Control of the BMAP/PH/1/K queue with group services

Dario A. Bini* Srinivas Chakravarthy† Beatrice Meini*

Abstract

We consider a finite capacity queue in which arrivals occur according to a batch Markovian process. Services are offered to groups of varying size and the service times are assumed to be of phase type. When the buffer size exceeds N, the service rate is increased by a factor $\theta > 1$, $L \leq N < K$ and the service is brought back to its normal rate upon completion of the current service. For this queuing model the steady state analysis is performed. The duration in normal as well as fast periods are shown to be of phase type. Efficient algorithms for computing system performance measures, based on regular splittings, are presented. A comparison of this algorithm to (block) Gauss-Seidel method is provided. Some numerical examples are discussed.

1 Introduction and model description

In order to attract and serve customers (or messages or jobs) efficiently it may be necessary to increase the service rate whenever the number of waiting customers in the system exceeds a certain threshold value, say N, and then bring back to normal rate. Variations and special cases of a single server queuing model with control of services have been studied in the literature [5, 11, 12, 13, 14, 15, 17, 22, 25, 26, 28]. While, in controlled queuing systems involving switching costs (switching between normal and fast service rates) determination of optimal operating policy is in general difficult [26], Neuts and Rao [22] develop algorithms for a finite capacity M/G/1 queue with phase-type services, that permit to search for optimal values interactively.

*Dipartimento di Matematica, Università di Pisa, Italy, bini,meini@dm.unipi.it

†Department of Industrial and Manufacturing Engineering and Business, Kettering University, Flint, MI, USA, schakrav@kettering.edu

These results were generalized in [5] for MAP arrivals. All these papers dealt with service scheme in which customers are served *one* at a time.

In the context of a finite capacity queueing system with finite buffer of size K, a different type of service scheme in which groups of varying size, ranging from a predetermined threshold L to the maximum size K, was introduced in Chakravarthy [1]. The pre-assigned number $L \geq 1$, called the *threshold* operates as follows. An idle server finding fewer than L customers in the queue remains idle. However, if there are i, $L \leq i \leq K$, customers present, the idle server initiates a service for the entire group. If a service has to be initiated through an arrival of a batch (for example, i jobs, $i < L$, are in the queue with the server being idle, and the arriving batch has at least $L - i$ jobs), then the server initiates a service for a group of size no larger than K, and any remaining customers in the batch are considered lost.

Service scheme of this type in the context of finite capacity GI/PH/1 and MAP/G/1 with single arrivals, BMAP/M/c models were studied extensively in [1, 2, 3, 8, 10, 4, 9, 6, 5, 11, 12, 7]. Some potential applications of this type of service mechanism in manufacturing processes were given in those papers.

In this paper, we consider a single server queue with a limited buffer of size K. Arrivals of customers occur according to a batch Markovian arrival process (BMAP), a tractable class of Markov renewal processes. Customers in a batch are admitted to the extent of available buffer and all others are considered lost. In general, in production and manufacturing areas, jobs (or customers) arrive in batches from various sources to a common processing center and so the arrival process can no longer be assumed to form a renewal process. Hence, modeling the arrival processes with BMAP seems to be a natural choice. It should be noted that by appropriately choosing the parameters of the BMAP the underlying arrival process could be made as a renewal process with batch arrivals.

Services are offered to groups of varying size i, $L \leq i \leq K$, where L is a predetermined threshold. The period when the server serves at a normal rate is called the normal service period and the period when the server serves at a fast rate will be referred to as the fast service period. The normal service time follows a phase-type distribution (PH-distribution) with representation $(\boldsymbol{\beta}, S)$ of order n. When the buffer size exceeds N, the service rate is increased by a factor $\theta > 1$, $L \leq N < K$, and the service is brought back to its normal rate upon completion of the current service. Thus, during the fast service mode, the service time has the representation $(\boldsymbol{\beta}, \theta S)$ of order n. Although our analysis can be carried out for the case when the fast service time has the representation, say, $(\boldsymbol{\beta}^{(1)}, S^{(1)})$ of order $n^{(1)}$, we consider the special representation to avoid introducing more elaborate notation in our analysis. Note that the normal service rate, μ, is given by $\mu = (-\boldsymbol{\beta} S^{-1} \boldsymbol{e})^{-1}$.

1.1 Batch Markovian Arrival Process

The BMAP, a special class of tractable Markov renewal process, is a rich class of point processes that includes many well-known processes such as Poisson, PH-renewal processes, and Markov-modulated Poisson process. One of the most significant features of the BMAP is the underlying Markovian structure and fits ideally in the context of matrix-analytic solutions to stochastic models. Matrix-analytic methods were first introduced and studied by Neuts [19, 20]. As is well known, Poisson processes are the simplest and most tractable ones used extensively in stochastic modeling. The idea of the BMAP is to significantly generalize the Poisson processes and still keep the tractability for modeling purposes. Furthermore, in many practical applications, notably in communications engineering, production and manufacturing engineering, the arrivals do not usually form a renewal process. So, BMAP is a convenient tool to model both renewal and non-renewal arrivals. While BMAP is defined for both discrete and continuous times, here we will need only the continuous time case.

The BMAP in continuous time is described as follows. Let the underlying Markov chain be irreducible and let $Q = (q_{i,j})$ be the generator of this Markov chain. At the end of a sojourn time in state i, that is exponentially distributed with parameter $\lambda_i \geq -q_{i,i}$, one of the following two events could occur: with probability $p_{i,j}(k)$ the transition corresponds to an arrival of group size k, $k \geq 1$, and the underlying Markov chain is in state j with $1 \leq i, j \leq m$; with probability $p_{i,j}(0)$ the transition corresponds to no arrival and the state of the Markov chain is j, $j \neq i$. Note that the Markov chain can go from state i to state i only through an arrival. Also, we have

$$\sum_{k=1}^{\infty} \sum_{j=1}^{m} p_{i,j}(k) + \sum_{\substack{j=1 \\ j \neq i}}^{m} p_{i,j}(0) = 1, \quad \text{for all} \quad 1 \leq i \leq m.$$

For $k \geq 0$, define matrices $D_k = (d_{i,j}(k))$ such that $d_{i,i}(0) = -\lambda_i$, $1 \leq i \leq m$, $d_{i,j}(0) = \lambda_i p_{i,j}(0)$, for $j \neq i$, $1 \leq i, j \leq m$, and $d_{i,j}(k) = \lambda_i p_{i,j}(k)$. By assuming D_0 to be a nonsingular matrix, the interarrival times will be finite with probability one and the arrival process does not terminate. Hence, we see that D_0 is a stable matrix. The generator Q is then given by $Q = \sum_{k=0}^{\infty} D_k$.

Thus, the BMAP is described by the matrices D_k's with D_0 governing the transitions corresponding to *no* arrival and D_k governing those corresponding to arrivals of group size k, $k \geq 1$. It can be shown that BMAP is equivalent to Neuts' versatile Markovian point process. The point process described by the BMAP is a special class of semi-Markov processes with transition probability matrix given by

$$\int_0^x e^{D_0 t} dt \; D_k = [I - e^{D_0 x}](-D_0)^{-1} D_k, \quad \text{for} \quad k \geq 1. \tag{1}$$

Let $\boldsymbol{\pi}$ be the stationary probability vector of the Markov process with generator Q. That is, $\boldsymbol{\pi}$ is the unique (positive) probability vector satisfying

$$\boldsymbol{\pi} Q = \mathbf{0} \quad \text{and} \quad \boldsymbol{\pi} \boldsymbol{e} = 1.$$

Let $\boldsymbol{\alpha}$ be the initial probability vector of the underlying Markov chain governing the BMAP. Then choosing $\boldsymbol{\alpha}$ appropriately we can model the time origin to be

1. an arbitrary arrival point;
2. the end of an interval during which there are at least k arrivals;
3. the point at which the system is in specific state such as the busy period ends or busy period begins.

The most interesting case is the one in which we get stationary version of the BMAP by $\boldsymbol{\alpha} = \boldsymbol{\pi}$. The constant $\lambda = \boldsymbol{\pi} \sum_{k=1}^{\infty} k D_k \boldsymbol{e}$, referred to as the *fundamental rate*, gives the expected number of arrivals per unit of time in the stationary version of the MAP. The quantity $\lambda_g = \boldsymbol{\pi}(-D_0)\boldsymbol{e}$, gives the arrival rate of groups. Note that for a BMAP with single arrivals $\lambda = \lambda_g$. For further details on BMAP and their usefulness in stochastic modeling, we refer to Lucantoni [16], and Neuts [18, 19, 20, 21].

2 The Markov process

The model outlined in Section 1 can be studied as a continuous-time Markov chain (CTMC). Denote by (i, j), $0 \leq i \leq L-1$, $1 \leq j \leq m$, the state that corresponds to the server being idle with i customers waiting in the queue and the arrival process is in phase j. By (i, j, k), for $0 \leq i \leq K$, $1 \leq j \leq m$, $1 \leq k \leq n$, we denote the state in which there are i customers waiting in the queue with the arrival process in phase j and the current service is in phase k. We will denote the set of states $\{(i, j) : \; 1 \leq j \leq m\}$ by $\underline{\boldsymbol{i}}^*$, for $0 \leq i \leq L-1$, and the set of states $\{(i, j, k) \; : \; 1 \leq j \leq m, \; 1 \leq k \leq n\}$ by $\underline{\boldsymbol{i}}$, for $0 \leq i \leq K$. The CTMC describing the model has the state space given by $\Delta = \{(i, j), \; 0 \leq i \leq L-1, \; 1 \leq j \leq m\} \cup \{(i, j, k), \; 0 \leq i \leq K, \; 1 \leq j \leq$

$m,\ 1 \le k \le n\}$. The generator, Q^*, of the CTMC is given by

$$\left[\begin{array}{ccccccccccccc}
D_0 & \dots & D_{L-1} & \widetilde{D}_L \otimes \boldsymbol{\beta} & & & & & & & & & \\
 & \ddots & \vdots & \vdots & & & & & & & & & \\
 & & D_0 & \widetilde{D}_1 \otimes \boldsymbol{\beta} & & & & & & & & & \\
I \otimes \boldsymbol{S}^0 & & & A_0 & \dots & A_{L-1} & A_L & \dots & A_N & A_{N+1} & \dots & A_{K-1} & \widetilde{A}_K \\
 & \ddots & & & \ddots & \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\
 & & I \otimes \boldsymbol{S}^0 & & & A_0 & A_1 & & \vdots & \vdots & & \vdots & \vdots \\
 & & & I \otimes \boldsymbol{S}^0\boldsymbol{\beta} & & & A_0 & \ddots & \vdots & \vdots & & \vdots & \vdots \\
 & & & \vdots & & & & \ddots & A_1 & \vdots & & \vdots & \vdots \\
 & & & I \otimes \boldsymbol{S}^0\boldsymbol{\beta} & & & & & A_0 & A_1 & & \vdots & \vdots \\
 & & & \theta I \otimes \boldsymbol{S}^0\boldsymbol{\beta} & & & & & & E_0 & \ddots & \vdots & \vdots \\
 & & & \vdots & & & & & & & \ddots & A_1 & \vdots \\
 & & & \theta I \otimes \boldsymbol{S}^0\boldsymbol{\beta} & & & & & & & & E_0 & \widetilde{A}_1 \\
 & & & \theta I \otimes \boldsymbol{S}^0\boldsymbol{\beta} & & & & & & & & & E_1
\end{array}\right]$$

where the matrices appearing in Q^* are given by

$$\begin{array}{l}
A_0 = D_0 \otimes I + I \otimes S,\ A_i = D_i \otimes I,\ 1 \le i \le K-1, \\
\widetilde{A}_i = \widetilde{D}_i \otimes I,\ 1 \le i \le K, \\
E_0 = D_0 \otimes I + \theta I \otimes S,\ E_1 = Q \otimes I + \theta I \otimes S.
\end{array} \tag{2}$$

Let $\boldsymbol{x}$ be the steady-state probability vector of Q^*. That is, $\boldsymbol{x}$ is the unique solution to $\boldsymbol{x}Q^* = \boldsymbol{0}$ and $\boldsymbol{x}\boldsymbol{e} = 1$. Partition $\boldsymbol{x}$ into

$$\boldsymbol{x} = (\boldsymbol{u}(0), \dots, \boldsymbol{u}(L-1), \boldsymbol{v}(0), \dots, \boldsymbol{v}(K))$$

where the vectors $\boldsymbol{u}(i)$, $0 \le i \le L-1$, are of order m, and the vectors $\boldsymbol{v}(i)$, $0 \le i \le K$ are of order mn.

The coefficient matrices appearing in Q^* are highly sparse and exploitation of these will result in manipulating with matrices of lesser dimensions. Our algorithmic procedures take this into account and some details are given in the next section.

3 Computation of the steady state vector

The generator matrix Q^* is sparse and strongly structured; moreover, the block entries themselves are structured. In this section we propose an iterative algorithm, based on a suitable regular splitting, for computing the steady state vector $\boldsymbol{x}$, that fully exploits the sparsity and the structure of

Q^*. We recall [27] that, given an M-matrix W, the splitting $W = F - G$ is called a *regular splitting* if F is nonsingular and the matrices G and F^{-1} are nonnegative. Regular splittings of singular and nonsingular M-matrices have been extensively studied in the literature [27, 23, 24] and are particularly suitable for solving large linear systems with structured matrices.

In our case the matrix $-Q^*$ is a singular M-matrix and, if $-Q^* = F - G$ is a regular splitting, then the system $\boldsymbol{x}Q^* = \boldsymbol{0}$ can be equivalently rewritten in the form $\boldsymbol{x} = GF^{-1}\boldsymbol{x}$. The latter equation can be used to generate the sequence of vectors $\boldsymbol{x}_{h+1} = GF^{-1}\boldsymbol{x}_h$, $h \geq 0$: if the sequence $\{\boldsymbol{x}_h\}_h$ converges to a vector $\boldsymbol{x}^*$, then $\boldsymbol{x}^*$ is such that $\boldsymbol{x}^*Q^* = \boldsymbol{0}$, i.e., up to a multiplicative constant, $\boldsymbol{x}^*$ is the steady state vector.

In the computation of the null vector of a singular M-matrix, a nontrivial problem is the choice of the regular splitting. Indeed, on one hand we would like to have a high convergence rate; on the other hand, sparsity or structural properties of GF^{-1} can lead to a great reduction of the computational cost at each step. Moreover, for any regular splitting $F - G$, the iteration matrix GF^{-1} has spectral radius equal to one, and thus not all regular splittings generate convergent sequences of vectors [23, 24].

For the computation of the steady state vector $\boldsymbol{x}$ we have choosen a regular splitting $-Q^* = F - G$ where F is a structured matrix that can be easily inverted, and G is a matrix of rank mn. In this way the computational cost of each iteration is just $O(m^2n^2)$ arithmetic operations and, from the numerical experiments that we have perfomed, also the convergence rate is high. More precisely, we take

$$G = W_1 W_2^T,\ W_1 = \begin{bmatrix} \widetilde{D}_L \otimes \boldsymbol{\beta} \\ \vdots \\ \widetilde{D}_1 \otimes \boldsymbol{\beta} \\ 0 \\ \vdots \\ 0 \\ I \otimes \boldsymbol{S}^0\boldsymbol{\beta} \\ \vdots \\ I \otimes \boldsymbol{S}^0\boldsymbol{\beta} \\ \theta I \otimes \boldsymbol{S}^0\boldsymbol{\beta} \\ \vdots \\ \theta I \otimes \boldsymbol{S}^0\boldsymbol{\beta} \end{bmatrix},\ W_2^T = \left[\begin{array}{ccc|c} 0 & \dots & 0 & H_1 \end{array}\right],\ F = -Q^* + G$$

where $H_1 = [I, 0, \dots, 0]$ is the first block row of the identity matrix of block size K, with blocks of size mn. The resulting matrix F can be repartitioned as

$$F = \begin{bmatrix} F_{1,1} & 0 & 0 & 0 \\ F_{2,1} & F_{2,2} & F_{2,3} & F_{2,4} \\ 0 & 0 & F_{3,3} & F_{3,4} \\ 0 & 0 & 0 & -E_1 \end{bmatrix}$$

where $F_{1,1}$, $F_{2,2}$, $F_{3,3}$ are block upper triangular block Toeplitz matrices of block size L, N, $K-N-1$ respectively; $F_{2,3}$ is an $N \times (K-N-1)$ block Toeplitz matrix; $F_{2,1}$ is an $N \times L$ block diagonal matrix; $F_{2,4}$ and $F_{3,4}$ are block column vectors of size N and $K-N-1$ respectively. More precisely,

$$F_{1,1} = -\begin{bmatrix} D_0 & \dots & D_{L-1} \\ & \ddots & \vdots \\ & & D_0 \end{bmatrix}, \quad F_{2,2} = -\begin{bmatrix} A_0 & \dots & A_N \\ & \ddots & \vdots \\ & & A_0 \end{bmatrix}, \quad F_{3,4} = -\begin{bmatrix} \widetilde{A}_{K-N} \\ \vdots \\ \widetilde{A}_1 \end{bmatrix},$$

$$F_{3,3} = -\begin{bmatrix} E_0 & A_1 & \dots & A_{K-N-2} \\ & \ddots & \ddots & \vdots \\ & & E_0 & A_1 \\ & & & E_0 \end{bmatrix}, \quad F_{2,3} = -(A_{j-i+N+1})_{i=0,N;j=0,K-N-2},$$

$$F_{2,1} = -\begin{bmatrix} I \otimes \boldsymbol{S}^{(0)} & & \\ & \ddots & \\ & & I \otimes \boldsymbol{S}^{(0)} \\ & & \end{bmatrix}, \quad F_{2,4} = -\begin{bmatrix} \widetilde{A}_K \\ \vdots \\ \widetilde{A}_{K-N+1} \end{bmatrix}$$

The iteration matrix is thus $GF^{-1} = W_1 W_2^T F^{-1}$, and each iterative step is reduced to perform the product $\boldsymbol{w}_{h+1} = \boldsymbol{w}_h R$ where $R = W_2^T F^{-1} W_1$ is a matrix of size mn and $\boldsymbol{w}_h$ is an mn dimensional vector. The matrix R is nonnegative matrix with spectral radius equal to 1, and the steady state vector $\boldsymbol{x}$ is given by $\boldsymbol{x} = \boldsymbol{w}_* W_2^T F^{-1}$, where $\boldsymbol{w}_* = \lim_h \boldsymbol{w}_h$.

For the computation of the block vector $V^T = W_2^T F^{-1}$ we partition V^T into two subvectors of size L and $K+1$, respectively, i.e., $V^T = (V_1^T, V_2^T)$. Then V_2^T is computed by solving the system

$$V_2^T \begin{bmatrix} F_{2,2} & F_{2,3} & F_{2,4} \\ 0 & F_{3,3} & F_{3,4} \\ 0 & 0 & E_1 \end{bmatrix} = [I, 0, 0].$$

V_1^T is obtained by solving the system $V_1^T F_{1,1} = V_{2,1}^T F_{2,1}$ where $V_{2,1}^T$ is the first block component of V_2^T. Using the Kronecker structure of the blocks, the vector V^T can be computed with $O(m^3n^3 + K^2m^3n^2)$ arithmetic operations. Alternatively, using the block Toeplitz structure, we may compute V^T in $O(m^2n^2K \log K + Km^3n^3)$ arithmetic operations by means of FFT.

4 Analysis of idle time, normal and fast service periods

Let Y_I, Y_N, and Y_F, respectively, denote the idle time of the server, the time spent in normal service and the time spent in fast service periods. In this section we will show that these random variables have phase type distributions.

Theorem 1 *The random variable Y_I is of phase type with representation $(\boldsymbol{\xi}_I, R_I)$ of dimension Lm, where*

$$\boldsymbol{\xi}_I = \frac{1}{\sum_{i=0}^{L-1} \boldsymbol{v}(i)(\boldsymbol{e} \otimes \boldsymbol{S}^0)}[\boldsymbol{v}(0)(I \otimes \boldsymbol{S}^0), \ldots, \boldsymbol{v}(L-1)(I \otimes \boldsymbol{S}^0)],$$

$$R_I = \begin{bmatrix} D_0 & D_1 & \ldots & D_{L-1} \\ & D_0 & \ddots & \vdots \\ & & \ddots & D_1 \\ & & & D_0 \end{bmatrix}.$$

Proof. On noting that (a) the busy server becomes idle if and only if the number of customers waiting in the queue is less than L upon completion of the current service and (b) the server becomes busy again when the queue size increases to L or more, the stated result follows. □

Theorem 2 *The random variable Y_N is of phase type with representation $(\boldsymbol{\xi}_N, R_N)$ of dimension Lmn, where*

$$\boldsymbol{\xi}_N = \frac{1}{\sum_{i=0}^{L-1} \boldsymbol{u}(i)\widetilde{D}_{L-i}\boldsymbol{e}}[\boldsymbol{u}(0)\widetilde{D}_L \otimes \boldsymbol{\beta}, \ldots, \boldsymbol{u}(L-1)\widetilde{D}_1 \otimes \boldsymbol{\beta}],$$

$$R_N = \begin{bmatrix} A_0 & A_1 & \ldots & A_N \\ & A_0 & \ddots & \vdots \\ & & \ddots & A_1 \\ & & & A_0 \end{bmatrix} + \begin{bmatrix} 0_L \\ I \\ \vdots \\ I \end{bmatrix} [I \otimes \boldsymbol{S}^0\boldsymbol{\beta}, 0, \ldots, 0]$$

where 0_L is a null block vector with L null block components.

Proof. The result follows immediately on noting that (a) an arbitrary normal service period starts at the beginning of a service. If the system is in state $\boldsymbol{i}^*$, for $0 \le i \le L-1$, then the vector $\boldsymbol{u}(i)\widetilde{D}_{L-i} \otimes \boldsymbol{\beta}$, appropriately normalized, gives the probabilities that the server initiates a service; and (b) the normal service mode ends either through the server becoming idle due to lack of customers or the server enters into a fast service mode. □

Theorem 3 *The random variable Y_F is of phase type with representation $\left(\sum_{r=0}^{N} \boldsymbol{v}(r)(\widetilde{D}_{N-r}\boldsymbol{e} \otimes I), \theta S\right)$ of dimension n.*

Proof. First note that the vector $(\sum_{r=0}^{N} \boldsymbol{v}(r)(\widetilde{D}_{N-r}\boldsymbol{e} \otimes I)$ suitably normalized gives the probabilities that the current (normal) service will be accelerated from appropriate states, by the factor θ to become a fast service. Upon completion of this service the server returns to a normal service mode due to the type of service rule assumed. □

5 Numerical examples

In this section we will discuss a few interesting examples to bring out the qualitative behavior of the model under study. For $(\boldsymbol{\beta}, S)$ we consider three sets

- Exponential services: $n = 1$, $\boldsymbol{\beta} = 1$, $S = -10$;
- Erlang services: $n = 10$, $\boldsymbol{\beta} = (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$, S has diagonal entries equal to -100 and upper diagonal entries equal to 100 and is zero elsewhere;
- Hyperexponential services: $n = 2$,$\boldsymbol{\beta} = (0.9, 0.1)$, S is a diagonal matrix with diagonal entries -100, $-1/0.91$.

For the arrival process, we consider the following special class of BMAP. Let $D_k = Da_k$, for $k \geq 1$, where $\{a_k\}$ is a legitimate probability mass function. Thus, for this special BMAP, we need to specify the matrices D_0 and D of order m and the probability mass function $\{a_k\}$. The matrices D_0 and D are generated using two phase type distributions with representations $(\boldsymbol{\alpha}_1, T_1)$ and $(\boldsymbol{\alpha}_2, T_2)$. Let the phase type representation $(\boldsymbol{\alpha}_1, T_1)$ of order 9 denote an Erlang of order 9 with parameter η and let the representation $(\boldsymbol{\alpha}_2, T_2)$ of order 1 stand for an exponential distribution with parameter 10η. Note that $\boldsymbol{\alpha}_2 = 1$, $T_2 = -10\eta$. We take D_0 and D to be

$$D_0 = \begin{bmatrix} T_1 & 0 \\ 0 & -100\eta \end{bmatrix}, \quad D = \begin{bmatrix} p_1 \boldsymbol{T}_1^0 \boldsymbol{\alpha}_1 & (1-p_1)\boldsymbol{T}_1^0 \\ 100\eta(1-p_2)\boldsymbol{\alpha}_1 & 100 p_2 \eta \end{bmatrix},$$

where $\boldsymbol{T}^0$ is such that $T\boldsymbol{e} + \boldsymbol{T}^0 = \boldsymbol{0}$ and $0 \leq p_1, p_2 \leq 1$. By varying η, $\{a_k\}$, p_1 and p_2, we consider the following eight BMAPs.

BMAP_1 $a_5 = 1$ and $a_k = 0$ for all $k \neq 5$, $p_1 = p_2 = 0.01$.

BMAP_2 $a_5 = 1$ and $a_k = 0$ for all $k \neq 5$, $p_1 = p_2 = 0.99$.

	$K = 25$		$K = 50$		$K = 100$		$K = 200$	
BMAP	η	var	η	var	η	var	η	var
1	45.05	0.0122	45.05	0.0122	45.05	0.0122	45.05	0.0122
2	45.05	0.0122	45.05	0.0122	45.05	0.0122	45.05	0.0122
3	37.5416	0.0175	37.5416	0.0175	37.5416	0.0175	37.5416	0.0175
4	37.5416	0.0175	37.5416	0.0175	37.5416	0.0175	37.5416	0.0175
5	34.3817	0.0209	20.2689	0.0601	11.6845	0.1810	6.6201	0.5637
6	34.3817	0.0209	20.2689	0.0601	11.6845	0.1810	6.6201	0.5637
7	17.3269	0.0823	8.8333	0.3166	4.4604	1.2418	2.2412	4.9180
8	17.3269	0.0823	8.8333	0.3166	4.4604	1.2418	2.2412	4.9180

Table 1: Values of the parameter η and of the variance

BMAP_3 $a_k = de^{-5}5^{k-1}/(k-1)!$, for $1 \le k \le K$ and $a_k = 0$ for all other values of k, $p_1 = p_2 = 0.01$, and d is the normalizing constant. That is, $\{a_k\}$ is the truncated Poisson with parameter 5.

BMAP_4 $a_k = de^{-5}5^{k-1}/(k-1)!$, for $1 \le k \le K$ and $a_k = 0$ for all other values of k, $p_1 = p_2 = 0.99$, and d is the normalizing constant.

BMAP_5 $a_k = d/k$, for $1 \le k \le K$ and $a_k = 0$ for all other values of k, $p_1 = p_2 = 0.01$, and d is the normalizing constant.

BMAP_6 $a_k = d/k$, $1 \le k \le K$, and $a_k = 0$ for all other values of k, $p_1 = p_2 = 0.99$, and d is the normalizing constant.

BMAP_7 $a_k = d/K$ for all $1 \le k \le K$, $a_k = 0$ for other values of k, $p_1 = p_2 = 0.01$, and d is the normalizing constant.

BMAP_8 $a_k = d/K$ for all $1 \le k \le K$, $a_k = 0$ for other values of k, $p_1 = p_2 = 0.99$, and d is the normalizing constant.

All these eight BMAPs have the same total arrival rate of 50 jobs per unit of time. However, these are qualitatively different in that they have different variance and correlation structure. Since the arrival rate is fixed to be 50 for all BMAPs, the value of η is dependent on $\{a_k\}$.

In tables, 1, 2 we list the values of η the variance and correlation of these BMAPs in the stationary version. Note that higher the variance is more the variability in the arrival times. Also, a larger positive correlation indicates that longer interarrival times are followed by longer ones and shorter interarrival times are followed by shorter ones.

We have implemented the iterative algorithm of section 3 in Fortran 90. For the computation of the iteration matrix we did not use the FFT-based

BMAP	Correlation	BMAP	Correlation
1	-0.801492	5	-0.801492
2	0.801492	6	0.801492
3	-0.801492	7	-0.801492
4	0.801492	8	0.801492

Table 2: Correlation of BMAPs

algorithms. This is mainly to exploit the Kronecker structure appearing in the coefficient matrices.

We vary $K = 25, 50, 100, 200$ and take $L = \lceil K/4 \rceil$, $N = \lceil K/2 \rceil$. For arrival process, we consider the eight BMAPs listed above, and for the service times we consider the three PH-distributions mentioned at the beginning of this section. In Table 3 we report the number of iterations, the residual error and the CPU time needed to run the program on Pentium II, 450 MHz, with Linux operating system. As stopping criterion we choose the condition $||\boldsymbol{z}_{h+1} - \boldsymbol{z}_h|| < 10^{-13}$, where $||\boldsymbol{v}|| = \sum_i |v_i|$ denotes the 1-norm of $\boldsymbol{v} = (v_i)$.

We observe that the CPU time grows as $O(K^2)$ and is dominated by the computation of the iteration matrix. The cost of the iterative part of the algorithm seems to be unaffected.

We compared our algorithm with the customary (block) Gauss-Seidel iteration. The performance of the latter algorithm is strongly related to the number of iteration steps since the cost of each iteration, i.e., $O(K^2m^2n)$, is much larger than the corresponding cost of our algorithm, i.e., $O(m^2n^2)$.

In table 4 we report the performance measures in the case of BMAP_1 and BMAP_2. The full set of output data can be provided upon request. The parameters that we report are: the troughput THR, the probability P_I that a server is idle, the probability P_L that at least one customer is lost, the mean value E_L of the lost customers, the fraction FNR in normal rate and the fraction FFR in fast rate, the mean queue length MQL and the s. d. of queue length $SDQL$.

References

[1] S. Chakravarthy. A finite capacity GI/PH/1 queue with group services. *Naval Research Logistics Quarterly*, 39:345–357, 1992.

[2] S. Chakravarthy. Analysis of a finite MAP/G/1 queue with group services. *Queueing Sist. Appl.*, 13:385–407, 1993.

	Exponential			Erlang			Hyperexponential		
K	Steps	Time (s.)	Residual	Steps	Time (s.)	Residual	Steps	Time (s.)	Residual
BMAP_1									
25	111	0.5	3.7e-13	249	4.1	4.0e-13	202	0.6	3.6e-13
50	332	2.0	2.8e-13	500	14.5	2.9e-13	219	2.3	2.4e-13
100	285	7.6	1.6e-13	295	52.8	1.7e-13	189	9.2	1.4e-13
200	137	30.1	6.8e-14	137	203.5	7.0e-14	117	36.4	6.1e-14
BMAP_2									
25	40	0.5	6.7e-14	31	4.1	7.8e-14	112	0.6	2.1e-13
50	37	1.9	5.7e-14	30	14.3	3.0e-14	101	2.3	1.2e-13
100	32	7.5	3.9e-14	27	52.8	2.3e-14	85	9.3	6.5e-14
200	28	30.0	1.0e-14	22	203.8	1.1e-14	63	36.0	3.2e-14
BMAP_3									
25	18	0.5	1.6e-14	24	4.1	5.7e-14	14	0.6	2.8e-14
50	14	1.9	2.0e-14	14	14.3	5.5e-15	13	2.3	7.4e-15
100	9	7.5	8.4e-16	9	52.0	5.5e-16	10	9.1	3.9e-16
200	5	30.1	3.6e-16	5	203.0	3.5e-16	7	36.0	2.2e-16
BMAP_4									
25	40	0.5	1.1e-13	32	4.1	5.2e-14	112	0.6	2.2e-13
50	37	1.9	8.0e-14	30	14.2	4.0e-14	102	2.3	1.4e-13
100	31	7.5	3.3e-14	27	52.3	1.7e-14	84	9.1	7.7e-14
200	28	29.6	1.3e-14	22	202.6	7.5e-15	63	36.0	3.0e-14
BMAP_5									
25	15	0.5	6.1e-15	16	4.1	5.4e-15	19	0.6	1.5e-14
50	15	1.9	7.3e-15	16	14.2	2.1e-14	17	2.4	1.3e-14
100	15	7.5	8.5e-15	16	53.3	2.9e-15	16	9.3	1.8e-15
200	15	29.6	1.6e-15	15	206.1	2.0e-15	14	36.1	3.9e-15
BMAP_6									
25	47	0.5	9.4e-14	39	4.1	6.6e-14	126	0.6	1.6e-13
50	65	2.0	6.0e-14	59	14.3	4.3e-14	149	2.4	8.3e-14
100	89	7.5	3.3e-14	82	54.1	3.0e-14	171	9.2	4.8e-14
200	133	29.6	2.1e-14	127	206.3	2.0e-14	196	36.3	2.5e-14
BMAP_7									
25	58	0.5	1.8e-13	63	4.2	1.7e-13	50	0.6	1.6e-13
50	58	1.9	8.0e-14	59	14.3	7.1e-14	52	2.3	5.9e-14
100	54	7.5	2.8e-14	54	53.8	3.2e-14	51	9.1	2.5e-14
200	53	29.8	1.3e-14	53	207.3	1.5e-14	51	36.0	1.7e-14
BMAP_8									
25	81	0.5	9.0e-14	72	4.0	1.1e-13	176	0.6	1.5e-13
50	131	1.9	6.3e-14	125	14.2	6.1e-14	231	2.4	8.0e-14
100	212	7.5	3.8e-14	202	52.6	3.5e-14	310	9.1	4.1e-14
200	354	29.6	2.1e-14	348	205.1	2.1e-14	422	36.0	2.2e-14

Table 3: Steps, CPU time and residuals for splitting algorithm

K	THR	P_I	P_L	E_L	FNR	FFR	MQL	$SDQL$
BMAP_1: Exponential service								
50	50.000	0.673851	0.000000	0.0000	0.325916	0.000333	5.460	3.393
200	50.000	0.900001	0.000000	0.000	0.09999	0.000000	22.499	12.474
BMAP_1: Erlang service								
50	50.000	0.666774	0.000000	0.000	0.333226	0.000000	5.048	2.916
200	50.000	0.900000	0.000000	0.000	0.100000	0.000000	22.500	12.475
BMAP_1: Hyperexponential service								
50	49.839	0.810415	0.000643	0.161	0.160277	0.029308	6.847	6.500
200	50.000	0.909410	0.000000	0.000	0.088779	0.001812	23.363	14.386
BMAP_2: Exponential service								
50	29.867	0.829705	0.080532	20.133	0.168423	0.001872	5.072	3.437
200	39.382	0.945040	0.042472	10.618	0.053606	0.001354	22.463	13.874
BMAP_2: Erlang service								
50	29.226	0.829722	0.083094	20.774	0.168726	0.001552	5.045	3.344
200	38.410	0.945327	0.046361	11.590	0.053516	0.001157	22.412	13.666
BMAP_2: Hyperexponential service								
50	34.516	0.867194	0.061937	15.484	0.120063	0.012744	5.964	5.256
200	44.092	0.944790	0.023633	5.908	0.052057	0.003152	23.062	15.767

Table 4: BMAP_1 and BMAP_2: Performance Measures

[3] S. Chakravarthy. A finite capacity queuing network with single and multiple processing nodes. In R. F. Onvural and I. F. Akyildiz, editors, *Queuing Networks with Finite Capacity*, pages 197–211. North-Holland, Netherlands, 1993.

[4] S. Chakravarthy. Two finite queues in series with nonrenewal input and group services. In *Proceedings of the Seventh International Symposium on Applied Stochastic Models and Data Analysis*, pages 78–87. 1995.

[5] S. Chakravarthy. Analysis of the MAP/PH/1/K queue with service control. *Applied Stochastic Models and Data Analysis*, 12:179–191, 1996.

[6] S. Chakravarthy. Analysis of a priority polling system with group services. *Commun. Statist. Stochastic Models*, 14:25–49, 1998.

[7] S. Chakravarthy. Analysis of a multi-server queue with batch Markovian arrivals and group services *to appear in Engineering Simulation*,1999.

[8] S. Chakravarthy and A. S. Alfa. A finite capacity queue with Markovian arrivals and two servers with group services. *J. of Appl. Math. and Stochastic Analysis*, 7:161–178, 1994.

[9] S. Chakravarthy and L. Bin. A finite capacity queue with nonrenewal input and exponential dynamic group services. *INFORMS Journal on Computing*, 9:276–287, 1997.

[10] S. Chakravarthy and S. Y. Lee. An optimization problem in a finite capacity ph/ph/1 queue with group services. In et al G. V. Krishna Reddy, editor, *Stochastic Models, Optimization Techniques and Computer Applications*, pages 3–13. 1994.

[11] J. W. Cohen. On the optimal switching-level for an M/G/1 queueing system. *Stochastic Processes and Applications*, 4:297–316, 1976.

[12] T. B. Crabill. Optimal control of a service facility with variable exponential service times and constant arrival rate. *Management Science*, 18:560–566, 1972.

[13] T. B. Crabill, D. Gross, and M. J Magazine. A classified bibliography of research on optimal design and control of queues. *Oper. Res.*, 25:219–232, 1977.

[14] A. Federgruen and H.C. Tijms. Computation of the stationary distribution of the queue size in an M/G/1 queueing system with variable size. *J. Appl. Probability*, 17:515–522, 1980.

[15] R. F. Gebhard. A queueing process with bilevel hysteretic service-rate control. *Naval Research Logistics Quarterly*, 14:55–67, 1967.

[16] D. M. Lucantoni. New results on the single server queue with a batch Markovian arrival process. *Commun. Statist. Stochastic Models*, 7:1–46, 1971.

[17] D. M. Lucantoni, K. S. Meier-Hellstern, and M. F. Neuts. A single server queue with server vacations and a class of non-renewal arrival processes. *Adv. Appl. Prob.*, 22:676–705, 1990.

[18] M. F. Neuts. A versatile Markovian point process. *J. Appl. Probability*, 16:764–769, 1979.

[19] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach.* The Johns Hopkins University Press, Baltimore, MD, 1981.

[20] M. F. Neuts. *Structured Stochastic Matrices of M/G/1 Type and Their Applications.* Marcel Dekk., New York, 1989.

[21] M. F. Neuts. Models based on the Markovian arrival process. *IEICE Transactions on Communications*, E75B:1255–1265, 1992.

[22] M. F. Neuts and B. M. Rao. On the design of a finite-capacity queue with Phase-type service times and hysteretic control. *European J. of Operational Research*, 62:221–240, 1992.

[23] D. J. Rose. Convergent regular splittings for singular M-matrices. *SIAM J. Alg. Discr. Meth.*, 5:133–144, 1984.

[24] P. Semal. Iterative algorithms for large stochastic matrices. *Linear Algebra Appl.*, 154:65–103, 1991.

[25] H. C. Tijms. On a switch-over policy for controlling the work load in a queueing system with two constant service rates and fixed switch-over costs. *Zeitschrift für Operations Research*, 21:19–32, 1977.

[26] H. C. Tijms. A unified steady-state analysis for controlled markov drift processes in inventory, queueing and replacement problems. In P. Bartfai and J. Tomco, editors, *Point Processes and Queueing Problems*. Amsterdam, 1991.

[27] R. S. Varga. *Matrix Iterative Analysis*. Prentice Hall, New Jersey, 1962.

[28] M. Yadin and P. Naor. On queueing systems with variable service capacities. *Naval Research Logistics Quarterly*, 14:43–53, 1967.