Hindawi Publishing Corporation
Advances in Operations Research
Volume 2016, Article ID 4024950, 17 pages
http://dx.doi.org/10.1155/2016/4024950

Hindawi

Research Article

Phase-Type Arrivals and Impatient Customers in Multiserver Queue with Multiple Working Vacations

Cosmika Goswami and N. Selvaraju

Department of Mathematics, Indian Institute of Technology Guwahati, Guwahati 781 039, India

Correspondence should be addressed to Cosmika Goswami; cosmika.goswami@gmail.com

Received 27 October 2015; Accepted 14 February 2016

Academic Editor: Viliam Makis

We consider a PH/M/c queue with multiple working vacations where the customers waiting in queue for service are impatient. The working vacation policy is the one in which the servers serve at a lower rate during the vacation period rather than completely ceasing the service. Customer's impatience is due to its arrival during the period where all the servers are in working vacations and the arriving customer has to join the queue. We formulate the system as a nonhomogeneous quasi-birth-death process and use finite truncation method to find the stationary probability vector. Various performance measures like the average number of busy servers in the system during a vacation as well as during a nonvacation period, server availability, blocking probability, and average number of lost customers are given. Numerical examples are provided to illustrate the effects of various parameters and interarrival distributions on system performance.

1. Introduction

In communication networks, multiple servers are used to reduce the traffic congestion and improve the system performance. Multiple services are also used in highly efficient bandwidth-intensive applications. Different services may require different channel capacities and capacity of a channel depends upon the number of resources allocated to it. To understand the network behavior and to make intelligent decisions in their management, these systems can be modelled as multiserver queueing systems with server vacations. Levy and Yechiali [1] first discussed an M/M/c queue with exponentially distributed vacations. Tian and Li [2], Tian et al. [3], and Tian and Zhang [4] studied a variety of vacation models with multiple servers. They established the conditional stochastic decomposition properties on the steady-state queue length and the waiting time when all the servers are busy and obtained the stationary distributions for queue length and waiting times. Tian and Zhang [5] considered a two-threshold vacation policy in the context of a multiserver queueing model M/M/c. A multiserver queueing system with identical unreliable servers with PH-distributed service times is considered by Yang and Alfa [6]. Chakravarthy [7] studied an MAP/M/c queueing system, in which a group of servers take a simultaneous PH vacation.

The phenomenon of customer impatience is commonly observed in queueing systems, where customers leave a service system before receiving service due to the long waiting time or due to uncertainty of receiving service. Customer impatience or reneging represents loss in revenues and customer goodwill to the service provider. The problem of queues with impatient customers was first analyzed by Palm [8]. A bibliography can be found in Gross et al. [9]. Perel and Yechiali [10] considered a two-phase service impatient model where the customers become impatient if the server is in slow service phase. There are situations where customer's impatience is due to the absence of the server, more precisely due to the server being on vacation, and is independent of the customers in system. Altman and Yechiali [11, 12] studied the customer impatience in a classical vacation model and system with additional task, respectively.

Economou and Kapodistria [13] considered an unreliable queue where the customers leave the system at system failure times.

Multiserver queues with impatience, however, have attracted much attention in queueing literature possibly because of explosive demands to efficiently design and manage call or contact centres. Baccelli et al. [14] studied the waiting time distribution in M/M/c queue with general impatience bound on queueing times by constructing a simple Markov process and also gave the waiting time distribution in the M/G/1 queue with general impatience on queueing times. Yechiali [15] considered an M/M/c system which as a whole suffers occasionally a disastrous breakdown, upon which all present customers (waiting and served) are cleared from the system and lost. Stationary distribution of a multiserver vacation queue with constant impatient times is studied by Sakuma and Inoie [16]. Chen et al. [17] studied M/M/m/k queue with preemptive resume and impatience of the prioritised customers and derived the queue length distraction in stationary state and performance measures using the method of matrix analysis.

In communication systems, wavelength division multiplexing (WDM) is a method of transmitting packets from different sources, over the same fiber optic link, to the destination. A WDM network divides the available fiber bandwidth into WDM channels, details in Ho and Woei [18] and also in Wang [19]. This division of bandwidth or channel allocation is based on the capacities required for various services. For a high performance system, WDM channel allocation should lead to optimized resource utilization in a given network, which is physically feasible and cost-effective. A reconfigurable WDM system can be modelled as a queue with working vacations (WVs) as explained in Goswami and Selvaraju [20]. This vacation cannot be put in a classical vacation framework because here, unless the system is empty, the service does not cease completely. Servi and Finn [21] were the first to model such a WDM network into a WV queueing model. Liu et al. [22] studied the M/M/1/WV model with multiple WVs whereas the single WV model is analyzed by Tian et al. [23]. The same model is studied by Xu et al. [24] and also by Xiu et al. [25] with single WV and setup times. Wang et al. [26] presented the M/M/1/WV model using Newton's method to compute the steady-state probabilities and system performance measures. Wu and Takagi [27] extended Servi and Finn's work to M/G/1/WV model with generally distributed service times and vacation duration times. Baba [28] considered the GI/M/1/WV system with general independent arrival process where the distributions of the vacation duration times and service times are exponential. Chen et al. [29, 30] proposed an N-policy WV and a cyclic polling system for WDM taking the service times as exponential and PH distribution, respectively. Lin and Ke [31] considered a multiserver M/M/c queue and a cost model is derived to determine the optimal values of the number of servers and the WV rate simultaneously, in order to minimize the total expected cost per unit time.

Short distance networks, like local area networks (LANs), mostly use multimode WDM links. Multimode link is a single fiber link that supports many propagation paths or transverse modes through it. Aronson et al. [32] explained how the bandwidth of the fiber is multiplied by the number of paths used by using WDM in multimode fiber. LAN over Internet Protocol (IP) allows the forwarding of LAN packets over the Internet or an intranet network. One of the most critical performance measures in LAN over IP is the percentage of packets that are transmitted within hard delay bound or time constraint. If quality of service requirements is not met within the time bound, end users may terminate the Internet connections. A connection is terminated by pressing the stop button, refreshing the connection, or following a different link. This behavior can be termed the impatience of a user in LANs. To study the effect of multiple servers and user impatience on the performance in a WDM network, we consider in this paper a multiserver model with asynchronous multiple working vacation (AMWV) policy and impatient customers. In an AMWV policy, the servers take vacations individually and continue taking vacations till they do not find any customer in the system. An M/M/1/WV impatient model with single and multiple WV policies is studied by Selvaraju and Goswami [33]. Analysis of a finite buffer M/M/2 working vacations queue with balking and reneging wherein the servers operate under a triadic $(0, Q, N, M)$ policy is done recently by Laxmi and Jyothsna [34]. Lin and Ke [31] presented a multiserver WV queue with exponential interarrivals but none of these models represent systems with nonexponential arrivals or state-dependent systems. To study the role of arrival processes in a multiserver model having impatient customers, we consider here the PH arrival process. PH distribution is a general, nonexponential distribution characterized by a Markov chain. Importance of considering PH interarrivals is the fact that PH distribution is able to capture the nonexponential effects on arrivals while information flows in modern communication systems are rarely exponential. PH distribution is able to capture the profound effect of arrivals in system performance measures and makes the mathematical model more convincing to fit a real world scenario.

The paper is organized as follows. In Section 2, we formulate the system as a three-dimensional continuous-time Markov chain whose generator matrix is a level-dependent quasi-birth-death (QBD) process. Section 3 gives the finite truncation method used to find the stationary probability vector of the level-dependent process. The various performance measures are listed in Section 4 and in Section 5 the numerical illustrations of the system are presented.

2. Model Description

We consider a PH/M/c queue with multiple WVs and impatient customers. The interarrival times of customers follow a PH distribution, $\mathrm{PH}(\alpha, T)$, of dimension n and with arrival rate λ. A PH distribution denotes the distribution of time

until absorption in a finite Markov chain whose transition rate matrix is of type

$$P = \begin{bmatrix} T & T^0 \\ \mathbf{0} & 0 \end{bmatrix} \tag{1}$$

and α is the initial probability vector satisfying $\alpha \mathbf{e}_n = 1$ and $T^0 = -T\mathbf{e}_n$, where $\mathbf{e}_n$ is the column vector of dimension n with all the entries equal to one. The matrix T is a nonsingular square matrix with $(T)_{ii} < 0$, $1 \le i \le n$, and $(T)_{ij} \ge 0$, $1 \le i \ne j \le n$. The matrix T^0 is a nonnegative, n-dimensional column vector, grouping the absorption rates from any state to the absorbing one. The matrix $T^0\alpha$ gives the transition from one phase to another with an arrival of a customer to the system.

The customers are served according to FCFS basis. An arriving customer who finds all the c servers busy has to wait in queue; that is, when the number of customers in the system is more than c, a queue begins to form. The servers work independently of each other. The service times of each server during the nonvacation period follow an exponential distribution with rate μ_b, denoted by Exp(μ_b). A server goes to a WV as soon as it completes a service and finds no customer to serve in the system. For each server, the duration of WVs follows Exp(θ) distribution. During a WV period of a server, if a customer arrives to that server, it will serve the customer with Exp(μ_v) distribution, where $\mu_v < \mu_b$; that is, the customer will be served at a lower service rate. When a server returns from its vacation, if it finds at least one customer in queue waiting for service or finds an ongoing service in that server, the server switches its service rate from μ_v to μ_b and a nonvacation period starts. Otherwise, if the server finds an empty queue, after returning from one vacation, it immediately leaves for another WV.

An arriving customer gets service immediately upon its arrival, if it finds any of the c servers empty. But if all the servers are busy, the customer has to wait in a queue. A waiting customer becomes impatient when it finds all the servers serving at rate μ_v; that is, if the waiting customer finds all the servers in their WV period, the customer activates an impatient timer X. This impatient timer X follows Exp(ξ) distribution and is independent of the number of customers in the queue at that moment. If no server returns from its WV period by the time X expires, the customer leaves the system and never returns. Otherwise, if any of the servers returns from its vacation before the time X expires, the customer stops the timer and stays in the system until its service is completed. Here, the customer's impatience depends not only on waiting time in a queue but also on the number of servers that are in WVs. The interarrival times, service times, vacation duration times, and the impatient times all are taken to be mutually independent.

To model this system, we define a continuous-time Markov chain:

$$\Delta = \{(N_t, Q_t, J_t),\ t \ge 0\}, \tag{2}$$

where N_t denotes the total number of customers in the system, Q_t denotes the number of busy servers in nonvacation state, and J_t gives the phase of the arrival process. The state space of this Markov chain is

$$\begin{aligned} E = {} & \{(i,j,k);\ 0 \le i < c,\ j \le i,\ 1 \le k \le n\} \\ & \cup \{(i,j,k);\ i \ge c,\ 0 \le j \le c,\ 1 \le k \le n\}. \end{aligned} \tag{3}$$

The lexicographical order of the states, that is, $(0,0,1),\ldots,(0,0,n)$, $(1,0,1),\ldots,(1,0,n)$, $(1,1,1),\ldots,(1,1,n)$, $(2,0,1),\ldots,(2,0,n)$, $(2,1,1),\ldots,(2,1,n)$, $(2,2,1),\ldots,(2,2,n),\ldots$, $(c,0,1),\ldots,(c,c,n)$, $(c+1,0,1),\ldots,(c+1,c,n),\ldots$, gives the infinitesimal generator matrix of the Markov chain as with

$$Q = \begin{pmatrix} A_1^{(0)} & A_0^{(0)} & & & \\ A_2^{(1)} & A_1^{(1)} & A_0^{(1)} & & \\ & A_2^{(2)} & A_1^{(2)} & A_0^{(2)} & \\ & & A_2^{(3)} & A_1^{(3)} & A_0^{(3)} \\ & & & \ddots & \ddots & \ddots \end{pmatrix}, \tag{4}$$

where, for $1 \le i < c$,

$$A_2^{(i)} = \begin{bmatrix} i\mu_v I & & & & \\ & (\mu_b + (i-1)\mu_v) I & & & \\ & & (2\mu_b + (i-2)\mu_v) I & & \\ & & & \ddots & \\ & & & & ((i-1)\mu_b + \mu_v) I \\ & & & & & i\mu_b I \end{bmatrix}_{(i+1)n \times (i+1)n} \tag{5}$$

and, for $0 \le i < c$,

$$A_1^{(i)} = \begin{bmatrix} T-(c\theta+i\mu_v)I & c\theta I & & & \\ & T-((c-1)\theta+\mu_b+(i-1)\mu_v)I & (c-1)\theta I & & \\ & & \ddots & & \\ & & & T-i\mu_b I \end{bmatrix}_{(i+1)n\times(i+1)n},$$

$$A_0^{(i)} = \begin{bmatrix} T^0\alpha & & & \\ & T^0\alpha & & \\ & & \ddots & \\ & & & T^0\alpha \end{bmatrix}_{(i+1)n\times(i+1)n}. \tag{6}$$

For $i \geq c$,

$$A_2^{(c+l)} = \begin{bmatrix} (c\mu_v+l\xi)I & & & & \\ & (\mu_b+(c-1)\mu_v)I & & & \\ & & (2\mu_b+(c-2)\mu_v)I & & \\ & & & \ddots & \\ & & & & c\mu_b I \end{bmatrix}_{(c+1)n\times(c+1)n},$$

$$A_1^{(c+l)} = \begin{bmatrix} T-(c(\theta+\mu_v)+l\xi)I & c\theta I & & & \\ & \ddots & \ddots & & \\ & & T-(\theta+\mu_v+(c-1)\mu_b)I & \theta I & \\ & & & T-(c\mu_b)I \end{bmatrix}_{(c+1)n\times(c+1)n}, \tag{7}$$

$$A_0^{(c+l)} = \begin{bmatrix} T^0\alpha & & & \\ & T^0\alpha & & \\ & & \ddots & \\ & & & T^0\alpha \end{bmatrix}_{(c+1)n\times(c+1)n}.$$

The matrix I is an identity matrix of dimension n. Here, for $0 \leq i \leq c-1$, dimension of the matrices $A_0^{(i)}$, $A_1^{(i)}$, and $A_2^{(i)}$ increases with the levels; and for $i \geq c$, the matrices are of dimensions $(c+1)n \times (c+1)n$ each. It can be observed that Q given above is the generator of a nonhomogeneous QBD process, which we assume to be irreducible, with levels denoting the number of customers in the system.

3. Stationary Distribution

The queueing system under study is stable for $\rho = \lambda/c\mu_b < 1$ [28, 35].

Let x be the stationary probability vector associated with Q satisfying

$$xQ = 0,$$
$$x\mathbf{e} = 1. \tag{8}$$

Aggregating terms depending on levels, we get $x = [x_0 \; x_1 \; x_2 \; \cdots]$. Further depending on number of busy servers in nonvacation, we get, for $0 \leq i \leq c-1$, $x_i = [x_{i1}, x_{i2}, \ldots, x_{ii}]$, which are row $(i+1)n$-vectors and, for $i \geq c$, x_i are row $(c+1)n$-vectors. Each x_{ij} vector is an n-dimensional row vector, $x_{ij} = [x_{ij1}, \ldots, x_{ijn}]$ for $j \leq i$, depending on the phases of arrivals.

In this model, the generator matrix Q is spatially nonhomogeneous and a closed-form analytical solution or a direct algorithmic computation of the stationary probability vector x is quite difficult, if not impossible. For such level-dependent QBDs (LDQBDs), the stationary vectors are usually approximated by using various numerical approximation methods like finite truncation method (Artalejo et al. [36] and Chakravarthy et al. [37]), generalized truncation method (Falin [38] and Artalejo and Pozo [39]), truncation method using LDQBD processes (Bright and Taylor [40] and Krishnamoorthy et al. [41]), and matrix-geometric approximations (Neuts and Rao [42]).

Different methods have different levels of computable efficiency but it is expected that whichever method is used, the general behavior of the performance measures of a system with a change in system parameters is not affected by the method used. Since the finite truncation method is comparatively tractable compared to the others, we choose this method to derive the stationary distributions of the nonhomogeneous QBD with the generator matrix given by (4).

In the finite truncation method, the infinite generator matrix is truncated at a finite level K. That is, the system of equations given by $xQ = 0$ and $x\mathbf{e} = 1$ is truncated at a sufficiently large value, say K, and the resulting finite system is solved for the equilibrium probability vector. The level K is arbitrary but fixed and it is chosen such that customer loss probability due to truncation is small. As for higher dimension generator matrices, the level K is difficult to find analytically; a trial-and-error approach needs to be adopted. An appropriate level, say K_f, is determined by starting with a reasonable initial value for K and increasing it progressively until an appropriately chosen cut-off criterion is met. Stationary probability vector x can then be evaluated by an iterative method, such as that by Gauss-Seidel [43], which takes advantage of the sparsity and structure of Q. For each new value of K, the previously computed vector x is used as the initial solution to reduce the number of iterations required [42]. Thus, the numerical implementation of the approximation based on finite truncation implies the determination of an appropriate cut-off level K_f. Here, we use the algorithm given by Artalejo et al. [36], the steps of which are described below.

For K as the cut-off point, the modified generator will be

$$\widehat{Q}(K) = \begin{pmatrix} A_1^{(0)} & A_0^{(0)} & & & & & \\ A_2^{(1)} & A_1^{(1)} & A_0^{(1)} & & & & \\ & A_2^{(2)} & A_1^{(2)} & A_0^{(2)} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & A_2^{(K-1)} & A_1^{(K-1)} & A_0^{(K-1)} & \\ & & & & A_2^{(K)} & \phi_1^{(K)} \end{pmatrix}, \quad (9)$$

where $\phi_1^{(K)} = A_1^{(K)} + A_0^{(K)}$. Let π be the stationary distribution of $\widehat{Q}(K)$ which satisfies

$$\begin{aligned} \pi\widehat{Q}(K) &= 0, \\ \pi\mathbf{e} &= 1, \end{aligned} \quad (10)$$

where $\pi = [\pi(0), \pi(1), \ldots, \pi(K)]$, by aggregating terms of the QBD $\widehat{Q}(K)$, depending on levels. Define $z = [z_0(K), z_1(K)]$ with

$$\begin{aligned} z_0(K) &= [\pi(0), \pi(1), \ldots, \pi(K-1)], \\ z_1(K) &= \pi(K). \end{aligned} \quad (11)$$

And $z(K, i) = \pi(i)$, $0 \le i \le K$. Here, $z_0(K)$ is a row vector of dimension $m = nc(c+1)/2 + n(c+1)(K-c)$ and $z_1(K)$ is a row vector with dimension $n(c+1)$. By partitioning $\widehat{Q}(K)$ according to $z_0(K)$ and $z_1(K)$, we have

$$(z_0(K), z_1(K)) \begin{pmatrix} B_{00}(K) & B_{01}(K) \\ B_{10}(K) & B_{11}(K) \end{pmatrix} = (0_m, 0_{n(c+1)}), \quad (12)$$

where $B_{00}(K)$ is the matrix obtained by deleting the last column matrices and last row matrices from $\widehat{Q}(K)$, $B_{01}(K) = \text{trans}[0, 0, \ldots, 0, A_0^{(K-1)}]$, $B_{10}(K) = [0, 0, \ldots, 0, A_2^{(K)}]$, and $B_{11}(K) = \phi_1^{(K)}$. These are block structured matrices with $(K \times K)$, $(K \times 1)$, $(1 \times K)$, and (1×1) blocks, respectively. 0_m and $0_{n(c+1)}$ are row vectors of dimensions m and $n(c+1)$, respectively, with all entries equal to zero. From (12), we find that

$$z_1(K) B_{10}(K) B_{00}^{-1}(K) = -z_0(K), \quad (13)$$

$$z_1(K) \left[B_{11}(K) - B_{10}(K) B_{00}^{-1}(K) B_{01}(K)\right] = 0_{n(c+1)}. \quad (14)$$

Further, we can have

$$B_{00}(K) = \begin{pmatrix} B_{00}(K-1) & B_{01}(K-1) \\ C_0(K-1) & C_1(K-1) \end{pmatrix}, \quad (15)$$

where

$$\begin{aligned} C_0(K-1) &= \left[0_n, 0_{2n}, \ldots, 0, A_2^{(K-1)}\right], \\ C_1(K-1) &= -A_1^{(K-1)}. \end{aligned} \quad (16)$$

The inverse of matrix $B_{00}(K)$ can be determined, using methods given in Hunter [44] as

$$B_{00}^{-1}(K) = \begin{pmatrix} D_{00}(K) & D_{01}(K) \\ D_{10}(K) & D_{11}(K) \end{pmatrix}, \quad (17)$$

where

$$
\begin{aligned}
D_{00}(K) &= \left[B_{00}(K-1)\right. \\
&\quad \left. - B_{10}(K-1) C_1^{-1}(K-1) C_0(K-1)\right]^{-1}, \\
D_{10}(K) &= -C_1^{-1}(K-1) C_0(K-1) D_{00}(K), \\
D_{11}(K) &= \left[C_1(K-1)\right. \\
&\quad \left. - C_0(K-1) B_{00}^{-1}(K-1) B_{01}(K-1)\right]^{-1}, \\
D_{01}(K) &= -B_{00}^{-1}(K-1) B_{01}(K-1) D_{11}(K).
\end{aligned}
\tag{18}
$$

As the dimensions of the matrices increase in each iteration, the calculations to compute the above matrices involve multiplications and inversion of increasingly large matrices. If we exploit the structure of matrices in the above equations, we notice that the sparse blocks of $B_{01}(K-1)$ and $C_0(K-1)$ simplify the calculations. $B_{01}(K-1)$ has only one nonzero square matrix $A_0^{(K-1)}$ of dimension $n(c+1)$ in the last rows and $C_0(K-1)$ has one $A_2^{(K-1)}$ in the last columns. So $B_{00}^{-1}(K-1)B_{01}(K-1)$ can be written in simplified form as $[D_{10}(K-1), D_{11}(K-1)]A_0^{(K-1)}$. Further, $C_0(K-1)B_{00}^{-1}(K-1)B_{01}(K-1)$ becomes $A_2^{(K-1)}D_{11}(K-1)A_0^{(K-1)}$. These substitutions make the remaining operations in $D_{01}(K)$ and $D_{11}(K)$ simple, as they involve multiplications and inversion of only known simple matrices of size $n(c+1)$. The key step is to compute the matrix $D_{00}(K)$. The inverse in the definition of $D_{00}(K)$ can be computed by using small-rank adjustment; that is, if we have the inverse of a matrix A and we want the inverse of its adjustment $B = A + XWY$, where W is a matrix of smaller order than A, then we have

$$
B^{-1} = \left[I - A^{-1}X\left(W^{-1} + YA^{-1}X\right)^{-1}Y\right]A^{-1}. \tag{19}
$$

Here, we have $A = B_{00}(K-1)$, $X = -B_{01}(K-1)$, $W^{-1} = C_1^{-1}(K-1)$, and $Y = C_0(K-1)$. Thus, we obtain that

$$
\begin{aligned}
D_{00}(K) &= B^{-1} \\
&= \left[I - D_{01}(K) C_0(K-1)\right] B_{00}^{-1}(K-1),
\end{aligned}
\tag{20}
$$

so D_{00} is obtained by multiplications and additions of already computed matrices. Finally, we have

$$
\begin{aligned}
&D_{11}(K) \\
&\quad = \left[C_1(K-1) - A_2^{(K-1)} D_{11}(K-1) A_0^{(K-1)}\right]^{-1}, \\
&D_{01}(K) \\
&\quad = -\left[D_{10}(K-1), D_{11}(K-1)\right] A_0^{(K-1)} D_{11}(K),
\end{aligned}
$$

```
(1)  K := c + 1;
(2)  compute B_00^{-1}(K)
(3)  compute z_1(K) by (14) and (22)
(4)  compute z_0(K) by (13)
(5)  store B_00(K), B_00^{-1}(K), B_01(K) and B_10(K)
(6)  K := K + 1
(7)  while K ≤ K_f
(8)    do
(9)      compute B_00^{-1}(K) by (17)
(10)     compute z_1(K) by (14) and (22)
(11)     compute z_0(K) by (13)
(12)     update B_00(K), B_00^{-1}(K), B_01(K) and B_10(K)
(13)     if max_{0≤i≤K} ‖z(K,i) − z(K−1,i)‖_∞ < ε
(14)       then K_f = K
(15)            break
(16)       else K := K + 1
```

Algorithm 1: Finite truncation method.

$$
\begin{aligned}
D_{00}(K) &= \left[I - D_{01}(K) C_0(K-1)\right] B_{00}^{-1}(K-1), \\
D_{10}(K) &= -C_1^{-1}(K-1) C_0(K-1) D_{00}(K).
\end{aligned}
\tag{21}
$$

So, the computation of vector $z_1(K)$ is reduced to solving system (14) subject to the normalization condition

$$
\pi(K)\left[\mathbf{e}_{n(c+1)} - B_{10}(K) B_{00}^{-1}(K) \mathbf{e}_{nK(c+1)}\right] = 1, \tag{22}
$$

where $\mathbf{e}_{n(c+1)}$ and $e_{nK(c+1)}$ are column vectors of dimensions $n(c+1)$ and $nK(c+1)$, respectively, with all entries equal to one. Finally, the vector $z_0(K)$ can be solved substituting $z_1(K)$ in (13). To get the cut-off value, successive increments of K are made, starting from $K = c+1$, and we stop at the point $K = K_f$ when

$$
\max_{0 \le i \le K_f} \left\| z\left(K_f, i\right) - z\left(K_f - 1, i\right) \right\|_\infty < \epsilon, \tag{23}
$$

where ϵ is an infinitesimal quantity and $\|\cdot\|_\infty$ is the infinity norm. The whole method of computing the stationary distribution using the finite truncation method is summarized in Algorithm 1.

4. Performance Measures

The performance measures give the qualitative behavior of the model under study. In a multiserver queueing model, the efficiency of the model depends upon the mean number of busy servers, the mean queue length, the blocking probability, and the mean number of customers lost due to impatience.

In our model, the server serves even during its vacation. Therefore, the number of busy servers will be i, $0 \le i \le c$, if

there are i customers in the system and when the system has more than c customers, all the servers will be busy serving customers either in WV or in nonvacation with rates μ_v and μ_b, respectively. The mean number of servers busy in nonvacation is

$$B_s = \sum_{i=0}^{c-1} \sum_{j=1}^{i} j x_{ij} \mathbf{e}_n + \sum_{i=c}^{\infty} \sum_{j=1}^{c} j x_{ij} \mathbf{e}_n. \tag{24}$$

The mean queue length of the system under study is

$$L = E(N) = \sum_{i=1}^{c-1} i x_i \mathbf{e}_{(i+1)n} + \sum_{i=c}^{\infty} i x_i \mathbf{e}_{(c+1)n}. \tag{25}$$

Availability of the server, R, is the probability that an arrival finds a server free. It can happen only if the number of total customers in the system in less than c and is given by

$$R = P(N < c) = \sum_{i=0}^{c-1} x_i \mathbf{e}_{(i+1)n}. \tag{26}$$

The blocking probability of a multiserver queue is the probability of refraining a customer from service. In our model, a customer is kept waiting in the queue for service when all the servers are in busy state, either in WV or in nonvacation, that is, when the number of customers in the system is more than c:

$$B_p = P(N > c) = 1 - \sum_{i=0}^{c-1} x_i \mathbf{e}_{(i+1)n} = 1 - R. \tag{27}$$

The mean number of customers lost by the system is the average of customers who have abandoned the system as a result of waiting in a queue ($i > c$) with all servers in WV ($j = 0$); therefore,

$$N_c = \sum_{i=c+1}^{\infty} i x_{i0} \mathbf{e}_n. \tag{28}$$

5. Numerical Examples

Let us illustrate the behavior of our PH/M/c/WV queue with the help of some numerical examples. Algorithm 1 is coded in MATLAB©. The algorithm computes the stationary distribution and its main objective is to find the termination criteria of the level K_f. We start with an initial value $K \geq c + 1$ and progressively increase the value of K until a change in the stationary probability z is sufficiently small due to increased K. We choose the smallest value of K_f such that $\max_{0 \leq i \leq K_f} \| z(K_f, i) - z(K_f - 1, i) \|_\infty < \epsilon$, for $\epsilon = 10^{-6}$. With this selection criterion, we find the values of K_f, the mean queue lengths, and the blocking probabilities for various sets of parameter values and for different arrival processes. Here, we take some examples (from Chakravarthy et al. [37]) of well known distributions and give their PH representations below:

(1) Exponential (Exp):

$$\begin{aligned} T &= -1, \\ T^0 &= 1, \\ \alpha &= 1. \end{aligned} \tag{29}$$

(2) Erlang-2 (Erl):

$$\begin{aligned} T &= \begin{pmatrix} -2 & 2 \\ 0 & -2 \end{pmatrix}, \\ T^0 &= \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \\ \alpha &= \begin{bmatrix} 1 & 0 \end{bmatrix}. \end{aligned} \tag{30}$$

(3) Hyperexponential-2 (Hyp):

$$\begin{aligned} T &= \begin{pmatrix} -1.9 & 0 \\ 0 & -0.19 \end{pmatrix}, \\ T^0 &= \begin{pmatrix} 1.9 \\ 0.19 \end{pmatrix}, \\ \alpha &= \begin{bmatrix} 0.9 & 0.1 \end{bmatrix}. \end{aligned} \tag{31}$$

All these PH distributions have the same mean arrival rate $\lambda = 1$. The standard deviations of the three distributions are 1.0, 0.70711, and 2.24472, respectively. The service rate during nonvacation period, μ_b, is calculated for specific values of ρ using the formula $\rho = \lambda / c\mu_b$. We have chosen $\rho = 0.1$, 0.5, and 0.9 for given values of c ($c = 1, 3, 6$). The effect of parameters on system performance is illustrated here. We will mention the models having interarrivals as exponential, Erlang, and hyperexponentially distributed as exponential model, Erlang model, and hyperexponential model, respectively.

5.1. Effect on Cut-Off Value K_f. We have illustrated here the effect of the parameters, namely, traffic intensity (ρ), rate of vacation duration (θ), service rate during WV (μ_v), and the type of arrival process, on the truncation cut-off value K_f. For three different values of c ($c = 1, 3, 6$), different tables are presented. The impatient rate is fixed at $\xi = 0.1$ for the tables. We have the following observations from Tables 1, 2, and 3:

(1) The cut-off value increases with the increase in the variance of the distribution of the interarrival times. For Erlang model, the termination is the fastest whereas for the hyperexponential one it is the slowest. This behavior seems to be the same for all sets of parameter values and for all c.

Table 1: Multiserver model with $c = 1$.

ρ	θ	μ_v	K_f			L			B_p		
			Erl	Exp	Hyp	Erl	Exp	Hyp	Erl	Exp	Hyp
0.1	0.1	0.0	25	28	36	6.4229	6.4127	6.3354	0.9354	0.9143	0.7955
		3.0	10	13	18	1.3951	1.4544	1.5802	0.3211	0.3173	0.3088
		6.0	7	9	13	1.1771	1.1963	1.2238	0.1652	0.1647	0.1639
		9.0	6	8	11	1.1149	1.1244	1.1358	0.1109	0.1108	0.1107
	1	0.0	14	16	22	2.0525	2.0533	2.0481	0.6062	0.5479	0.4325
		3.0	8	11	15	1.3040	1.3342	1.3793	0.2650	0.2562	0.2421
		6.0	6	8	12	1.1656	1.1811	1.2011	0.1561	0.1545	0.1524
		9.0	6	7	11	1.1137	1.1229	1.1337	0.1099	0.1097	0.1094
	100	0.0	5	6	10	1.1131	1.1215	1.1305	0.1097	0.1089	0.1082
		3.0	5	6	10	1.1099	1.1184	1.1273	0.1066	0.1061	0.1056
		6.0	5	6	10	1.1068	1.1153	1.1243	0.1037	0.1034	0.1031
		9.0	5	6	10	1.1040	1.1124	1.1214	0.1009	0.1008	0.1008
0.5	0.1	0.0	25	28	42	6.4099	6.5109	6.9481	0.9579	0.9412	0.8236
		0.5	21	25	41	4.4283	4.5761	5.2829	0.9062	0.8788	0.7457
		1.0	17	21	40	2.9028	3.1115	3.9708	0.7716	0.7451	0.6520
		1.5	15	20	38	2.1227	2.3112	3.0542	0.6121	0.5987	0.5579
	1	0.0	17	19	38	2.7191	2.8952	3.6542	0.7863	0.7442	0.6270
		0.5	16	21	38	2.3956	2.5839	3.3851	0.7113	0.6784	0.5949
		1.0	16	21	38	2.1386	2.3326	3.1440	0.6331	0.6121	0.5616
		1.5	15	20	38	1.9441	2.1339	2.9316	0.5605	0.5505	0.5280
	100	0.0	14	18	37	1.8203	2.0117	2.8092	0.5061	0.5049	0.5027
		0.5	14	20	37	1.8178	2.0080	2.8066	0.5046	0.5037	0.5021
		1.0	15	20	37	1.8146	2.0055	2.8041	0.5030	0.5025	0.5014
		1.5	15	20	37	1.8121	2.0030	2.8016	0.5015	0.5012	0.5007
0.9	0.1	0.0	74	99	246	11.2890	13.2254	28.5672	0.9893	0.9846	0.9509
		0.3	73	98	253	10.0678	11.9600	27.1965	0.9805	0.9735	0.9350
		0.6	71	97	260	8.7709	10.6206	25.7831	0.9593	0.9495	0.9139
		0.9	75	102	265	7.4716	9.2882	24.3478	0.9142	0.9057	0.8880
	1	0.0	73	96	231	8.7160	10.8465	27.2102	0.9574	0.9479	0.9187
		0.3	72	96	233	8.4580	10.5880	26.9545	0.9445	0.9361	0.9135
		0.6	72	96	234	8.2028	10.3341	26.7009	0.9289	0.9228	0.9080
		0.9	71	95	235	7.9573	10.0868	26.4490	0.9111	0.9080	0.9022
	100	0.0	81	110	265	7.8679	10.0268	26.8945	0.9013	0.9010	0.9004
		0.3	81	110	265	7.8651	10.0240	26.8905	0.9010	0.9007	0.9003
		0.6	81	110	265	7.8623	10.0213	26.8865	0.9006	0.9005	0.9002
		0.9	81	110	265	7.8596	10.0185	26.8825	0.9003	0.9002	0.9001

(2) For a particular arrival process when the traffic load is small, the value of K_f decreases with increase in μ_v and also with the increase in θ. But for high ρ (> 0.5) and high θ, it shows the reverse property for all arrival processes and for any number of servers c; that is, when the system load is heavy and the system has small vacation duration, the cut-off value seems to be high for all types of arrival processes and any number of servers.

(3) When the vacation duration rate θ is too high (=100), K_f value remains unaffected by vacation-service rate μ_v for any number of servers.

These observations show that the cut-off value depends on the system parameters and also on the arrival process but becomes independent of vacation-service rates when we have systems with small vacation duration.

Table 2: Multiserver model with $c = 3$.

ρ	θ	μ_v	K_f			L			B_p		
			Erl	Exp	Hyp	Erl	Exp	Hyp	Erl	Exp	Hyp
0.1	0.1	0.0	22	25	33	4.1401	4.1552	4.1155	0.4953	0.4909	0.4452
		1.0	10	13	18	1.8312	1.8449	1.8849	0.0376	0.0606	0.1064
		2.0	8	10	13	1.4721	1.4743	1.4805	0.0050	0.0133	0.0294
		3.0	7	8	11	1.3301	1.3307	1.3326	0.0013	0.0049	0.0117
	1	0.0	9	12	16	1.6336	1.6535	1.6721	0.0203	0.0421	0.0730
		1.0	7	10	13	1.4723	1.4792	1.4875	0.0056	0.0163	0.0343
		2.0	7	9	12	1.3787	1.3811	1.3847	0.0022	0.0078	0.0181
		3.0	6	8	11	1.3165	1.3172	1.3189	0.0011	0.0044	0.0107
	100	0.0	6	7	11	1.3037	1.3051	1.3062	0.0009	0.0039	0.0096
		1.0	6	7	11	1.3026	1.3039	1.3049	0.0009	0.0039	0.0095
		2.0	6	7	11	1.3015	1.3027	1.3036	0.0009	0.0038	0.0093
		3.0	6	7	11	1.3005	1.3016	1.3023	0.0009	0.0037	0.0092
0.5	0.1	0.0	36	41	58	6.1463	6.5540	8.1834	0.7544	0.7535	0.7094
		0.2	26	30	48	4.4310	4.7592	6.1809	0.5820	0.5952	0.6062
		0.4	20	24	43	3.3472	3.5687	4.6449	0.3792	0.4116	0.4878
		0.6	16	20	39	2.7521	2.8807	3.5933	0.2308	0.2694	0.3736
	1	0.0	17	19	37	3.1041	3.2843	4.0155	0.3311	0.3663	0.4414
		0.2	16	19	37	2.9373	3.0968	3.8141	0.2832	0.3217	0.4116
		0.4	17	21	37	2.7875	2.9234	3.6269	0.2421	0.2821	0.3825
		0.6	16	21	37	2.6620	2.7814	3.4534	0.2074	0.2473	0.3544
	100	0.0	16	19	37	2.6304	2.7513	3.4302	0.1989	0.2385	0.3468
		0.2	16	21	37	2.6286	2.7437	3.4280	0.1984	0.2380	0.3463
		0.4	16	21	37	2.6267	2.7418	3.4257	0.1978	0.2375	0.3459
		0.6	16	21	37	2.6249	2.7399	3.4234	0.1973	0.2370	0.3455
0.9	0.1	0.0	78	100	224	14.0668	16.3843	33.6740	0.9618	0.9582	0.9403
		0.1	75	98	224	12.7466	14.9913	31.9614	0.9424	0.9390	0.9251
		0.2	73	95	229	11.3801	13.5544	30.1618	0.9089	0.9077	0.9052
		0.3	71	93	234	9.9871	12.0651	28.2970	0.8541	0.8593	0.8796
	1	0.0	76	101	239	9.7276	11.7783	28.3667	0.8612	0.8643	0.8806
		0.1	76	100	238	9.5526	11.6045	28.1630	0.8466	0.8523	0.8758
		0.2	75	100	237	9.3873	11.4217	27.9598	0.8312	0.8397	0.8708
		0.3	75	99	237	9.2075	11.2421	27.7421	0.8149	0.8265	0.8656
	100	0.0	82	111	269	9.1219	11.1422	29.3701	0.8041	0.8176	0.8622
		0.1	82	111	269	9.1194	11.1398	29.3614	0.8038	0.8175	0.8621
		0.2	82	111	269	9.1169	11.1375	29.3527	0.8036	0.8173	0.8621
		0.3	82	111	269	9.1144	11.1351	29.3440	0.8033	0.8171	0.8620

5.2. Effect on Mean Queue Length

(1) The mean queue length of the system depends upon the arrival process. Tables 1, 2, and 3 show that systems with interarrival distributions of high variance have higher number of customers in the queue for any number of servers. We have fixed the impatient rate at $\xi = 0.1$. For $c = 1, 3, 6$ with $\rho = 0.5$, we have Figures 1, 2, and 3, respectively, and with $\rho = 0.9$, we have Figures 4, 5, and 6, where the changes in mean queue lengths are given for increasing vacation-service rates. A hyperexponential model always has the highest mean queue length compared to the corresponding Erlang and exponential models, irrespective of the number of servers.

(2) When the traffic load is heavy, $\rho = 0.9$, increase in vacation-service rate does not affect the mean

Table 3: Multiserver model with $c = 6$.

ρ	θ	μ_v	K_f			L			B_p		
			Erl	Exp	Hyp	Erl	Exp	Hyp	Erl	Exp	Hyp
0.1	0.1	0.0	20	23	32	3.2118	3.2976	3.4094	0.0679	0.0887	0.1405
		0.5	11	13	18	2.1893	2.2065	2.2152	0.0008	0.0034	0.0159
		1.0	9	11	14	1.8365	1.8407	1.8401	0.0000	0.0003	0.0024
		1.5	8	9	12	1.6458	1.6464	1.6462	0.0000	0.0001	0.0005
	1	0.0	8	11	15	1.7766	1.7958	1.8184	0.0000	0.0005	0.0035
		0.5	8	10	14	1.7120	1.7219	1.7334	0.0000	0.0002	0.0016
		1.0	8	9	13	1.6587	1.6631	1.6680	0.0000	0.0001	0.0008
		1.5	8	9	12	1.6136	1.6145	1.6155	0.0000	0.0000	0.0004
	100	0.0	8	8	12	1.6022	1.6050	1.6039	0.0000	0.0000	0.0004
		0.5	8	8	12	1.6016	1.6041	1.6028	0.0000	0.0000	0.0004
		1.0	8	8	12	1.6009	1.6032	1.6016	0.0000	0.0000	0.0004
		1.5	8	8	12	1.6002	1.6024	1.6005	0.0000	0.0000	0.0004
0.5	0.1	0.0	30	34	52	6.3770	6.8052	8.4877	0.4072	0.4526	0.5456
		0.1	24	30	46	5.3485	5.6091	6.9287	0.2546	0.3073	0.4411
		0.2	21	25	41	4.6431	4.8028	5.7143	0.1445	0.1924	0.3376
		0.3	18	22	39	4.1697	4.2487	4.8192	0.0797	0.1170	0.2476
	1	0.0	18	20	39	4.5180	4.6593	5.1256	0.1208	0.1617	0.2889
		0.1	18	20	39	4.3618	4.4869	4.9560	0.1007	0.1399	0.2679
		0.2	19	23	38	4.2119	4.2803	4.8001	0.0837	0.1208	0.2477
		0.3	18	22	38	4.0862	4.1494	4.6449	0.0696	0.1042	0.2284
	100	0.0	17	20	38	4.0659	4.1500	4.6956	0.0662	0.1001	0.2233
		0.1	17	20	38	4.0633	4.1474	4.6923	0.0659	0.0998	0.2230
		0.2	18	23	38	4.0514	4.1099	4.6889	0.0657	0.0995	0.2226
		0.3	18	23	38	4.0489	4.1074	4.6856	0.0655	0.0993	0.2223
0.9	0.1	0.00	78	101	229	15.1572	17.3703	35.4516	0.9045	0.9054	0.9082
		0.05	76	99	227	14.2268	16.3514	34.0044	0.8719	0.8759	0.8903
		0.10	75	98	225	13.1999	15.2452	32.4807	0.8265	0.8360	0.8684
		0.15	73	96	222	12.1491	14.0835	30.9017	0.7663	0.7840	0.8418
	1	0.00	79	105	246	12.0928	13.9233	31.2097	0.7748	0.7881	0.8391
		0.05	79	104	245	11.9039	13.7614	30.9647	0.7595	0.7758	0.8343
		0.10	78	103	245	11.7512	13.5946	30.6805	0.7436	0.7629	0.8292
		0.15	78	103	244	11.5535	13.4025	30.4553	0.7270	0.7496	0.8241
	100	0.00	84	114	272	11.5312	13.3305	35.6288	0.7160	0.7407	0.8207
		0.05	84	114	272	11.5273	13.3271	35.6049	0.7157	0.7405	0.8206
		0.10	84	114	272	11.5235	13.3237	35.5810	0.7155	0.7403	0.8205
		0.15	84	113	272	11.5196	13.3446	35.5572	0.7152	0.7401	0.8204

queue lengths, regardless of the arrival process or the number of servers (Figures 4, 5, and 6).

(3) For $\rho = 0.1$ and $c = 1, 3, 6$, we plot Figures 7, 8, and 9, respectively. Here, the hyperexponential model has the least queue length compared to the corresponding Erlang and exponential models. For $c = 6$, the queue lengths are the same and all of them are shown in Figure 9. Also, it can be seen that, for increased vacation-service rates, arrival processes do not have much influence on mean queue lengths.

(4) The impatient rate ξ affects the queue lengths significantly, especially when $\rho = 0.1$. In Figures 10, 11, and 12, the change in queue lengths with the increase in impatient rate is shown. When the

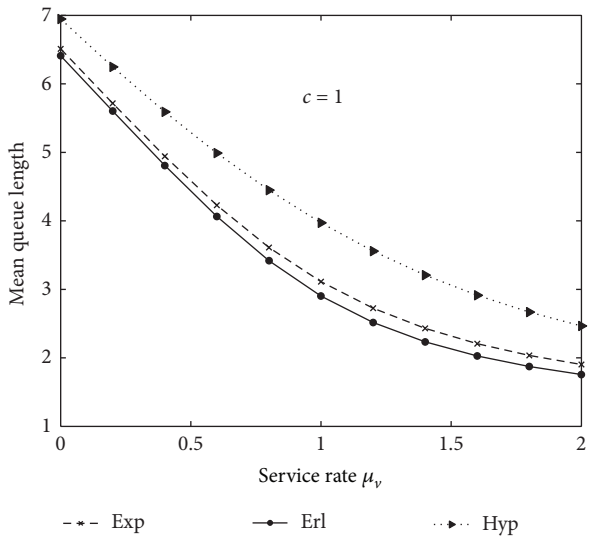

Figure 1: Mean queue length versus vacation-service rate with $\rho = 0.5$, $\xi = 0.1$, $\theta = 0.1$, and $c = 1$.

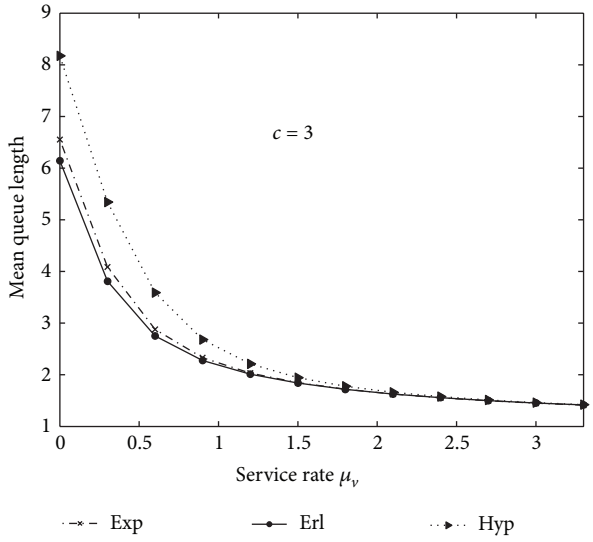

Figure 2: Mean queue length versus vacation-service rate with $\rho = 0.5$, $\xi = 0.1$, $\theta = 0.1$, and $c = 3$.

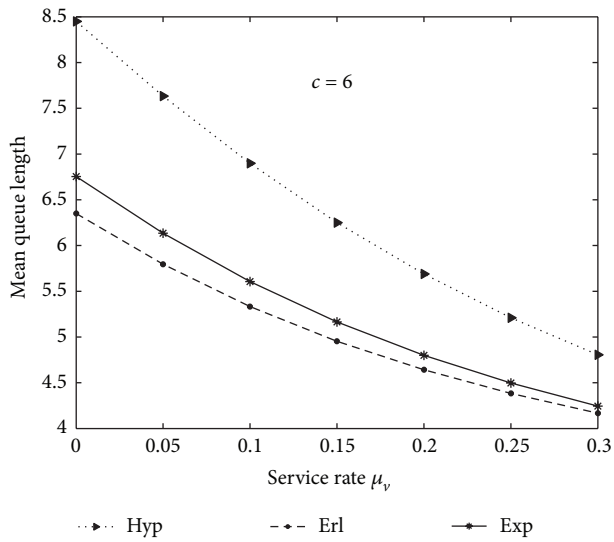

Figure 3: Mean queue length versus vacation-service rate with $\rho = 0.5$, $\xi = 0.1$, $\theta = 0.1$, and $c = 6$.

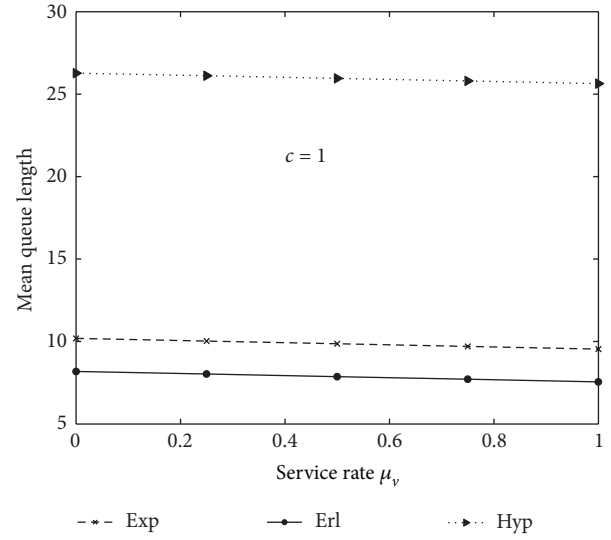

Figure 4: Mean queue length versus vacation-service rate with $\rho = 0.9$, $\xi = 1$, $\theta = 1$, and $c = 1$.

impatient rate is small, the mean queue length for Erlang model becomes minimum. As the impatient rate increases, it shows the reverse behavior. The point of inflection depends upon the service rate μ_v. But the impatient rate does not have much effect on queue lengths when the arrival process is Erlang, whereas for hyperexponential model, the mean queue length decreases significantly with the increase in impatient rate.

Therefore, systems with hyperexponential arrivals have the longest queues compared to corresponding Erlang or exponential arrivals. For light loaded systems ($\rho = 0.1$) and highly impatient customers ($1/\xi < 10$), hyperexponential arrivals give the minimum queue lengths. When the customer impatient rates are small, the system behavior depends on the vacation-service rates.

5.3. Effect on Blocking Probability. From the tables and the graphs plotted for blocking probability, we have seen the following properties for the models under study:

(1) Figure 13 gives that for a single-server system the blocking probability of a hyperexponential model is minimum and that for Erlang is maximum while $\theta = 1$, $\xi = 1$, and $\rho = 0.9$. This behavior is also observed in multiserver models when θ is too small. For higher values of θ, multiserver models follow the reverse nature; that is, Erlang model gives

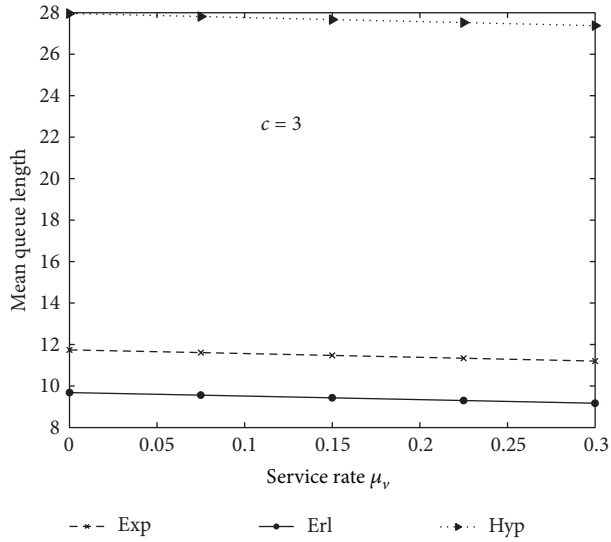

Figure 5: Mean queue length versus vacation-service rate with $\rho = 0.9$, $\xi = 1$, $\theta = 1$, and $c = 3$.

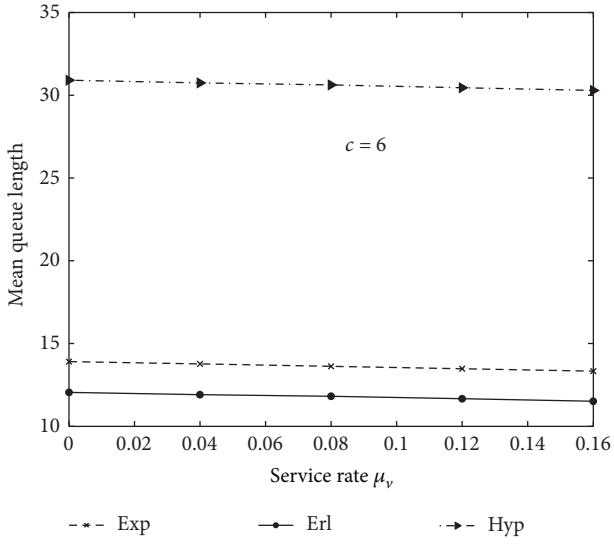

Figure 6: Mean queue length versus vacation-service rate with $\rho = 0.9$, $\xi = 1$, $\theta = 1$, and $c = 6$.

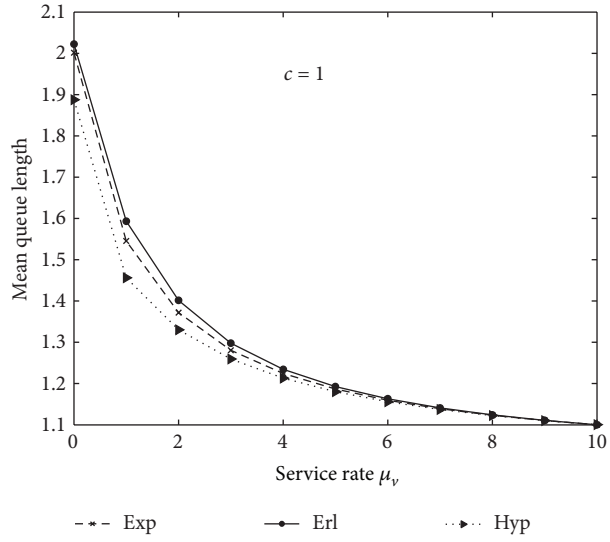

Figure 7: Mean queue length versus vacation-service rate with $\rho = 0.1$, $\xi = 10$, $\theta = 0.1$, and $c = 1$.

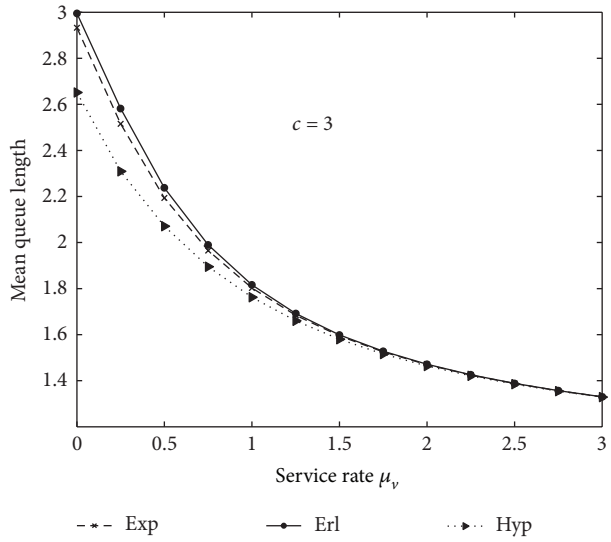

Figure 8: Mean queue length versus vacation-service rate with $\rho = 0.1$, $\xi = 10$, $\theta = 0.1$, and $c = 3$.

the minimum queue length and the hyperexponential one gives the maximum of those three different arrival models. That is, the chance of blocking a customer with Poisson arrival in a single-server as well as in a multiserver queue is always sandwiched between those with Erlang and hyperexponential arrivals.

(2) When we have a single-server Erlang model, the blocking probabilities seem to reduce up to 6% with an increased rate of service during vacation. Because of a single-server queue, the server rarely goes to vacation (as the systems are rarely empty) and the customers are served at a higher rate most of the times. And even when the system goes to vacation, because of the single-server queue, the queue started to form rapidly making the customers impatient and leave the system more often than a multiserver model. These contribute significantly to dropping the blocking probability in a single-server queue as seen in the plot.

(3) Figures 14 and 15 show that the hyperexponential model is not much affected by the vacation-service rate, whereas the Erlang model can reduce the blocking probability by up to 4% for increased vacation-service rates.

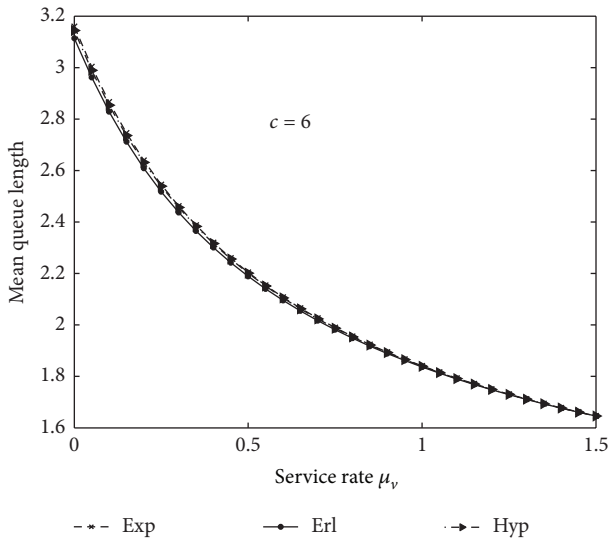

FIGURE 9: Mean queue length versus vacation-service rate with $\rho = 0.1$, $\xi = 10$, $\theta = 0.1$, and $c = 6$.

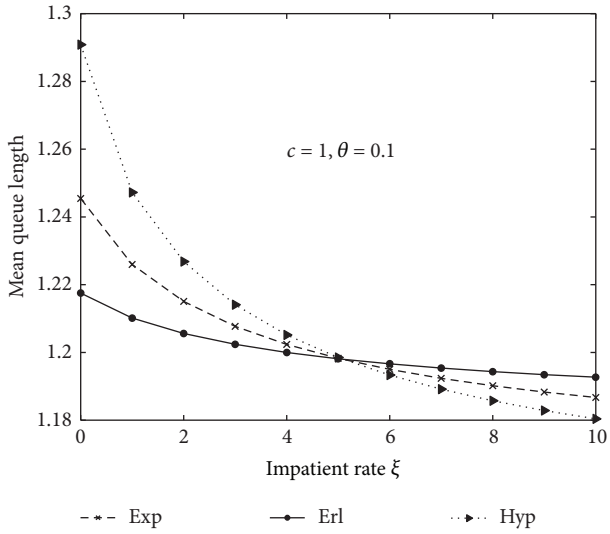

FIGURE 10: Mean queue length versus impatient rate with $\rho = 0.1$, $\mu_v = 0.5$, and $c = 1$.

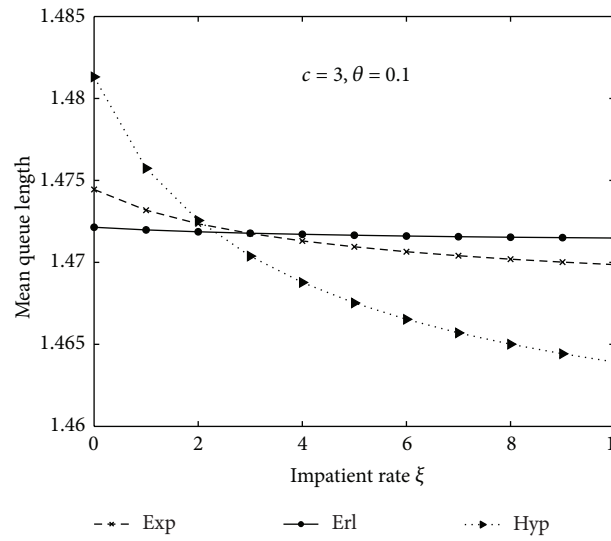

FIGURE 11: Mean queue length versus impatient rate with $\rho = 0.1$, $\mu_v = 0.2$, and $c = 3$.

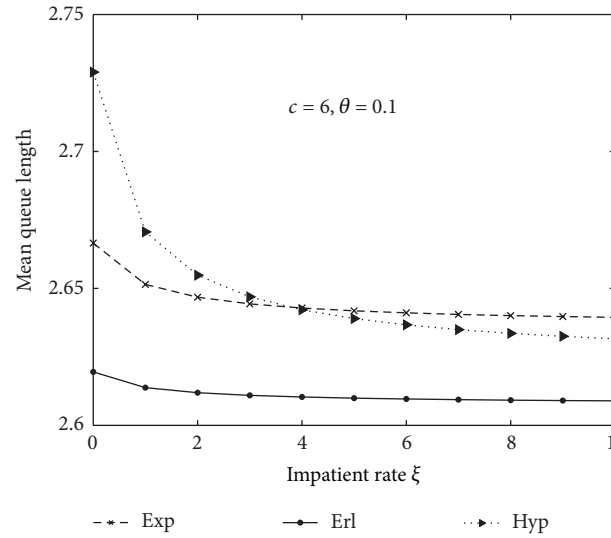

FIGURE 12: Mean queue length versus impatient rate with $\rho = 0.1$, $\mu_v = 0.2$, and $c = 6$.

A model with $c = 1$ and Erlang-2 arrival process has the maximum chance to make a customer wait in queue compared to exponential and hyperexponential arrivals, but as the number of servers is increased, hyperexponential arrival model has the highest blocking probability. The exponential arrival model always remains in between these two.

5.4. Average Number of Servers Busy in Nonvacation. The mean number of servers that are in working status during nonvacation period is shown in subsequent figures:

(1) Figure 16 is a plot of blocking probabilities with changing θ. Here, for an increase in vacation duration, the number of servers that remain busy is high, because the servers serve at a low rate but for longer time, and a new arrival will be served by an idle server if any. Consequently, it increases the number of busy servers in the system.

(2) Figure 17 shows that if the service rate is fast, customers are served at a faster rate which results in a lower number of busy servers in nonvacation period. This is true for all the three types of arrival models.

(3) The impatience makes a customer leave the system unserved and for high impatient rates more servers remain idle (Figure 18). But if the impatient rate is increased beyond a certain value ($\xi > 6$), the mean number of busy servers remains unaffected.

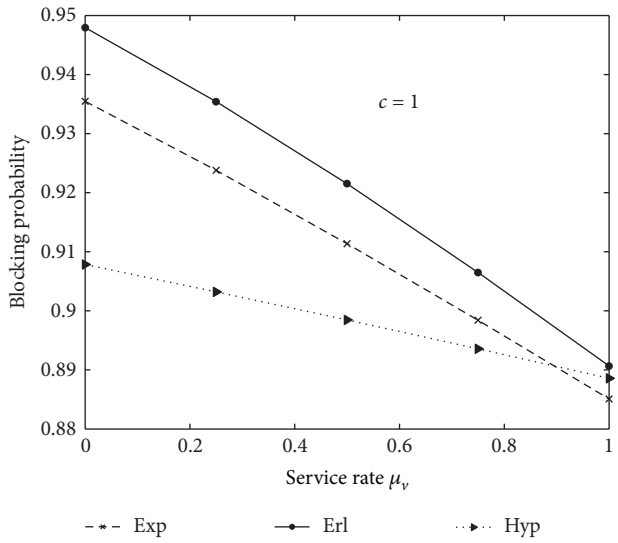

Figure 13: Blocking probability versus vacation-service rate with $\rho = 0.9, \theta = 1, \xi = 1$, and $c = 1$.

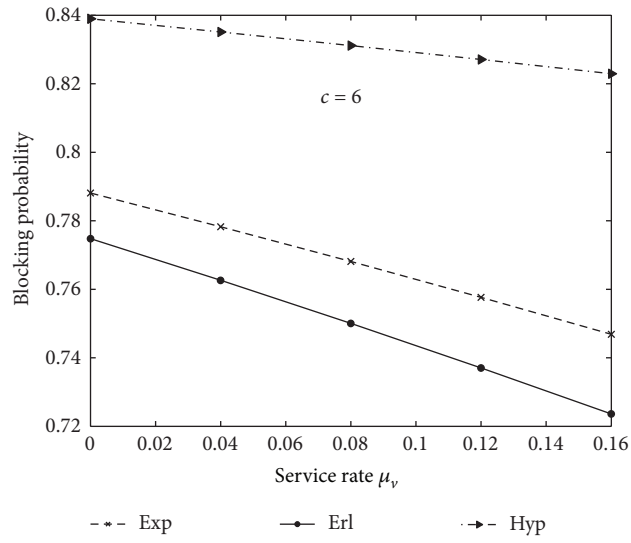

Figure 15: Blocking probability versus vacation-service rate with $\rho = 0.9, \theta = 1, \xi = 1$, and $c = 6$.

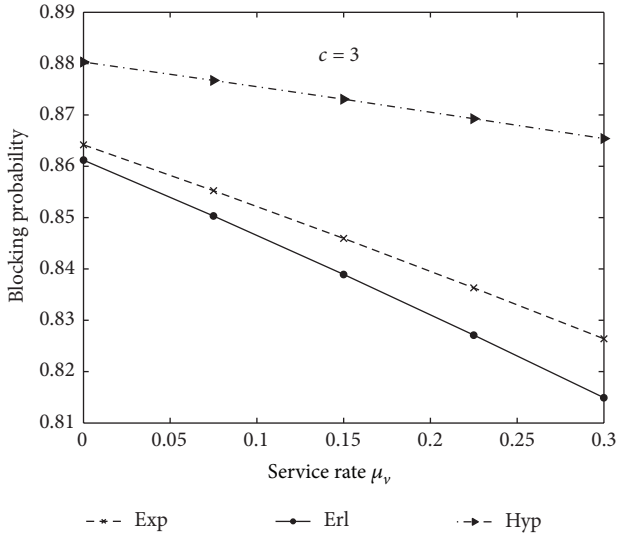

Figure 14: Blocking probability versus vacation-service rate with $\rho = 0.9, \theta = 1, \xi = 1$, and $c = 3$.

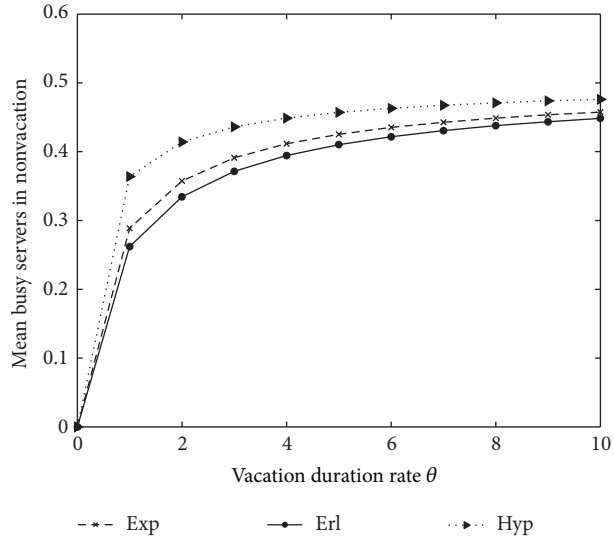

Figure 16: Mean number of busy servers versus vacation duration rate with $c = 1$, $\rho = 0.5$, $\xi = 0.1$, $\mu_b = 10$, and $\mu_v = 0.4$.

5.5. Average Customer Loss. We plot the mean number of customers who abandon the queue without getting served in Figures 19 and 20. The values of θ for these plots are $\theta = 0.1$ and $\theta = 1$, respectively, keeping the other parameters fixed for both cases.

When $\theta = 0.1$, that is, the system has longer vacations, the number of lost customers is less compared to the corresponding model for $\theta = 1$. In both cases, the hyperexponential models have the maximum customer loss, which is up to 60% more than the Erlang model. Also, the effect of impatient rates on customer loss is negligible for a system having small vacation duration.

For Figure 19, when vacation duration rate $\theta = 0.1$, we can see local maxima for Erlang and exponential models but minima for hyperexponential one at the point where impatient rate is equal to one. From Figure 18, we can see that when $c = 3$, the number of busy servers in nonvacation drops sharply until impatient rate becomes one and then it is almost consistent thereafter. This drop is more significant for hyperexponential model. This suggests that as impatient rates increase from zero to one, the number of busy servers in nonvacation period becomes less, which increases the probability of losing more customers for Erlang and exponential models. As the impatient rate increases beyond one, the number of busy servers in nonvacation remains consistent and the loss of customer is influenced mainly by the increased rate of impatience. But for the hyperexponential model, this behaviour alters because of the

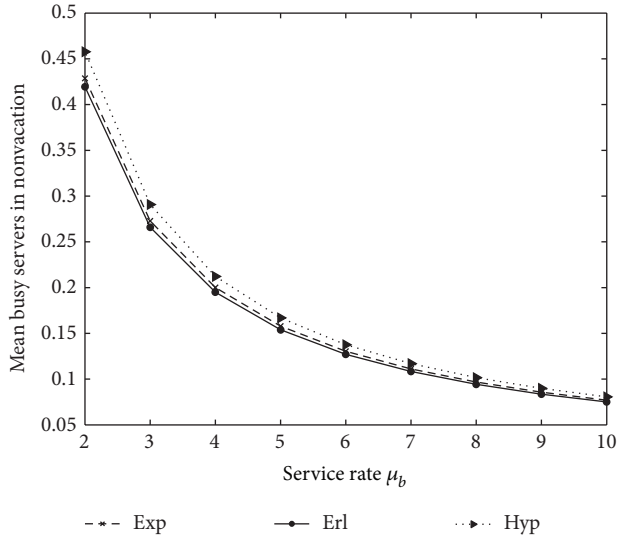

Figure 17: Mean number of busy servers versus service rate with $c = 1$, $\rho = 0.5$, $\xi = 0.1$, and $\mu_v = 0.4$.

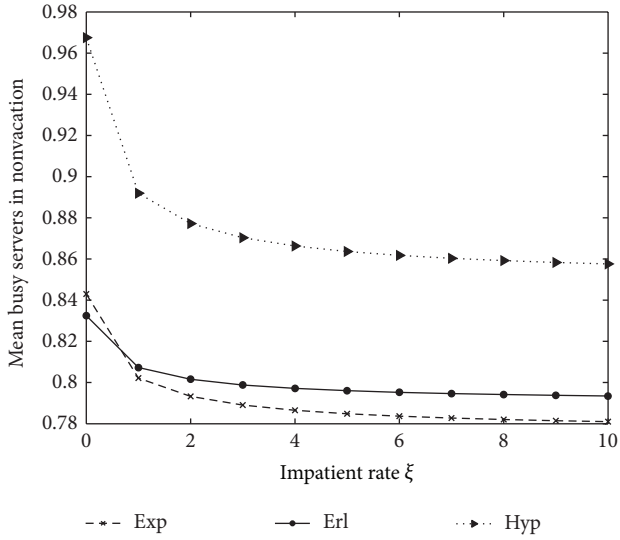

Figure 18: Mean number of busy servers versus impatient rate with $c = 3$, $\rho = 0.5$, $\mu_v = 0.4$, and $\theta = 0.1$.

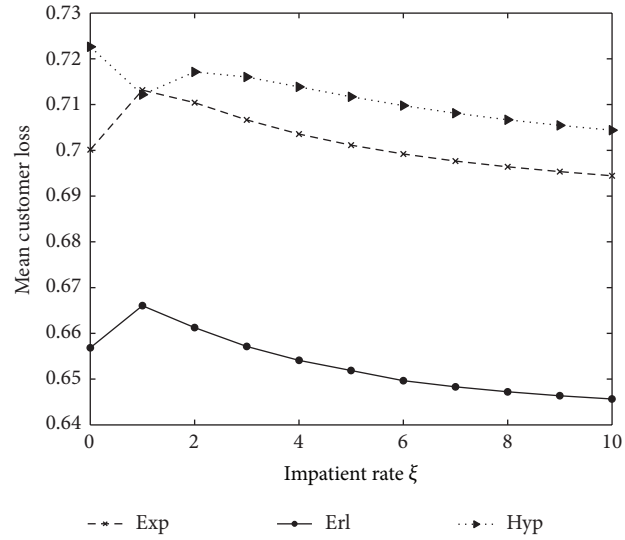

Figure 19: Mean customer loss versus impatient rate with $c = 3$, $\rho = 0.5$, $\mu_v = 0.4$, and $\theta = 0.1$.

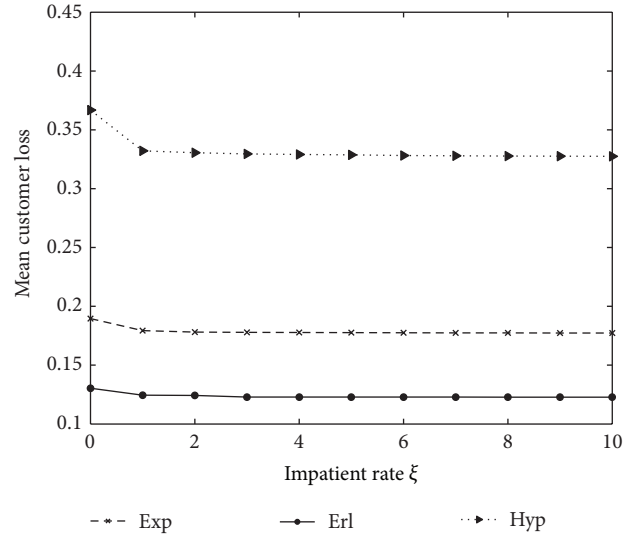

Figure 20: Mean customer loss versus impatient rate with $c = 3$, $\rho = 0.5$, $\mu_v = 0.4$, and $\theta = 1$.

change in mean queue lengths with the change of impatient rates (Section 5.2(4)). Its influence can be seen more towards the point of inflection where the blocking probability of the hyperexponential model becomes the same as that of the exponential one. But as the impatient rate increases, the customers leave the system, increasing the customer loss.

Thus, we have seen the role of various parameters on system performances and we are now in a position to handle them to enhance the system efficiency.

6. Conclusion

In this paper, we have analyzed the nonhomogeneous QBD model of a PH/M/c queue with impatient customers and multiple working vacations. We have used the finite truncation method to determine the stationary distribution. The effects of system parameters on the performance measures of the model are illustrated with the help of some numerical examples. Comparisons are made for different interarrival time distributions and the effects of the parameters on those distributions are also presented.

Disclosure

The present address of Cosmika Goswami is School of Engineering, University of Glasgow, Rankine Building, Glasgow G12 8LT, UK.

Competing Interests

The authors declare that there are no competing interests.

References

[1] Y. Levy and U. Yechiali, "An M/M/c queue with servers vacations," *INFOR*, vol. 14, no. 2, pp. 153–163, 1976.

[2] N. Tian and Q. Li, "The M/M/c queue with PH synchronous vacations," *System in Mathematical Science*, vol. 13, no. 1, pp. 7–16, 2000.

[3] N. Tian, Q.-L. Li, and J. Cao, "Conditional stochastic decompositions in the M/M/c queue with server vacations," *Communications in Statistics. Stochastic Models*, vol. 15, no. 2, pp. 367–377, 1999.

[4] N. Tian and Z. Zhang, "M/M/c queue with synchronous vacations of some servers and its application to electronic commerce operations," Working Paper, 2000.

[5] N. Tian and Z. Zhang, "A two threshold vacation policy in multiserver queueing systems," *European Journal of Operational Research*, vol. 168, pp. 153–163, 2006.

[6] X. Yang and A. S. Alfa, "A class of multi-server queueing system with server failures," *Computers and Industrial Engineering*, vol. 56, no. 1, pp. 33–43, 2009.

[7] S. R. Chakravarthy, "Analysis of a multi-server queue with Markovian arrivals and synchronous phase type vacations," *Asia-Pacific Journal of Operational Research*, vol. 26, no. 1, pp. 85–113, 2009.

[8] C. Palm, "Methods of judging the annoyance caused by congestion," *Telecommunications*, vol. 4, pp. 153–163, 1953.

[9] D. Gross, J. F. Shortle, J. M. Thompson, and C. M. Harris, *Fundamentals of Queueing Theory*, John Wiley & Sons, 4th edition, 2008.

[10] N. Perel and U. Yechiali, "Queues with slow servers and impatient customers," *European Journal of Operational Research*, vol. 201, no. 1, pp. 247–258, 2010.

[11] E. Altman and U. Yechiali, "Analysis of customers' impatience in queues with server vacations," *Queueing Systems*, vol. 52, no. 4, pp. 261–279, 2006.

[12] E. Altman and U. Yechiali, "Infinite-server queues with system's additional tasks and impatient customers," *Probability in the Engineering and Informational Sciences*, vol. 22, no. 4, pp. 477–493, 2008.

[13] A. Economou and S. Kapodistria, "Synchronized abandonments in a single server unreliable queue," *European Journal of Operational Research*, vol. 203, no. 1, pp. 143–155, 2010.

[14] F. Baccelli, P. Boyer, and G. Hebuterne, "Single server queues with impatient customers," *Advances in Applied Probability*, vol. 16, no. 4, pp. 887–905, 1984.

[15] U. Yechiali, "Queues with system disasters and impatient customers when system is down," *Queueing Systems*, vol. 56, no. 3-4, pp. 195–202, 2007.

[16] Y. Sakuma and A. Inoie, "Stationary distribution of a multiserver vacation queue with constant impatient times," *Operations Research Letters*, vol. 40, no. 4, pp. 239–243, 2012.

[17] P.-S. Chen, Y.-J. Zhu, and X. Geng, "M/M/m/k queue with preemptive resume and impatience of the prioritized customers," *Systems Engineering and Electronics*, vol. 30, no. 6, pp. 1069–1073, 2008.

[18] C.-S. Ho and C. Woei, "Study of re-provisioning mechanism for dynamic traffic in WDM optical networks," in *Proceedings of the 10th International Conference on Advanced Communication Technology*, pp. 288–291, Gangwon-Do, Republic of Korea, February 2008.

[19] J. Wang, "Queueing analysis of WDM-based access networks with reconfiguration delay," in *Proceedings of the 4th International Conference on Queueing Theory and Network Applications*, Article No. 13, Singapore, July 2009.

[20] C. Goswami and N. Selvaraju, "The discrete-time MAP/PH/1 queue with multiple working vacations," *Applied Mathematical Modelling*, vol. 34, no. 4, pp. 931–946, 2010.

[21] L. D. Servi and S. G. Finn, "M/M/1 queues with working vacations (M/M/1/WV)," *Performance Evaluation*, vol. 50, no. 1, pp. 41–52, 2002.

[22] W.-Y. Liu, X.-L. Xu, and N.-S. Tian, "Stochastic decompositions in the M/M/1 queue with working vacations," *Operations Research Letters*, vol. 35, no. 5, pp. 595–600, 2007.

[23] N. Tian, X. Zhao, and K. Wang, "The M/M/1 queue with single working vacation," *International Journal of Information and Management Sciences*, vol. 19, no. 4, pp. 621–634, 2008.

[24] X. Xu, Z. Zhang, and N. Tian, "The M/M/1 queue with single working vacation and set-up times," *International Journal of Operational Research*, vol. 6, no. 3, pp. 420–434, 2009.

[25] C. Xiu, N. Tian, and Y. Liu, "The M/M/1 queue with single working vacation serving at a slower rate during the start-up period," *Journal of Mathematics Research*, vol. 2, no. 1, pp. 98–102, 2010.

[26] K.-H. Wang, W.-L. Chen, and D.-Y. Yang, "Optimal management of the machine repair problem with working vacation: Newton's method," *Journal of Computational and Applied Mathematics*, vol. 233, no. 2, pp. 449–458, 2009.

[27] D.-A. Wu and H. Takagi, "M/G/1 queue with multiple working vacations," *Performance Evaluation*, vol. 63, no. 7, pp. 654–681, 2006.

[28] Y. Baba, "Analysis of a GI/M/1 queue with multiple working vacations," *Operations Research Letters*, vol. 33, no. 2, pp. 201–209, 2005.

[29] H. Chen, F. Wang, N. Tian, and D. Lu, "Study on N-policy working vacation polling system for WDM," in *Proceedings of IEEE International Conference on Communication Technology*, pp. 394–397, Hangzhou, China, November 2008.

[30] H. Chen, F. Wang, N. Tian, and J. Qian, "Study on working vacation polling system for WDM with PH distribution service time," in *Proceedings of the International Symposium on Computer Science and Computational Technology (ISCSCT '08)*, pp. 426–429, Shanghai, China, December 2008.

[31] C.-H. Lin and J.-C. Ke, "Multi-server system with single working vacation," *Applied Mathematical Modelling*, vol. 33, no. 7, pp. 2967–2977, 2009.

[32] L. B. Aronson, B. E. Lemoff, L. A. Buckman, and D. W. Dolfi, "Low-cost multimode WDM for local area networks up to 10 Gb/s," *IEEE Photonics Technology Letters*, vol. 10, no. 10, pp. 1489–1491, 1998.

[33] N. Selvaraju and C. Goswami, "Impatient customers in an M/M/1 queue with single and multiple working vacations," *Computers & Industrial Engineering*, vol. 65, no. 2, pp. 207–215, 2013.

[34] P. V. Laxmi and K. Jyothsna, "Analysis of a working vacations queue with impatient customers operating under a triadic policy," *International Journal of Management Science and Engineering Management*, vol. 9, no. 3, pp. 191–200, 2014.

[35] N. Tian and Z. G. Zhang, "Stationary distributions of GI/M/c queue with PH type vacations," *Queueing Systems*, vol. 44, no. 2, pp. 183–202, 2003.

[36] J. R. Artalejo, A. Economou, and A. Gómez-Corral, "Algorithmic analysis of the Geo/Geo/c retrial queue," *European Journal of Operational Research*, vol. 189, no. 3, pp. 1042–1056, 2008.

[37] S. R. Chakravarthy, A. Krishnamoorthy, and V. C. Joshua, "Analysis of a multi-server retrial queue with search of customers from the orbit," *Performance Evaluation*, vol. 63, no. 8, pp. 776–798, 2006.

[38] G. I. Falin, "Calculation of probability characteristics of a multiline system with repeated calls," *Moscow University Computational Mathematics and Cybernetics*, vol. 1, pp. 43–49, 1983.

[39] J. R. Artalejo and M. Pozo, "Numerical calculation of the stationary distribution of the main multiserver retrial queue," *Annals of Operations Research*, vol. 116, pp. 41–56, 2002.

[40] L. Bright and P. G. Taylor, "Calculating the equilibrium distribution in level dependent quasi-birth-and-death processes," *Communications in Statistics. Stochastic Models*, vol. 11, no. 3, pp. 497–525, 1995.

[41] A. Krishnamoorthy, S. Babu, and V. C. Narayanan, "The MAP/(PH/PH)/1 queue with self-generation of priorities and non-preemptive service," *European Journal of Operational Research*, vol. 195, no. 1, pp. 174–185, 2009.

[42] M. F. Neuts and B. M. Rao, "Numerical investigation of a multiserver retrial model," *Queueing Systems*, vol. 7, no. 2, pp. 169–189, 1990.

[43] G. H. Golub and C. F. VanLoan, *Matrix Computations*, The John Hopkins University Press, London, UK, 1996.

[44] J. Hunter, *Mathematical Techniques of Applied Probability. Discrete Time Models: Basic Theory*, vol. 1, Academic Press, New York, NY, USA, 1983.