Optimization: A Journal of Mathematical Programming and Operations Research
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/gopt20

A heuristic algorithm for the optimization of a retrial system with Bernoulli vacation

Jau-Chuan Ke [a] , Chia-Huang Wu [b] & Wen Lea Pearn [b]
[a] Department of Applied Statistics , National Taichung Institute of Technology , Taichung , Taiwan , ROC
[b] Department of Industrial Engineering and Management , National Chiao Tung University , Taiwan , ROC
Published online: 25 May 2011.

Optimization, 2013
Vol. 62, No. 3, 299–321, http://dx.doi.org/10.1080/02331934.2011.579966

A heuristic algorithm for the optimization of a retrial system with Bernoulli vacation

Jau-Chuan Ke[a]*, Chia-Huang Wu[b] and Wen Lea Pearn[b]

[a]*Department of Applied Statistics, National Taichung Institute of Technology, Taichung, Taiwan, ROC;* [b]*Department of Industrial Engineering and Management, National Chiao Tung University, Taiwan, ROC*

(*Received 12 November 2010; final version received 6 April 2011*)

In this study, we consider an M/M/c retrial queue with Bernoulli vacation under a single vacation policy. When an arrived customer finds a free server, the customer receives the service immediately; otherwise the customer would enter into an orbit. After the server completes the service, the server may go on a vacation or become idle (waiting for the next arriving, retrying customer). The retrial system is analysed as a quasi-birth-and-death process. The sufficient and necessary condition of system equilibrium is obtained. The formulae for computing the rate matrix and stationary probabilities are derived. The explicit close forms for system performance measures are developed. A cost model is constructed to determine the optimal values of the number of servers, service rate, and vacation rate for minimizing the total expected cost per unit time. Numerical examples are given to demonstrate this optimization approach. The effects of various parameters in the cost model on system performance are investigated.

Keywords: Bernoulli vacation schedule; matrix-geometric method; quasi-Newton method; retrial; single vacation policy

1. Introduction

Retrial queueing system is characterized by the feature that the arriving customers who, on encountering the busy server, join a retrial queue called orbit. An arbitrary customer in the orbit generates a stream of repeated requests that is independent of the rest of customers in the orbit. This situation arises in telephony, where an arriving call is not allowed to await the termination of a busy signal. Such queueing systems play important roles in the analysis of many telephone switching systems, telecommunication networks and computer systems. Review of retrial queue literature could be found in Yang and Templeton [44], Falin and Templeton [23] and Artalejo [2]. A number of applications of retrial queues in science and engineering can be found in Kulkarni and Liang [28].

*Corresponding author. Email: jauchuan@ntit.edu.tw

Apart from its practical interest, due to its more accurate representation of several congestion phenomena, the multi-server retrial queue raises interesting mathematical and computational remarks. The investigation of the multi-server retrial queues is essentially more difficult than those with single server. Explicit formulae for the stationary distribution of M/M/c retrial queue are known only when the number of servers c is no more than two. Most multi-server retrial queues can be modelled by a level-dependent quasi-birth-and-death (QBD) process. The main feature of its infinitesimal generator is the spatial heterogeneity caused by transitions due to repeated attempts. This lack of homogeneity supports the analytical complexity of retrial models. Many interesting studies were devoted to an approximate approach of the stationary probabilities for system states (Falin [22], Bright and Taylor [9], Neuts and Rao [36], Stepanov [38], Artalejo and Pozo [7], Breuer et al. [8], Chakravarthy and Dudin [10]). Recently, Gomez-Corral [24] gave a detailed bibliographical guide to the analysis of retrial queues through matrix analytic techniques.

It is worth noting that the truncation models seem to be the most convenient method for obtaining reliable numerical solutions for the M/M/c retrial queue. For example, Falin [22] assumed that the retrial rate becomes infinite when the number of customers in orbit exceeds a level M. It means that, when the number of customers in the system is greater than M, the system performs as an ordinary M/M/1 queue with arrival rate λ and service rate $c\mu$, so that $\lambda < c\mu$ is a sufficient and necessary condition for system ergodicity. Neuts and Rao [36] and Artalejo and Pozo [7] proposed several models in this direction and provided efficient approximate solutions to the stationary distribution of the M/M/c retrial queue. As related works, a number of studies investigated the computation of the other system characteristics, such as the distributions of busy period, successful and blocked (unsuccessful) retrials, for the multi-server retrial queue of type M/M/c. The readers can refer to Artalejo et al. [4], Amador and Artalejo [1] and others. Artalejo et al. [5,6] presented an algorithmic analysis of the maximum number of customers in orbit (and in the system) during a busy period for the M/M/c retrial queue. The multi-server retrial queueing problems are extensively studied as mentioned earlier. However, in the literature, there are no detailed studies on multi-server retrial queue with a vacation at each service completion instantly.

Alternatively, queueing models with server vacations are practical models for performance analysis of manufacturing systems, local area networks and data communication systems. Past works on vacation queueing models include those with single-server and multiple-server systems. Surveys on the single-server vacation models have been reported by Doshi [21] and Takagi [41]. The variations and extensions of these vacation models were developed by several researchers such as Lee et al. [31,32], Krishna Reddy et al. [27], Choudhury [13,14], Shomrony and Yechiali [37], Ke and Chu [26] and many others. For the multiple-server vacation models, there are only a limited number of studies due to the complexity of the systems. The M/M/c queue with exponential vacations was first studied by Levy and Yechiali [33]. Chao and Zhao [11] investigated a GI/M/c vacation system and provided an algorithm to compute the performance measures. Tian et al. [42] 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 [45,46] and Xu and Zhang [43] analysed the M/M/c vacation systems with a 'partial server multiple vacation policy' in which some servers (only the idle ones) take single or multiple vacations.

Studies on various queueing models in the past are characterized by common feature; servers always serve the waiting customers in the queue until all customers are served exhaustively or the number of the waiting customers is dropped to predetermined level. In reality, however, it may occur that the service process requires to be temporarily stopped for overhauling at the end of a service. This overhauling can be utilized as a vacation in the presented model. For example, consider a production process with a number of machines (or c machines). A number of investigations (Madan et al. [34], Choudhury and Madan [18,19], Tadj et al. [39,40], and Choudhury et al. [20]) have recently appeared in queueing literature in which the single server provides to each service with Bernoulli schedule vacation (BSV). The so-called BSV means that when the service of a unit is completed, the server may leave for a vacation of random length with probability p to serve the next unit with probability $1-p$ (Choudhury and Madan [18,19]). Analytic steady-state solutions of a multi-sever retrial queue with Bernoulli schedules under a single vacation policy (BSV) have not been found. Multi-server vacation models are more flexible and applicable in practice than the queueing models with single server. Existing research works, including those mentioned above, have not addressed the analytical study and optimization issue in the multiple-server retrial queues in which the server may take a vacation upon his each service completion. This motivates us to discuss an M/M/c retrial queue with BSV by applying matrix analytic approach.

Recently, Choudhury [15,16] investigated the M/G/1 and $M^{[x]}$/G/1 queue with two phases of heterogeneous service and Bernoulli vacation schedule which operate under various retrial policies. Some extensive stationary analyses of the queueing system were carried out including the system size distribution and orbit size distribution. In the following year, Choudhury and Deka [17] dealt with the steady-state behaviour of $M^{[x]}$/G/1 retrial queue with second optional service, unreliable server and Bernoulli admission mechanism. The above-mentioned model generalizes both $M^{[x]}$/G/1 retrial queue with server breakdown and Bernoulli admission mechanism as well as $M^{[x]}$/G/1 queue with second optional service and unreliable server. Furthermore, Ke and Chang [25] derived the mathematical model of $M^{[x]}/(G_1, G_2)/1$ retrial queue under Bernoulli vacation schedules with general repeated attempts and starting failures. A practical mail system example was presented. Later, Langaris and Dimitriou [29] investigated a single-server queueing with n-phases of service and $(n-1)$ types of retrial customers. Some numerical results under exponentially distributed service time were provided. Artalejo [3] presented a bibliography on retrial queues made during the past decade 2000–2009.

This study considers an M/M/c retrial queue where primary customers arrive as a Poisson process with parameter λ. An arriving primary customer finding one or more servers available (free) gets service immediately. On the other hand, if the primary customer finds all servers busy, he joins the orbit and tries to get the service later.

There are c channels (servers) that provide service for the arrivals and the service times are assumed to be exponentially distributed with mean $1/\mu$. Each server can serve only one customer at a time. At each service completion instant of a server, the server may take a vacation of random length with probability p or wait to serve the next arrival with probability $q(=1-p)$. The vacation times follow an exponentially distributed with a parameter η. Furthermore, each customer staying in the orbit makes repeated attempts independently and the inter-retrial time is assumed to be exponentially distributed with parameter σ. Upon requesting service from the orbit, customer who finds all c servers busy always rejoins the orbit; this manner continues until he is eventually served. It is assumed that the number of customers in the orbit that is allowed to conduct retrials have an upper bound N (Neuts and Rao [36] and Artalejo and Pozo [7]). This implies that the probability of a repeated attempt during $(t, t+\mathrm{d}t)$, given that j customers in the orbit at time t, is $\sigma_j\,\mathrm{d}t + o(\mathrm{d}t)$, where $\sigma_j = \min\{j, N\}\sigma$ Moreover, the process of primary arrivals, service times, and inter-retrial times are assumed to be mutually independent. Conveniently, we represent this multi-server system with Bernoulli vacation as M/M/c/BSV retrial queue.

This article is organized as follows. In Section 2, the QBD model of the M/M/c/BSV retrial queue is set up. The computable form of the rate matrix is derived and the stable condition is obtained using the matrix-geometric property. In Section 3, an efficient algorithm is developed to find the stationary probabilities by matrix-geometric method. In Section 4, some system performance measures are derived. In Section 5, a cost model is developed to determine the optimal number of servers, service rate and vacation rate, simultaneously, in order to minimize the total expected cost per unit time. The quasi-Newton method and direct search method are implemented to deal with the optimization tasks. Some numerical examples are provided to illustrate the optimization procedures. In Section 6, conclusions are made with some remarks.

2. M/M/c/BSV retrial queue

For M/M/c/BSV retrial queue system, the state of the system can be described by the pair (i, j, k), $i = 0, 1, 2, \ldots, c$, $j = 0, 1, 2, \ldots$ and $k = 0, 1, 2, \ldots, c - i$, where i denotes the number of busy servers, j the number of customers in orbit (sources of repeated demands) and k the number of vacation servers. According to system assumptions, the number of customers in orbit allowed to conduct retrials is restricted to an appropriate number N ($N > c$), so the retrial rate is $\sigma_j = \min\{j, N\}\sigma$, $j \geq 0$ and one server would go on vacation with probability p ($p > 0$) or resumes service with probability $q = 1 - p$ at a service completion instant. The customers upon the server get services immediately as $i + k < c$. The new arriving customer who finds all c servers busy ($i + k \geq c$) always rejoins the retrial group (orbit).

In steady state, the steady-state probability is defined as

$P_{i,j}^{k} \equiv$ probability, that is, there are i busy servers, j customers in orbit and k vacation servers, where $0 \leq i + k \leq c$ and $j = 0, 1, 2, \ldots$.

2.1. *Matrix representation of M/M/c/BSV retrial queue*

The infinitesimal generator $\mathbf{Q}$ of the QBD describing the M/M/c/BSV retrial queueing system is

$$\mathbf{Q} = \begin{bmatrix} \mathbf{A}_0 & \mathbf{B} & & & & & & & \\ \mathbf{C}_1 & \mathbf{A}_1 & \mathbf{B} & & & & & & \\ & \mathbf{C}_2 & \mathbf{A}_2 & \mathbf{B} & & & & & \\ & & \ddots & \ddots & \ddots & & & & \\ & & & & \mathbf{C}_N & \mathbf{A}_{N-1} & \mathbf{B} & & \\ & & & & & \mathbf{C}_{N-1} & \mathbf{A}_N & \mathbf{B} & \\ & & & & & & \mathbf{C}_N & \mathbf{A}_N & \mathbf{B} \\ & & & & & & & \ddots & \ddots & \ddots \end{bmatrix}. \tag{1}$$

The entries $\mathbf{B}$, $\mathbf{A}_j\,(j > 0)$ and $\mathbf{C}_j\,(j > 1)$ are block-diagonal matrices of order $(c+1)(c+2)/2$ defined by

$$\mathbf{B} = \begin{bmatrix} \mathbf{b}^0 & & & & \\ & \mathbf{b}^1 & & & \\ & & \ddots & & \\ & & & \mathbf{b}^{c-1} & \\ & & & & \mathbf{b}^c \end{bmatrix} \text{ and } \mathbf{C}_j = \begin{bmatrix} \mathbf{c}_j^0 & & & & \\ & \mathbf{c}_j^1 & & & \\ & & \ddots & & \\ & & & \mathbf{c}_j^{c-1} & \\ & & & & \mathbf{c}_j^c \end{bmatrix}, \ j = 1, 2, \ldots$$

where sub-matrices $\mathbf{b}^i$ and $\mathbf{c}_j^i$ are $(c+1-i) \times (c+1-i)$ square matrices with elements

$$\begin{cases} \mathbf{b}^i[c+1-i, c+1-i] = \lambda & \\ 0 & e.w \end{cases} \quad \text{and} \quad \begin{cases} \mathbf{c}_j^i[k, k+1] = \sigma_j, & 1 \le k \le c-i \\ 0 & e.w \end{cases}$$

$$\mathbf{A}_j = \begin{bmatrix} \mathbf{Y}_j^0 & \mathbf{X}^0 & & & & & \\ \mathbf{Z}^1 & \mathbf{Y}_j^1 & \mathbf{X}^1 & & & & \\ & \mathbf{Z}^2 & \mathbf{Y}_j^2 & \mathbf{X}^2 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \mathbf{Z}^{c-1} & \mathbf{Y}_j^{c-1} & \mathbf{X}^{c-1} \\ & & & & & \mathbf{Z}^c & \mathbf{Y}_j^c \end{bmatrix}, \quad j = 0, 1, 2, \ldots$$

where $\mathbf{X}^i$ is a $(c+1-i) \times (c-i)$ matrix with $\mathbf{X}^i[k+1, k] = kp\mu, 1 \le k \le c-i$, $\mathbf{Z}^i$ a $(c-i) \times (c+1-i)$ matrix with $\mathbf{Z}^i[k, k] = i\eta, 1 \le k \le c-i$ and $\mathbf{Y}_j^i$ a square matrix of order $(c+1-i)$ with elements

$$\begin{cases} \mathbf{Y}_j^i[k, k+1] = \lambda, & 1 \le k \le c-i \\ \mathbf{Y}_j^i[k+1, k] = k(1-p)\mu, & 1 \le k \le c-i \\ \mathbf{Y}_j^i[1, 1] = -[\lambda + (i-1)\eta + \sigma_j] & \\ \mathbf{Y}_j^i[k, k] = -[\lambda + (k+1)\mu + i\eta + \sigma_j], & 2 \le k \le c-i \\ \mathbf{Y}_j^i[c+1-i, c+1-i] = -[\lambda + i\eta + (c-i)\mu] & \end{cases}.$$

The detailed descriptions of the above matrices (for $c=3$) are given in the Appendix.

Let $\mathbf{\Pi}=[\mathbf{\Pi}_0,\mathbf{\Pi}_1,\mathbf{\Pi}_2,\ldots]$ with $\mathbf{\Pi}_i=[P^0_{0,i},P^0_{1,i},\ldots,P^0_{c,i},P^1_{0,i},P^1_{1,i},\ldots,P^1_{c-1,i},P^0_{0,i},\ldots,P^{c-1}_{0,i},P^{c-1}_{1,i},P^c_{0,i}]$, $i=0,1,2,\ldots$ be the unique solution to $\mathbf{\Pi Q}=\mathbf{0}$ and $\mathbf{\Pi e}=1$, where $\mathbf{e}$ is a column vector with all elements equal to 1. It is noted that the vector $\mathbf{\Pi}=[\mathbf{\Pi}_0,\mathbf{\Pi}_1,\mathbf{\Pi}_2,\mathbf{\Pi}_3,\ldots]$ with the following properties

$$\mathbf{\Pi}_{N+k}=\mathbf{\Pi}_N\mathbf{R}^k,\quad \text{for } k\geq 1. \tag{2}$$

The matrix $\mathbf{R}$ is the unique non-negative solution with spectral radius less than 1 of the equation

$$\mathbf{B}+\mathbf{R}\mathbf{A}_N+\mathbf{R}^2\mathbf{C}_N=\mathbf{0}. \tag{3}$$

From Neuts [35] and Latouche and Ramaswami [30], it is known that $\mathbf{R}$ is given by $\lim_{n\to\infty}\mathbf{R}_n$, where the sequence $\{\mathbf{R}_n\}$ is defined by

$$\mathbf{R}_0=\mathbf{0},\quad \text{and}\quad \mathbf{R}_{n+1}=-\mathbf{B}\mathbf{A}_N^{-1}-\mathbf{R}_n^2\mathbf{C}_N\mathbf{A}_N^{-1},\quad \text{for } n\geq 0. \tag{4}$$

The sequence $\{\mathbf{R}_n\}$ is monotone so that $\mathbf{R}$ could be evaluated from (4) by successive substitutions.

2.2. *Stability condition*

It is also known (Theorem 3.1.1 of Neuts [35]) that the steady-state probability vector exists if and only if

$$\mathbf{xBe}<\mathbf{xC}_N\mathbf{e}, \tag{5}$$

where $\mathbf{x}$ is the invariant probability of the matrix $\mathbf{F}=\mathbf{C}_N+\mathbf{A}_N+\mathbf{B}$. Here, $\mathbf{x}$ satisfies $\mathbf{xF}=\mathbf{0}$ and $\mathbf{xe}=1$. First we solve $\mathbf{xF}=\mathbf{0}$, where $\mathbf{x}=[\mathbf{x}^0_0,\mathbf{x}^0_1\ldots,\mathbf{x}^0_c,\ldots,\mathbf{x}^{c-1}_0,\mathbf{x}^{c-1}_1,\mathbf{x}^c_0]$. We can get the following $(c+1)(c+2)/2$ equations:

For $k=0$,

$$-(\lambda+N\sigma)\mathbf{x}^0_0+q\mu\mathbf{x}^0_1+\eta\mathbf{x}^1_0=0, \tag{6-1a}$$

$$(\lambda+N\sigma)\mathbf{x}^0_{i-1}-(\lambda+i\mu+N\sigma)\mathbf{x}^0_i+(i+1)q\mu\mathbf{x}^0_{i+1}+\eta\mathbf{x}^1_i=0,\quad 1\leq i\leq c-1, \tag{6-1b}$$

$$(\lambda+N\sigma)\mathbf{x}^0_{c-1}-c\mu\mathbf{x}^0_c=0. \tag{6-1c}$$

For $1\leq k\leq c-1$,

$$p\mu\mathbf{x}^{k-1}_1-(\lambda+N\sigma+k\eta)\mathbf{x}^k_0+q\mu\mathbf{x}^k_1+(k+1)\eta\mathbf{x}^{k+1}_0=0, \tag{6-2a}$$

$$\begin{aligned}&(\lambda+N\sigma)\mathbf{x}^k_{i-2}+ip\mu\mathbf{x}^{k-1}_i-[\lambda+N\sigma+(i-1)\mu+k\eta]\mathbf{x}^k_{i-1}\\&\quad+iq\mu\mathbf{x}^k_i+(k+1)\eta\mathbf{x}^{k+1}_{i-1}=0,\quad 2\leq i\leq c-k,\end{aligned} \tag{6-2b}$$

$$(c+1-k)p\mu\mathbf{x}^{k-1}_{c+1-k}+(\lambda+N\sigma)\mathbf{x}^k_{c-k-1}-[(c-k)\mu+k\eta]\mathbf{x}^k_{c-k}=0. \tag{6-2c}$$

For $k = c$,

$$p\mu \mathbf{x}_1^{c-1} - c\eta \mathbf{x}_0^c = 0. \tag{6-3a}$$

Using a effective Maple software to solve Equations (6-1a)–(6-2c), the following results are derived

$$\mathbf{x}_i^k = \frac{c!\eta^{c-k}}{i!k!(\lambda + N\sigma)^{c-i-k}\mu^i p^{c-k}} \mathbf{x}_0^c, \quad 0 \le i + k \le c. \tag{7}$$

Then, using the normalization condition $\mathbf{xe} = 1$, $\mathbf{x}_0^c$ can be determined as

$$\mathbf{x}_0^c = \left[\sum_{k=0}^{c} \sum_{i=0}^{c-k} \frac{c!\eta^{c-k}}{i!k!(\lambda + N\sigma)^{c-i-k}\mu^i p^{c-k}} \right]^{-1} \tag{8}$$

Substituting $\mathbf{B}$ and $\mathbf{C}_N$ into Equation (5) and doing some routine manipulations, then we have

$$N\sigma(1 - P_F) > \lambda P_{\text{Full}}, \tag{9}$$

where

$$\begin{aligned} P_{\text{Full}} &= \sum_{i=0}^{c} \mathbf{x}_i^{c-i} = \sum_{i=0}^{c} \frac{c!\eta^i}{i!(c-i)!\mu^i p^i} \mathbf{x}_0^c \\ &= \left(1 + \frac{\eta}{p\mu}\right)^c \left[\sum_{k=0}^{c} \sum_{i=0}^{c-k} \frac{c!\eta^{c-k}}{i!k!(\lambda + N\sigma)^{c-i-k}\mu^i p^{c-k}} \right]^{-1}, \end{aligned} \tag{10}$$

which is referred to the probability that all normal working (non-vacation) servers are busy (i.e. $i + k = c$). That is, the system will be stable if the expected successful retrial rate is greater then the expected arrival rate of ‘orbit’.

3. Steady-state solution

Under the stability condition, the stationary probability vector $\mathbf{\Pi}$ exists. We deal with the steady-state equations using matrix technique. The steady-state equations are given by

$$\mathbf{\Pi}_0\mathbf{A}_0 + \mathbf{\Pi}_1\mathbf{C}_1 = \mathbf{0}, \tag{11a}$$

$$\mathbf{\Pi}_{i-1}\mathbf{B} + \mathbf{\Pi}_i\mathbf{A}_i + \mathbf{\Pi}_{i+1}\mathbf{C}_{i+1} = \mathbf{0}, \quad 1 \le i \le N-1, \tag{11b}$$

$$\mathbf{\Pi}_{N-1}\mathbf{B} + \mathbf{\Pi}_N\mathbf{A}_N + \mathbf{\Pi}_N\mathbf{R}\mathbf{C}_N = \mathbf{0}, \tag{11c}$$

$$\mathbf{\Pi}_N\mathbf{R}^{i-1-N}\mathbf{B} + \mathbf{\Pi}_N\mathbf{R}^{i-N}\mathbf{A}_N + \mathbf{\Pi}_N\mathbf{R}^{i+1-N}\mathbf{C}_N = \mathbf{0}, \quad N+1 \le i, \tag{11d}$$

$$\sum_{i=0}^{\infty} \mathbf{\Pi}_i\mathbf{e} = 1. \tag{12}$$

After doing some routine manipulations to Equations (11a)–(11c) recursively, we have

$$\begin{aligned}\mathbf{\Pi}_0 &= \mathbf{\Pi}_1\mathbf{C}_1(-\mathbf{A}_0)^{-1} = \mathbf{\Pi}_1\phi_1,\\ \mathbf{\Pi}_{i-1} &= \mathbf{\Pi}_i\mathbf{C}_i[-(\phi_{i-1}\mathbf{B}+\mathbf{A}_{i-1})]^{-1} = \mathbf{\Pi}_i\phi_i, \quad 2\le i\le N,\end{aligned} \tag{13}$$

and

$$\mathbf{\Pi}_N\phi_N\mathbf{B}+\mathbf{\Pi}_N\mathbf{A}_N+\mathbf{\Pi}_N\mathbf{R}\mathbf{C}_N=\mathbf{0}. \tag{14}$$

Consequently, $\mathbf{\Pi}_i(0\le i\le N-1)$ in Equation (13) can be written in terms of $\mathbf{\Pi}_N$ as $\mathbf{\Pi}_0=\mathbf{\Pi}_N\prod_{i=N}^{1}\phi_i$, $\mathbf{\Pi}_1=\mathbf{\Pi}_N\prod_{i=N}^{2}\phi_i,\ldots,\mathbf{\Pi}_{N-1}=\mathbf{\Pi}_N\prod_{i=N}^{N}\phi_i$ and the rest steady-state vector $[\mathbf{\Pi}_N,\mathbf{\Pi}_{N+1},\mathbf{\Pi}_{N+2},\ldots]$ can be determined recursively as $\mathbf{\Pi}_i=\mathbf{\Pi}_N\mathbf{R}^{i-N}$, for $i\ge N$. Therefore, once the steady-state probability $\mathbf{\Pi}_N$ is obtained, the steady-state solutions $[\mathbf{\Pi}_0,\mathbf{\Pi}_1,\mathbf{\Pi}_2,\ldots,\mathbf{\Pi}_{N-1},\mathbf{\Pi}_N,\mathbf{\Pi}_{N+1},\ldots]$ are determined. The steady-state probability $\mathbf{\Pi}_N$ can be solved by Equation (14) with the following normalization equation

$$\begin{aligned}\sum_{i=0}^{\infty}\mathbf{\Pi}_i\mathbf{e} &= [\mathbf{\Pi}_0+\mathbf{\Pi}_1+\cdots+\mathbf{\Pi}_{N-1}+\mathbf{\Pi}_N+\mathbf{\Pi}_{N+1}+\mathbf{\Pi}_{N+2}+\cdots]\mathbf{e}\\ &= [\mathbf{\Pi}_N\prod_{i=N}^{1}\phi_i+\mathbf{\Pi}_N\prod_{i=N}^{2}\phi_i+\cdots+\mathbf{\Pi}_N\prod_{i=N}^{N}\phi_i+\mathbf{\Pi}_N+\mathbf{\Pi}_N\mathbf{R}+\mathbf{\Pi}_N\mathbf{R}^2+\cdots]\mathbf{e}\\ &= \mathbf{\Pi}_N\left[\sum_{k=1}^{N}\prod_{i=N}^{k}\phi_i+(\mathbf{I}-\mathbf{R})^{-1}\right]\mathbf{e}=1.\end{aligned} \tag{15}$$

where $\mathbf{I}$ denotes the identity matrix with suitable size. Solving Equations (14) and (15) in accordance with Cramer's rule, $\mathbf{\Pi}_N$ can be obtained. Then, the prior state probabilities $[\mathbf{\Pi}_0,\mathbf{\Pi}_1,\mathbf{\Pi}_2,\ldots,\mathbf{\Pi}_{N-1}]$ are computed from (13) and $[\mathbf{\Pi}_{N+1},\mathbf{\Pi}_{N+2},\mathbf{\Pi}_{N+3},\ldots]$ are gained by the formula $\mathbf{\Pi}_i=\mathbf{\Pi}_N\mathbf{R}^{i-N}$, $i\ge N+1$. The solution procedure of steady-state probabilities is summarized as follows:

Algorithm Recursive Solver

Step 1 Set $\phi_1=\mathbf{C}_1(-\mathbf{A}_0)^{-1}$

Step 2 For i from 2 to N, set $\phi_i=\mathbf{C}_i[-(\phi_{i-1}\mathbf{B}+\mathbf{A}_{i-1})]^{-1}$.

Step 3 For k from 1 to N, set $\mathbf{\Phi}_k=\prod_{i=N}^{k}\phi_i$.

Step 4 Solving $\mathbf{\Pi}_N\phi_N\mathbf{B}+\mathbf{\Pi}_N\mathbf{A}_N+\mathbf{\Pi}_N\mathbf{R}\mathbf{C}_N=\mathbf{0}$, $\mathbf{\Pi}_N[\sum_{k=1}^{N}\mathbf{\Phi}_k+(\mathbf{I}-\mathbf{R})^{-1}]\mathbf{e}=1$ and obtain steady-state probability $\mathbf{\Pi}_N$.

Step 5 Construct steady-state probability $\mathbf{\Pi}_i$ as follows:

(a) if $0\le i\le N$, assign $\mathbf{\Pi}_i=\mathbf{\Pi}_N\mathbf{\Phi}_{i+1}$,
(b) if $N\le i$, assign $\mathbf{\Pi}_{i+1}=\mathbf{\Pi}_iR$,

4. System performance measures

There are several system descriptors (system performance measures) of the M/M/c/BSV retrial queue, such as the expected number of busy servers (denoted by $E[B]$), the expected number of vacation servers (denoted by $E[V]$) and the expected number of customers in orbit (denoted by $E[\text{Orbit}]$), which can be evaluated from the steady-state probabilities. The explicit expressions for $E[B]$, $E[V]$, and $E[\text{Orbit}]$ are given by

$$\begin{aligned} E[B] &= \sum_{j=0}^{\infty} \mathbf{\Pi}_j \mathbf{v} = \sum_{j=0}^{N-1} \mathbf{\Pi}_j \mathbf{v} + \mathbf{\Pi}_N \mathbf{v} + \mathbf{\Pi}_N \mathbf{R} \mathbf{v} + \mathbf{\Pi}_N \mathbf{R}^2 \mathbf{v} + \cdots \\ &= \sum_{j=0}^{N-1} \mathbf{\Pi}_N \mathbf{\Phi}_{j+1} \mathbf{v} + \mathbf{\Pi}_N \mathbf{v} + \mathbf{\Pi}_N \mathbf{R} \mathbf{v} + \mathbf{\Pi}_N \mathbf{R}^{\mathbf{2}} \mathbf{v} + \cdots \\ &= \mathbf{\Pi}_N \left[\sum_{j=1}^{N} \mathbf{\Phi}_{\mathbf{j}} + (\mathbf{I} - \mathbf{R})^{-1} \right] \mathbf{v} \end{aligned} \tag{16}$$

$$\begin{aligned} E[V] &= \sum_{j=0}^{\infty} \mathbf{\Pi}_j \mathbf{u} = \sum_{j=0}^{N-1} \mathbf{\Pi}_j \mathbf{u} + \mathbf{\Pi}_N \mathbf{u} + \mathbf{\Pi}_N \mathbf{R} \mathbf{u} + \mathbf{\Pi}_N \mathbf{R}^2 \mathbf{u} + \cdots \\ &= \sum_{j=0}^{N-1} \mathbf{\Pi}_N \mathbf{\Phi}_{j+1} \mathbf{u} + \mathbf{\Pi}_N (\mathbf{I} - \mathbf{R})^{-1} \mathbf{u} \\ &= \mathbf{\Pi}_N \left[\sum_{j=1}^{N} \mathbf{\Phi}_j + (\mathbf{I} - \mathbf{R})^{-1} \right] \mathbf{u} \end{aligned} \tag{17}$$

$$\begin{aligned} E[\text{Orbit}] &= \sum_{j=1}^{\infty} j \mathbf{\Pi}_j \mathbf{e} = \sum_{j=1}^{N-1} j \mathbf{\Pi}_N \mathbf{\Phi}_{j+1} \mathbf{e} + N \mathbf{\Pi}_N \mathbf{e} + (N+1) \mathbf{\Pi}_N \mathbf{R} \mathbf{e} + (N+2) \mathbf{\Pi}_N \mathbf{R}^2 \mathbf{e} + \cdots \\ &= \sum_{j=2}^{N} (j-1) \mathbf{\Pi}_N \mathbf{\Phi}_j \mathbf{e} + \mathbf{\Pi}_N [N(\mathbf{I} - \mathbf{R})^{-1} + \mathbf{R}(\mathbf{I} - \mathbf{R})^{-2}] \mathbf{e} \\ &= \Pi_N \left[\sum_{j=2}^{N} (j-1) \mathbf{\Phi}_j + N(\mathbf{I} - \mathbf{R})^{-1} + \mathbf{R}(\mathbf{I} - \mathbf{R})^{-2} \right] \mathbf{e} \end{aligned} \tag{18}$$

where

$$\mathbf{v} = \left[\underbrace{0, 1, \ldots, c}_{\#=c+1}, \underbrace{0, 1, \ldots, c-1}_{\#=c}, \ldots, \underbrace{0, 1}_{\#=2}, 0 \right] \quad \text{and}$$

$$\mathbf{u} = \left[\underbrace{0, 0, \ldots, 0}_{\#=c+1}, \underbrace{1, 1, \ldots, 1}_{\#=c}, \ldots, \underbrace{c-1, c-1}_{\#=2}, c \right]$$

are column vectors with dimension $(c+1)(c+2)/2$.

4.1. *System performance versus system parameters*

For an M/M/c/BSV retrial queue, the numerical results of $E[\text{Orbit}]$ are obtained by considering the following four cases with different values of c.

Case 1 $N = 30$, $\lambda = 5$, $\eta = 10$, $p = 0.5$, $\sigma = 5$, varying μ from 10 to 15.

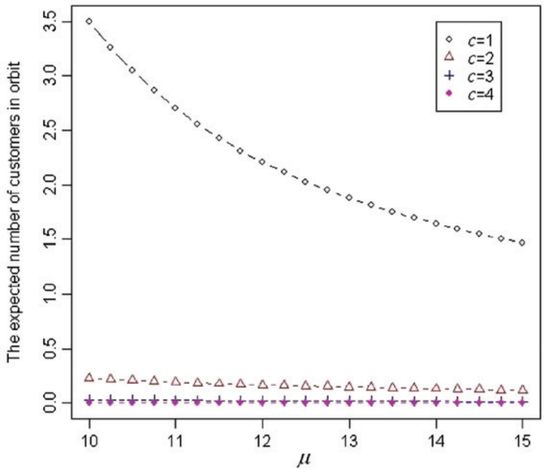

Figure 1. The expected number of customers in orbit E[Orbit] versus μ.

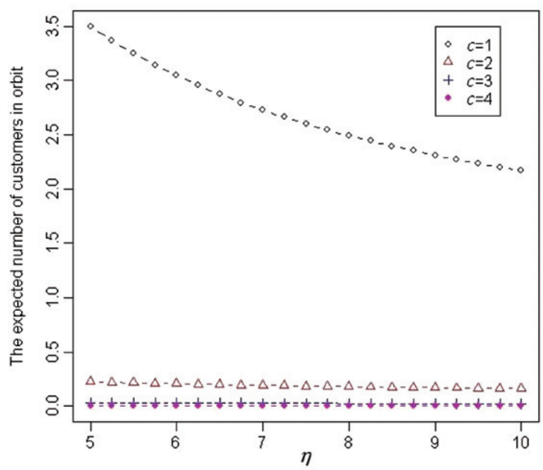

Figure 2. The expected number of customers in orbit E[Orbit] versus η.

Case 2 $N = 30$, $\lambda = 5$, $\mu = 10$, $p = 0.5$, $\sigma = 10$, varying η from 10 to 15.

Case 3 $N = 30$, $\mu = 15$, $\eta = 15$, $p = 0.5$, $\sigma = 10$, varying λ from 5 to 10.

Case 4 $N = 30$, $\lambda = 5$, $\mu = 15$, $\eta = 15$, $p = 0.5$, varying σ from 10 to 15.

Results of E[Orbit] are depicted in Figures 1–4 for Cases 1–4, respectively. One sees from Figures 1 and 2 that E[Orbit] drastically decreases as μ or η increases for $c = 1$, while E[Orbit] is not sensitive to μ or η for $c \geq 2$. It reveals from Figure 3 that E[Orbit] increases violently as λ increases for $c = 1$, while E[Orbit] slightly increases as λ increases for $c \geq 2$. Figure 4 reports that E[Orbit] decreases as σ increases for $c = 1$, while E[Orbit] is not sensitive to σ for $c \geq 2$.

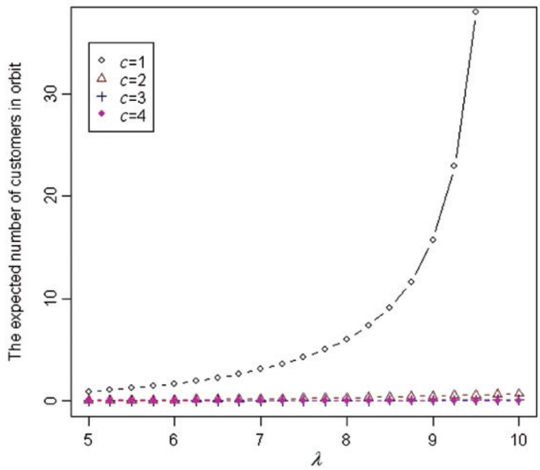

Figure 3. The expected number of customers in orbit E[Orbit] versus λ.

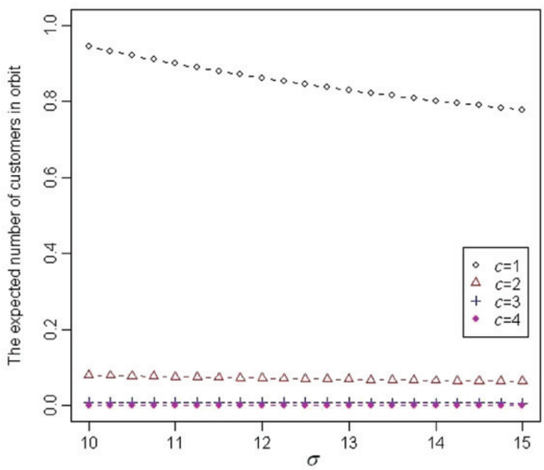

Figure 4. The expected number of customers in orbit E[Orbit] versus σ.

There are several general descriptors of retrial queues, some of which are listed below:

(1) The overall rate of retrials

$$\begin{aligned}\sigma_1^* &= \sum_{j=1}^{N} j\sigma \sum_{k=0}^{c} \sum_{i=0}^{c-k} P_{i,j}^k + \sum_{j=N+1}^{\infty} N\sigma \sum_{k=0}^{c} \sum_{i=0}^{c-k} P_{i,j}^k = \sum_{j=1}^{N} j\sigma \Pi_j e + \sum_{j=N+1}^{\infty} N\sigma \Pi_N R^{j-N} e \\ &= \sum_{j=1}^{N} j\sigma \mathbf{\Pi}_j \mathbf{e} + N\sigma \mathbf{\Pi}_N \mathbf{R}(\mathbf{I}-\mathbf{R})^{-1}\mathbf{e} = \sigma\left[\sum_{j=1}^{N} j\mathbf{\Pi}_j + N\mathbf{\Pi}_N \mathbf{R}(\mathbf{I}-\mathbf{R})^{-1}\right]\mathbf{e}. \\ &= \sigma \mathbf{\Pi}_N \left[\sum_{j=1}^{N-1} j\mathbf{\Phi}_{j+1} + N(\mathbf{I}-\mathbf{R})^{-1}\right]\mathbf{e} \end{aligned} \tag{19}$$

(2) The rate of retrials that are successful

$$\sigma_2^* = \sum_{j=1}^{N} j\sigma \sum_{k=0}^{c} \sum_{i=0}^{c-k-1} P_{i,j}^k + \sum_{j=N+1}^{\infty} N\sigma \sum_{k=0}^{c} \sum_{i=0}^{c-k-1} P_{i,j}^k. \tag{20}$$

(3) The fraction of retrials that are successful

$$F = \frac{\sigma_2^*}{\sigma_1^*}. \tag{21}$$

(4) The marginal distribution of the number of busy servers

$$\sum_{j=0}^{\infty} P_{i,j}^k, \quad 0 \le i + k \le c. \tag{22}$$

(5) Busy period: The busy period T of a retrial queue is defined as the period that starts at the epoch when an arriving customer finds an empty system (all servers are idle and no customer in the orbit) and ends at the departure epoch at which the system is empty again.

The mean busy period

$$E(T) = \frac{1}{\lambda}\left(\frac{1}{P_{0,0}^0} - 1\right) = \frac{1}{\lambda}\left(\frac{1}{\mathbf{\Pi}_N \mathbf{\Phi}_1[1]} - 1\right) \tag{23}$$

where the symbol '$\mathbf{\Pi}_N \mathbf{\Phi}_1[1]$' denotes the first element of the column vector $\mathbf{\Pi}_N \mathbf{\Phi}_1$.

(6) Vain retrials: A vain retrial is an unsuccessful retrial when all servers are busy.

The steady-state probability of vain retrial P_V

$$P_V = \frac{\sum_{j=1}^{\infty} \sum_{i+k=c} P_{i,j}^k}{\sum_{j=1}^{\infty} \sum_{k=0}^{c} \sum_{i=0}^{c-k} P_{i,j}^k} = \frac{\sum_{j=1}^{\infty} \sum_{i+k=c} P_{i,j}^k}{1 - \mathbf{\Pi}_0 \mathbf{e}}. \tag{24}$$

4.2. *System performance versus truncated parameters*

To understand how system performance measures listed above vary with N, we also perform a numerical investigation to the measures based on changing the value of N from 5 to 25, which is based on $\lambda = 5$, $\mu = 15$, $p = 0.5$, $\sigma = 10$, and $\eta = 10$. The numerical illustration is graphically presented in Figures 5–8.

From Figures 5–8, it is clear that increasing the retrial rate beyond a certain point does not result in a commensurate improvement in the system performance, which is according with the result of Neuts and Rao [36].

5. Optimization analysis

In this section, we construct the total expected cost function per unit time based on the system performance measures for the M/M/c/BSV retrial queue, in which the

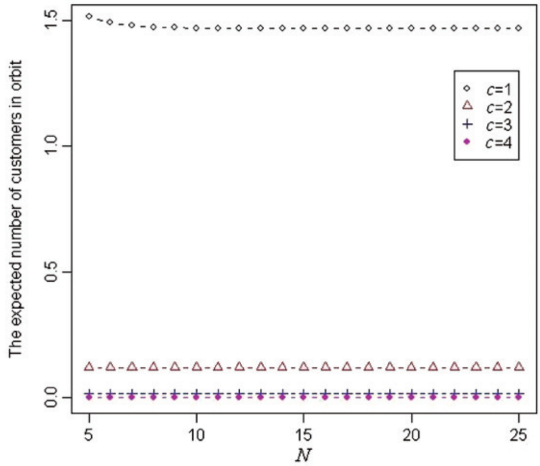

Figure 5. The expected number of customers in orbit E[Orbit] versus N.

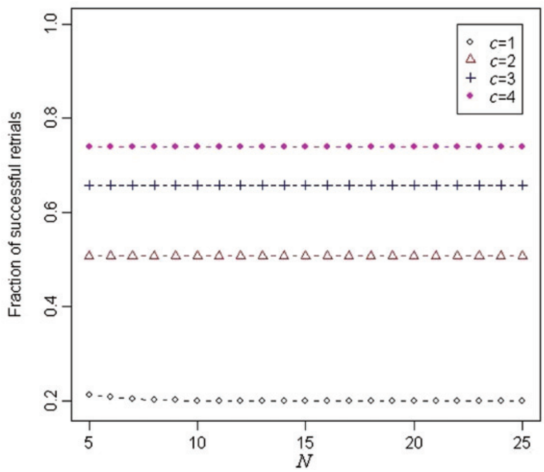

Figure 6. The fraction of successful retrials F versus N.

number of servers (c) is a discrete decision variable, and the service rate (μ) and the vacation rate (η) are continuous decision variables. Let us define the following cost elements:

$C_h \equiv$ holding cost per unit time per customer present in orbit;

$C_s \equiv$ cost per unit time of providing a service rate μ;

$C_v \equiv$ cost per unit time when one server is on vacation;

$C_r \equiv$ cost per unit time of providing a vacation rate η; and

$C_p \equiv$ fixed cost for purchasing one server.

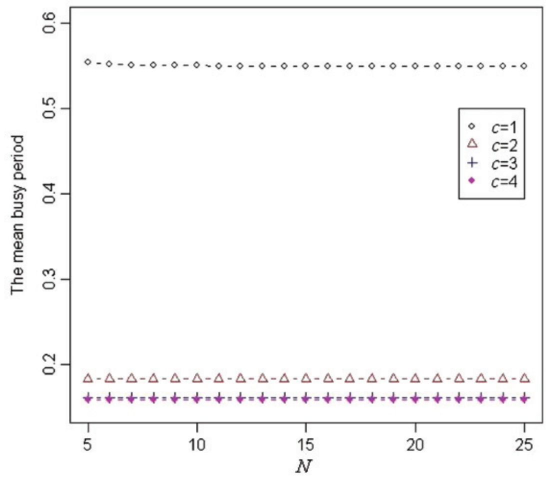

Figure 7. The mean busy period $E[T]$ versus N.

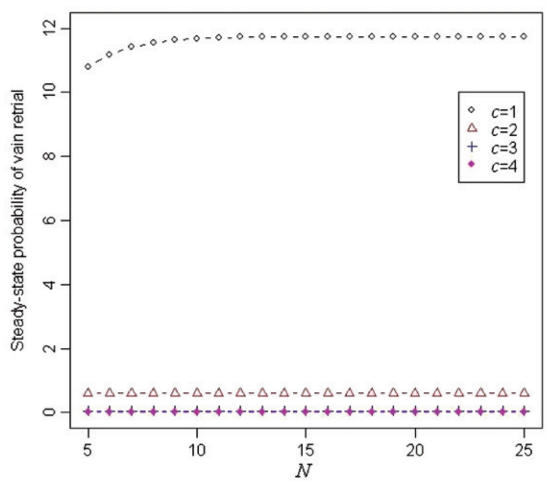

Figure 8. The steady-state probability of vain retrial P_V versus N.

Based on the definition of the cost parameters, the total expected cost function per unit time can be expressed as

$$F(c, \mu, \eta) = C_h E[\text{Orbit}] + C_s\mu + C_v E[V] + C_r\eta + C_p c \tag{25}$$

where $E[\text{Orbit}]$ and $E[V]$ are defined previously.

The main objective is to find the optimal number of servers c^*, and the optimal values of service rate and vacation rate (μ^*, η^*) simultaneously which minimize the cost function $F(c, \mu, \eta)$. The analytical study of the optimization behaviour of the expected cost function would have been an arduous task to undertake since the decision variables appear in an expression which is a highly non-linear and complex and non-linear in terms of (c, μ, η). Next, two methods are provided to deal with this problem heuristically.

In the next section, we first use the quasi-Newton method to find the approximate optimal value of continuous variable (μ, η), say (μ^*, η^*), and then use direct search method to search the optimal value of discrete variable c, say c^*.

5.1. *Quasi-Newton method for optimal* (μ, η)

For practice situation of purchase budget, the number of servers is bounded by a positive integer $c_U \geq 1$. We want to find the joint optimal value (μ^*,η^*) for each given c in the feasible set $\{1, 2, \ldots, c_U\}$. The cost minimization problem can be illustrated mathematically as

$$F(c, \mu^*, \eta^*) = \min_{\text{and s.t. (9)}} \{F(c, \mu, \eta)|c\}, \quad c = 1, 2, \ldots, c_U \tag{26}$$

For the problem of (26), we should show the convexity of $F(c, \mu, \eta)$ in (μ, η). However, this study is difficult to implement. It is noted that the derivative of the cost function F with respect to (μ, η) indicates the direction at which the cost function increases. It means that, the optimal value (μ^*, η^*) can be found along this opposite direction of the gradient (Chong and Zak [12]). That is, for a fixed c, quasi-Newton method is employed to search (μ, η) until the approximate minimum value of $F(c, \mu, \eta)$ is achieved, say $F(c, \mu^*, \eta^*)$. An effective procedure that makes it possible to calculate the optimal value (c, μ^*, η^*) is presented as follows:

Algorithm Quasi-Newton Method

Step 1 Set the initial trial solution for $\vec{\boldsymbol{\theta}}^{(0)}$, and compute $F(c, \mu^{(0)}, \eta^{(0)})$.

Step 2 Compute the cost gradient $\vec{\nabla}F(\vec{\boldsymbol{\theta}}) = [\partial F/\partial\mu, \partial F/\partial\eta]^{\mathrm{T}}$ and the cost Hessian matrix

$$H(\vec{\boldsymbol{\theta}}) = \begin{bmatrix} \partial^2 F/\partial\mu^2 & \partial^2 F/\partial\mu\partial\eta \\ \partial^2 F/\partial\eta\partial\mu & \partial^2 F/\partial\eta^2 \end{bmatrix} \quad \text{at point } \vec{\boldsymbol{\theta}}^{(i)}.$$

Step 3 While $|\partial F/\partial\mu| > \varepsilon$ or $|\partial F/\partial\eta| > \varepsilon$, set the new trial solution $\vec{\boldsymbol{\theta}}^{(i+1)} = \vec{\boldsymbol{\theta}}^{(i)} - [H(\vec{\boldsymbol{\theta}}^{(i)})]^{-1}\vec{\nabla}F(\vec{\boldsymbol{\theta}}^{(i)})$ and return to Step 2.

To demonstrate the validness and the approximate optimization solution, we perform some computation and analysis on the examples given in Table 1 by considering the following cost parameters as

$C_h =$ \$25/customer/unit time, $C_s =$ \$45/unit time,
$C_v =$ \$120/server/unit time, $C_r =$ \$90/unit time, $C_p =$ \$120/server

From Table 1, it can be seen that the minimum expected cost per unit time of 1474.377 is achieved at $(\mu^*, \eta^*) = (11.54626, 6.305710)$ by using six iterations, which is based on Case (i) with initial value $(c, \mu, \eta) = (1, 15, 5)$. Based on Case (ii) with

Table 1. The illustration of the implement process of quasi-Newton method.

Iterations	0	1	2	3	4	5	6
Case (i): $(\lambda, p, \sigma) = (5, 0.5, 10)$ with initial value $(c, \mu, \eta) = (1, 15, 5)$							
$F(c, \mu, \eta)$	1544.435	1517.015	1482.721	1474.921	1474.380	1474.377	1474.377
μ	15	10.74763	11.11560	11.41594	11.53441	11.54617	11.54626
η	5	5.932174	6.131345	6.263916	6.303111	6.305700	6.305710
$\frac{\partial F}{\partial \mu}$	15.31879	−78.2392	−25.8695	−5.64068	−0.43039	−0.00300	-7.8×10^{-8}
$\frac{\partial F}{\partial \eta}$	−73.2424	−133.720	−43.6031	−9.22994	−0.66640	−0.00424	-1.5×10^{-7}
E[Orbit]	7.177405	10.75622	8.070767	6.782341	6.418249	6.388411	6.388210
$E[V]$	0.500000	0.421422	0.407740	0.399111	0.396630	0.396467	0.396466
Case (ii): $(\lambda, p, \sigma) = (10, 0.8, 15)$ with initial value $(c, \mu, \eta) = (2, 10, 10)$							
$F(c, \mu, \eta)$	2037.910	1988.860	1971.630	1968.793	1968.692	1968.692	1968.692
μ	10	11.05421	11.93856	12.42039	12.52661	12.53093	12.53093
η	10	9.256253	8.869115	8.722289	8.697166	8.696282	8.696281
$\frac{\partial F}{\partial \mu}$	−98.0608	−41.9620	−13.3913	−2.29042	−0.09060	−0.00016	-7.7×10^{-9}
$\frac{\partial F}{\partial \eta}$	−35.0235	−22.3227	−9.22534	−1.86890	−0.08050	−0.00014	1.6×10^{-9}
E[Orbit]	9.276428	7.785777	6.717369	6.192268	6.074761	6.069724	6.069715
$E[V]$	0.799990	0.862781	0.902006	0.917190	0.919840	0.919933	0.919933

Table 2. The optimal value (μ^*, η^*) and the corresponding minimum expected cost.

c	Initial value	Coverage value (μ^*, η^*)	Iteration	Cost*
Case (i) $(\lambda, p, \sigma) = (10, 0.8, 15)$				
1	[25, 15]	[25.13488, 16.43305]	6	3118.635
2	[10, 10]	[12.53093, 8.696281]	6	1968.692
3	[10, 5]	[8.214208, 6.210196]	6	1725.728
4	[5, 5]	[5.999552, 5.046493]	7	1708.284
5	[5, 5]	[4.652035, 4.414643]	7	1779.094
Case (ii) $(\lambda, p, \sigma) = (15, 0.5, 20)$				
1	[30, 20]	[33.17698, 17.35916]	6	3601.021
2	[15, 10]	[16.60255, 9.183037]	5	2210.467
3	[10, 5]	[10.97471, 6.530226]	10	1882.075
4	[6, 6]	[8.099802, 5.265980]	8	1819.241
5	[5, 5]	[6.347280, 4.561196]	7	1861.652

initial value $(c, \mu, \eta) = (2, 10, 10)$, the minimum expected cost per unit time of 1968.692 is achieved at $(\mu^*, \eta^*) = (12.53093, 8.696281)$ using six iterations.

5.2. *Direct search method for optimal c*

After obtaining the joint approximate optimal value (μ^*, η^*) of the continuous variable (μ, η), we use direct search method to obtain the optimal c such that the expected cost function $F(c, \mu^*, \eta^*)$ attains a minimum, say $F(c^*, \mu^*, \eta^*)$. Therefore, the cost minimization problem can be illustrated mathematically as

$$F(c^*, \mu^*, \eta^*) = \min_{1 \le c \le c_U} \left\{ F(c, \mu^*, \eta^*) \right\} \tag{27}$$

The procedure to find the optimal solution is described in the following. A numerical example given in Table 2 is based on (i) $(\lambda, p, \sigma) = (10, 0.8, 15)$ and (ii) $(\lambda, p, \sigma) = (15, 0.5, 20)$.

Algorithm Direct Search Method

Step 1 Set $F^* = M$ which M is a sufficiently large number.

Step 2 For each i from 1 to c_U, set a initial trial solution (μ, η) and use Quasi-Newton method to find the optimal value (μ^*, η^*) and the cost function $F(c, \mu^*, \eta^*)$.

Step 3 If the quasi-Newton method diverges, try another initial trial solution and back to Step 1.

Step 4 If $F(c, \mu^*, \eta^*) < F^*$, set $F^* = F(c, \mu^*, \eta^*)$ and $S^* = (c, \mu^*, \eta^*)$.

It is noted that the optimal value is $(c^*, \mu^*, \eta^*) = (4, 5.999552, 5.046493)$ and the corresponding minimum cost is $F^* = 1708.284$ for Case (i). For Case (ii), $(c^*, \mu^*, \eta^*) = (4, 8.099802, 5.265980)$ and $F^* = 1819.241$ are optimal.

Table 3. The optimal value (c^*, μ^*, η^*) and the minimum expected cost for various values of λ and p.

(λ, p, σ)	(5, 0.2, 10)	(10, 0.2, 10)	(20, 0.2, 10)	(5, 0.8, 10)	(10, 0.8, 10)	(20, 0.8, 10)
c^*	2	3	4	4	4	5
(μ^*, η^*)	[4.965695, 2.123714]	[6.427349, 2.781059]	[9.416220, 3.974561]	[2.997995, 2.998664]	[6.062298, 5.075460]	[9.609657, 7.689420]
$F(c^*, \mu^*, \eta^*)$	901.7296	1245.806	1727.201	1325.523	1716.873	2386.602
E[Orbit]	2.825372	3.199280	4.199710	1.626472	3.125312	4.497047
$E[V]$	0.470873	0.719505	1.006400	1.333927	1.576212	2.080781
(λ, p, σ)	(5, 0.2, 10)	(5, 0.5, 10)	(5, 0.8, 10)	(10, 0.2, 15)	(10, 0.5, 15)	(10, 0.8, 15)
c^*	2	3	4	3	3	4
(μ^*, η^*)	[4.965695, 2.123714]	[3.774111, 2.689427]	[2.997995, 2.998664]	[6.347744,2.767427]	[7.295827, 4.645567]	[5.999552, 5.046493]
$F(c^*, \mu^*, \eta^*)$	901.7296	1116.483	1325.523	1237.045	1511.634	1708.284
E[Orbit]	2.825372	2.122060	1.626472	3.024207	3.662626	2.955528
$E[V]$	0.470873	0.929566	1.333927	0.722693	1.076295	1.585259
(λ, p, σ)	(10, 0.2, 5)	(10, 0.2, 10)	(10, 0.2, 15)	(10, 0.8, 5)	(10, 0.8, 10)	(10, 0.8, 15)
c^*	2	3	3	4	4	4
(μ^*, η^*)	[10.00245, 3.820378]	[6.427349, 2.781059]	[6.347744,2.767427]	[6.232824, 5.154912]	[6.062298, 5.075460]	[5.999552, 5.046493]
$F(c^*, \mu^*, \eta^*)$	1361.503	1245.806	1237.045	1739.966	1716.873	1708.284
E[Orbit]	5.789514	3.199280	3.024207	3.572681	3.125312	2.955528
$E[V]$	0.5235084	0.719505	0.722693	1.551918	1.576212	1.585259

Finally, we perform a sensitivity investigation on the optimal values (c^*, μ^*, η^*). For various values of λ and p, the minimum expected cost $F(c^*, \mu^*, \eta^*)$, the system performance measures L_s and $E[V]$ at the optimum values (c^*, μ^*, η^*) are given in Table 3.

From Table 3, it can be seen that (1) c^* is insensitive to λ or p; (2) μ^* increases as λ increases; and (3) η^* increases as λ or p increases. Moreover, the minimum expected cost increases $F(c^*, \mu^*, \eta^*)$as λ or p increases.

6. Conclusions

An M/M/c retrial queue with Bernoulli vacation (M/M/c/BSV retrial queue) was investigated using the matrix-geometric method. The queueing system was formulated as a QBD process. The sufficient and necessary condition for the stability of the system was discussed. The stationary probability vectors were obtained. We also obtained some system performance in matrix forms. A cost model was constructed to calculate the optimal number of servers, the optimal service rate and vacation rate, so that the cost function is minimized. Two methods were provided to deal with the optimization problem heuristically. We performed a sensitivity analysis of the joint optimal values (c^*, μ^*, η^*) with respect to specific values of λ, p and σ.

Acknowledgements

The authors gratefully acknowledge the constructive comments of editors and the anonymous reviewers.

References

[1] J. Amador and J.R. Artalejo, *On the distribution of the successful and blocked events in the M/M/c retrial queue: A computational approach*, Appl. Math. Comput. 190 (2007), pp. 1612–1626.
[2] J.R. Artalejo, *A classified bibliography of research on retrial queues: Progress in 1990–1999*, Top 7 (1999), pp. 187–211.
[3] J.R. Artalejo, *Accessible bibliography on retrial queues: Progress in 2000–2009*, Math. Comput. Model. 51 (2010), pp. 1071–1081.
[4] J.R. Artalejo, S.R. Chakravarthy, and M.J. Lopez-Herrero, *The busy period and the waiting time analysis of a MAP/M/c queue with finite retrial group*, Stoch. Anal. Appl. 25 (2007), pp. 445–469.
[5] J.R. Artalejo, A. Economou, and M.J. Lopez-Herrero, *Algorithmic approximations for the busy period distribution of the M/M/c retrial queue*, Eur. J. Oper. Res. 176 (2007), pp. 1687–1702.
[6] J.R. Artalejo, A. Economou, and M.J. Lopez-Herrero, *Algorithmic analysis of the maximum queue length in a busy period for the M/M/c retrial queue*, INFORMS J. Comput. 19 (2007), pp. 121–126.
[7] J.R. Artalejo and M. Pozo, *Numerical calculation of the stationary distribution of the main multiserver retrial queue*, Ann. Oper. Res. 116 (2002), pp. 41–56.
[8] L. Breuer, A.N. Dudin, and V.I. Klimenok, *A retrial BMAP/PH/N system*, Queueing Syst. 40 (2002), pp. 433–457.

[9] L. Bright and P.G. Taylor, *Calculating the equilibrium distribution in level dependent quasi-birth-and-death processes*, Stoch. Models 11 (1995), pp. 497–525.

[10] S.R. Chakravarthy and A.N. Dudin, *Multiserver retrial queue with BMAP arriving and group services*, Queueing Syst. Theory Appl. 42 (2002), pp. 5–31.

[11] X. Chao and Y. Zhao, *Analysis of multiple-server queues with station and server vacations*, Eur. J. Oper. Res. 110 (1998), pp. 392–406.

[12] E.K.P. Chong and S.H. Zak, *An Introduction to Optimization*, 2nd ed., John Wiley and Sons, Inc., New York, NY, 2001.

[13] G. Choudhury, *An* $M^{[x]}/G/1$ *queueing system with a setup period and a vacation period*, Queueing Syst. Theory Appl. 36 (2000), pp. 23–38.

[14] G. Choudhury, *A batch arrival queue with a vacation time under single vacation policy*, Comput. Oper. Res. 29 (2002), pp. 1941–1955.

[15] G. Choudhury, *A two phase batch arrival retrial queueing system with Bernoulli vacation schedule*, Appl. Math. Comput. 188 (2007), pp. 1455–1466.

[16] G. Choudhury, *Steady state analysis of an* $M/G/1$ *queue with linear retrial policy and two phase service under Bernoulli vacation schedule*, Appl. Math. Model. 32 (2008), pp. 2480–2489.

[17] G. Choudhury and K. Deka, *An* $M^{[x]}/G/1$ *unreliable retrial queue with two phases of service and Bernoulli admission mechanism*, Appl. Math. Comput. 215 (2009), pp. 936–949.

[18] G. Choudhury and K.C. Madan, *A two phase batch arrival queueing system with a vacation time under Bernoulli schedule*, Appl. Math. Comput. 149 (2004), pp. 337–349.

[19] G. Choudhury and K.C. Madan, *A two-stage batch arrival queueing system with a modified Bernoulli schedule vacation under N-policy*, Math. Comput. Model. 42 (2005), pp. 71–85.

[20] G. Choudhury, L. Tadj, and M. Paul, *Steady state analysis of an* $M^{[x]}/G/1$ *queue with two phase service and Bernoulli vacation schedule under multiple vacation policy*, Appl. Math. Model. 31 (2007), pp. 1079–1091.

[21] B.T. Doshi, *Queueing systems with vacations-a survey*, Queueing Syst. 1 (1986), pp. 29–66.

[22] G.I. Falin, *Calculations of probability characteristics of a multiline system with repeated calls*, Moscow Univ. Comput. Math. Cybernet. 1 (1983), pp. 43–49.

[23] G.I. Falin and J.G.C. Templeton, *Retrial Queues*, Chapman and Hall, London, 1997.

[24] A. Gomez-Corral, *A bibliographical guide to the analysis of retrial queues through matrix analytic techniques*, Ann. Oper. Res. 141 (2006), pp. 177–207.

[25] J.C. Ke and F.M. Chang, $M^{[x]}/(G_1, G_2)/1$ *retrial queue under Bernoulli vacation schedules with general repeated attempts and starting failures*, Appl. Math. Model., 33 (2009), pp. 3186–3196.

[26] J.C. Ke and Y.K. Chu, *A modified vacation model* $M^{[x]}/G/1$ *system*, Appl. Stoch. Bus. Ind. 22 (2006), pp. 1–16.

[27] G.V. Krishna Reddy, R. Nadarajan, and R. Arumuganathan, *Analysis of a bulk queue with N-policy multiple vacations and setup times*, Comput. Oper. Res. 25 (1998), pp. 957–967.

[28] V.G. Kulkarni and H.M. Liang, *Retrial queues revisited*, in *Frontiers in Queueing: Models and Applications in Science and Engineering*, J.H. Dshalalow, ed., CRC Press, Inc., Boca Raton, FL, 1997, pp. 19–34.

[29] C. Langaris and I. Dimitriou, *A queueing system with n-phases of service and (n-1)-types of retrial customers*, Eur. J. Oper. Res. 205 (2010), pp. 638–649.

[30] G. Latouche and V. Ramaswami, *Introduction to Matrix Analytic Methods in Stochastic Modeling* (ASA-SIAM Series on Statistics and Applied Probability), Elsevier Science, Amsterdam, Netherlands, 1999.

[31] H.W. Lee, S.S. Lee, J.O. Park, and K.C. Chae, *Analysis of $M^{[x]}/G/1$ queue with N policy and multiple vacations*, J. Appl. Prob. 31 (1994), pp. 467–496.
[32] S.S. Lee, H.W. Lee, S.H. Yoon, and K.C. Chae, *Batch arrival queue with N policy and single vacation*, Comput. Oper. Res., 22 (1995), pp. 173–189.
[33] Y. Levy and Y.U. Yechiali, *An M/M/c queue with servers' vacations*, INFOR 14 (1976), pp. 153–163.
[34] K.C. Madan, W. Abu-Dayyeh, and F. Taiyyan, *A two server queue with Bernoulli schedules and a single vacation policy*, Appl. Math. Comput. 145 (2003), pp. 59–71.
[35] M.F. Neuts, *Matrix Geometric Solutions in Stochastic Models: An Algorithmic Approach*, The John Hopkins University Press, Baltimore, MD, 1981.
[36] M.F. Neuts and B.M. Rao, *Numerical investigation of a multiserver retrial model*, Queueing Syst. Theory Appl. 7 (1990), pp. 169–190.
[37] M. Shomrony and U. Yechiali, *Burst arrival queues with server vacations and random timers*, Math. Meth. Oper. Res. 53 (2001), pp. 117–146.
[38] S.N. Stepanov, *Markov models with retrials: The calculation of stationary performance measures based on the concept of truncation*, Math. Comput. Model. 30 (1999), pp. 207–228.
[39] L. Tadj, G. Choudhury, and C. Tadj, *A quorum queueing system with a random setup time under N-policy and with Bernoulli vacation schedule*, Qual. Technol. Quant. Manag. 3 (2006), pp. 145–160.
[40] L. Tadj, G. Choudhury, and C. Tadj, *A bulk quorum queueing system with a random setup time under N-policy and with Bernoulli vacation schedule*, Stoch.: Int. J. Probab. Stoch. Process. 78 (2006), pp. 1–11.
[41] H. Takagi, *Queueing Analysis: A Foundation of Performance Evaluation*, Vol. I, *Vacation and Priority Systems, Part I*, North-Holland, Amsterdam, 1991.
[42] N. Tian, Q. Li, and J. Cao, *Conditional stochastic decompositions in the M/M/c queue with server vacations*, Stoch. Models, 14 (1999), pp. 367–377.
[43] X. Xu and Z.G. Zhang, *Analysis of multiple-server queue with a single vacation (e,d)-policy*, Perform. Eval. 63 (2006), pp. 825–838.
[44] T. Yang and J.G.C. Templeton, *A survey on retrial queues*, Queueing Syst. Theory Appl. 2 (1987), pp. 201–233.
[45] Z.G. Zhang and N. Tian, *Analysis of queueing systems with synchronous single vacation for some servers*, Queueing Syst. Theory Appl., 45 (2003), pp. 161–175.
[46] Z.G. Zhang and N. Tian, *Analysis on queueing systems with synchronous vacations of partial servers*, Perform. Eval. 52 (2003), pp. 269–282.

Appendix

For instance, for $c = 3$, the sub-matrices of $\mathbf{B}$ are

$$\mathbf{b}^0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda \end{bmatrix}, \quad \mathbf{b}^1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \lambda \end{bmatrix}, \quad \mathbf{b}^2 = \begin{bmatrix} 0 & 0 \\ 0 & \lambda \end{bmatrix}, \quad \mathbf{b}^3 = \lambda.$$

The sub-matrices of $\mathbf{C}_1$, $\mathbf{C}_2$ and $\mathbf{C}_3$ are

$$\mathbf{c}_1^0=\begin{bmatrix}0&\sigma&0&0\\0&0&\sigma&0\\0&0&0&\sigma\\0&0&0&0\end{bmatrix},\quad \mathbf{c}_1^1=\begin{bmatrix}0&\sigma&0\\0&0&\sigma\\0&0&0\end{bmatrix},\quad \mathbf{c}_1^2=\begin{bmatrix}0&\sigma\\0&0\end{bmatrix},\quad \mathbf{c}_1^3=0.$$

$$\mathbf{c}_2^0=\begin{bmatrix}0&2\sigma&0&0\\0&0&2\sigma&0\\0&0&0&2\sigma\\0&0&0&0\end{bmatrix},\quad \mathbf{c}_2^1=\begin{bmatrix}0&2\sigma&0\\0&0&2\sigma\\0&0&0\end{bmatrix},\quad \mathbf{c}_2^2=\begin{bmatrix}0&2\sigma\\0&0\end{bmatrix},\quad \mathbf{c}_2^3=0.$$

$$\mathbf{c}_3^0=\begin{bmatrix}0&3\sigma&0&0\\0&0&3\sigma&0\\0&0&0&3\sigma\\0&0&0&0\end{bmatrix},\quad \mathbf{c}_3^1=\begin{bmatrix}0&3\sigma&0\\0&0&3\sigma\\0&0&0\end{bmatrix},\quad \mathbf{c}_3^2=\begin{bmatrix}0&3\sigma\\0&0\end{bmatrix},\quad \mathbf{c}_3^3=0.$$

The diagonal sub-matrices of $\mathbf{A}_j$, where $j=0,1,2,3$ are described as follows.
For $\mathbf{A}_0$:

$$\mathbf{Y}_0^0=\begin{bmatrix}-\lambda&\lambda&&\\(1-p)\mu&-(\lambda+\mu)&\lambda&\\&2(1-p)\mu&-(\lambda+2\mu)&\lambda\\&&3(1-p)\mu&-(\lambda+3\mu)\end{bmatrix},$$

$$\mathbf{Y}_0^1=\begin{bmatrix}-(\lambda+\eta)&\lambda&\\(1-p)\mu&-(\lambda+\mu+\eta)&\lambda\\&2(1-p)\mu&-(\lambda+2\mu+\eta)\end{bmatrix},$$

$$\mathbf{Y}_0^2=\begin{bmatrix}-(\lambda+2\eta)&\lambda\\(1-p)\mu&-(\lambda+\mu+2\eta)\end{bmatrix},\quad \mathbf{Y}_0^3=-(\lambda+3\eta).$$

For $\mathbf{A}_1$:

$$\mathbf{Y}_1^0=\begin{bmatrix}-(\lambda+\sigma)&\lambda&&\\(1-p)\mu&-(\lambda+\mu+\sigma)&\lambda&\\&2(1-p)\mu&-(\lambda+2\mu+\sigma)&\lambda\\&&3(1-p)\mu&-(\lambda+3\mu)\end{bmatrix},$$

$$\mathbf{Y}_1^1=\begin{bmatrix}-(\lambda+\eta+\sigma)&\lambda&\\(1-p)\mu&-(\lambda+\mu+\eta+\sigma)&\lambda\\&2(1-p)\mu&-(\lambda+2\mu+\eta)\end{bmatrix},$$

$$\mathbf{Y}_1^2=\begin{bmatrix}-(\lambda+2\eta+\sigma)&\lambda\\(1-p)\mu&-(\lambda+\mu+2\eta)\end{bmatrix},\quad \mathbf{Y}_1^3=-(\lambda+3\eta)$$

For $\mathbf{A}_2$:

$$\mathbf{Y}_2^0 = \begin{bmatrix} -(\lambda+2\sigma) & \lambda & & \\ (1-p)\mu & -(\lambda+\mu+2\sigma) & \lambda & \\ & 2(1-p)\mu & -(\lambda+2\mu+2\sigma) & \lambda \\ & & 3(1-p)\mu & -(\lambda+3\mu) \end{bmatrix},$$

$$\mathbf{Y}_1^1 = \begin{bmatrix} -(\lambda+\eta+\sigma) & \lambda & \\ (1-p)\mu & -(\lambda+\mu+\eta+\sigma) & \lambda \\ & 2(1-p)\mu & -(\lambda+2\mu+\eta) \end{bmatrix},$$

$$\mathbf{Y}_2^2 = \begin{bmatrix} -(\lambda+2\eta+2\sigma) & \lambda \\ (1-p)\mu & -(\lambda+\mu+2\eta) \end{bmatrix}, \quad \mathbf{Y}_2^3 = -(\lambda+3\eta).$$

For $\mathbf{A}_3$:

$$\mathbf{Y}_3^0 = \begin{bmatrix} -(\lambda+3\sigma) & \lambda & & \\ (1-p)\mu & -(\lambda+\mu+3\sigma) & \lambda & \\ & 2(1-p)\mu & -(\lambda+2\mu+3\sigma) & \lambda \\ & & 3(1-p)\mu & -(\lambda+3\mu) \end{bmatrix},$$

$$\mathbf{Y}_3^1 = \begin{bmatrix} -(\lambda+\eta+3\sigma) & \lambda & \\ (1-p)\mu & -(\lambda+\mu+\eta+3\sigma) & \lambda \\ & 2(1-p)\mu & -(\lambda+2\mu+\eta) \end{bmatrix},$$

$$\mathbf{Y}_3^2 = \begin{bmatrix} -(\lambda+2\eta+3\sigma) & \lambda \\ (1-p)\mu & -(\lambda+\mu+2\eta) \end{bmatrix}, \quad \mathbf{Y}_3^3 = -(\lambda+3\eta).$$

For $\mathbf{A}_0$, $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$, the first super-diagonal sub-matrices and the first sub-diagonal sub-matrices are given by

$$\mathbf{X}^0 = \begin{bmatrix} 0 & 0 & 0 \\ p\mu & 0 & 0 \\ 0 & 2p\mu & 0 \\ 0 & 0 & 3p\mu \end{bmatrix}, \quad \mathbf{X}^1 = \begin{bmatrix} 0 & 0 \\ p\mu & 0 \\ 0 & 2p\mu \end{bmatrix}, \quad \mathbf{X}^2 = \begin{bmatrix} 0 \\ p\mu \end{bmatrix},$$

and

$$\mathbf{Z}^1 = \begin{bmatrix} \eta & 0 & 0 & 0 \\ 0 & \eta & 0 & 0 \\ 0 & 0 & \eta & 0 \end{bmatrix}, \quad \mathbf{Z}^2 = \begin{bmatrix} 2\eta & 0 & 0 \\ 0 & 2\eta & 0 \end{bmatrix}, \quad \mathbf{Z}^3 = \begin{bmatrix} 3\eta & 0 \end{bmatrix},$$

respectively.