iterative computation in step 3 is nearly the same as the time needed for QR operations in step 4. This tendency is valid over all the sets of system parameters. This situation is demonstrated in *Fig. 5*, where the individual time of each step is depicted for different system loads. The figures also give explanation about the load-insensitive nature of the algorithm. It is observable from the figures that the system load only influences the time of step 3, but that effect is indeed very small. Step 4 and step 5 are not affected by the system load at all. The time of step 5 in all cases only takes about 10 percent of the total amount of time needed.
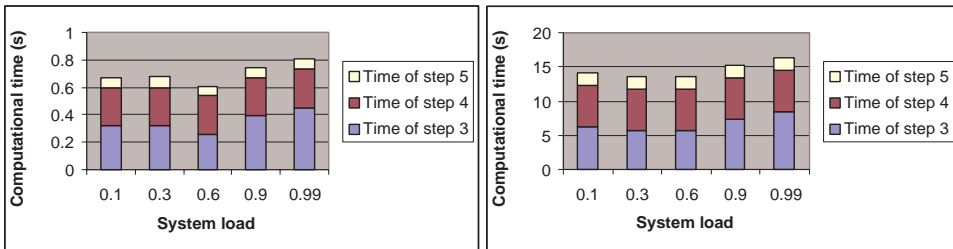

Fig. 5. Contribution of each step in the GIS method to the total computational time ($y_1 = 3$, $y_2 = 2$, $N = 10, 30$)

For the accuracy of computation of each method we check the infinity norm of the residual of the solution vector $||\underline{v} - \underline{v}P||_\infty$, where $\underline{v} = (\underline{v}_0, \underline{v}_1, \ldots, \underline{v}_M, \ldots)$, P is the transition probability matrix of the QBD-M process. Let us denote $\underline{v}P = (\underline{x}_0, \underline{x}_1, \ldots, \underline{x}_M, \ldots)$. Since both $\underline{v}$ and $\underline{v}P$ are of infinite size, we only take their first $M+1$ sub-vectors into account and compute $max_j ||\underline{v}_j - \underline{x}_j||_\infty$ for $0 \leq j \leq M$. From *Table 1* and *Table 2*, it turns out that the residual error of the GIS method is slightly increasing when a system load is approaching 1. However, compared to the other algorithms, in a wide range of system parameters, the GIS method has the smallest residual error. It is also noteworthy that unlike the methods applied after re-blocking (NA and SE after BL), the achieved accuracy of the GIS method does not suffer deterioration observed in some cases (see *Table 2* for $y_2 \geq 6$ cases) and only exhibits negligible fluctuations at varying system parameters.

Table 1. Computation error versus system load ($N = 30$, $y_1 = 7$, $y_2 = 2$)

Load	NA after BL	SE after BL	ITE	GIS
0.1	8.161874e-16	5.987853e-09	1.837527e-14	1.955901e-16
0.2	1.395965e-15	7.346733e-14	1.089881e-12	2.769594e-16
0.3	4.025900e-15	8.085196e-08	9.767948e-12	1.628065e-16
0.4	3.331434e-15	1.956147e-12	3.687938e-11	1.037581e-16
0.5	1.380291e-15	1.819238e-08	7.485544e-11	1.563794e-16
0.6	2.395972e-15	4.972458e-13	1.240864e-10	1.173762e-16
0.7	5.839951e-16	8.120562e-11	1.882883e-10	2.880204e-16
0.8	1.097247e-15	2.355659e-11	2.463647e-10	4.775837e-16
0.9	8.208073e-16	6.714674e-10	3.119365e-10	2.224955e-15
0.95	7.823879e-16	2.977887e-12	3.473892e-10	1.787801e-15
0.99	6.468402e-16	3.345734e-06	3.773965e-10	2.866264e-14

Another benefit of the GIS method relies on the form (23). Using that form of level probability vectors, one has no difficulty to provide closed form expressions for performance measures by which easy implementation of computation is available. For example, by some simple algebraic manipulations, the average number of jobs in the systems can be expressed as

$$K_{avg} = \sum_{j=0}^{M-y_1-1} j\underline{v}_j\underline{e} + \underline{w}_0 U_2 \left[(I - F)^{-2} + (M - y_1 - 1)(I - F)^{-1} \right] L_2 T\underline{e}. \quad (28)$$

For the sake of completeness, the comparative results are summarized in *Table 3*.

Table 2. Computation error versus maximum departure batch size ($N = 30, y_1 = 10, load = 0.6$)

y_2	NA after BL	SE after BL	ITE	GIS
1	1.496724e-15	7.008743e-05	1.304302e-10	3.930233e-17
2	1.179457e-15	7.282278e-14	2.084226e-10	1.087955e-16
3	3.816726e-15	2.170779e-15	1.839186e-10	6.130667e-16
4	1.120249e-15	2.443699e-13	1.656975e-10	1.789820e-16
5	1.319358e-15	9.813286e-14	1.609332e-10	2.146837e-16
6	4.780242e-04	4.780242e-04	1.478210e-10	8.572997e-16
7	9.414644e-04	9.414644e-04	1.434201e-10	3.099843e-16
8	1.880715e-03	1.880715e-03	1.348634e-10	4.624654e-16
9	3.764807e-03	3.764807e-03	1.283634e-10	4.950161e-16

Table 3. Summary of comparative results

ASPECTS\METHOD	GIS	ITE	NA after BL	SE after BL
Robustness to system load	yes	no	yes, but not strictly	yes
Time complexity	moderately good	the fastest under certain conditions	good	the most time consuming
Numerical accuracy	the best	acceptably good	good	good

6. Conclusions

The paper has two main contributions. First, we have revealed that beyond the results of previous work published so far in [1, 3], the application range of the theory of generalised invariant subspaces can also be extended to steady state analysis of such queueing models, in which batch arrivals and batch departures with arbitrary size below a given threshold occur. Such queueing systems (referred to as QBD-M processes) have a wide range of applicability. They are frequently arising in modeling and performance analysis of telecommunication networks and computer systems, therefore the computational method we have proposed for their steady state solution indeed deserves practical credit.

The second main contribution of the paper is twofold. The application of QBD-M model to a system of homogeneous processors subject to breakdowns and repairs, with bursty arrivals and bursty departures of jobs has been justified. Moreover, performance comparison between the proposed computational method

(abbreviated as GIS) and other ones developed for QBD-M process has been carried out through this case study.

Based on our negotiation, the advantages of the GIS method can be summarized as follows:

- It is numerically stable and robust to traffic load.
- The GIS method is more advisable to use than the spectral expansion method not only because it performs faster but also because it avoids the calculation of eigenvalues and associated eigenvectors, therefore contingent numerical problems are no longer faced.
- In contrast with the new iterative method proposed in [18], the GIS method does not suffer any restriction stemming from the relation between upper-bounds of arrival and departure batches.
- In most cases, the proposed method produces results with the smallest computational error.
- The use of the proposed method makes it easy to give a closed form expression for performance measures.

Although the proposed method is not considered as the fastest one, the aforementioned benefits will make its utilization and implementation reasonable for numerical analysis of a wide range of problems.

A. A Case Study – Multiprocessor System with Breakdowns and Repairs

We take the system of homogeneous processors whose number is N. The processors break down from time to time. Single and independent failures of processors, as well as multiple and simultaneous failures are possible. Failed processor returns to operative state after successful repair. Single and independent repairs of processors, as well as multiple and simultaneous repairs are possible. Failure and repair times are assumed to be exponentially distributed. The inter-arrival time of job arrivals, as well as the service time of each processor is assumed to have GE distribution. The system has a buffer of infinite size.

This system is modeled by a continuous time, two dimensional Markov process. At any time t, the state of the system is denoted by a couple of integers $I(t), J(t)$ in the following way:

- $I(t)$ is the operative state of the system, representing the number of operative processors at time t, $I(t) = 1, \ldots, N$.
- $J(t)$ is the number of jobs in the system at time t, $J(t) = 0, 1, \ldots$

First of all, we construct the generator matrix of this process. Similar to the discrete case, let us introduce the following transition matrixes

- A_j^*: purely phase transitions – From state (i, j) to state (k, j) $(0 \leq i, k \leq N; i \neq k; j = 0, 1, \ldots)$

- $B_{j,s}^*$: bounded s−step upward transitions–From state (i, j) to state $(k, j+s)$ $(0 \le i, k \le N; 1 \le s \le y_1; y_1 \ge 1; j = 0, 1, \ldots)$
- $C_{j,s}^*$: bounded s−step downward transitions–From state (i, j) to state $(k, j - s)$ $(0 \le i, k \le N; s \le j; 1 \le s \le y_2; y_2 \ge 1; j = 0, 1, \ldots)$. If $j < s$ then $C_{j,s} = 0$.

As one can see, we have assumed that the upward and downward transitions in level dimension j are not only one step, but may be up to y_1 and y_2 steps. Note, that there will be a boundary M, above which the transition matrixes become level-independent.

The task now is building up the transition matrixes (A^*, B^* and C^*). Let us notice that when a new batch arrives or when a completed batch departs from the system, the operative state does not change, unless there is an independent coincidence towards such a change. Hence, change in the operative state of the system is reflected only in the matrixes A^* and A_j^*.

A.1. Building up A* Matrixes

As we mentioned before, there are two kinds of possible failures and repairs.

- The individual processors break down independently at rate ξ and are repaired independently at rate η.
- The global simultaneous breakdowns of all currently operative processors occur at rate ξ_0 and the global simultaneous repairs of all currently inoperative processors occur at rate η_N.

The A_j^* matrixes are j-independent and are given by:

$$A^* = A_j^*(j = 0, 1, \ldots) = \begin{bmatrix} 0 & N\eta & & & & \eta_N \\ \xi_0 + \xi & 0 & (N-1)\eta & & & \eta_N \\ \xi_0 & 2\xi & 0 & & & \eta_N \\ & & & \ddots & & \\ & & & & & \eta + \eta_N \\ \xi_0 & & & & N\xi & 0 \end{bmatrix}$$
(29)

For numerical negotiation, the concrete values were chosen as $\xi = \xi_0 = 0.05$, $\eta = \eta_N = 0.1$.

A.2. Building up B* Matrixes

The arrival process is assumed to be an N phase MMCPP (Markov Modulated Composed Poisson Process) with (σ_i, θ_i) parameters relating to GE inter-arrival time distribution in phase i. Due to the interpretation of GE (see [12]), the arrival

point process is Poisson with batches arriving at each point having geometric size. The probability that a batch has size s in phase i is given by $(1 - \theta_i)\theta_i^{s-1}$.

Furthermore, suppose that when the operative state $I(t) = i$, the arrival process is in phase i. The B^* matrixes are now ready to be written

$$B_{j,s}^* = \operatorname{diag}[\sigma_0, \sigma_1, \ldots, \sigma_N] * P(\text{batch size} = s). \tag{30}$$

Using the GE distribution and limiting the maximum batch size to y_1, the distribution of batch size in phase i is defined by

$$P(\text{batch size}=s) = \begin{cases} (1 - \theta_i)\theta_i^{s-1} & 1 \leq s \leq y_1 - 1, \\ \theta_i^{y_1-1} & s = y_1. \end{cases} \tag{31}$$

It follows

$$B_{j,s}^* = \operatorname{diag}[(1 - \theta_0)\theta_0^{s-1}\sigma_0, (1 - \theta_1)\theta_1^{s-1}\sigma_1, \ldots, (1 - \theta_N)\theta_N^{s-1}\sigma_N]$$
$$\text{if} \quad 1 \leq s \leq y_1 - 1 \tag{32}$$

and

$$B_{j,s}^* = \operatorname{diag}[\theta_0^{y_1-1}\sigma_0, \theta_1^{y_1-1}\sigma_1, \ldots, \theta_N^{y_1-1}\sigma_N] \quad \text{if} \quad s = y_1. \tag{33}$$

For numerical negotiation all $\sigma_i = \sigma$, $\theta_i = \{0.1 * i\}$ where $\{x\}$ operator means a fraction of x.

A.3. Building up C^* Matrixes

Let us assume that each operative processor has GE service time with parameters (μ, ϕ). The actual departure rate depends on the actual state $I(t) = i$, $J(t) = j$ and that is given by the entry $C_{j,s}^*(i, i)$ of matrix $C_{j,s}^*$. Similarly to [7], the following remarks are considered:

- The batch size associated with a service completion is bounded by one more than the number of jobs waiting to commence service at the departure instant.
- Therefore if $j > i$ ($j =$ number of jobs, $i =$ number of operative processors), the maximum batch size is $j - i + 1$. Combined with the assumption that the maximum of departure batch size is y_2, we get the form

$$P(\text{batch size}=s) = \begin{cases} (1 - \phi)\phi^{s-1} & 1 \leq s \leq \min(j - i + 1, y_2) - 1, \\ \phi^{\min(j-i+1,y_2)-1} & s = \min(j - i + 1, y_2). \end{cases} \tag{34}$$

  Furthermore, in this case the service rate relating with batch departure is $i.\mu$.
- For $j \leq i$, since there is no job waiting to be serviced the departing batch has size 1 with probability one. That is

$$P(\text{batch size} = s) = \begin{cases} 1 & \text{if } s = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{35}$$

In this case the service rate relating with batch departure (with size one) is $j.\mu$.

Taking all the aforementioned aspects into consideration we arrive at

$$C_{j,s}^* = \text{diag}\,[0, \min(j, 1).\mu, \ldots, \min(j, N).\mu] * P(\text{batch size} = s). \qquad (36)$$

Note that $C_{j,s}^*(i, i)$ does not depend on j if

- $j \geq i$ (note that the maximum value of i is N)
- the term $P(\text{batch size} = s)$ does not depend on j, which is fulfilled when $j - i + 1 \geq y_2$.

The first condition is fulfilled if $j \geq N$, the latter one is true when $j \geq y_2 + N - 1$. That is for $j \geq N + y_2 - 1$

$$C_{j,s}^* = C_s^* = \text{diag}\,[0, \mu, \ldots, N.\mu] * P(\text{batch size} = s) \quad 1 \leq s \leq y_2 \qquad (37)$$

is j-independent. Taking this point into account it follows that the threshold M is given by $M = N + y_2 - 1$.

The choice for numerical test is $\mu = 10/N$ and $\phi = 0.5$.

Based on matrixes A^*, B^*, C^*, the generator matrix of the continuous time QBD-M process can be easily defined. Using the transformation delineated in [13] we obtain the embedded discrete time QBD-M process which has the same steady state distribution as the original continuous one and for which the computational methods have been applied.

References

[1] AKAR, Nail – CEM OGUZ, Nihat – SOHRABY, Khosrow, Matrix-geometric Solution of M/G/1 Type Markov Chains: A Unifying Generalized State-space Approach, *IEEE JSAC*, **16** (1998), pp. 626–639.

[2] AKAR, Nail – SOHRABY, Khosrow, Finite and Infinite QBD Chains: A Simple and Unifying Algorithmic Approach, In *Proceedings of IEEE INFOCOM*, (1997), pp. 1105–1113.

[3] AKAR, Nail – SOHRABY, Khosrow, Matrix Geometric Solutions in M/G/1 Type Markov Chains with Multiple Boundaries: A Generalized State-Space Approach, *International Teletraffic Congress*, March, 1997.

[4] BINI, Dario – MEINI, Beatrice, Improved Cyclic Reduction for Solving Queueing Problems, *Numerical Algorithms*, **15** (1997), pp. 57–74.

[5] BLONDIA, C. – CASALS, O., Statical Multiplexing of VBR Source: A Matrix-Analytic Approach, *Performance Evaluation*, **16** (1992), pp. 5–20.

[6] CHAKKA, R., Performance and Reliability Modelling of Computing System Using Spectral Expansion, University of Newcastle upon Tyne, PhD Thesis, 1995.

[7] CHAKKA, R. – HARRISON, P.G., The MMCPP/GE/c Queue as a Node Model for ATM Networks, In *UK Performance Evaluation Workshop 97*, 1997, June 19.

[8] CHIEN, M. – ORUC, Y., High Performance Concentrators and Superconcentrators using Multiplexing Schemes, *IEEE Transactions on Communications*, **42** (1994), pp. 3045–3050.

[9] DUMMLER, Mathias, Analysis of the Departure Process of a Batch Service Queueing System, Universität Würzburg, 1998, September.

[10] LATOUCHE, G. – RAMASWAMI, V., A Logarithmic Reduction Algorithm for Quasi-Birth-Death Processes, *Journal of Applied Probability*, **30** (1993), pp. 650–674.

[11] NAOUMOV, Valeri – KRIEGER, Udo R. – WARNER, Dietmar, Analysis of a Multi-Server Delay-Loss System With a General Markovian Arrival Process, *Matrix-Analytic Methods in Stochastic Models*, 1997.

[12] NATARAJAN, K., On the G/GE/1 Queue, Technical report, Universität Trier Mathematic/Informatik, 1993.

[13] NELSON, RANDOLPH, *Probability, Stochastic Processes and Queueing Theory*, Springer-Verlag, New York, 1995.

[14] NEUTS, M. F., *Matrix Geometric Solutions in Stochastic Model*, Johns Hopkins University Press, Baltimore, 1981.

[15] NIU, Zhisheng – TAKAHASHI, Yoshitaka, An Extended Queueing Model for SVC-based IP-over-ATM Networks and its Analysis, In *Proceedings of IEEE GLOBECOM*, (1998), pp. 490–495.

[16] ROSTI, E. et al., On Processor Saving Scheduling Policies for Multiprocessor Systems, *IEEE Trans. Comp.*, **47** (2), (1998), pp. 178–189.

[17] STEWART J., William, On the Use of Numerical Methods for ATM Models, In *IFIP Workshop: Modelling and Performance Evaluation of ATM Technology*, (1993), pp. 4.1.1-4.1.19.

[18] TRAN T., Hung – DO V., Tien, A New Iterative Method for Systems with Batch Arrivals and Batch Departures, In *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, CNDS'00*, San Diego, USA, 2000.