Mathematical and Computer Modelling 57 (2013) 1122–1130

Contents lists available at SciVerse ScienceDirect

Mathematical and Computer Modelling

journal homepage: www.elsevier.com/locate/mcm

Multi-threshold policy for a multi-server queue with synchronous single vacation

Chia-Huang Wu [a], Jau-Chuan Ke [b,*]

[a] Department of Industrial Engineering and Management, National Chiao Tung University, Taiwan, ROC

[b] Department of Applied Statistics, National Taichung University of Science and Technology, No. 129, Sec. 3, Sanmin Rd., Taichung 404, Taiwan, ROC

ARTICLE INFO

Article history:
Received 19 September 2011
Received in revised form 28 September 2012
Accepted 7 October 2012

Keywords:
$M/M/c$ queue
Multiple-threshold
Synchronous single vacation

ABSTRACT

This paper considers an infinite buffer $M/M/c$ queueing system in which servers follow a multi-threshold vacation policy. With such a policy, at a service completion instant, if the number of customers in the system is less than a prefixed threshold value, part of servers together take a single vacation (or leave for a random amount of time doing other secondary job). At the vacation completion instant, they return to the system for serving the customers. Some practical production and inventory systems or call centers could be modeled as this Markovian queue with a multi-threshold vacation policy. Using the Markovian process model, we obtain the exact closed-form expression of rate matrix and the stationary distribution of the number of customers in the system. A cost model is developed to search the joint optimal values of the thresholds of vacation policy and service rate of each server, which minimizes the long-term average cost. Some numerical results are presented to illustrate the optimization procedures.

1. Introduction

Wu and Wu [1] analyzed the reliability of a two-unit cold standby repairable system under Poisson shocks consideration. The explicit expressions of some reliability indices such as the steady-state availability, the mean up-time, and the steady-state failure frequency of a system consisting of one switch unit and one repairman were derived. GI/BMSP/1/∞ queues with and without state-dependent arrival were investigated by Banik [2]. The steady-state queue length distributions at pre-arrival and arbitrary epochs were derived by implementing matrix-geometric method, the argument of Markov renewal theory and semi-Markov process. Moreover, queueing models with server vacations are effective tools for performance analysis of manufacturing systems, local area networks, and data communication systems. For example, consider an airline company where a group of employees is trained to load/unload baggages (primary tasks) as well as the jobs like drivers (aerial ladders), machine repair and runway maintenance which are regarded as vacations. The employees would be partitioned into several groups and operation by the groups. In this case, the synchronous vacation policy could be applied.

Past works on vacation queueing models are either single server or multiple server systems. Excellent surveys on the single server vacation models have been reported by Doshi [3] and Takagi [4]. The variations and extensions of these vacation models were developed by several researchers such as Krishna Reddy et al. [5], Choudhury [6,7], Shomrony and Yechiali [8], Yechiali [9], Tadj and Choudhury [10], Ke [11,12], and Ke et al. [13] and many others. Later, Tian and Zhang [14] studied an $M^{[x]}/G/1$ queueing system with a controllable N policy, in which the server takes at most J vacations during the idle period. For the multiple server vacation models, there are only a limited number of studies due to the complexity of the

* Corresponding author.
E-mail address: jauchuan@nutc.edu.tw (J.-C. Ke).

0895-7177/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2012.10.012

systems. The $M/M/c$ queue with exponential vacations was first studied by Levy and Yechiali [15]. Chao and Zhao [16] investigated a GI/M/c vacation system and provided an algorithm to compute the performance measures. Tian et al. [17] and Tian and Xu [18] gave a detailed study of the $M/M/c$ vacation systems in which all servers take multiple vacation policy when the system is empty. Zhang and Tian [19,20] and Xu and Zhang [21] analyzed the $M/M/c$ vacation systems that partial servers with a multiple vacation policy in which some servers (only the idle ones) take single or multiple vacations, which is called the synchronous vacation policy. Multi-server vacation models are more flexible and applicable in practice than single server models. Later, a discrete-time multi-server queueing system with infinite buffer size, Bernoulli arrival process, and multi-threshold service policy was discussed by Peng [22]. An algorithm for computing the stationary waiting time distribution was presented, and the stationary waiting time distribution was also derived. More recent, Ke et al. [23] used quasi-birth–death and matrix-analytical method to investigate an infinite capacity $M/M/c$ queueing system with modified Bernoulli vacation under a single vacation policy.

Existing research works mentioned earlier, have not addressed the optimization issue of vacation criteria in the infinite buffer multiple-server systems, in which some servers take vacations together. Since the arrival is random, the system may blow up or overload sometimes. Nevertheless, it may be unmanned (even empty) once in some time interval. The manager could employ more employees to deal with this variation. However, it need more labor cost and more time spending. In order to reduce the cost of companies and increase the working efficiency of employees, adopting an appropriate allocation strategy is essential and necessary. For example, the operator in facility would leave the service station (product line) to take some secondary job when the system is not hard-pressed. Perhaps, they may be assigned to execute other second tasks as they are idly in the system. This situation could be regarded as the vacation of servers. Therefore, we investigate a multi-server queueing system with synchronous vacation policy which is described formally later. In this study, we wish to develop a computational model that helps managers to answer the following important questions: (1) Under a certain cost structure, what is the optimal multi-threshold vacation policy that minimize the expected cost function; That is how many servers are needed and how to utilize their idle time when the customers in the queue are served till some predetermined thresholds. (2) When the optimal multi-threshold vacation policy is determined, how to adjust the service rate on the minimum cost reduced.

The paper is organized as follows; In Section 2, the mathematical model and the quasi-birth–death (QBD) model of an infinite capacity $M/M/c$ queue with a synchronous single vacation are set up. The closed-form of the rate matrix for the QBD model is derived explicitly in Section 3. By using the matrix-analytical method, the stationary probabilities are obtained in Section 4. In Section 5, some system performance measures are derived. A cost model is further developed to determine the optimal values of number of servers and service rate, simultaneously which minimize the total expected cost per unit time. We use direct search method and Quasi-Newton method to deal with the optimization tasks. Some numerical examples are provided to illustrate the two optimization methods. Section 6 concludes.

2. Model description and QBD process

We consider a multi-server $M/M/c$ queueing system with mean arrival rate λ, mean service rate μ, and a multiple-threshold synchronous vacation policy. The system allows at most s orders (groups) of synchronous vacation. When the number of customers in the system is less than a pre-determined threshold value h_i ($h_i < c$), k_i ($k_i < c$ and $\sum_{i=1}^{s} k_i < c$) of servers (partial servers) would take a synchronous vacation with mean vacation time $1/\theta_i$, $i = 1, 2, \ldots, s$. At a vacation completion instant, the returned servers start servicing the customers in the system or wait idly for serving new arrivals if system is empty, that is, the single vacation policy. In addition, the inter-arrival time, the service time and the vacation time are all exponentially distributed and mutually independent. The service discipline is assumed to be First Come First Served (FCFS).

Let $S(t)$ and $N(t)$ denote, respectively, the number of normal (not on vacation) servers and the number of customers in the system at time t. The state $S(t) = i$ indicates that there are $(c - i)$ servers on vacation. The process $\{S(t), N(t) : t \geq 0\}$ is a continuous-time Markov chain with state space given by $\Omega = \bigcup\{(i, n) : i = c, c - k_1, \ldots, c - \sum_{j=1}^{s} k_j, n \geq 0\}$. For analysis convenience, it is assumed that $\theta_1 > \theta_2 > \cdots > \theta_s$. Following the concepts by Neuts [24], using the lexicographical sequence for the states (i, n), the infinitesimal generator for the Markov process could be written as a block-partitioned form:

$$\mathbf{Q} = \begin{pmatrix} \mathbf{A}_0 & \mathbf{B} & & & & & & & \\ \mathbf{C}_1 & \mathbf{A}_1 & \mathbf{B} & & & & & & \\ & \mathbf{C}_2 & \mathbf{A}_2 & \mathbf{B} & & & & & \\ & & \ddots & \ddots & \ddots & & & & \\ & & & \ddots & \ddots & \ddots & & & \\ & & & & \mathbf{C}_{c-1} & \mathbf{A}_{c-1} & \mathbf{B} & & \\ & & & & & \mathbf{C}_c & \mathbf{A}_c & \mathbf{B} & \\ & & & & & & \mathbf{C}_c & \mathbf{A}_c & \mathbf{B} \\ & & & & & & & \ddots & \ddots & \ddots \end{pmatrix}, \quad (1)$$

where each entry of $\mathbf{Q}$ is a square matrix of size $(s+1)$. $\mathbf{B} = \lambda \mathbf{I}_{s+1}$ with $\mathbf{I}_{s+1}$ denotes the identity matrix with size $(s+1)$. For $1 \le i \le c$, $i \notin \{h_1, h_2, \ldots, h_s\}$ (*i.e.*, i does not belong to the threshold values), $\mathbf{C}_i$ is a diagonal matrix with diagonal element $\mathbf{C}_i[j,j] = \min(i, c - \sum_{\ell=1}^{j-1} k_\ell)\mu$, $1 \le j \le s+1$. The remainder matrix $\mathbf{C}_{h_r}$ $(r = 1, 2, \ldots, s)$ is the same as $\mathbf{C}_i$ but the element of $\mathbf{C}_{h_r}[r,r]$ is shifted to the position of $\mathbf{C}_{h_r}[r, r+1]$ (*i.e.*, $\mathbf{C}_{h_r}[r, r+1] = \min(h_r, c - \sum_{\ell=1}^{r-1} k_\ell)\mu$ and $\mathbf{C}_{h_r}[r,r] = 0$). Consequently, matrix $\mathbf{C}_i$ could be obtained by a general formula as follows:

$$\mathbf{C}_i = \begin{pmatrix} (1-I_i^1)\mu_i^c & I_i^1\mu_i^c & & & & \\ & (1-I_i^2)\mu_i^{c-k_1} & I_i^2\mu_i^{c-k_1} & & & \\ & \ddots & \ddots & & & \\ & & \ddots & \ddots & & \\ & & & (1-I_i^{s-1})\mu_i^{c-\sum_{\ell=1}^{s-1}k_\ell} & I_i^{s-1}\mu_i^{c-\sum_{\ell=1}^{s-1}k_\ell} & \\ & & & & \mu_i^{c-\sum_{\ell=1}^{s}k_\ell} & \end{pmatrix}_{(s+1)\times(s+1)}, \quad i \ge 0 \tag{2}$$

with $I_a^b = 1$ if $a = b$, *i.e.*, the indicator function. The symbol $\mu_a^b = \min(a, b)\mu$ denotes the mean service rate corresponding to various statuses. In addition, the matrix $\mathbf{A}_i$ is given by

$$\mathbf{A}_i = \begin{pmatrix} * & & & & & \\ \theta_1 & * & & & & \\ & \theta_2 & * & & & \\ & & \ddots & \ddots & & \\ & & & \theta_{s-1} & * & \\ & & & & \theta_s & * \end{pmatrix}_{(s+1)\times(s+1)}, \quad i \ge 0. \tag{3}$$

The diagonal elements of matrix $\mathbf{A}_i$ (or $\mathbf{Q}$), indicated by "*", are magnitudes satisfy that the sum of each row of $\mathbf{Q}$ is zero. From the matrix structure of $\mathbf{Q}$, we find that the Markov process $\{S(t), N(t)\}$ is a QBD process (see Neuts [24] and Latouche and Ramaswami [25]).

Based on the Theorem 3.1.1 in Neuts [24], the queueing system would be stable and the steady state probability exists if and only if

$$\mathbf{xBe} < \mathbf{xC_c e} \tag{4}$$

where $\mathbf{e}$ is a column vector of dimension $s+1$ with all elements equal to one. $\mathbf{x} = [x_c, x_{c-k_1}, \ldots, x_{c-k_1\cdots-k_s}]$ is the invariant probability of the matrix $\mathbf{F} = \mathbf{C}_c + \mathbf{A}_c + \mathbf{B}$ which satisfies $\mathbf{xF} = \mathbf{0}$ and $\mathbf{xe} = 1$. Solving $\mathbf{xF} = \mathbf{0}$ and $\mathbf{xe} = 1$ implies $x_c = 1$ and other $x_i = 0$, for $i \ne c$. Substituting $\mathbf{B}$, $\mathbf{C}_c$ and $\mathbf{x}$ into Eq. (4) and doing some routine manipulations, then we have

$$c\mu > \lambda \quad \text{or} \quad 1 > \frac{\lambda}{c\mu} = \rho, \tag{5}$$

which is a common and reasonable conclusion.

3. Steady state results

As $\rho = \lambda(c\mu)^{-1} < 1$, this QBD process could be investigated in steady-state. Let $\{S, N\}$ be the stationary random variables for the status of the vacation servers and the number of customers in the system. Denote the stationary probability by

$$p_{i,n} \equiv P\{S = i, N = n\} = \lim_{t\to\infty} P\{S(t) = i, N(t) = n\}, \quad (i,n) \in \Omega. \tag{6}$$

Let $\boldsymbol{\Pi}$ denotes the corresponding steady-state probability vector of $\mathbf{Q}$. By partitioning the vector $\boldsymbol{\Pi}$ as $\boldsymbol{\Pi} = [\boldsymbol{\Pi}_0, \boldsymbol{\Pi}_1, \ldots, \boldsymbol{\Pi}_{c-1}, \boldsymbol{\Pi}_c, \boldsymbol{\Pi}_{c+1}, \ldots]$, where each sub-vector $\boldsymbol{\Pi}_i = [p_{c,i}, p_{c-k_1,i}, p_{c-k_1-k_2,i}, \ldots, p_{c-k_1\cdots-k_s,i}]$ is a row vector with dimension $(s+1)$. Then, the steady state probability vector $\boldsymbol{\Pi}$ is the unique solution that satisfies $\boldsymbol{\Pi}\mathbf{Q} = \mathbf{0}$ and the normalization condition $\sum_{n=0}^{\infty} \boldsymbol{\Pi}_n \mathbf{e} = 1$ (see Neuts [24] and Latouche and Ramaswami [25]). It is noted that the vector $\boldsymbol{\Pi} = [\boldsymbol{\Pi}_0, \boldsymbol{\Pi}_1, \ldots, \boldsymbol{\Pi}_{c-1}, \boldsymbol{\Pi}_c, \boldsymbol{\Pi}_{c+1}, \ldots]$ has the following properties

$$\boldsymbol{\Pi}_{c+\ell} = \boldsymbol{\Pi}_c \mathbf{R}^\ell, \quad \text{for } \ell \ge 1 \tag{7}$$

where the matrix $\mathbf{R}$, called "rate matrix", is the minimal non-negative solution of the matrix quadratic equation

$$\mathbf{R}^2\mathbf{C}_c + \mathbf{R}\mathbf{A}_c + \mathbf{B} = \mathbf{0}. \tag{8}$$

The rate matrix **R** could be explicitly determined using an efficient Maple computer program. Because the coefficient matrices of Eq. (8) are all lower-triangular, **R** is also lower-triangular. We develop the explicit formula for rate matrix **R** as following:

$$\mathbf{R} = \begin{pmatrix} r_{1,1} & & & & & \\ r_{2,1} & r_{2,2} & & & & \\ r_{3,1} & r_{3,2} & r_{3,3} & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ r_{s,1} & r_{s,2} & r_{s,3} & \cdots & r_{s,s} & \\ r_{s+1,1} & r_{s+1,2} & r_{s+1,3} & \cdots & r_{s+1,s} & r_{s+1,s+1} \end{pmatrix} \tag{9}$$

where $r_{i,i} = \frac{\lambda+\theta_{i-1}+\mu_c^{i-1}-\sqrt{[\lambda+\theta_{i-1}+\mu_c^{i-1}]^2-4\lambda\,\mu_c^{i-1}}}{2\mu_c^{i-1}}$, for $1 \le i \le s+1$,

$$r_{i,j} = \frac{r_{i,j+1}\theta_j + \mu_c^{j-1} \sum_{\ell=j+1}^{i-1} r_{i,\ell}\, r_{\ell,j}}{\lambda + \theta_{j-1} + \mu_c^{j-1}(1 - r_{i,i} - r_{j,j})}, \quad \text{for } 1 \le j \le i \le s+1,$$
$$r_{i,j} = 0, \quad \text{for } j > i,$$

$\theta_0 = 0$, $\mu_c^i = (c - \sum_{\ell=1}^{i} k_\ell)\mu$ and empty summation $(\sum_i^j, j < i)$ is defined to be zero.

Theorem 1. *If* $\rho = \lambda(c\mu)^{-1} < 1$, *the spectral radius of rate matrix* **R**, sp(**R**), *is less than* 1.

Proof. It is noted that the diagonal elements are the corresponding eigen-values of the matrix **R**. Firstly, because of the assumption of $\theta_0 = 0$

$$\mathbf{R}[1,1] = \frac{\lambda + c\mu - |\lambda - c\mu|}{2c\mu} = \frac{\lambda + c\mu - (c\mu - \lambda)}{2c\mu} = \lambda(c\mu)^{-1} < 1. \tag{10}$$

For $2 \le i \le s+1$, the diagonal element $\mathbf{R}[i,i] = r_{i,i}$ is obtained from the quadratic equation

$$f(x) = \mu_c^{i-1}x^2 - (\lambda + \theta_{i-1} + \mu_c^{i-1})x + \lambda = 0. \tag{11}$$

By the intermediate value theorem, there exists exact one real root in (0, 1) because

$$\begin{aligned} &f(0) = \lambda > 0, \\ &f(1) = \mu_c^{i-1} - (\lambda + \theta_{i-1} + \mu_c^{i-1}) + \lambda = -\theta_{i-1} < 0. \end{aligned} \tag{12}$$

From (10) and (12), all diagonal elements (eigen-values) of rate matrix **R** are less than 1. Therefore, the spectral radius of rate matrix **R**, $\mathrm{sp}(\mathbf{R}) = \max_{1\le i\le s+1}\{r_{i,i}\}$ is less than 1. □

Theorem 2. *The rate matrix* **R** *satisfies* $\mathbf{RT} = \lambda\mathbf{e}$, *where* $\mathbf{T} = [\mu_c^c, \mu_c^{c-k_1}, \ldots, \mu_c^{c-\sum_{i=1}^{s} k_i}]$.

Proof. Multiplying both sides of Eq. (8) by **e** gives

$$\begin{aligned} (\mathbf{R}^2\mathbf{C}_c + \mathbf{R}\mathbf{A}_c + \mathbf{B})\mathbf{e} &= \mathbf{R}^2\mathbf{C}_c\mathbf{e} - \mathbf{R}(\mathbf{B} + \mathbf{C}_c)\mathbf{e} + \lambda\mathbf{e} \\ &= \mathbf{R}^2\mathbf{T} - \mathbf{R}(\lambda\mathbf{e} + \mathbf{T}) + \lambda\mathbf{e} = (\mathbf{I} - \mathbf{R})(\lambda\mathbf{e} - \mathbf{RT}) = \mathbf{0}, \end{aligned} \tag{13}$$

since $(\mathbf{B} + \mathbf{A}_c + \mathbf{C}_c)\mathbf{e} = \mathbf{0}$ and $\mathbf{C}_c\mathbf{e} = \mathbf{T}$. However, $\mathbf{I} - \mathbf{R}$ is invertible, hence $\lambda\mathbf{e} = \mathbf{RT}$.

Once the rate matrix **R** is determined, the steady state probability $\boldsymbol{\Pi}_i$, $i > c$ could be evaluated recursively. Furthermore, the steady-state equations $\boldsymbol{\Pi}\mathbf{Q} = \mathbf{0}$ are given by

$$\boldsymbol{\Pi}_0\mathbf{A}_0 + \boldsymbol{\Pi}_1\mathbf{C}_1 = \mathbf{0}, \tag{14}$$
$$\boldsymbol{\Pi}_{i-1}\mathbf{B} + \boldsymbol{\Pi}_i\mathbf{A}_i + \boldsymbol{\Pi}_{i+1}\mathbf{C}_{i+1} = \mathbf{0}, \quad 1 \le i \le c, \tag{15}$$
$$\boldsymbol{\Pi}_c\mathbf{R}^{i-1-c}(\mathbf{B} + \mathbf{R}\mathbf{A}_c + \mathbf{R}^2\mathbf{C}_c) = \mathbf{0}, \quad c+1 \le i, \tag{16}$$

and the following normalization condition

$$\sum_i \sum_n p_{i,n} = \sum_i \boldsymbol{\Pi}_i\mathbf{e} = 1. \tag{17}$$

Eqs. (14)–(15) could be manipulated routinely, we have

$$\begin{aligned} &\boldsymbol{\Pi}_0 = \boldsymbol{\Pi}_1\mathbf{C}_1(-\mathbf{A}_0)^{-1} = \boldsymbol{\Pi}_1\phi_0, \\ &\boldsymbol{\Pi}_i = \boldsymbol{\Pi}_{i+1}\mathbf{C}_{i+1}[-(\phi_{i-1}\mathbf{B} + \mathbf{A}_i)]^{-1} = \boldsymbol{\Pi}_{i+1}\phi_i, \quad 1 \le i \le c-1, \end{aligned} \tag{18}$$

and

$$\boldsymbol{\Pi}_c\phi_{c-1}\mathbf{B} + \boldsymbol{\Pi}_c\mathbf{A}_c + \boldsymbol{\Pi}_c\mathbf{R}\mathbf{C}_c = \mathbf{0}. \tag{19}$$

Consequently, $\boldsymbol{\Pi}_i$ $(1 \le i \le c-1)$ in Eqs. (14)–(15) could be written in terms of $\boldsymbol{\Pi}_c$ as $\boldsymbol{\Pi}_i = \boldsymbol{\Pi}_c\boldsymbol{\Phi}_i$ where $\boldsymbol{\Phi}_i = \phi_{c-1}\phi_{c-2}\cdots\phi_{i+1}\phi_i$, $0 \le i \le c-1$ and $\phi_0 = \mathbf{C}_1(-\mathbf{A}_0)^{-1}$, $\phi_i = \mathbf{C}_{i+1}[-(\phi_{i-1}\mathbf{B}+\mathbf{A}_i)]^{-1}$, $1 \le i \le c-1$. Once the steady-state probability $\boldsymbol{\Pi}_c$ is obtained, $\boldsymbol{\Pi} = [\boldsymbol{\Pi}_0, \boldsymbol{\Pi}_1, \ldots, \boldsymbol{\Pi}_{c-1}, \boldsymbol{\Pi}_c, \boldsymbol{\Pi}_{c+1}, \ldots]$ is determined. $\boldsymbol{\Pi}_c$ could be derived by simultaneously solving Eq. (19) and the following normalization condition

$$\begin{aligned}\sum_i\sum_n p_{i,n} &= \sum_i \boldsymbol{\Pi}_i\mathbf{e} = (\boldsymbol{\Pi}_0 + \boldsymbol{\Pi}_1 + \cdots + \boldsymbol{\Pi}_{c-1} + \boldsymbol{\Pi}_c + \boldsymbol{\Pi}_{c+1} + \boldsymbol{\Pi}_{c+2}\cdots)\mathbf{e} \\ &= (\boldsymbol{\Pi}_c\boldsymbol{\Phi}_0 + \boldsymbol{\Pi}_c\boldsymbol{\Phi}_1 + \cdots + \boldsymbol{\Pi}_c\boldsymbol{\Phi}_{c-1} + \boldsymbol{\Pi}_c + \boldsymbol{\Pi}_c\mathbf{R} + \boldsymbol{\Pi}_c\mathbf{R}^2\cdots)\mathbf{e} \\ &= \boldsymbol{\Pi}_c\left[\sum_{i=0}^{c-1}\boldsymbol{\Phi}_i + (\mathbf{I}-\mathbf{R})^{-1}\right]\mathbf{e} = 1. \quad \square\end{aligned} \tag{20}$$

4. Special case of $s = 1$

Our model could be reduced to the queue system with synchronous single vacation for some servers, which was one vacation threshold investigated by Zhang and Tian [19]. As $s = 1$, set $k_1 = d$, $h_1 = c - d$ and $\theta_1 = \theta$, we have

$$\mathbf{C}_c = \begin{pmatrix} c\mu & 0 \\ 0 & (c-d)\mu \end{pmatrix}, \qquad \mathbf{A}_c = \begin{pmatrix} -(\lambda + c\mu) & 0 \\ \theta & -[\lambda + (c-d)\mu] \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}.$$

Substituting these three matrices into Eq. (8) gives the system of equations

$$c\mu r_{1,1}^2 - (\lambda + c\mu)r_{1,1} + \lambda = 0, \tag{21}$$

$$c\mu r_{2,1}(r_{1,1} + r_{2,2}) - (\lambda + c\mu)r_{2,1} + \lambda = 0, \tag{22}$$

$$(c-d)\mu r_{2,2}^2 - [\lambda + (c-d)\mu]r_{2,2} + \lambda = 0. \tag{23}$$

Then $\mathbf{R} = \begin{pmatrix} r_{11} & 0 \\ r_{21} & r_{22} \end{pmatrix}$ is the minimal nonnegative solution of the system of Eqs. (21)–(23). Also, the rate matrix $\mathbf{R}$ could be found from Eq. (9),

$$\begin{aligned} r_{1,1} &= \frac{\lambda + c\mu - \sqrt{[\lambda + c\mu]^2 - 4\lambda c\,\mu}}{2c\mu} = \frac{\lambda}{c\mu} = \rho, \\ r_{2,2} &= \frac{\lambda + \theta + (c-d)\mu - \sqrt{[\lambda + \theta + (c-d)\mu]^2 - 4\lambda\,(c-d)\mu}}{2(c-d)\mu} \quad (\text{set} = r), \\ r_{2,1} &= \frac{r_{2,2}\theta}{\lambda + c\mu(1 - r_{1,1} - r_{2,2})} = \frac{\theta r}{\lambda + c\mu(1-\rho-r)} = \frac{\theta r}{c\mu(1-r)}. \end{aligned}$$

Consequently, the rate matrix is

$$\mathbf{R} = \begin{pmatrix} r_{11} & 0 \\ r_{21} & r_{22} \end{pmatrix} = \begin{pmatrix} \rho & 0 \\ \dfrac{\theta r}{c\mu(1-r)} & r \end{pmatrix}. \tag{24}$$

By resorting the state sequence, we get the $\mathbf{R}$ that is consistent with Eq. (4) in Zhang and Tian [19].

5. Performance measures and cost model

In this section, some performance measures of the system are given. Based on these measures, we develop a cost model to determine the optimal vacation policy $\mathbf{K} = [k_1, k_2, \ldots, k_s]$ when the threshold values $\mathbf{H} = [h_1, h_2, \ldots, h_s]$ and the vacation rate $\boldsymbol{\Theta} = [\theta_1, \theta_2, \ldots, \theta_s]$ are given. Various system measures of our model are defined as follows. Let

$L_s \equiv$ the average number of customers in the system
$L_q \equiv$ the average number of customers in the queue
$E[V] \equiv$ the average number of servers on vacation
$E[I] \equiv$ the average number of idle servers
$E[B] \equiv$ the average number of busy servers
$O.U. \equiv$ the operative utilisation.

The expressions for L_s, L_q, $E[V]$, $E[I]$, and $E[B]$ are given as follows:

$$\begin{aligned} L_s &= \boldsymbol{\Pi}_1\mathbf{e} + 2\boldsymbol{\Pi}_2\mathbf{e} + \cdots + (c-1)\boldsymbol{\Pi}_{c-1}\mathbf{e} + c\boldsymbol{\Pi}_c\mathbf{e} + (c+1)\boldsymbol{\Pi}_{c+1}\mathbf{e} + \cdots \\ &= \boldsymbol{\Pi}_c\boldsymbol{\Phi}_1\mathbf{e} + 2\boldsymbol{\Pi}_2\boldsymbol{\Phi}_2\mathbf{e} + \cdots + (c-1)\boldsymbol{\Pi}_c\boldsymbol{\Phi}_{c-1}\mathbf{e} + c\boldsymbol{\Pi}_c\mathbf{e} + (c+1)\boldsymbol{\Pi}_c\mathbf{R}\mathbf{e} + \cdots \\ &= \boldsymbol{\Pi}_c\left(\sum_{i=1}^{c-1} i\boldsymbol{\Phi}_i + c(\mathbf{I}-\mathbf{R})^{-1} + \mathbf{R}(\mathbf{I}-\mathbf{R})^{-2}\right)\mathbf{e} \end{aligned} \tag{25}$$

$$\begin{aligned} L_q &= \sum_{i=1}^{c-1}\boldsymbol{\Pi}_i\begin{bmatrix} \max\{i-c,0\} \\ \max\{i-(c-k_1),0\} \\ \vdots \\ \max\left\{i-\left(c-\sum_{j=1}^{s}k_j\right),0\right\}\end{bmatrix} + \boldsymbol{\Pi}_c\begin{bmatrix}0\\k_1\\\vdots\\\sum_{j=1}^{s}k_j\end{bmatrix} + \boldsymbol{\Pi}_c\mathbf{R}\left(\begin{bmatrix}0\\k_1\\\vdots\\\sum_{j=1}^{s}k_j\end{bmatrix} + \mathbf{e}\right) + \cdots \\ &= \sum_{i=1}^{c-1}\boldsymbol{\Pi}_i\begin{bmatrix} \max\{i-c,0\} \\ \max\{i-(c-k_1),0\} \\ \vdots \\ \max\left\{i-\left(c-\sum_{j=1}^{s}k_j\right),0\right\}\end{bmatrix} + \boldsymbol{\Pi}_c(\mathbf{I}-\mathbf{R})^{-1}\begin{bmatrix}0\\k_1\\\vdots\\\sum_{j=1}^{s}k_j\end{bmatrix} + \boldsymbol{\Pi}_c\mathbf{R}(\mathbf{I}-\mathbf{R})^{-2}\mathbf{e} \end{aligned} \tag{26}$$

$$E[V] = \sum_{i=0}^{\infty}\boldsymbol{\Pi}_i\begin{bmatrix}0\\k_1\\\vdots\\\sum_{j=1}^{s}k_j\end{bmatrix} = \boldsymbol{\Pi}_c\left(\sum_{i=1}^{c-1}\boldsymbol{\Phi}_i + (\mathbf{I}-\mathbf{R})^{-1}\right)\begin{bmatrix}0\\k_1\\\vdots\\\sum_{j=1}^{s}k_j\end{bmatrix} \tag{27}$$

$$E[I] = \sum_{i=0}^{\infty}\boldsymbol{\Pi}_i\begin{bmatrix}\max\{c-i,0\}\\\max\{c-k_1-i,0\}\\\vdots\\\max\left\{c-\sum_{j=1}^{s}k_j-i,0\right\}\end{bmatrix} = \sum_{i=0}^{c-1}\boldsymbol{\Pi}_c\boldsymbol{\Phi}_i\begin{bmatrix}\max\{c-i,0\}\\\max\{c-k_1-i,0\}\\\vdots\\\max\left\{c-\sum_{j=1}^{s}k_j-i,0\right\}\end{bmatrix} \tag{28}$$

$$E[B] = c - E[V] - E[I], \qquad O.U. = E[B]/c. \tag{29}$$

Next, we construct a total expected cost function per unit time based on these system performance measures. Let

$C_h \equiv$ cost per unit time when one customer in the system,
$C_b \equiv$ cost per unit time when one server is busy,
$C_i \equiv$ cost per unit time when one server is idle,
$C_o \equiv$ loss cost of operative untilisation.

Using the definitions of these cost elements listed above, the total expected cost function per unit time is given by

$$F(\mathbf{K}) = C_hL_s + C_bE[B] + C_iE[I] + C_o(1-O.U.). \tag{30}$$

Our objective is to determine the optimum vacation policy, say $\mathbf{K}^*$, so as to minimize this function. Due to the discrete property of $\mathbf{K}$, a direct search method may be adopted. We use direct substitution of successive values of $k_1, k_2, \ldots, k_s$ ($\sum_{i=1}^{s} k_i < c$) into the cost function until all feasible combinations (solutions) are computed. The specific steps in the direct search algorithm to establish the optimal value $\mathbf{K}^*$ are as follows:

Step 1. Let M be a sufficient large number and set candidate $= M$ (Initialize).
Step 2. For each variable k_i, $1 \le i \le s$, make a do loop with lower bound 1 and upper bound $c - s - \sum_{\ell=1}^{i-1} k_\ell$.
Step 3. Calculate the cost function $F(\mathbf{K})$ along this do loop and replace candidate by $F(\mathbf{K})$ if $F(\mathbf{K}) <$ candidate.
Step 4. Candidate is the optimal solution, output.

An example is provided to illustrate the direct search algorithm described above.

Example. An example (such as the airline company mentioned in Section 1) is provided to illustrate the direct search procedure. For example:

- There are $c = 6$ employees who are responsible for baggage loading/unloading.
- The baggages arrive follows a Poisson process with rate $\lambda = 1.5$/min.

Table 1
System performance measures of the multi-server queueing system under a multiple-threshold vacation policy ($c = 10, s = 3, \lambda = 2.5, \mu = 0.3$ and $\boldsymbol{\Theta} = [0.05, 0.2, 0.5]$).

H	[1, 2, 3]	[2, 4, 6]	[3, 6, 9]	[1, 3, 5]	[2, 5, 8]	[1, 5, 9]
K	[2, 3, 2]	[2, 2, 1]	[1, 3, 1]	[2, 2, 2]	[2, 2, 1]	[2, 2, 1]
F	776.665	769.793	753.675	776.574	769.564	776.450
L_s	10.8406	11.3788	12.1737	10.8496	11.4833	10.8798
L_q	2.50731	3.04551	3.84032	2.51631	3.14997	2.54650
$E[V]$	0.01935	0.15551	0.42291	0.02135	0.16966	0.02609
$E[I]$	1.64732	1.51116	1.24376	1.64532	1.49701	1.64058
$E[B]$	8.33333	8.33333	8.33333	8.33333	8.33333	8.33333
$O.U$	0.83333	0.83333	0.83333	0.83333	0.83333	0.83333
K	[5, 3, 1]	[3, 3, 3]	[1, 3, 5]	[4, 3, 2]	[2, 2, 2]	[2, 3, 4]
H	[1, 2, 9]	[4, 5, 6]	[6, 7, 8]	[2, 3, 9]	[6, 7, 8]	[5, 6, 7]
F	778.588	759.305	720.953	776.954	720.166	736.143
L_s	11.2481	17.8398	16.2541	12.8384	17.5490	17.2506
L_q	2.91480	9.50645	7.92072	4.50505	9.21566	8.91728
$E[V]$	0.04326	0.98991	1.23986	0.23811	1.32582	1.18181
$E[I]$	1.62341	0.67676	0.42681	1.42856	0.34085	0.48486
$E[B]$	8.33333	8.33333	8.33333	8.33333	8.33333	8.33333
$O.U$	0.83333	0.83333	0.83333	0.83333	0.83333	0.83333

- The baggage loading/unloading time are according to an exponential distribution with rate $\mu = 0.3$/min.
- When the number of baggage is less than fixed threshold values $\mathbf{H} = [1, 2, 3]$, $\mathbf{K} = [k_1, k_2, k_3]$ employees would go away from the primary task to deal with other secondary job (taking vacation).
- The vacation rates are $\boldsymbol{\Theta} = [0.05, 0.2, 0.5]$ (1/min).
- Holding cost $C_h = \$10$/unit, $C_b = \$60$/person, $C_i = \$90$/person, $C_o = \$120$.

Step 1. Let $M = 5000$ and set candidate $= M$.
Step 2. Make three do loops: k_1 from 1 to 3, k_2 from 1 to 3-k_1, and k_3 from 1 to 3-k_1-k_2.
Step 3. Calculate $F(1, 1, 1) = 482.592 < 5000$, set candidate $= 482.592$.
Calculate $F(1, 1, 2) = 482.506 < 482.592$, set candidate $= 482.506$.
...
Calculate $F(3, 1, 1) = 497.537 > 482.375$.
Step 4. $F(1, 2, 1) = 482.375$ is the optimal solution.

Analogously, the optimal value of **H** could be solved by adopting a similar optimization procedure. More system measures and optimal values of **K** (**H**) given **H** (**K**) are shown in Table 1. The manager could use the information provided in Table 1 to decide the discipline and the staff mobility in order to minimize the total cost. In practice, the service rate may be adjusted to reduce the total cost as other system parameters are determined. After the determination of the discrete system parameters, the Quasi-Newton method is employed to search the optimal service rate μ^* until the minimum cost is achieved. To find the optimal value μ^*, we should show the convexity of F. However, it is very difficult to implement. Note that the derivative of the cost function F with respect to μ indicates the direction which cost function increases. Therefore, the (local) minimum of F could be found along this opposite direction of the gradient (see Chong and Zak [26]). An effective procedure to calculate μ^* is presented as follows:

Step 1. Set an initial trial solution $\mu^{(0)}$ and give a tolerance $\varepsilon > 0$.
Step 2. Compute $F(\mu^{(i)})$, $F'(\mu^{(i)}) = \partial F/\partial\mu|_{\mu^{(i)}}$, and $F''(\mu^{(i)}) = \partial^2 F/\partial\mu^2|_{\mu^{(i)}}$.
Step 3. If $\left|F'(\mu^{(i)})\right| > \varepsilon$, find the new trial solution $\mu^{(i+1)} = \mu^{(i)} - F'(\mu^{(i)})/F''(\mu^{(i)})$ and back to step 2. Otherwise, the approximate optimal solution is found.

In the following, two examples are provided to illustrate this optimization procedure are presented in Table 2. From Table 1 and Table 2, (i) the average number of busy servers is equal to the traffic intensity λ/μ, which is a reasonable result; (ii) minimum cost may be achieved by a combination of various values of **K** and **H**, that is, the optimal decision may be very sensitive to the original system setting and parameters; and (iii) the Quasi-Newton method is effective to deal with the continuous variable optimization problem (the approximation solution is found by repeating five times of the procedure listed above), *i.e.*, it is easily converged to the optimum values. It is very useful and helpful in deal with cost reduction problem. Finally, a sensitivity investigation to the optimal values of **K** and μ for various values of λ and **H** is performed. The corresponding system performance measures and cost are shown in Table 3. From Table 3, we observe that (i) as λ becomes larger, the system loading becomes heavy, the optimal mean service rate μ^* also increases in order to keep the service quality and the total cost acceptable; (ii) as expected, the expected number of customer in the system L_s and the cost function F also increase if λ increases; and (iii) the optimal vacation policy $\mathbf{K}^*$ seems insensitive to the change of **H**. It means that $\mathbf{K}^*$ could be adopted to be a (near) optimal solution in various values of **H** (i.e., it is robust for the optimal values of number of vacation servers in various thresholds).

Table 2
The illustration of the optimization procedure of the Newton-Quasi method when $c = 10$, $s = 3$, $\lambda = 2.5$, $\varepsilon = 10^{-6}$ and $\boldsymbol{\Theta} = [0.05, 0.2, 0.5]$.

Iterations	0	1	2	3	4	5
Case (i): $\mathbf{K} = [2, 2, 1]$, $\mathbf{H} = [2, 4, 6]$, $\mu^{(0)} = 0.5$						
F	635.047	590.297	587.832	587.737	587.736	**587.736**
$\mu^{(i)}$	0.5	0.61844	0.65604	0.66536	0.66586	**0.66586**
$\partial F/\partial\mu$	−710.452	−117.628	−19.7837	−0.96597	−0.00270	3×10^{-7}
$\partial^2 F/\partial\mu^2$	5922.60	3128.30	2124.00	1918.40	1907.40	1908.30
L_s	5.80376	4.47601	4.15654	4.08406	4.08022	4.08021
$E[V]$	2.58879	3.38533	3.48536	3.50326	3.50417	3.50417
$E[I]$	2.41121	2.57221	2.70389	2.73935	2.74129	2.74129
$E[B]$	5.00000	4.04245	3.81076	3.75739	3.75455	3.75454
$O.U.$	0.50000	0.40425	0.38108	0.37574	0.37545	0.37545
Case (ii): $\mathbf{K} = [3, 3, 3]$, $\mathbf{H} = [4, 5, 6]$, $\mu^{(0)} = 1.0$						
F	474.127	459.192	451.623	451.462	451.462	**451.462**
$\mu^{(i)}$	1.0	0.68881	0.77674	0.79205	0.79241	**0.79241**
$\partial F/\partial\mu$	195.707	−157.878	−20.7297	−0.47171	−0.00034	8.8×10^{-8}
$\partial^2 F/\partial\mu^2$	628.900	1795.60	1353.80	1292.70	1290.70	1291.70
L_s	4.93867	7.64658	6.78833	6.64795	6.64464	6.64463
$E[V]$	5.44733	5.38707	5.56754	5.58278	5.58309	5.58309
$E[I]$	2.05267	0.98348	1.21386	1.26085	1.26199	1.26199
$E[B]$	2.50000	3.62944	3.21860	3.15638	3.15492	3.15492
$O.U.$	0.25000	0.36294	0.32186	0.31564	0.31549	0.31549

Table 3
The optimal service rate and the system performance measures for various value of λ and $\mathbf{H}$ when $c = 10$, $s = 3$, $\mu = 0.5$, $\boldsymbol{\Theta} = [0.05, 0.2, 0.5]$.

λ	1.0	2.0	3.0	1.0	2.0	3.0
$\mathbf{H}$	[1, 4, 7]	[1, 4, 7]	[1, 4, 7]	[2, 5, 8]	[2, 5, 8]	[2, 5, 8]
$\mathbf{K}^*$	[7, 1, 1]	[4, 2, 1]	[3, 1, 1]	[7, 1, 1]	[5, 1, 1]	[3, 1, 1]
μ^*	0.87879	0.79420	0.44896	0.57347	0.68213	0.72502
F	396.566	513.513	789.693	389.022	468.591	575.750
L_s	1.37236	2.92643	7.51312	2.69259	3.79130	4.78850
$E[V]$	6.54850	4.77761	0.27544	6.49628	5.17976	3.53719
$E[I]$	2.31358	2.70415	3.04246	1.75995	1.88825	2.32501
$E[B]$	1.13793	2.51825	6.68210	1.74377	2.93199	4.13780
$O.U.$	0.11379	0.25182	0.66821	0.17438	0.29320	0.41378
$\mathbf{H}$	[3, 6, 9]	[3, 6, 9]	[3, 6, 9]	[5, 7, 9]	[5, 7, 9]	[5, 7, 9]
$\mathbf{K}^*$	[7, 1, 1]	[5, 1, 1]	[2, 3, 1]	[7, 1, 1]	[6, 1, 1]	[3, 2, 1]
μ^*	0.45851	0.60457	0.68623	0.33455	0.63715	0.65375
F	402.340	468.538	545.198	433.021	446.713	524.935
L_s	4.00764	4.72612	6.22994	6.87781	7.41322	7.26313
$E[V]$	6.29034	5.10869	3.92765	5.79862	5.82133	4.16624
$E[I]$	1.52863	1.58319	1.70062	1.21225	1.03968	1.24484
$E[B]$	2.18098	3.30812	4.37173	2.98913	3.13898	4.58892
$O.U.$	0.21810	0.33081	0.43717	0.29891	0.31389	0.45889

6. Conclusions

In this paper, we have investigated a multi-server queueing system under multiple-threshold synchronous vacation policy, where partial servers may take a synchronous single vacation when the number of customers in the system is less than a pre-determined threshold. This system was formulated as a QBD process, and the necessary and sufficient condition for the stability of system was discussed. The steady-state probability and the closed-form expression of the rate matrix were obtained using matrix-analytical method with the aid of computer software. We proved the convergence property of the rate matrix and showed the explicit form of the rate matrix under a special case of single threshold. Based on the derived system performance measures, a cost model was developed to search for the optimal vacation policy and the optimal service rate, which minimize total expected cost function per unit time. Two approaches were implemented to deal with optimization problems of the discrete variables and continuous variables, respectively. Two examples were provided to illustrate the optimization procedure for each approach. We finally performed a sensitivity analysis between the optimal values of (μ, $\mathbf{K}$) and specific values of (λ, $\mathbf{H}$). The analysis presented in this paper would be helpful for decision makers to promote the competitiveness and profits of enterprize.

Acknowledgments

The authors are grateful to two referees and the Editor whose constructive comments have led to a substantial improvement in the presentation of the paper. The work was partly supported by National Science Council of Taiwan, under Contract No. 99-2221-E025-002-MY3.

References

[1] Q. Wu, S. Wu, Reliability analysis of two-unit cold standby repairable systems under Poisson shocks, Applied Mathematics and Computation 218 (2011) 171–182.
[2] A.D. Banik, Analyzing state-dependent arrival in GI/BMSP/1/∞ queues, Mathematical and Computer Modelling 53 (2011) 1229–1246.
[3] B.T. Doshi, Queueing systems with vacations-a survey, Queueing Systems 1 (1986) 29–66.
[4] H. Takagi, Queueing analysis: a foundation of performance evaluation, in: Vacation and Priority Systems, Part I, Vol. I, North-Holland, Amsterdam, 1991.
[5] G.V. Krishna Reddy, R. Nadarajan, R. Arumuganathan, Analysis of a bulk queue with N-policy multiple vacations and setup times, Computers & Operations Research 25 (1998) 957–967.
[6] G. Choudhury, An $M^{[x]}/G/1$ queueing system with a setup period and a vacation period, Queueing Systems 36 (2000) 23–38.
[7] G. Choudhury, A batch arrival queue with a vacation time under single vacation policy, Computers & Operations Research 29 (2002) 1941–1955.
[8] M. Shomrony, U. Yechiali, Burst arrival queues with server vacations and random timers, Mathematical Methods of Operations Research 53 (2001) 117–146.
[9] U. Yechiali, On the $M^{[x]}/G/1$ queue with a waiting server and vacations, Sankhyā 66 (2004) 159–174.
[10] L. Tadj, G. Choudhury, Optimal design and control of queues, TOP 13 (2) (2005) 359–412.
[11] J.C. Ke, Modified T vacation policy for an $M/G/1$ queueing system with an un-reliable server and startup, Mathematical and Computer Modelling 41 (2005) 1267–1277.
[12] J.C. Ke, Optimal NT policies for $M/G/1$ system with a startup and unreliable server, Computers and Industrial Engineering 50 (2006) 248–262.
[13] J.C. Ke, H.I. Huang, Y.K. Chu, Batch arrival queue with N-policy and at most J vacations, Applied Mathematical Modelling 34 (2010) 451–466.
[14] N. Tian, Z.G. Zhang, Vacation Queuing Models: Theory and Applications, Springer, New York, 2006.
[15] Y. Levy, U. Yechiali, An $M/M/c$ queue with servers' vacations, INFOR 14 (1976) 153–163.
[16] X. Chao, Y. Zhao, Analysis of multiple-server queues with station and server vacations, European Journal of Operational Research 110 (1998) 392–406.
[17] N. Tian, Q. Li, J. Cao, Conditional stochastic decompositions in the $M/M/c$ queue with server vacations, Stochastic Models 14 (2) (1999) 367–377.
[18] N. Tian, X. Xu, $M/M/c$ queue with synchronous multiple vacation of partial servers, OR Transactions 5 (3) (2001) 85–94.
[19] Z.G. Zhang, N. Tian, Analysis of queueing systems with synchronous single vacation for some servers, Queueing Systems 45 (2003) 161–175.
[20] Z.G. Zhang, N. Tian, Analysis on queueing systems with synchronous vacations of partial servers, Performance Evaluation 52 (2003) 269–282.
[21] X. Xu, Z.G. Zhang, Analysis of multiple-server queue with a single vacation (e, d)-policy, Performance Evaluation 63 (2006) 825–838.
[22] W. Peng, Algorithm for waiting time distribution of a discrete-time multiserver queue with deterministic service times and multi-threshold service policy, Procedia Computer Science 4 (2011) 1383–1392.
[23] J.C. Ke, C.H. Wu, W.L. Pearn, Algorithmic analysis of the multi-server system with a modified Bernoulli vacation schedule, Applied Mathematical Modelling 35 (2011) 22196–22208.
[24] M. Neuts, Matrix-Geometric Solutions in Stochastic Models, John Hopkins University Press, Baltimore, 1981.
[25] G. Latouche, V. Ramaswami, Introduction to Matrix Analytical Methods in Stochastic Modeling, SIAM, Philadelphia, PA, 1999.
[26] E.K.P. Chong, S.H. Zak, An Introduction to Optimization, second ed., John Wiley and Sons, Inc., 2001.