Computers & Industrial Engineering 65 (2013) 216–225

Contents lists available at SciVerse ScienceDirect

Computers & Industrial Engineering

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

Analysis of an infinite multi-server queue with an optional service

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

[a] Department of Applied Statistics, National Taichung University of Science and Technology, Taichung, Taiwan, ROC

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

A R T I C L E   I N F O

Article history:
Received 19 June 2012
Received in revised form 18 February 2013
Accepted 22 February 2013
Available online 13 March 2013

Keywords:
First essential channel
Linear progressing algorithm
Particle Swarm Optimization
Quasi-Newton method
Second optional channel

A B S T R A C T

This paper deals with an infinite-capacity multi-server queueing system with a *second optional service* (SOS) channel. The inter-arrival times of arriving customers, the service times of the first essential service (FES) and the SOS channel are all exponentially distributed. A customer may leave the system after the FES channel with a probability $(1-\theta)$, or the completion of the FES may immediately require a SOS with a probability θ $(0 \leqslant \theta \leqslant 1)$. The formulae for computing the rate matrix and stationary probabilities are derived by means of a matrix analytical approach. A cost model is developed to simultaneously determine the optimal values of the number of servers and the two service rates at the minimal total expected cost per unit time. Quasi-Newton method and *Particle Swarm Optimization* (PSO) method are employed to deal with the optimization problem. Under optimal operating conditions, numerical results are provided from which several system performance measures are calculated based on the assumed numerical values of the system parameters.

1. Introduction

In day to day life, one encounters numerous examples of queueing models through which all arriving customers need an essential service but only some require an additional optional service. In this paper, the *quasi-birth–death* (QBD) process and matrix analytic methods are used to analyze an M/M/R queue with a second optional service channel. An algorithm is developed to calculate the rate matrix and the stationary probabilities of the QBD process. A cost model is also constructed to search the optimal parameters at a minimum cost.

A possible application of our model is in a manufacturing system for a pump based on the work of Yang, Wang, and Kuo (2011). Consider a pump manufacturing industry that manufactures different kinds of pumps which require shafts of various dimensions. The arrival of shafts from the turning center to the CNC (computer numerical control) copy turning center follows a random process, in which the center owns multiple CNC machines. The mechanics set up the template in these CNC machines to perform the copy turning process for shafts, i.e., the first essential service. The good quality items are kept in storage and are sold. Some of the served (processed) shafts are defective. The defective ones need to be reworked (re-served) to meet the required specifications. In this scenario, the mechanics (including the CNC-machines) and the re-served action of defective items correspond to the channels and the second optional service, respectively.

It is assumed that customers arrive according to a Poisson process with parameter λ. Customers arriving at the system form a single waiting line and are served in the order of their arrival, that is, first-come-first-served. There are R channels (servers) that provide the *first essential service* (FES) as well as the *second optional service* (SOS) to arriving customers. The FES is needed by all arriving customers. The service times of FES and the SOS are exponentially distributed with means $1/\mu_1$ and $1/\mu_2$, respectively. As soon as the FES of a customer is completed, they may leave the system with a probability $(1-\theta)$ or, opt for the SOS provided by the same server with a probability θ $(0 \leqslant \theta \leqslant 1)$. Each channel can serve only one customer at a time and it also provides only either FES or SOS at one time. Customers who, upon entry into the channel facility, find that all channels are busy are required to wait in the queue until a channel becomes available. The various stochastic (arrival or service) processes involved in the system are independent of each other.

Analytic steady-state solutions of an M/M/R queue with a second optional service channel have not been found. A pioneering work in this queueing situation was proposed by Madan (2000), who first introduced the concept of the SOS. Madan (2000) studied an M/G/1 queue with SOS using the supplementary variable technique in which he considered a general service time distribution for the FES and an exponential service time distribution for the SOS. He also cited some important applications of this model in many real-life situations. Medhi (2002) derived the transient

* Corresponding author. Address: Department of Applied Statistics, National Taichung University of Science and Technology, No. 129, Sec. 3, Sanmin Rd., Taichung 404, Taiwan, ROC. Tel.: +886 422196077; fax: +886 422196331.

E-mail addresses: jauchuan@nutc.edu.tw (J.-C. Ke), duckboy614583@gmail.com (C.-H. Wu), wlpearn@mail.nctu.edu.tw (W.L. Pearn).

0360-8352/$ - see front matter
http://dx.doi.org/10.1016/j.cie.2013.02.017

solution and steady-state solution for the ordinary M/G/1 queue with the SOS using the same technique. Medhi's M/G/1 model was also investigated by Al-Jararha and Madan (2003), in which they developed the time-dependent probability generating functions involved in Laplace transforms and further obtained the corresponding steady-state results. Choudhury and Madan (2005) and Choudhury and Paul (2006) studied the queue size distribution at a random epoch as well as at a departure epoch for an $M^{[x]}/G/1$ queueing system with a second optional channel and different considerations under *N*-policy. They also derived a simple procedure to obtain the optimal stationary policy under a suitable linear cost structure. Tadj, Choudhury, and Tadj (2006a, 2006b) investigated some bulk service queueing systems under *N* policy. The reliability measures were examined by Wang (2004) regarding the ordinary M/G/1 queue with channel breakdowns and SOS. Recently, Ke (2008) investigated a batch arrival $M^{[x]}/G/1$ queueing system with *J* optional services. He derived the steady-state results, including the system size distribution at a random epoch and at a departure epoch, the distributions of idle and busy periods, and the waiting time distribution in the queue. Choudhury and Tadj (2009) generalized this type of model by introducing the concept of a server breakdown and a delay-repair-period. Recently, Choudhury and Tadj (2011) studied the optimal management of an $M^x/G/1$ unreliable server queue with optional service under a Bernoulli vacation schedule. Existing work with optional service mentioned earlier, mainly focused on *single-server queue*. The main reason for this is such that the steady-state probability vector of a multi-server queue with SOS is not easily derived. This motivates us to investigate an infinite capacity M/M/*R* queueing system with a second optional service channel which includes parameter optimization at a minimum cost.

The paper is organized as follows. In Section 2, the steady-state equations and the *quasi-birth–death* (QBD) model of an infinite capacity M/M/*R* queue with SOS channel are set up. The matrix-geometric property (matrix analytic method) is used to calculate the rate matrix in Section 3. In Section 4, we develop an efficient algorithm to obtain the stationary probabilities using a matrix-geometric and a recursive method. Some system performance measures are derived in Section 5. In Section 6, a cost model is developed to determine the optimal values of the number of channels and the two different service rates used to minimize the total expected cost per unit time. We use the Quasi-Newton method and the direct search method to implement the optimization tasks. The *Particle Swarm Optimization* (PSO) method is compared with the Quasi-Newton method in establishing the heuristic solution. Some numerical examples are provided to illustrate these two optimization methods. In Section 7, we offer our conclusions.

2. Markov chain model

For an infinite capacity M/M/*R* queueing system with *second optional service* (SOS) channel, the states of the system are described by the pair (i,j), $i = 0,1,2,\ldots$ and $j = 0,1,2,\ldots,R$, where *i* denotes the number of customers in the FES channel (including customers waiting in the queue) and *j* denotes the number of customers in the SOS channel. If $(i+j) \leqslant R$, i.e., there are available servers, the customers upon arrival to the server will get service immediately. If $(i+j) > R$, i.e., all servers are busy, the newly arriving customer must wait in the queue until a server becomes available. We define the following notations in steady-state:

$P_{i,j} \equiv$ probability that there are *i* customers in the FES channel and there are *j* customers in the SOS channel, where $i = 0,1,2,\ldots$ and $j = 0,1,2,\ldots,R$.

Referring to the steady transition-rate diagram shown in Fig. 1 and using the birth-and-death process, the steady-state equations governing the M/M/*R* queueing system are:

(i) $j = 0$

$$\lambda P_{0,0} = (1-\theta)\mu_1 P_{1,0} + \mu_2 P_{0,1}, \quad (1)$$

$$(\lambda + i\mu_1)P_{i,0} = \lambda P_{i-1,0} + (i+1)(1-\theta)\mu_1 P_{i+1,0} + \mu_2 P_{i,1}, \quad 1 \leqslant i \leqslant R-1, \quad (2)$$

$$(\lambda + R\mu_1)P_{i,0} = \lambda P_{i-1,0} + R(1-\theta)\mu_1 P_{i+1,0} + \mu_2 P_{i,1}, \quad R \leqslant i. \quad (3)$$

(ii) $1 \leqslant j \leqslant R-1$

$$(\lambda + j\mu_2)P_{0,j} = \theta\mu_1 P_{1,j-1} + (1-\theta)\mu_1 P_{1,j} + (j+1)\mu_2 P_{0,j+1}, \quad (4)$$

$$(\lambda + i\mu_1 + j\mu_2)P_{i,j} = \lambda P_{i-1,j} + (i+1)\theta\mu_1 P_{i+1,j-1} + (i+1)(1-\theta)\mu_1 P_{i+1,j} + (j+1)\mu_2 P_{i,j+1}, \quad 1 \leqslant i \leqslant R-j-1, \quad (5)$$

$$[\lambda + (R-j)\mu_1 + j\mu_2]P_{i,j} = \lambda P_{i-1,j} + (R+1-j)\theta\mu_1 P_{i+1,j-1} + (R-j)(1-\theta)\mu_1 P_{i+1,j} + (j+1)\mu_2 P_{i,j+1}, \quad R-j \leqslant i. \quad (6)$$

(iii) $j = R$

$$(\lambda + R\mu_2)P_{0,R} = \theta\mu_1 P_{1,R-1}, \quad (7)$$

$$(\lambda + R\mu_2)P_{i,R} = \lambda P_{i-1,R} + \theta\mu_1 P_{i+1,R-1}, \quad 1 \leqslant i. \quad (8)$$

There is no way of solving Eq. (1)–(8) in a recursive manner to develop the explicit expressions for the steady-state probabilities $P_{i,j}$, where $i = 0,1,2,\ldots$ and $j = 0,1,2,\ldots,R$. Alternatively, the infinitesimal generator $\mathbf{Q}$ of the *quasi birth-and-death* (QBD) process describing the M/M/*R* queueing system with SOS channel is of the block-tridiagonal form (see Neuts, 1981):

$$\mathbf{Q} = \begin{array}{c} \ell(0) \\ \ell(1) \\ \ell(2) \\ \vdots \\ \ell(R-1) \\ \ell(R) \\ \ell(R+1) \\ \vdots \end{array} \overset{\begin{array}{cccccccccc} \ell(0) & \ell(1) & \ell(2) & \cdots & \cdots & \cdots & \ell(R-2) & \ell(R-1) & \ell(R) & \ell(R+1) & \cdots \end{array}}{\begin{bmatrix} \mathbf{A}_0 & \mathbf{B} & & & & & & \\ \mathbf{C}_1 & \mathbf{A}_1 & \mathbf{B} & & & & & \\ & \mathbf{C}_2 & \mathbf{A}_2 & \mathbf{B} & & & & \\ & & \ddots & \ddots & \ddots & & & \\ & & & \mathbf{C}_{R-1} & \mathbf{A}_{R-1} & \mathbf{B} & & \\ & & & & \mathbf{C}_R & \mathbf{A}_R & \mathbf{B} & \\ & & & & & \mathbf{C}_R & \mathbf{A}_R & \mathbf{B} \\ & & & & & & \ddots & \ddots & \ddots \end{bmatrix}} \quad (9)$$

Each entry of the matrix $\mathbf{Q}$ is a square matrix of order $R+1$ listed as follows:

$$\mathbf{B} = \lambda\mathbf{I}, \quad (10)$$

$$\mathbf{A}_i = \begin{bmatrix} a_{i,0} & & & & \\ \mu_2 & a_{i,1} & & & \\ & 2\mu_2 & a_{i,2} & & \\ & & \ddots & \ddots & \\ & & & R\mu_2 & a_{i,R} \end{bmatrix}, \quad i = 0,\ldots,R \quad (11)$$

$$\mathbf{C}_i = \begin{bmatrix} c_{i,0} & d_{i,0} & & & \\ & c_{i,1} & d_{i,1} & & \\ & & c_{i,2} & d_{i,2} & \\ & & \ddots & \ddots & \\ & & & c_{i,R-1} & d_i \\ & & & & 0 \end{bmatrix}, \quad i = 1,\ldots,R \quad (12)$$

where $\mathbf{I}$ is the identity matrix of order $R+1$, and

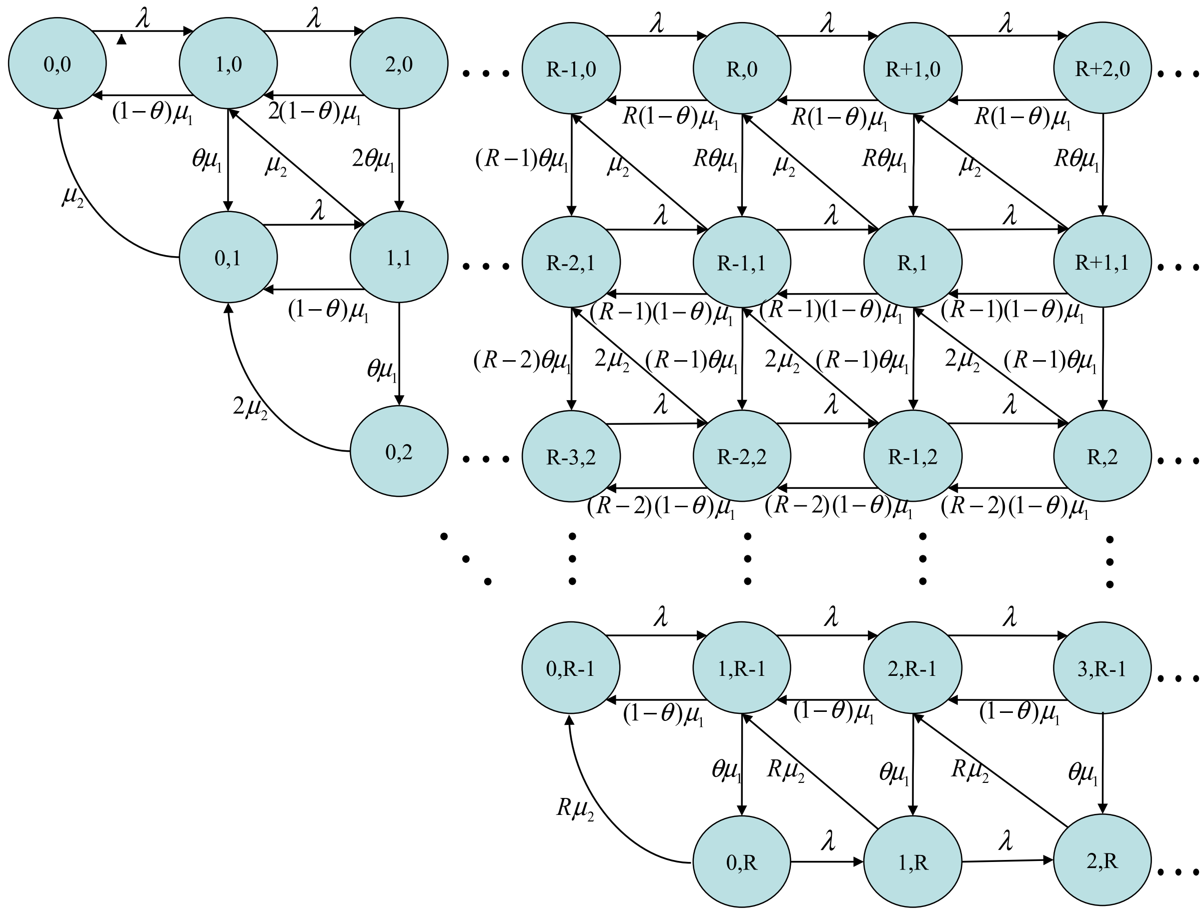

Fig. 1. Steady-transition-rate diagram for an M/M/R queueing system with second optional service channel.

$$a_{i,j} = \begin{cases} -(\lambda + i\mu_1 + j\mu_2), & 1 \leqslant i+j \leqslant R, \\ -[\lambda + (R-j)\mu_1 + j\mu_2], & i+j > R, \end{cases} \tag{13}$$

$$c_{i,j} = \begin{cases} i(1-\theta)\mu_1, & 1 \leqslant i+j \leqslant R, \\ (R-j)(1-\theta)\mu_1, & i+j > R, \end{cases} \tag{14}$$

$$d_{i,j} = \begin{cases} i\theta\mu_1, & 1 \leqslant i+j \leqslant R, \\ (R-j)\theta\mu_1, & i+j > R. \end{cases} \tag{15}$$

Let

$$\mathbf{F} = \mathbf{C}_R + \mathbf{A}_R + \mathbf{B}, \tag{16}$$

It is clear that **F** is an irreducible generator (see Neuts, 1981). Let $\mathbf{X} = [x_0, x_1, \ldots, x_R]$, a $1 \times (R+1)$ vector, is the invariant vector of **F**. Then, **F** satisfies the linear equations:

$$\mathbf{XF} = \mathbf{0} \quad \text{and} \quad \mathbf{Xe} = 1, \tag{17}$$

where **0** and **e** are column vectors with dimensions $R+1$ and all elements are equal to zero and one, respectively. Solving the linear Eqs. (16) and (17), **X** could be obtained easily. Next, the stability condition could be established by Theorem 3.1.1 of Neuts (1981), the standard drift condition is:

$$\mathbf{XBe} < \mathbf{XC_R e}, \tag{18}$$

which is the necessary and sufficient condition for stability of the QBD-**Q** process. First, we solve $\mathbf{xF} = \mathbf{0}$ $x = [x_0, x_1, \ldots, x_R]$ and write following $(R+1)$ equations:

$$R\theta\mu_1 x_0 = x_1\mu_2, \tag{19a}$$

$$-(R-i+1)\theta\mu_1 x_{i-1} + [(R-i)\theta\mu_1 + i\mu_2]x_i - (i+1)\mu_2 x_{i+1} = 0, \quad 1 \leqslant i \leqslant R-1, \tag{19b}$$

$$\theta\mu_1 x_{R-1} = R\mu_2 x_R. \tag{19c}$$

Eq. (19a) implies that $x_1 = \frac{c\theta\mu_1}{\mu_2}x_0$. Solving Eqs. (19b) and (19c) recursively, we get:

$$x_{i+1} = \frac{(R-i)\theta\mu_1}{(i+1)\mu_2}x_i, \quad i = 1, \ldots, R-1. \tag{19d}$$

Finally, we have:

$$x_{i+1} = \frac{(c-i)\theta\mu_1}{(i+1)\mu_2}x_i = \frac{(c-i)(c-i-1)}{(i+1)i}\left(\frac{\theta\mu_1}{\mu_2}\right)^2 x_{i-1} = \cdots = \binom{c}{i+1}\left(\frac{\theta\mu_1}{\mu_2}\right)^{i+1} x_0, \quad i = 1, \ldots, R-1. \tag{19e}$$

Using the normalization condition $x_0 + x_1 + \cdots + x_{R-1} + x_R = 1$, x_0 can determine x_0 as:

$$x_0 = \left[\sum_{i=0}^{R}\binom{R}{i}\left(\frac{\theta\mu_1}{\mu_2}\right)^i\right]^{-1} = \left(1 + \frac{\theta\mu_1}{\mu_2}\right)^{-R}. \tag{19f}$$

Substituting **B** and $\mathbf{C}_R$ into Eq. (18) and using (19f), we have:

$$\mu_1(R - L_2) > \lambda, \tag{20a}$$

which is equivalent to:

$$\frac{\lambda}{\mu_1(R-L_2)} < 1, \tag{20b}$$

where

$$L_2 = x_1 + 2x_2 + \cdots + Rx_R = \sum_{i=1}^{R} ix_i = \sum_{i=1}^{R} i\binom{R}{i}\left(\frac{\theta\mu_1}{\mu_2}\right)^i x_0 = \frac{R\theta\mu_1}{\mu_2}x_0 = \frac{R\theta\mu_1}{\mu_2}\left(1+\frac{\theta\mu_1}{\mu_2}\right)^{-R} \quad (21)$$

denotes the expected number of customers in the SOS channel. It is noted that if $\theta = 0$ or $\mu_2 \to \infty$ (i.e., $L_2 = 0$), Eq. (20) could be reduced to the stability condition for the ordinary M/M/R queueing system without the SOS channel.

3. The matrix geometric property

Because the infinitesimal generator given in (9) is a special case of Eq. (5.2.1) of Neuts (1981), we know that the QBD is periodic and a positive recurrent. Denote by **P** the stationary probability vector of **Q**. This implies that the unique solution of the system $\mathbf{PQ} = \mathbf{0}$ under stability conditions. We partition the vector **P** as $\mathbf{P} = [\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_{R-1}, \mathbf{P}_R, \mathbf{P}_{R+1}, \ldots]$, where $P_i = [P_{i,0}, P_{i,1}, \ldots, P_{i,R}]$ is a row vector with a dimension of $R + 1$. Our aim is to obtain the steady-state vector **P** by means of the matrix analytic method and normalization. By applying the matrix-geometric method, the steady-state probabilities $[\mathbf{P}_{R+1}, \mathbf{P}_{R+2}, \mathbf{P}_{R+3}, \ldots]$ can be obtained as $\mathbf{P}_i = \mathbf{P}_R\mathbf{T}^{i-R}$, for $i \geqslant R + 1$, where **T** is the minimal nonnegative solution to the matrix quadratic equation:

$$\mathbf{T}^2\mathbf{C}_R + \mathbf{T}\mathbf{A}_R + \mathbf{B} = \mathbf{0}. \quad (22)$$

The matrix **T** is a very important matrix needed in the evaluation of the performance measures of a QBD process. It is known as the rate matrix of the Markov chain **Q**. Developing a closed-form solution for the rate matrix by taking the nonlinear Eq. (22) is very difficult because the matrix structure of $\mathbf{A}_R$, **B** and $\mathbf{C}_R$ is not consistent. We develop some matrix analytic properties to approximate the rate matrix **T**.

Let us decompose the level space into two groups as $\ell(J) = \{\ell(0), \ell(1), \ldots, \ell(R)\}$ and $\ell(K) = \{\ell(R + 1), \ell(R + 2), \ldots\}$. The QBD model of this paper is partially level-dependent up to a certain level (group $\ell(J)$) and thereafter becomes a infinite level-independent (group $\ell(K)$). An infinite level-independent QBD has a matrix-geometric form which can be solved from the matrix quadratic equation (Latouche & Ramaswami, 1999). The level-independent structure in our paper can be solved by Cramer's rule. Thus, we can use the finite level-dependent algorithm first and then the algorithm of infinite level-independent QBDs to derive the state probabilities.

Note from the matrix (9) that starting from level $\ell(R)$ the matrices $\mathbf{C}_{R-1}$ and $\mathbf{A}_{R-1}$ change to $\mathbf{C}_R$ and $\mathbf{A}_R$, respectively. This implies that the process holds an infinite level-independent QBD with group $\ell(K)$. First, we reduce the QBD-**Q** into a finite level-dependent QBD-$\mathbf{Q}^*$ as:

$$\mathbf{Q}^* = \begin{matrix} & \ell(0) & \ell(1) & \cdots & \cdots & \ell(R) & \ell(R+1) \\ \ell(0) & \mathbf{A}_0 & \mathbf{B} & 0 & \cdots & 0 & 0 \\ \ell(1) & \mathbf{C}_1 & \mathbf{A}_1 & \mathbf{B} & \cdots & 0 & 0 \\ \ell(2) & 0 & \mathbf{C}_2 & \mathbf{A}_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \ell(R) & 0 & 0 & 0 & \cdots & \mathbf{A}_R & \mathbf{B} \\ \ell(R+1) & 0 & 0 & 0 & \cdots & \mathbf{C}_R & \mathbf{H} \end{matrix}. \quad (23)$$

From Neuts (1981), we know that the QBD-$\mathbf{Q}^*$ is a periodic and irreducible infinitesimal generator with finite dimensions. The matrix **H** in (23) represents the transitions between the states belonging to the imaginary level group $\ell(K)$. The boundary steady-state probability vector $\mathbf{P}_{R+1}$ based on $\ell(R + 1)$ is given by solving the following equations:

$$\mathbf{P}_R\mathbf{B} + \mathbf{P}_{R+1}\mathbf{H} = \mathbf{P}_{R+1}, \quad (\text{from QBD-}\mathbf{Q}^*) \quad (24a)$$

$$\mathbf{P}_R\mathbf{B} + \mathbf{P}_{R+1}\mathbf{A}_R + \mathbf{P}_{R+2}\mathbf{C}_R = \mathbf{P}_{R+1}. \quad (\text{from QBD-}\mathbf{Q}) \quad (24b)$$

Solving Eqs. (24a) and (24b), we obtain:

$$\mathbf{H} = \mathbf{A}_R + \mathbf{T}\mathbf{C}_R. \quad (25)$$

Substituting (25) into Eq. (23) yields the following system of linear equations:

$$\Lambda\mathbf{Q}^* = [\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_{R+1}] \begin{bmatrix} \mathbf{A}_0 & \mathbf{B} & & & & \\ \mathbf{C}_1 & \mathbf{A}_1 & \mathbf{B} & & & \\ & \mathbf{C}_2 & \mathbf{A}_2 & & & \\ & & \ddots & \ddots & \ddots & \\ & & & & \mathbf{A}_R & \mathbf{B} \\ & & & & \mathbf{C}_R & \mathbf{A}_R + \mathbf{T}\mathbf{C}_R \end{bmatrix} = \mathbf{0}, \quad (26)$$

where $\mathbf{P}_i = [\mathbf{P}_{i,0}, \mathbf{P}_{i,1}, \mathbf{P}_{i,2}, \ldots, \mathbf{P}_{i,R}]$, $i = 0, 1, 2, \ldots, R + 1$.

By the arguments of Latouche and Ramaswami (1999), there exists an infinitesimal generator **U** of the transient continuous-time Markov chain that is restricted to level $\ell(R + 2)$ before it reaches $\ell(R + 1)$ from group level $\ell(J)$. It is given by:

$$\mathbf{U} = \mathbf{A}_R + \mathbf{B}(-\mathbf{U})^{-1}\mathbf{C}_R = \mathbf{A}_R + \mathbf{BG} = \mathbf{A}_R + \mathbf{T}\mathbf{C}_R,$$

where

$$\mathbf{T} = \mathbf{B}(-\mathbf{U})^{-1},$$
$$\mathbf{G} = (-\mathbf{U})^{-1}\mathbf{C}_R,$$
$$\mathbf{H} = \mathbf{U}.$$

Based on the analysis above, we summarize an algorithm to obtain the approximation for the rate matrix **T** (see Latouche & Ramaswami, 1999).

Algorithm 1. Linear Progression Algorithm

INPUT **B**, $\mathbf{A}_R$, $\mathbf{C}_R$, $\mathbf{e} = [1, \ldots, 1]^{\mathrm{T}}$, **I** is the identity matrix, and the tolerance δ.
OUTPUT approximate solution **T**
Step 1 $\mathbf{G} = (-\mathbf{A}_R)^{-1}\mathbf{C}_R$
Step 2 while $\|\mathbf{e} - \mathbf{Ge}\| \geqslant \delta$ do Step 3–4
Step 3 set $\mathbf{H} = \mathbf{A}_R + \mathbf{BG}$
Step 4 and $\mathbf{G} = (-\mathbf{H})^{-1}\mathbf{C}_R$
Step 5 set $\mathbf{T} = \mathbf{B}(-\mathbf{H})^{-1}$
Step 6 OUTPUT

4. Probability computation

By solving Eq. (24) recursively, we obtain:

$$\mathbf{P}_0 = \mathbf{P}_1\mathbf{C}_1(-\mathbf{A}_0)^{-1} = \mathbf{P}_1\phi_1, \quad (27a)$$

$$\mathbf{P}_1 = \mathbf{P}_2\mathbf{C}_2[-(\phi_1\mathbf{B} + \mathbf{A}_1)]^{-1} = \mathbf{P}_2\phi_2, \quad (27b)$$

$$\mathbf{P}_2 = \mathbf{P}_3\mathbf{C}_3[-(\phi_2\mathbf{B} + \mathbf{A}_2)]^{-1} = \mathbf{P}_3\phi_3, \quad (27c)$$

$$\vdots$$

$$\mathbf{P}_{R-1} = \mathbf{P}_R\mathbf{C}_R[-(\phi_{R-1}\mathbf{B} + \mathbf{A}_{R-1})]^{-1} = \mathbf{P}_R\phi_R, \quad (27d)$$

$$\mathbf{P}_R = \mathbf{P}_{R+1}\mathbf{C}_R[-(\phi_R\mathbf{B} + \mathbf{A}_R)]^{-1} = \mathbf{P}_{R+1}\phi_{R+1}, \quad (27e)$$

$$\mathbf{P}_R\mathbf{T}[\phi_{R+1}\mathbf{B} + \mathbf{H}] = 0. \quad (27f)$$

where $\phi_1 = \mathbf{C}_1(-\mathbf{A}_0)^{-1}, \phi_2 = \mathbf{C}_2[-(\phi_1\mathbf{B} + \mathbf{A}_1)]^{-1}, \ldots, \phi_i = \mathbf{C}_i[-(\phi_{i-1}\mathbf{B} + \mathbf{A}_{i-1})]^{-1}$, and $\phi_{R+1} = \mathbf{C}_R[-(\phi_R\mathbf{B} + \mathbf{A}_R)]^{-1}$. Consequently, the $\mathbf{P}_i$ $(0 \leqslant i \leqslant R - 1)$ steady-state probabilities $\mathbf{P}_i$ $(0 \leqslant i \leqslant R - 1)$ in Eqs. (27a)–(27f) can be written in terms of $\mathbf{P}_R$ as $\mathbf{P}_0 = \mathbf{P}_R\prod_{i=R}^{1}\phi_i$,

$\mathbf{P}_1 = \mathbf{P}_R\prod_{i=R}^{2}\phi_i, \ldots, \mathbf{P}_{R-1} = \mathbf{P}_R\prod_{i=R}^{R}\phi_i$, and the rest of the steady-state vector $[\mathbf{P}_R, \mathbf{P}_{R+1}, \mathbf{P}_{R+2}, \ldots]$ can be determined recursively using $\mathbf{P}_i = \mathbf{P}_R\mathbf{T}^{i-R}$, for $i \geqslant R$. Once the level probability $\mathbf{P}_R$ is obtained, the steady-state solutions $[\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_{R-1}, \mathbf{P}_R, \mathbf{P}_{R+1}, \ldots]$ can be determined. The steady-state probability $\mathbf{P}_R$ can be solved by using the following normalization equation:

$$\sum_{n=0}^{\infty}\mathbf{P}_n\mathbf{e} = \left[\mathbf{P}_R\prod_{i=R}^{1}\phi_i + \mathbf{P}_R\prod_{i=R}^{2}\phi_i + \ldots + \mathbf{P}_R\prod_{i=R}^{R}\phi_i + \mathbf{P}_R + \mathbf{P}_R\mathbf{T} + \mathbf{P}_R\mathbf{T}^2 + \ldots\right]\mathbf{e}$$
$$= \mathbf{P}_R\left[\sum_{k=1}^{R}\prod_{i=R}^{k}\phi_i + \mathbf{I} + \mathbf{T}(\mathbf{I}-\mathbf{T})^{-1}\right]\mathbf{e} = 1. \tag{28}$$

It is clear that we need $O(R+2)$ equations to obtain the steady-state probability $\mathbf{P}_R$. Solving Eqs. (27f) and (28) in accordance with Cramer's rule, we obtain $\mathbf{P}_R$. Next, computing the prior state probabilities $[\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_{R-1}]$ from (27) we obtain $[\mathbf{P}_{R+1}, \mathbf{P}_{R+2}, \ldots]$ by the formula, $\mathbf{P}_i = \mathbf{P}_R\mathbf{T}^{i-R}$, $i \geqslant R+1$. We summarize the procedure below:

Algorithm 2. Recursive Solver

INPUT R, $\mathbf{B}$, $\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_R$, $\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_R$, $\mathbf{T}$, and $\mathbf{H}$
OUTPUT approximate solution $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \ldots$
Step 1 set $\phi_1 = \mathbf{C}_1(-\mathbf{A}_0)^{-1}$
Step 2 for $i = 2$ to R
  Step 3 set $\phi_i = \mathbf{C}_i[-(\phi_{i-1}\mathbf{B} + \mathbf{A}_{i-1})]^{-1}$
Step 4 end
Step 5 set $\phi_{R+1} = \mathbf{C}_R[-(\phi_R\mathbf{B} + \mathbf{A}_R)]^{-1}$
Step 6 for $k = 0$ to $R-1$
  Step 7 set $\mathbf{\Phi}_k = \prod_{i=R}^{k+1}\phi_i$
Step 8 end
Step 9 Solving
$\mathbf{P}_R\mathbf{T}[\phi_{R+1}\mathbf{B} + \mathbf{H}] = 0,\ \mathbf{P}_R\left[\sum_{k=1}^{R}\prod_{i=R}^{k}\phi_i + \mathbf{I} + \mathbf{T}(\mathbf{I}-\mathbf{T})^{-1}\right]\mathbf{e} = 1$
Step 10 for $i = 0$ to $R-1$
  Step 11 set $\mathbf{P}_i = \mathbf{P}_R\mathbf{\Phi}_i$
Step 12 end
Step 13 for $i = R+1$ to …
  Step 14 set $\mathbf{P}_{i+1} = \mathbf{P}_i\mathbf{T}$
Step 15 end
Step 16 OUTPUT

5. System performance measures

The system performance measures, such as the expected number of customers in the FES channel (denoted by L_1), the expected number of customers in the SOS channel (denoted by L_2), the expected number of idle servers (denoted by $E[I]$) and the expected number of busy servers in the system (denoted by $E[B]$), can be evaluated from the steady-state probabilities $\mathbf{P}_i = [\mathbf{P}_{i,0}, \mathbf{P}_{i,1}, \mathbf{P}_{i,2}, \ldots, \mathbf{P}_{i,R}]$. The expressions for L_1, L_2, $E[I]$ and $E[B]$ are given by:

$$L_1 = \sum_{i=1}^{\infty} i\mathbf{P}_i\mathbf{e} = \left[\sum_{i=1}^{R-1} i\mathbf{P}_i + R\mathbf{P}_R + (R+1)\mathbf{P}_R\mathbf{T} + \cdots\right]\mathbf{e}$$
$$= \left[\sum_{i=1}^{R-1} i\mathbf{P}_i + R\mathbf{P}_R(\mathbf{I}-\mathbf{T})^{-1} + \mathbf{P}_R\mathbf{T}(\mathbf{I}-\mathbf{T})^{-2}\right]\mathbf{e}, \tag{29}$$

$$L_2 = \sum_{i=0}^{\infty}\mathbf{P}_i\mathbf{J} = \left[\sum_{i=1}^{R-1}\mathbf{P}_i + \mathbf{P}_R + \mathbf{P}_R\mathbf{T} + \cdots\right]\mathbf{J}$$
$$= \left[\sum_{i=1}^{R-1}\mathbf{P}_i + \mathbf{P}_R(\mathbf{I}-\mathbf{T})^{-1}\right]\mathbf{J}, \tag{30}$$

$$L_s = L_1 + L_2$$
$$= \sum_{i=1}^{R-1}\mathbf{P}_i(i\mathbf{e} + \mathbf{J}) + \mathbf{P}_R(\mathbf{I}-\mathbf{T})^{-1}(R\mathbf{e} + \mathbf{J}) + \mathbf{P}_R\mathbf{T}(\mathbf{I}-\mathbf{T})^{-2}\mathbf{e}, \tag{31}$$

$$E[I] = \sum_{i=0}^{R-1}\mathbf{P}_i\mathbf{v}_i, \tag{32}$$

$$E[B] = R - E[I]. \tag{33}$$

where $\mathbf{J}$ and $\mathbf{e}$ are column vectors with dimension $R+1$ as $[0,1,2,\ldots,R]^{\mathrm{T}}$ and $[1,\ldots,1]^{\mathrm{T}}$, respectively. For each $0 \leqslant i \leqslant R-1$, the jth element of vector $\mathbf{v}_i$ is $\max(0, R-i-j+1)$, $j = 1,2,\ldots,R$. That is,

$$\mathbf{v}_i = \begin{bmatrix}\max(0,R-i-1+1)\\ \max(0,R-i-2+1)\\ \max(0,R-i-3+1)\\ \vdots\\ \max(0,R-i-R+1)\end{bmatrix} = \begin{bmatrix}\max(0,R-i)\\ \max(0,R-i-1)\\ \max(0,R-i-2)\\ \vdots\\ \max(0,-i+1)\end{bmatrix} \tag{34}$$

For an infinite capacity M/M/R queueing system with the SOS channel, the numerical results of L_s are obtained by considering the following three cases with different values of R:

Case 1: $\mu_1 = 15$, $\mu_2 = 5$, $\theta = 0.05$, vary the values of λ from 0.5 to 10.
Case 2: $\lambda = 10$, $\mu_1 = 15$, $\theta = 0.05$, vary the values of μ_2 from 2.5 to 10.
Case 3: $\lambda = 10$, $\mu_2 = 5$, $\theta = 0.05$, vary the values of μ_1 from 15 to 25.

Results for L_S are depicted in Figs. 2–4 for Cases 1–3, respectively. One sees from Fig. 2 that L_s drastically increases as λ increases for $R = 1$, while L_s slightly increases as λ increases for $R \geqslant 2$. From Figs. 3 and 4 we can see that L_s drastically decreases as μ_1 or μ_2 increases for $R = 1$, while L_S is not sensitive to μ_1 or μ_2 for $R \geqslant 2$.

6. Optimization analysis

We construct the total expected cost function per customer per unit time based on the system performance measures presented in the previous section. Our main objective is to determine the optimum number of servers R, say R^*, and the optimal value of the service rate $\mu = (\mu_1, \mu_2)$, say $\mu^* = (\mu_1^*, \mu_2^*)$, simultaneously, so that the expected cost function is minimized. To do this, we define the following cost elements:

$C_h \equiv$ cost per unit time per customer present in the system,
$C_1 \equiv$ cost per unit time when one server is busy,
$C_2 \equiv$ cost per unit time of providing a service rate μ_1,
$C_3 \equiv$ cost per unit time of providing a service rate μ_2,
$C_4 \equiv$ fixed cost for purchase of one server.

Using these cost elements listed above, the expected cost function $F(R, \mu_1, \mu_2)$ per customer per unit time is given by:

$$F(R, \mu_1, \mu_2) = C_hL_s + C_1E[B] + C_2\mu_1 + C_3\mu_2 + C_4R. \tag{35}$$

The cost function in (30) is assumed to be linear in the mean number of indicated quantity and it would have been a difficult task to develop analytic results for the optimum value (R^*, μ_1^*, μ_2^*) because the expected cost function is highly complex and non-linear in terms of (R, μ_1, μ_2). In the next section, we first use the Quasi-Newton method to find the optimal value of the continuous variable

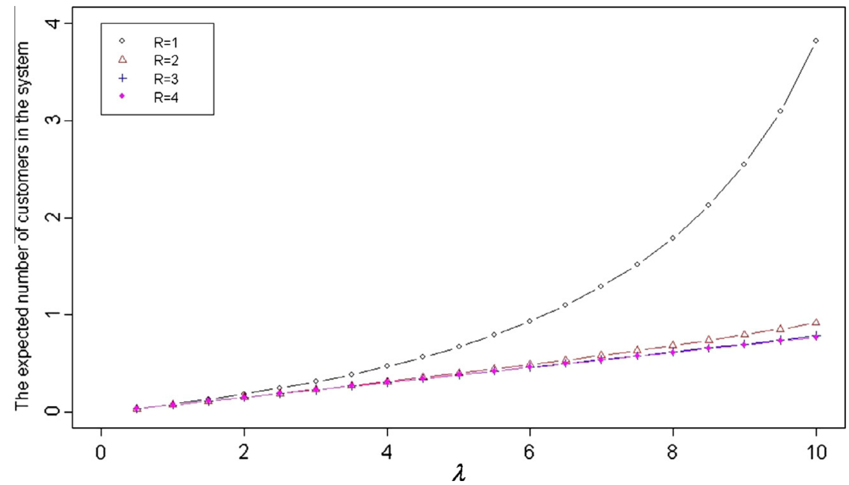

Fig. 2. The expected number of customers in the system versus λ.

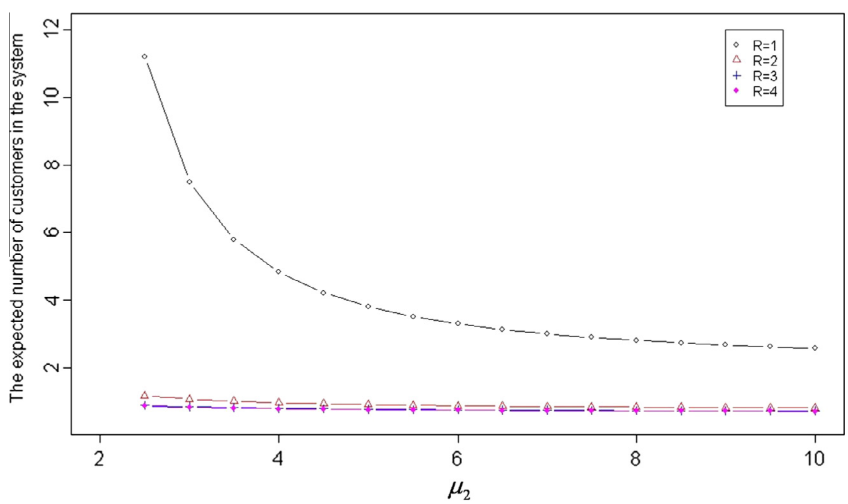

Fig. 3. The expected number of customers in the system versus μ_2.

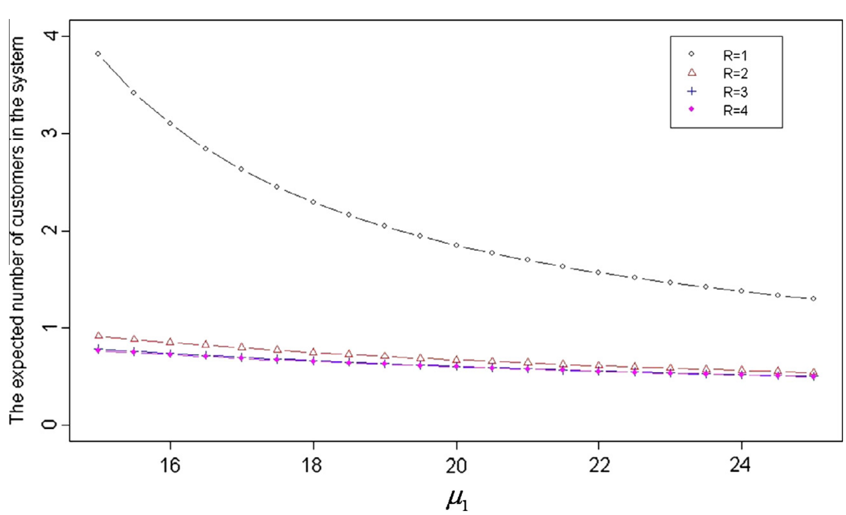

Fig. 4. The expected number of customers in the system versus μ_1.

(μ_1, μ_2), say (μ_1^*, μ_2^*), and then use the direct search method to search for the optimal value of discrete variable R, say R^*.

6.1. Quasi-Newton method

In practice, the number of servers is bounded by a positive integer $R_U \geqslant 1$ because of the purchase budget. We want to find the joint optimal value (μ_1^*, μ_2^*) for each given R in the feasible set $\{1, 2, \ldots, R_U\}$. The cost minimization problem can be illustrated mathematically as:

$$F(R, \mu_1^*, \mu_2^*) = \min_{\text{and s.t.}(20)} \{F(R, \mu_1, \mu_2)|R\}, \quad R = 1, 2, \ldots, R_U \tag{36}$$

For the optimization problem in (31), it is difficult to show the convexity of $F(R, \mu_1, \mu_2)$ in (μ_1, μ_2).We note that the derivative of the cost function F with respect to (μ_1, μ_2) indicates the direction in which the cost function increases. It means that the optimal value (μ_1^*, μ_2^*) can be found along this opposite direction of the gradient (see Chong & Zak (2001)). That is, for a fixed R, the Quasi-Newton method is employed to search (μ_1, μ_2) until the minimum value of $F(R, \mu_1, \mu_2)$ is achieved, say $F(R, \mu_1^*, \mu_2^*)$. An effective procedure that makes it possible to calculate the optimal value (R, μ_1^*, μ_2^*) is presented as follows:

Algorithm 3. Quasi-Newton Method

INPUT Cost function $F(R, \mu_1, \mu_2)$, R, λ, θ initial value $\boldsymbol{\mu}^{(0)} = \left[\mu_1^{(0)}, \mu_2^{(0)}\right]^{T}$, and the tolerance ε.
OUTPUT approximation solution $\left[\mu_1^*, \mu_2^*\right]^{T}$.
Step 1 Set the initial trial solution for $\boldsymbol{\mu}^{(0)}$, and compute $F(\boldsymbol{\mu}^{(0)})$.
Step 2 While $|\partial F/\partial \mu_1| > \varepsilon$ or $|\partial F/\partial \mu_2| > \varepsilon$ do Step 3–4
Step 3 Compute the cost gradient $\vec{\nabla} F(\boldsymbol{\mu}) = [\partial F/\partial \mu_1, \partial F/\partial \mu_2]^{T}$ and the cost Hessian matrix $H(\boldsymbol{\mu}) = \begin{bmatrix} \partial^2 F/\partial \mu_1^2 & \partial^2 F/\partial \mu_1 \partial \mu_2 \\ \partial^2 F/\partial \mu_2 \partial \mu_1 & \partial^2 F/\partial \mu_2^2 \end{bmatrix}$ at point $\vec{\mu}^{(i)}$.
Step 4 Find the new trial solution $\boldsymbol{\mu}^{(i+1)} = \boldsymbol{\mu}^{(i)} - [H(\boldsymbol{\mu}^{(i)})]^{-1} \vec{\nabla} F(\boldsymbol{\mu}^{(i)})$.
Step 5 OUTPUT

To demonstrate the validity and the process of the optimization method, some examples are given in Table 1 which consider the following cost parameters

$$C_h = \$250/\text{customer/unit-time}, \quad C_1 = \$180/\text{server/unit-time},$$
$$C_2 = \$15/\text{unit-time}, \quad C_3 = \$30/\text{unit-time}, \quad \text{and} \quad C_4 = \$60/\text{server}.$$

Under other given parameters, one can find from Table 1 that the minimum expected cost per unit time of **1682.21** is achieved at $(\mu_1^*, \mu_2^*) = (\mathbf{27.3756}, \mathbf{14.0267})$ using six iterations, which is $R = 3$ based on Case (i) with the initial value $(\mu_1, \mu_2) = (20, 10)$. Based on Case (ii) with $R = 2$ and initial value $(\mu_1, \mu_2) = (20, 20)$, the minimum expected cost per unit time of **1737.30** is achieved at $(\mu_1^*, \mu_2^*) = (\mathbf{28.8310}, \mathbf{18.7206})$ using six iterations.

6.1.1. Direct search method

After we obtain the joint optimal value (μ_1^*, μ_2^*) of the continuous variable (μ_1, μ_2), we will use the direct search method to obtain the optimal R such that the expected cost function $F(R, \mu_1^*, \mu_2^*)$ attains a minimum, say $F(R^*, \mu_1^*, \mu_2^*)$. Therefore, the cost minimization problem can be illustrated mathematically as:

$$F(R^*, \mu_1^*, \mu_2^*) = \min_{R \in \{1, 2, \ldots, R_U\}} \{F(R, \mu_1^*, \mu_2^*)\} \tag{37}$$

The procedure to find the optimal solution is described below. A numerical example is shown in Table 2 and is based on: (i) $(\lambda, \theta) = (15, 0.5)$ and (ii) $(\lambda, \theta) = (20, 0.8)$.

Algorithm 4. Direct Search Method

INPUT R_U, $F^* = M$ which M is a sufficiently large number
OUTPUT approximation solution $S^* = (R^*, \mu_1^*, \mu_2^*)$ and $F^* = F(R^*, \mu_1^*, \mu_2^*)$.
Step 1 for $R = 1$ to R_U
Step 2 Set a initial trial solution (μ_1, μ_2)
Step 3 Use Quasi-Newton method to find the optimal value (μ_1^*, μ_2^*) and the cost function $F(R, \mu_1^*, \mu_2^*)$
Step 4 If the algorithm is diverge, back to step 2 end if
Step 5 If $F(R, \mu_1^*, \mu_2^*) < F^*$
Step 6 $F^* = F(R, \mu_1^*, \mu_2^*)$ and $S^* = (R, \mu_1^*, \mu_2^*)$
Step 7 end if
Step 8 end
Step 9 OUTPUT S^* and F^*

Based on Table 2, it is noted that the optimal value $(R^*, \mu_1^*, \mu_2^*) = (\mathbf{3}, \mathbf{22.86016}, \mathbf{11.64466})$ and the corresponding minimum cost $F^* = \mathbf{1463.830}$ for Case (i). For Case (ii), $(R^*, \mu_1^*, \mu_2^*) = (\mathbf{4}, \mathbf{25.40649}, \mathbf{16.13801})$ and $F^* = \mathbf{1891.530}$ are optimal. Finally, we perform a sensitivity investigation to the optimal value (R^*, μ_1^*, μ_2^*) based on changes in specific values of the system parameters. The numerical results are shown in Table 3 for various values of θ and λ. We find that (i) R^* increases as λ or θ increases; and (ii) $\mu_1^*(\mu_2^*)$ increases as $\lambda(\theta)$ increases. Moreover, the minimum expected cost increases as λ or θ increases.

6.2. Particle Swarm Optimization

In this section, the PSO method is implemented to deal with the cost optimization problem. A comparison between the Quasi-Newton method and the PSO method are also performed. The PSO algorithm, introduced by Kennedy and Eberhart (1995) and Kennedy, Eberhart, and Shi (2001), works by having a population of particles and includes the idea of exploitation and exploration searches. Each particle having a position and velocity denotes a candidate solution. Each particle's movement is influenced by its best known local and global positions in the search-space.

Algorithm 5. Particle Swarm Optimization

INPUT R, initial solution $\mathbf{X}$, learning parameter w and the tolerance δ
OUTPUT approximate solution $\hat{\boldsymbol{\mu}} = [\hat{\mu}_1, \hat{\mu}_2]$ and $F(R, \hat{\boldsymbol{\mu}})$
Step 1 Initialization, do Step 2 to Step 4
Step 2 Initialize partial best solution $\mathbf{PB} = \mathbf{X}$
Step 3 Initialize global best solution $GB = \arg\min_{\mathbf{x}}\{F(R, \mathbf{x}); \mathbf{x} \in \mathbf{PB}\}$
Step 4 Initialize velocity $\mathbf{V} = \mathbf{0}$

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

Iterations	0	1	2	3	4	5	6
Case (i): $(\lambda,\theta) = (20,0.5)$ with $R = 3$ and initial value $(\mu_1,\mu_2) = (20,10)$							
$F(R^*,\mu_1,\mu_2)$	1862.22	1735.76	1689.68	1682.43	1682.21	1682.22	1682.21
μ_1	20	22.7766	25.5320	27.0701	27.3668	27.3756	**27.3756**
μ_2	10	11.4360	12.9115	13.8155	14.0192	14.0267	**14.0267**
$\frac{\partial F}{\partial \mu_1}$	−32.3746	−12.3033	−3.53768	−0.51339	−0.01504	−0.00001	-3×10^{-9}
$\frac{\partial F}{\partial \mu_2}$	−74.8311	−28.6228	−8.49048	−1.33248	−0.04521	−0.00005	-4.7×10^{-9}
L_s	2.88890	2.22232	1.83577	1.67455	1.64478	1.64379	1.64379
$E[B]$	2.00000	1.75253	1.55783	1.46265	1.44412	1.44350	1.44350
Case (ii): $(\lambda,\theta) = (15,0.8)$ with $R = 2$ and initial value $(\mu_1,\mu_2) = (20,20)$							
$F(R^*,\mu_1,\mu_2)$	1829.50	1760.25	1739.61	1737.33	1737.30	1737.30	1737.30
μ_1	20	23.8016	27.0887	28.6094	28.8273	28.8310	**28.8310**
μ_2	20	19.2062	18.8294	18.7303	18.7207	18.7206	**18.7206**
$\frac{\partial F}{\partial \mu_1}$	−30.3036	−10.9797	−2.84444	−0.32356	−0.00538	-1.8×10^{-6}	0.
$\frac{\partial F}{\partial \mu_2}$	−6.92424	−3.14200	−1.01269	−0.13222	−0.00232	-9.9×10^{-7}	$-3 \times 10^{(-10)}$
L_s	2.26602	1.72458	1.73603	1.66634	1.65691	1.65674	1.65674
$E[B]$	1.35000	1.25501	1.19104	1.16498	1.16134	1.16128	1.16128

The bold value means that the optimum value of μ_i.

Table 2
The optimal value (μ_1,μ_2) and the corresponding minimum expected cost.

R	Initial value	Coverage value (μ_1^*,μ_2^*)	Iteration	Cost*
(i) $(\lambda,\theta) = (15,0.5)$				
$R = 1$	[30,25]	[44.20521,24.33688]	7	2022.146
$R = 2$	[20,20]	[27.50290,14.50211]	6	1527.743
$R = 3$	[15,15]	[22.86016,11.64466]	6	1463.830
$R = 4$	[15,10]	[21.33382,10.71376]	6	1492.969
$R = 5$	[15,10]	[20.88151,10.44900]	5	1545.927
(ii) $(\lambda,\theta) =$ (20,0.8)				
$R = 1$	[50,30]	[61.14970,40.31473]	9	2890.717
$R = 2$	[40,30]	[35.80379,23.29807]	8	2056.578
$R = 3$	[30,25]	[28.23610,18.09640]	8	1896.310
$R = 4$	[25,20]	[25.40649,16.13801]	5	1891.530
$R = 5$	[20,15]	[24.38956,15.44162]	5	1933.145

Step 5 Generate two random numbers $U_1 \sim U(0,1)$ and $U_2 \sim U(0,1)$
Step 6 Update the particle's velocity and positions as:

$$\mathbf{V} = w\mathbf{V} + U_1(\mathbf{PB} - \mathbf{X}) + U_2(\mathbf{e}GB - \mathbf{X}) \text{ and } \mathbf{X} = \mathbf{X} + \mathbf{V}.$$

Step 7 Update the partial best solution **PB** and $GB = \arg\min_{\mathbf{x}}\{F(R,\mathbf{x}); \mathbf{x} \in \mathbf{PB}\}$
Step 8 Repeat Step 5 to Step 7 until $\max\{F(R,\mathbf{x}_1) - F(R,\mathbf{x}_2); \mathbf{x}_1,\mathbf{x}_2 \in \mathbf{PB}\} < \delta$
Step 9 Output approximate solution $\hat{\boldsymbol{\mu}} = GB$ and $F(R,\hat{\boldsymbol{\mu}})$

Since the PSO approach does not include the computation of the gradient, it is suitable for the non-differentiable objective function. On the basis of system parameters $\theta = 0.5$ and $R = 2,3,\ldots,7$, the Quasi-Newton method is implemented with $\lambda = 10$, 15, 20, and initial solution $\boldsymbol{\mu}^{(0)} = [\lambda, 10]^T$. Based on same conditions, the PSO algorithm is executed with an initial solution with 20 particles (randomly generated) and learning parameter $w = 0.2$. Some numerical results, including the approximate optimal solutions, computation time (in seconds), iterations needed to reach convergence and the minimum cost obtained by the two approaches are shown in Table 4. All numerical results are obtained by the mathematical program MAPLE 9, which are performed using a computer with a CPU-Pentium i3-2100, RAM 4.00 GB.

6.3. Comparison

A comparative analysis for the two methods is shown in Table 4, with changes in initial values of the decision variables for given system parameters. It is noted that (i) the approximate optimal solutions and the corresponding minimum cost established by Quasi-Newton method and PSO algorithm are very close and (ii) the computation time and convergence iterations of Quasi-Newton method are evidently less than those of PSO method. That is, the Quasi-Newton method is significantly more effective than PSO method because the PSO method has numerous computations/calculations in updating the candidate solution matrix.

Table 3
The optimal value (R^*,μ_1^*,μ_2^*) and minimum expected value $F(R^*,\mu_1^*,\mu_2^*)$ for various values of λ and θ.

(λ,θ)	(5,0.2)	(10,0.2)	(20,0.2)	(5,0.8)	(10,0.8)	(20,0.8)
R^*	2	2	3	2	3	4
(μ_1^*,μ_2^*)	[13.0953,4.35200]	[19.9021,6.80977]	[26.3424,8.64436]	[13.7175,8.80645]	[18.2622,11.6276]	[25.4065,16.1380]
$F(R^*,\mu_1^*,\mu_2^*)$	729.6488	1011.985	1391.119	976.8809	1356.801	1897.530
L_s	0.690286	0.983412	1.346797	0.958229	1.326524	1.864544
$E[B]$	0.611596	0.796155	1.221962	0.818713	1.235596	1.778650
(λ,θ)	(10,0.2)	(10,0.5)	(10,0.8)	(20,0.2)	(20,0.5)	(20,0.8)
R^*	2	3	3	3	3	4
(μ_1^*,μ_2^*)	[19.9021,6.80977]	[17.9854,9.09991]	[18.2622,11.6276]	[26.3424,8.64436]	[27.37559,14.02674]	[25.4065,16.1380]
$F(R^*,\mu_1^*,\mu_2^*)$	1011.985	1215.012	1356.801	1391.119	1682.213	1897.530
L_s	0.983412	1.173003	1.326524	1.346797	1.643788	1.864544
$E[B]$	0.796155	1.105460	1.235596	1.221962	1.443501	1.778650

Table 4
The comparison between the Quasi-Newton method and the PSO method with various values of λ and initial solutions.

R	2	3	4	5	6	7
(i) $\left(\lambda,\mu_1^{(0)},\mu_2^{(0)}\right)=(10,10,10)$						
Quasi-Newton method						
Iterations	9	7	7	7	7	7
CPU time	17.110	20.173	21.015	26.844	35.424	47.046
μ_1^*	20.82313	17.98540	17.18035	16.98173	16.94016	16.93263
μ_2^*	10.88468	9.099914	8.608544	8.493474	8.470412	8.466353
$F(R,\mu_1^*,\mu_2^*)$	1228.797	1215.012	1259.429	1316.463	1375.963	1435.886
Particle Swarm Optimization method						
Iterations	22	23	18	13	27	15
CPU time	119.765	187.532	159.459	158.098	417.968	327.624
μ_1^*	20.00670	17.96513	17.17655	16.98197	16.73007	16.75662
μ_2^*	9.544196	9.161628	8.615406	8.493446	8.270186	8.534586
$F(R,\mu_1^*,\mu_2^*)$	1238.469	1215.027	1259.429	1316.464	1376.148	1435.930
(ii) $\left(\lambda,\mu_1^{(0)},\mu_2^{(0)}\right)=(15,15,10)$						
Quasi-Newton method						
Iterations	10	7	7	6	6	6
CPU time	19.798	19.344	21.265	21.000	28.017	34.047
μ_1^*	27.50290	22.86016	21.33382	20.88151	20.76724	20.74225
μ_2^*	14.50211	11.64466	10.71376	10.44900	10.38488	10.37130
$F(R,\mu_1^*,\mu_2^*)$	1527.743	1463.830	1492.969	1545.927	1604.499	1664.242
Particle Swarm Optimization method						
Iterations	25	14	16	21	17	29
CPU time	150.061	115.296	133.749	248.909	256.893	559.404
μ_1^*	23.42697	22.86019	21.33448	20.95719	20.76728	20.42189
μ_2^*	14.68239	11.64474	10.71396	10.31709	10.38477	10.90717
$F(R,\mu_1^*,\mu_2^*)$	1542.834	1463.831	1492.968	1545.984	1604.499	1665.103
(iii) $\left(\lambda,\mu_1^{(0)},\mu_2^{(0)}\right)=(20,20,10)$						
Quasi-Newton method						
Iterations	15	7	6	6	6	6
CPU time	27.359	17.998	16.859	21.357	27.64	38.028
μ_1^*	33.86672	27.37559	25.03292	24.24485	24.01695	23.95997
μ_2^*	17.95195	14.02674	12.60472	12.14069	12.01170	11.98050
$F(R,\mu_1^*,\mu_2^*)$	1799.006	1682.213	1693.087	1740.360	1797.412	1856.803
Particle Swarm Optimization method						
Iterations	15	24	20	18	15	23
CPU time	83.075	189.732	168.342	206.8	233.689	455.797
μ_1^*	33.86676	27.42405	25.05725	24.24761	24.01718	23.95270
μ_2^*	17.95193	13.96756	12.60232	12.12005	12.01238	11.95651
$F(R,\mu_1^*,\mu_2^*)$	1799.007	1682.223	1693.087	1740.360	1797.414	1856.803

7. Concluding remarks

In this paper, we modeled an infinite capacity M/M/R queue using some practical situations wherein arrivals may require an additional optional service (second optional channel by the server). The stationary probabilities were able to be efficiently computed by using the two algorithms, in matrix and using matrix approach with the aid of computer software. We also presented efficient search approaches to determine the optimal number of channels and the optimal service rates simultaneously to incur minimum cost and we evaluated various system performance measures under the optimal operating conditions. The comparison between the Quasi-Newton method and the PSO method were also undertaken. The results rendered are useful in the contexts of modeling banking service systems, computer job processing, automatic machine quality control services channels and many other related applications.

References

Al-Jararha, J., & Madan, K. C. (2003). An M/G/1 queue with second optional service with general service time distribution. *Information and Management Sciences, 14*, 47–56.

Chong, E. K. P., & Zak, S. H. (2001). *An introduction to optimization* (2nd ed.). John Wiley and Sons, Inc..

Choudhury, G., & Madan, K. C. (2005). A two-stage batch arrival queueing system with a modified Bernoulli schedule vacation under N-policy. *Mathematical and Computer Modelling, 42*, 71–85.

Choudhury, G., & Paul, M. (2006). A batch arrival queue with a second optional service channel under N-policy. *Stochastic Analysis and Applications, 24*, 1–21.

Choudhury, G., & Tadj, L. (2009). An M/G/1 queue with two phases of service subject to the server breakdown and delayed repair. *Applied Mathematical Modelling, 33*(6), 2699–2709.

Choudhury, G., & Tadj, L. (2011). The optimal control of an $M^x/G/1$ unreliable server queue with two phases of service and Bernoulli vacation schedule. *Mathematical and Computer Modelling, 54*(1-2), 673–688.

Ke, J.-C. (2008). An $M^{[x]}/G/1$ system with startup server and J additional options for service. *Applied Mathematical Modelling, 32*, 443–458.

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. *Proceedings IEEE International Conference on Neural Networks, 4*, 1942–1948.

Kennedy, J., Eberhart, R. C., & Shi, Y. (2001). *Swarm intelligence*. CA: Morgan Kaufmann.

Tadj, L., Choudhury, G., & Tadj, C. (2006a). A quorum queueing system with a random setup time under N-policy and with Bernoulli vacation schedule. *Quality Technology and Quantitative Management, 3*(2), 145–160.

Tadj, L., Choudhury, G., & Tadj, C. (2006b). A bulk quorum queueing system with a random setup time under N-policy and with Bernoulli vacation schedule. *Stochastics: An International Journal of Probability and Stochastics Processes, 78*(1), 1–11.

Latouche G. & Ramaswami V. (1999). Introduction to matrix analytic methods in stochastic modeling (ASA-SIAM series on statistics and applied probability).

Madan, K. C. (2000). An M/G/1 queue with second optional service. *Queueing Systems, 34*, 37–46.

Medhi, J. (2002). A single server Poisson input queue with a second optional channel. *Queueing Systems, 42*, 239–242.

Neuts, M. F. (1981). *Matrix geometric solutions in stochastic models: An algorithmic approach*. Baltimore: The John Hopkins University Press. 1981.

Wang, J. (2004). An M/G/1 queue with second optional service and server breakdowns. *Computers and Mathematics with Applications, 47*, 1713–1723.

Yang, D.-Y., Wang, K.-H., & Kuo, Y.-T. (2011). Economic application in a finite capacity multi-channel queue with second optional channel. *Applied Mathematics and Computation, 217*(18), 7412–7419.