Queueing Models and
Service Management
Vol. 9, No. 2, page 15-49, 2026

QMSM


# Retrial Queue with Balking, Synchronous Working Vacation Interruption, and Non-preemptive Priority

Ming-Chin Chen[1], Jau-Chuan Ke[2], Tzu-Hsin Liu[3,*], Fu-Min Chang[3,*]

[1] Ph.D. Program of Business Administration in Industrial Development
Department of Business Administration
Chaoyang University of Technology, Taichung 413, Taiwan

[2] Department of Applied Statistics
National Taichung University of Science and Technology, Taichung 404, Taiwan

[3] Department of Finance
Chaoyang University of Technology, Taichung 413, Taiwan

(*Received November 2025; accepted May 2026*)

**Abstract:** This paper deals with a multi-server priority retrial queue with vacation interruption, where the servers are not completely out of service during the vacation period. The system serves two customers classes, denoted $P_1$ and $P_2$. In the discussed system, there is no waiting space for class $P_1$ customers. Upon a class $P_1$ customer arrival epoch and all servers are occupied, the customer departs the system. On the other hand, upon a class $P_2$ customer arrival epoch and all servers are busy, he may balk the system or join the orbit. Whenever there are no customers in the system at the moment of service completion, all servers will be on vacation at the same time. After the vacation ends, the servers are reactivated only if at least $q$ customers are found on the orbit; otherwise, the servers continue with another vacation. The vacation will be interrupted if there are more than or equal to $q$ customers in orbit after the low-rate service is completed. For this system, the effect of system parameters on system characteristics is numerically illustrated. Finally, the optimization analysis is investigated to determine the optimal vacation rate and the optimal number of special servers to maximize profit.

**Keywords:** Balking, priority retrial queue, synchronous working vacation, vacation interruption.

## 1. Introduction

### *1.1. Relevant literature*

The widespread application of queueing systems with server vacations across various fields—from telecommunications to manufacturing—has led to significant research interest. Vacations are typically utilized for auxiliary tasks, maintenance, or individual requirements.

* Corresponding author
Email: t2011119@cyut.edu.tw; fmchang@cyut.edu.tw

To provide a clear overview of the existing body of work, this review categorizes relevant studies by their primary focus: retrial mechanisms, customer behavior, vacation policies, and priority schedules.

**(1) Retrial queues and customer impatience (balking and reneging)**

Retrial queues are characterized by customers who, upon finding all servers busy, leave the service area and join an orbit, from which they retry for service after a random time. Such models are widely applied in telephone systems, call centers, and computer networks. Extensive surveys and recent developments can be found in Kim and Kim [15], Yang and Wu [35], Ke et al. [16], Zhang [36], Xu et al. [34], Kumar et al. [20], and Sundarapandiyan and Nandhini [31].

In many real-life systems, customers exhibit impatience due to excessive waiting. "Reneging" refers to customers leaving after joining the queue, while "balking" refers to customers deciding not to enter upon observing congestion. Retrial queues incorporating impatience behaviors have been studied by Chang et al. [5], Zirem et al. [38], D'Arienzo et al. [8], Liu et al. [23], Peng and Wu [28], GnanaSekar and Kandaiyan [10], Melikov et al. [25] and Khan and Paramasivam [18]. Bouchentouf et al. [2, 3] further integrated impatience into multi-server systems with synchronous vacations and automated manufacturing contexts. Incorporating impatience into retrial systems significantly increases analytical complexity but enhances realism, particularly for service systems such as call centers and healthcare facilities.

**(2) Vacation and working vacation queues**

Queueing systems with server vacations have attracted considerable attention due to their wide applicability in manufacturing systems, telecommunication networks, computer services, and healthcare operations. Server vacations may represent periods devoted to secondary tasks, maintenance activities, breakdowns, or personal leave. Foundational surveys on server vacations were provided by Tian and Zhang [32] and Ke et al. [17].

Subsequent research has expanded into fuzzy environments and strategic behaviors. Kannadasan and Sathiyamoorth [14] analyzed a working vacation queue in a fuzzy environment using the supplementary variable technique and derived performance measures such as the busy period and waiting time distributions. Chakravarthy et al. [4] focused on effective service time during breakdowns and repairs. Kalita et al. [13] examined a modified vacation policy using the Laplace–Stieltjes transform. Zhang and Wang [37] studied customer strategic behavior and information disclosure levels in working vacation queues with server breakdowns. Li et al. [22] considered customer choice behaviors in discrete-time queues with different information levels. Using a simple mean value analysis, the authors constructed the overall profit functions for customers and social welfare. These studies demonstrate the practical importance of incorporating vacation mechanisms into queueing models, especially when service interruptions or reduced service rates occur.

**(3) Vacation interruption**

In recent years, increasing attention has been devoted to vacation interruption policies. Under such mechanisms, servers terminate their vacations prematurely when certain system indicators (e.g., orbit size or queue length) exceed a threshold. This strategy is particularly relevant in practice. For instance, a surgeon may interrupt a vacation to perform emergency

surgery when patient demand becomes urgent. If servers continue their vacations despite excessive waiting customers, substantial waiting costs may arise.

Li et al. [21] studied a single-server retrial queue with balking and Bernoulli-controlled working vacation interruption using supplementary variable and probability generating function techniques. Rajadurai [29] analyzed a priority retrial queue with vacation interruption. Gupta and Kumar [11, 12] examined retrial queues with working vacations, interruption policies, server breakdowns, and different balking behaviors, and conducted cost optimization analyses. These works highlight the operational and economic significance of incorporating interruption mechanisms.

**(4) Priority service schedules**

Further, service differentiation is essential in many applications where certain customers require urgent service. For example, emergency patients in hospitals must receive treatment before regular patients. Priority mechanisms are therefore crucial in queueing models. Two primary priority disciplines are widely studied: preemptive and non-preemptive priority. Under preemptive priority, high-priority customers interrupt the service of low-priority customers. Under non-preemptive priority, ongoing service cannot be interrupted.

Walraevens et al. [33] analyzed the asymptotic behavior of orbit length distributions in priority retrial queues. Ammar and Rajadurai [1] studied a retrial queue with working breakdowns under preemptive priority. Devos et al. [7] derived stationary distributions using singularity analysis of generating functions. Damodaran et al. [6] examined transient solutions under non-preemptive priority. More recent studies include Muthusamy et al. [26], Kumar et al. [19], Shi and Liu [30], Liu et al. [24], and Dhibar and Jain [9], which incorporate multiple customer classes, Bernoulli vacations, unreliable servers, delayed vacations, and two-way communication mechanisms.

### *1.2. Research gap and contribution*

Although significant progress has been made in retrial queues with vacations, priority service, and customer impatience, most existing studies focus primarily on single-server systems. However, multi-server queues are more representative of modern industrial and service applications. Due to the mathematical complexity of multi-server priority retrial queues, available studies remain limited.

To address this gap, we develop a multi-server non-preemptive priority retrial queue incorporating synchronous working vacations, vacation interruption, and customer impatience. The model captures realistic operational features and allows for both performance evaluation and economic analysis.

Specifically, this study:

- Develops a multi-server non-preemptive priority retrial queue with synchronous working vacations and vacation interruption.
- Employs the matrix-geometric method to derive the steady-state distribution of the system.
- Conducts numerical experiments to analyze system performance under parameter variations.
- Performs profit optimization analysis to determine optimal operational decisions.

The various sections of this article are as follows. Section 2 describes the model and its practical motivation. Section 3 derives the stationary distribution and key performance measures. Section 4 presents numerical experiments, sensitivity analysis, and profit optimization. A comparison between single-class and two-class systems based on expected profit is also provided. Finally, concluding remarks are given in Section 5.

## 2. Notations and Description of System

### *2.1. Notations*

$m =$ number of servers in the service facility
$m_1 =$ number of special servers who reserved exclusively for class $P_1$ customers
$\lambda_i (i = 1,2) =$ arrival rate of class $P_i$ customers
$\mu_i (i = 1,2) =$ service rate for class $P_i$ customers during regular busy periods
$\eta_i =$ service rate for class $P_i$ customers during vacation periods
$\alpha =$ retrial rate
$\theta =$ vacation rate
$1 - \beta =$ balking rate during the regular busy period
$1 - \delta =$ balking rate during the working vacation period
$q =$ threshold number of customers in orbit at which servers will interrupt their vacation
$C(t) =$ states of the servers at time *t*
$I_i(t) (i = 1,2) =$ number of the servers serving class $P_i$ customers in the system at time *t*
$Q(t) =$ number of class $P_2$ customers in orbit at time *t*
$\pi_{i,j,k}(n) =$ steady-state probability that the system is in state $(i, j, k, n)$

### *2.2. Model description*

#### *2.2.1. Problem description*

This paper addresses the operational challenges in service systems where two distinct classes of customers ($P_1$ and $P_2$) require different service priorities. High-priority (class $P_1$) customers often have strict latency requirements, while low-priority (class $P_2$) customers can tolerate delays through a retrial mechanism. The objective is to analyze a multi-server retrial queue system that balances these needs while incorporating resource-saving strategies like synchronous working vacations.

#### *2.2.2. System model*

We consider a multi-server retrial queue with $m$ total servers. To ensure priority service for class $P_1$ customers, a subset of $m_1$ $(1 \le m_1 < m)$ servers is reserved exclusively for them; these are referred to as named special servers.

(1) Class $P_1$ customers follow a non-preemptive priority rule but are subject to a loss discipline. If at least one server is idle, the class $P_1$ customer is served without preempting any class $P_2$ customer already in service.

(2) The behavior of arriving class $P_2$ customers is governed by the operational state of the service facility and a specific threshold $k = m_1 + 1$. An arriving class $P_2$ customer faces the following scenarios:
- Regular busy period: If the number of idle servers is fewer than $k$, upon a class $P_2$ customer's arrival, the customer cannot enter service immediately. In this case, the customer joins the orbit with probability $\beta$ or balks from the system with probability $1 - \beta$. This threshold mechanism ensures that a minimum number of servers remain available for potential high-priority $P_1$ arrivals.
- Working vacation period: If all available servers are busy (no idle servers), the $P_2$ customer either balks from the system with probability $1 - \delta$ or enters the orbit with probability $\delta$.

(3) Customers in orbit retry for service after a random time. A retrial is successful only if the "idle server" conditions mentioned above for the respective states are met. If a retrial fails, the retrial customer always be placed in orbit.

(4) If, upon service completion, the system becomes empty, all servers (including special servers) initiate a synchronous working vacation. During this period, special servers are inactive, while regular servers provide service at a lower service until the orbit reaches a threshold $q$, triggering a vacation interruption.

### *2.2.3. Model assumptions*

To ensure mathematical tractability, the following assumptions are adopted:

(1) Both class $P_1$ and $P_2$ customers arrive according to independent Poisson processes with rates $\lambda_1$ and $\lambda_2$, respectively.

(2) During regular busy period, service times follow exponential distributions with rates $\mu_1$ and $\mu_2$. During working vacations, these rates are reduced to $\eta_1$ and $\eta_2$.

(3) Retrial times from the orbit are exponentially distributed with rate $\alpha$.

(4) Vacation durations are exponentially distributed with rate $\theta$.

## *2.3. Practical application*

In light of the practical applications of vacation and priority, this study focuses on the analysis and optimization of a multi-server non-preemptive priority retrial queue with two classes of balking customers and server synchronous working vacation, in which the vacation may be interrupted. Such a queueing model has a potential application in a call center scenario. During COVID-19 pandemic, telecommunication, banks, call centers, and mobile network customers play a significant role. In a call center, an automatic call distributor (ACD) is responsible for distributing incoming calls to relevant consultants or agents. The available information of the agents is managed by a supervisor. People contact the call center by communicating with a customer service consultant or agent through voice calls ($P_1$ class calls). Besides, customers also contact the call center via alternative forms of calls, such as fax, e-mail, or live chat sessions ($P_2$ class calls). Due to limited resources, different customer services have different priorities. To do this, the call center reserves a certain number ($m_1$) of special agents to ensure that voice calls are served before other calls. When there are no calls in the call center after service completion, all agents simultaneously

take a vacation (performing the secondary job). During the vacation period, special agents cannot serve, but other agents provide service at a lower speed service rate if any calls arrive. All agents stop the vacation and go back to the normal service state when there are greater than or equal to a certain number call in orbit. Otherwise, all agents will continue the vacation. Upon a voice call arrival and all agents are unavailable, it will leave the system directly. Whereas if an arriving fax (or e-mail, live chat session) finds fewer than $m_1 + 1$ agents idle in normal service state or no idle agents in vacation state, the arrival either balks or goes to orbit. The calls in orbit retry for service after a random time. If more than $m_1 + 1$ agents are free during a normal service period or at least one idle agent during the vacation period, a retrial call will be served immediately, else he/she is always back to orbit. The proposed queueing model is a good approximation of this type of call centers. Figure 1 shows the illustration of the call center.

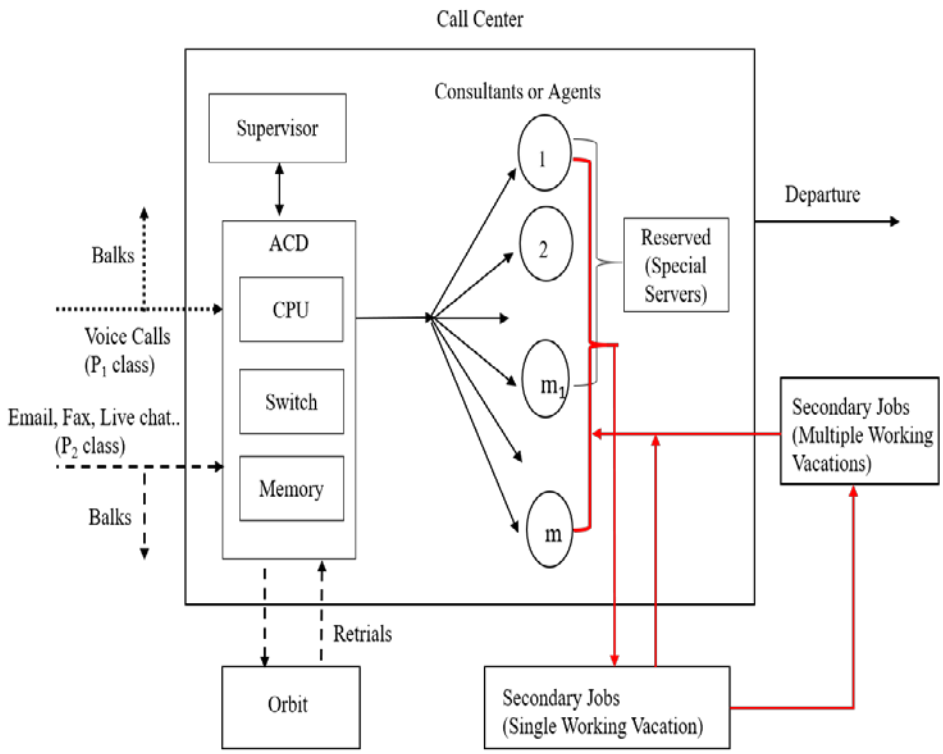


Figure 1. The illustration of the call center

## 3. Model Analysis

In this section, we derive the stationary distribution. Let $C(t)$ represent the state of the servers at time $t$, defined as follows: $C(t) = 0$ if all servers are in a working vacation period at time $t$ and $C(t) = 1$ if all servers are in a regular busy period at time $t$. Let $I_i(t), i = 1,2$, denote the number of servers that are busy serving class $P_i$ customers at time $t$. Furthermore, let $Q(t)$ be the number of class $P_2$ customers in orbit at time $t$. Apparently, the stochastic vector $\{C(t), I_1(t), I_2(t), Q(t)\}$ is a Markov process with state space $\Omega = \{(0, i, j, n), i = 0,1, \dots, m - m_1 - j, j = 0,1, \dots, m - m_1, n \geq 0\} \cup \{(1, i, j, n), i = 0, 1,$

$\dots, m-j, j=0,1,\dots,m-m_1, n \geq 0\}$. The proposed retrial queue is illustrated using the state transition rate diagram for the case of $m=2$, $m_1=1$ and $q=3$, shown in Figure 2.

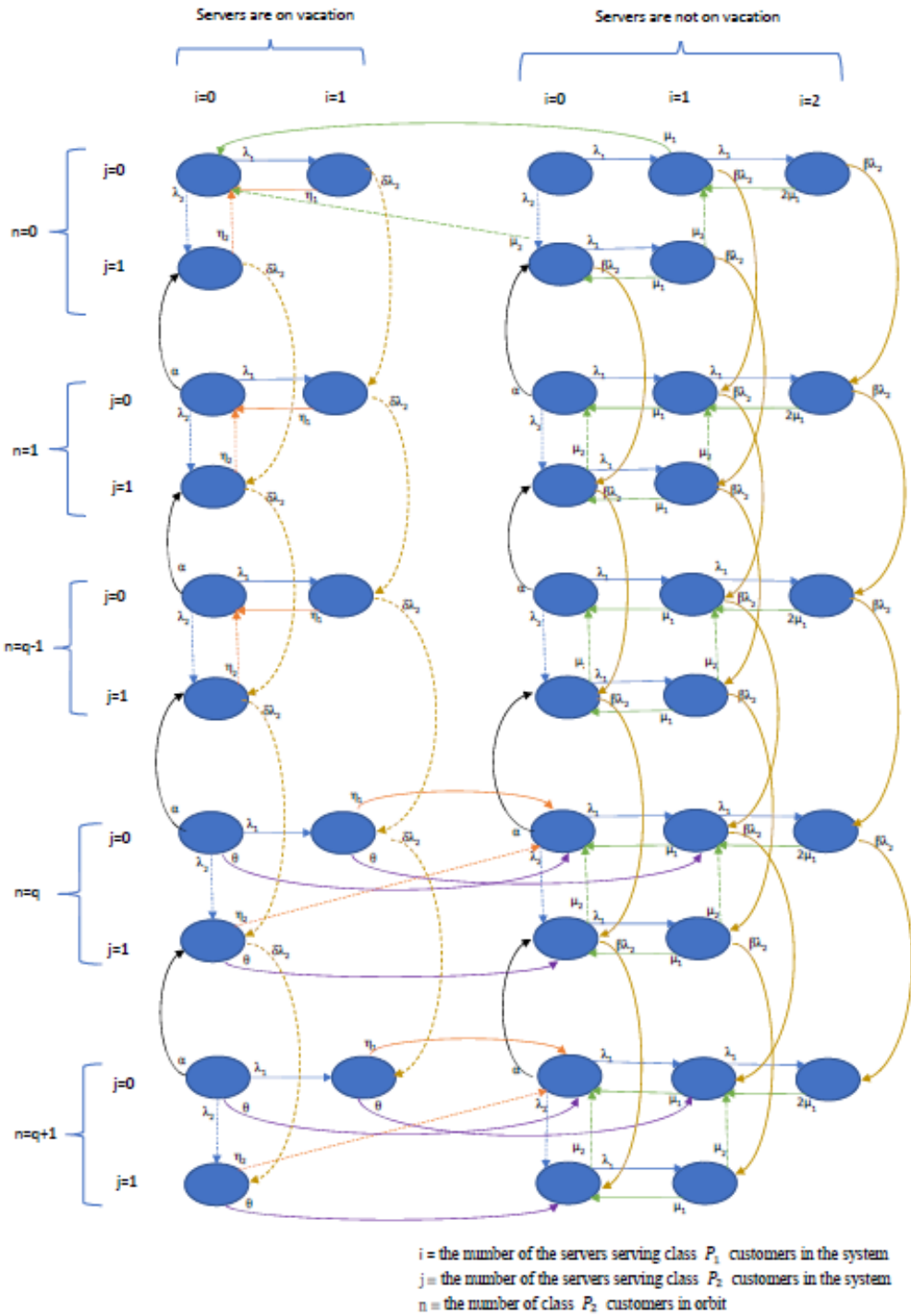


Figure 2. State transition rate diagram when $q=3$, $m_1=1$ and $m=2$.

In the following, the matrix geometric approach is utilized to determine the closed-form expression for the steady-state probabilities. By Neuts [27], the infinitesimal generator of the Markov chain $\{C(t), I_1(t), I_2(t), Q(t)\}$ is as follows:

$$\begin{array}{c} \\ 0 \\ 1 \\ \vdots \\ \boldsymbol{\Lambda} = q-1 \\ q \\ q+1 \\ \vdots \end{array} \begin{bmatrix} \mathbf{D}_0 & \mathbf{U} & & & & & \\ \mathbf{L} & \mathbf{D}_1 & \ddots & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \ddots & \mathbf{D}_1 & \mathbf{U} & & \\ & & & \mathbf{L} & \mathbf{D} & \mathbf{U} & \\ & & & & \mathbf{L} & \mathbf{D} & \ddots \\ & & & & & \ddots & \ddots \end{bmatrix}. \tag{1}$$

Due to the block structure of matrix $\boldsymbol{\Lambda}$, $\{C(t), I_1(t), I_2(t), Q(t)\}$ is called a quasi-birth-and-death process (see [27]). Let $\mathbf{O}_k$ be zero square matrix of order $k$, $\mathbf{0}_{k\times l}$ denote zero matrix with dimension $k \times l$ and $\mathbf{I}_k$ represent the identity matrix of order $k$. To provide better clarity on the dense mathematical derivations that follow, the physical roles of the submatrices $\mathbf{U}$, $\mathbf{L}$, $\mathbf{D}$, $\mathbf{D_1}$ and $\mathbf{D_0}$ in the infinitesimal generator $\boldsymbol{\Lambda}$ are summarized below:

- $\mathbf{U}$ is partitioned into block diagonal matrix as

  $$\mathbf{U} = diag\left(\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{m-m_1}\right),$$

  where

  $$\mathbf{u}_i = \left[\overbrace{0, \cdots, 0}^{\#=m-m_1-i}, \delta\lambda_2, \overbrace{0, \cdots, 0}^{\#=m-m_1-i}, \overbrace{\beta\lambda_2, \cdots, \beta\lambda_2}^{\#=m_1+1}\right].$$

  It represents the rate at which the number of class $P_2$ customers in orbit increases by one. This occurs when a class $P_2$ customer arrives but finds the system capacity (or regular server groups) full, forcing them into the orbit.

- $\mathbf{L}$ is partitioned into block super-diagonal matrix as

  $$\mathbf{L} = \begin{bmatrix} \mathbf{O}_{m+m_0+2} & \mathbf{H}_0 & & & \\ & \mathbf{O}_{m+m_0} & \mathbf{H}_1 & & \\ & & \ddots & \ddots & \\ & & & \mathbf{O}_{m_1+4} & \mathbf{H}_{m-m_1-1} \\ & & & & \mathbf{O}_{m_1+2} \end{bmatrix},$$

  where

$$\mathbf{H}_i = \alpha \begin{bmatrix} \mathbf{I}_{m-m_1-i} & & \\ \mathbf{0}_{1\times(m-m_1-i)} & & \\ & \mathbf{I}_{m-m_1-i} & \\ & \mathbf{0}_{(m_1+1)\times 1} & \end{bmatrix}.$$

It captures the rate at which the number of class $P_2$ customers in orbit decreases by one. This typically happens when an orbiting customer successfully retries and finds an available regular server.

- **D** is partitioned into the block tridiagonal matrix as

$$\mathbf{D} = \begin{bmatrix} \mathbf{A}_0 & \mathbf{B}_0 & & & \\ \mathbf{C}_1 & \mathbf{A}_1 & \mathbf{B}_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \mathbf{C}_{m-m_1-1} & \mathbf{A}_{m-m_1-1} & \mathbf{B}_{m-m_1-1} \\ & & & \mathbf{C}_{m-m_1} & \mathbf{A}_{m-m_1} \end{bmatrix},$$

where

$$\mathbf{B}_i = \lambda_2 \begin{bmatrix} \mathbf{I}_{m-m_1-i} & & \\ \mathbf{0}_{1\times(m-m_1-i)} & & \\ & \mathbf{I}_{m-m_1-i} & \\ & \mathbf{0}_{(m_1+1)\times(m-m_1-i)} & \end{bmatrix},$$

$$\mathbf{C}_i = \begin{bmatrix} i\eta_2\mathbf{I}_{m-m_1-i} & \mathbf{0}_{(m-m_1-i)\times 1} & & \\ & & i\mu_2\mathbf{I}_{m-i} & \mathbf{0}_{(m-i)\times 1} \end{bmatrix},$$

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{a}_i^{wv} & \mathbf{a}_i^{0} \\ & \mathbf{a}_i^{nb} \end{bmatrix},$$

$$\mathbf{a}_i^{wv} = \begin{bmatrix} -\omega_0 & \lambda_1 & & & \\ & -\omega_1 & \lambda_1 & & \\ & & \ddots & \ddots & \\ & & & -\omega_{m-m_1-i-1} & \lambda_1 \\ & & & & -\omega_{m-m_1-i} \end{bmatrix},$$

$$\mathbf{a}_i^{nb} = \begin{bmatrix} -\kappa_0 & \lambda_1 & & & \\ \mu_1 & -\kappa_1 & \lambda_1 & & \\ & \ddots & \ddots & \ddots & \\ & & (m-i-1)\mu_1 & -\kappa_{m-i-1} & \lambda_1 \\ & & & (m-i)\mu_1 & -\kappa_{m-i} \end{bmatrix},$$

$$\mathbf{a}_i^0 = \begin{bmatrix} \theta & & & & \\ \eta_1 & \theta & & & \\ & \ddots & \ddots & & \\ & & (m-m_1-i-1)\eta_1 & \theta & \\ & & & (m-m_1-i)\eta_1 & \theta \end{bmatrix},$$

$$\omega_k = \begin{cases} \lambda_1 + \lambda_2 + k\eta_1 + \theta + \alpha, k = 0,1,...,m-m_1-i-1, \\ \delta\lambda_2 + (m-m_1-i)\eta_1, k = m-m_1-i, \end{cases}$$

$$\kappa_k = \begin{cases} \lambda_1 + \lambda_2 + k\mu_1 + i\mu_2 + \alpha, k = 0,1,...,m-m_1-i-1, \\ \lambda_1 + \beta\lambda_2 + k\mu_1 + i\mu_2 + \alpha, k = m-m_1-i, m-m_1-i+1,...,m-i-1, \\ \beta\lambda_2 + k\mu_1 + i\mu_2, k = m-i. \end{cases}$$

It describes a transition where the orbit length remains unchanged. **D** handles standard transitions during regular busy periods, such as service completions or arrivals that are served immediately.

- $\mathbf{D}_1$ is partitioned into the block tridiagonal matrix as

$$\mathbf{D}_1 = \begin{bmatrix} \mathbf{Y}_0 & \mathbf{X}_0 & & & \\ \mathbf{Z}_1 & \mathbf{Y}_1 & \mathbf{X}_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \mathbf{Z}_{m-m_1-1} & \mathbf{Y}_{m-m_1-1} & \mathbf{X}_{m-m_1-1} \\ & & & \mathbf{Z}_{m-m_1} & \mathbf{Y}_{m-m_1} \end{bmatrix},$$

where

$$\mathbf{X}_i = \lambda_2 \begin{bmatrix} \mathbf{I}_{m-m_1-i} & & \\ \mathbf{0}_{1\times(m-m_1-i)} & & \\ & \mathbf{I}_{m-m_1-i} & \\ & \mathbf{0}_{(m_1+1)\times(m-m_1-i)} & \end{bmatrix},$$

$$\mathbf{Z}_i = \begin{bmatrix} \mathbf{0}_{(m-m_1-i)\times(m-m_1-i+1)} & i\eta_2\mathbf{I}_{m-m_1-i} & \mathbf{0}_{(m-m_1-i)\times(m_1+1)} & \mathbf{0}_{(m-i)\times 1} \\ \mathbf{0}_{(m-i)\times(m-m_1-i+1)} & & i\mu_2\mathbf{I}_{m-i} & \mathbf{0}_{(m-i)\times 1} \end{bmatrix},$$

$$\mathbf{Y}_i = \begin{bmatrix} \mathbf{y}_i^{wv} & \\ & \mathbf{y}_i^{nb} \end{bmatrix},$$

$$\mathbf{y}_i^{wv} = \begin{bmatrix} \theta-\omega_0 & \lambda_1 & & & & \\ \eta_1 & \theta-\omega_1 & \lambda_1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & (m-m_1-i-1)\eta_1 & \theta-\omega_{m-m_1-i-1} & \lambda_1 & \\ & & & (m-m_1-i)\eta_1 & \theta-\omega_{m-m_1-i} & \end{bmatrix},$$

$$\mathbf{y}_i^{nb} = \begin{bmatrix} \kappa_0 & \lambda_1 & & & \\ \mu_1 & \kappa_1 & \lambda_1 & & \\ & \ddots & \ddots & \ddots & \\ & & (m-i-1)\mu_1 & \kappa_{m-i-1} & \lambda_1 \\ & & & (m-i)\mu_1 & \kappa_{m-i} \end{bmatrix}.$$

It is similar to $\mathbf{D}$, but specifically incorporates the vacation interruption rate $\alpha$.

- $\mathbf{D_0}$ is the same as $\mathbf{D_1}$ ignoring $\alpha$, but the element of $\mathbf{D_0}[m-m_1+3, m-m_1+2]$ is shifted to the position of $\mathbf{D_0}[m-m_1+3,1]$ and the element of $\mathbf{D_0}[3m-2m_1+3, m-m_1+2]$ is shifted to the position of $\mathbf{D_0}[3m-2m_1+3,1]$.
  $\mathbf{D_0}$ describes the boundary transitions where the system shifts between vacation and normal states. The internal shifting of elements in $\mathbf{D_0}$ accounts for the specific boundary conditions when the system is at the edge of its defined state space.

### *3.1. Stability condition and special cases*

The stability condition is essential to ensure that the Markov chain is ergodic, and a unique steady-state distribution exists. For the Quasi-Birth-and-Death (QBD) process defined by the generator matrix $\mathbf{\Lambda}$ in (1), the stability is governed by the drift of the process.

Let $\mathbf{x} = [x_{0,0,0}, \dots, x_{m-m_1,0,0}, x_{0,0,1}, \dots, x_{m,0,1}, \dots, x_{0,m-m_1,0}, x_{0,m-m_1,1}, \dots, x_{m_1,m-m_1,1}]$ be the invariant probability vector of $(\mathbf{U}+\mathbf{L}+\mathbf{D})$ and $\mathbf{e}$ is a column vector with $(m+2)(m-m_1+1)$ dimensions and all its elements are equal to 1. Thus, the following linear equation is satisfied: $\mathbf{x}(\mathbf{U}+\mathbf{L}+\mathbf{D}) = \mathbf{0}_{1\times(m+2)(m-m_1+1)}$ and $\mathbf{xe} = 1$. Based on Theorem 3.1.1 in Neuts [27], the Markov chain is positive recurrent if and only if the following condition holds:

$$\mathbf{xUe} < \mathbf{xLe} \tag{2}$$

This condition represents that the mean rate of moving from state $n$ to $n+1$ (represented by **xUe**) must be strictly less than the mean rate of moving from state $n$ to $n-1$ (represented by **xLe**). Per Neuts [27], for a QBD process with a finite number of phases in each level, this mean drift condition is both necessary and sufficient for the existence of a stationary distribution.

Substituting **U** and **L** in equation (2) and after some algebraic manipulation, we obtain the necessary and sufficient condition

$$\left(\frac{\beta\lambda_2+\alpha}{\alpha}\right)P_G<1,$$

where $P_G=\sum_{j=0}^{m-m_1}\sum_{i=m-m_1-j}^{m-j}x_{i,j,1}$ is the probability that all servers except the special servers are occupied regardless of the status of the special servers when servers are not on vacation.

Let us consider a special case where $m=2$, $m_1=1$. We have $\mathbf{x}=\left[x_{0,0,0},x_{1,0,0},x_{0,0,1},x_{1,0,1},x_{2,0,1},x_{0,1,0},x_{0,1,1},x_{1,1,1}\right]$, and matrices **U**, **L** and **D** are as follows:

$$\mathbf{U}=\begin{bmatrix}0&0&0&0&0&0&0&0\\0&\delta\lambda_2&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&\beta\lambda_2&0&0&0&0\\0&0&0&0&\beta\lambda_2&0&0&0\\0&0&0&0&0&\delta\lambda_2&0&0\\0&0&0&0&0&0&\beta\lambda_2&0\\0&0&0&0&0&0&0&\beta\lambda_2\end{bmatrix},$$

$$\mathbf{L}=\begin{bmatrix}0&0&0&0&0&\alpha&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&\alpha&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0\end{bmatrix},$$

$$\mathbf{D}=\begin{bmatrix} -(\lambda_1+\lambda_2+\theta+\alpha) & \lambda_1 & \theta & 0 \\ 0 & -(\delta\lambda_2+\eta_1+\theta) & \eta_1 & \theta \\ 0 & 0 & -(\lambda_1+\lambda_2+\alpha) & \lambda_1 \\ 0 & 0 & \mu_1 & -(\lambda_1+\beta\lambda_2+\mu_1) \\ 0 & 0 & 0 & 2\mu_1 \\ 0 & 0 & \eta_2 & 0 \\ 0 & 0 & \mu_2 & 0 \\ 0 & 0 & 0 & \mu_2 \end{bmatrix}$$

$$\begin{bmatrix} 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_2 & 0 \\ \lambda_1 & 0 & 0 & 0 \\ -(\beta\lambda_2+2\mu_1) & 0 & 0 & 0 \\ 0 & -(\delta\lambda_2+\eta_2+\theta) & \theta & 0 \\ 0 & 0 & -(\lambda_1+\beta\lambda_2+\mu_2) & \lambda_1 \\ 0 & 0 & \mu_1 & -(\beta\lambda_2+\mu_1+\mu_2) \end{bmatrix}.$$

We can solve $\mathbf{x}(\mathbf{U}+\mathbf{L}+\mathbf{D})=\mathbf{0}_{1\times 8}$ and $\mathbf{xe}=1$ to obtain the invariant probability vector $\mathbf{x}$ as follows:

$$x_{0,0,0}=0, x_{1,0,0}=0, x_{0,1,0}=0, x_{1,0,1}=\frac{\lambda_1}{\mu_1}\left(1+\frac{\lambda_2+\alpha}{\lambda_1+\mu_1+\mu_2}\right)x_{0,0,1}$$

$$x_{2,0,1}=\frac{1}{2}\left(\frac{\lambda_1}{\mu_1}\right)^2\left(1+\frac{\lambda_2+\alpha}{\lambda_1+\mu_1+\mu_2}\right)x_{0,0,1}$$

$$x_{0,1,1}=\frac{\lambda_2+\alpha}{\mu_2}\frac{\mu_1+\mu_2}{\lambda_1+\mu_1+\mu_2}x_{0,0,1}, x_{1,1,1}=\frac{\lambda_2+\alpha}{\mu_2}\frac{\lambda_1}{\lambda_1+\mu_1+\mu_2}x_{0,0,1}$$

and

$$x_{0,0,1}=\left\{1+\frac{(\lambda_2+\alpha)}{\mu_2}+\left(\frac{\lambda_1}{\mu_1}+\frac{1}{2}\left(\frac{\lambda_1}{\mu_1}\right)^2\right)\frac{(\lambda_1+\lambda_2+\mu_1+\mu_2+\alpha)}{(\lambda_1+\mu_1+\mu_2)}\right\}^{-1}.$$

Substituting $\mathbf{x}$ into equation (2), the system's stability condition is simplified as follows:

$$\frac{\beta\lambda_2}{\alpha}\left(\left(1+\frac{\lambda_2+\alpha}{\lambda_1+\mu_1+\mu_2}\right)\left(\left(\frac{\lambda_1}{\mu_1}\right)+\frac{1}{2}\left(\frac{\lambda_1}{\mu_1}\right)^2\right)+\frac{\lambda_2+\alpha}{\mu_2}\right)<1. \quad (3)$$

Next, we consider another case in which $m = 1,\ m_1 = 0$ and matrices **U**, **L** and **D** are

$$\mathbf{D} = \begin{bmatrix} -(\lambda_1+\lambda_2+\theta+\alpha) & \lambda_1 & \theta \\ 0 & -(\delta\lambda_2+\eta_1+\theta) & \eta_1 \\ 0 & 0 & -(\lambda_1+\lambda_2+\alpha) \\ 0 & 0 & \mu_1 \\ 0 & 0 & \eta_2 \\ 0 & 0 & \mu_2 \end{bmatrix}$$

$$\begin{bmatrix} 0 & \lambda_2 & 0 \\ \theta & 0 & 0 \\ \lambda_1 & 0 & \lambda_2 \\ -(\beta\lambda_2+\mu_1) & 0 & 0 \\ 0 & -(\delta\lambda_2+\eta_2+\theta) & \theta \\ 0 & 0 & -(\beta\lambda_2+\mu_2) \end{bmatrix}$$

$$\mathbf{U} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \delta\lambda_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta\lambda_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta\lambda_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta\lambda_2 \end{bmatrix},\quad \mathbf{L} = \begin{bmatrix} 0 & 0 & 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Thus, the invariant probability vector $\mathbf{x} = [x_{0,0,0}, x_{1,0,0}, x_{0,0,1}, x_{1,0,1}, x_{0,1,0}, x_{0,1,1}]$ can be determined by solving for $\mathbf{x}(\mathbf{U}+\mathbf{L}+\mathbf{D}) = \mathbf{0}_{1\times 6}$ and $\mathbf{xe} = 1$, yielding

$$x_{0,0,0} = 0, x_{1,0,0} = 0, x_{1,0,1} = \frac{\lambda_1}{\mu_1} x_{0,0,1}, x_{0,1,0} = 0, x_{0,1,1} = \frac{(\lambda_2+\alpha)}{\mu_2} x_{0,0,1}$$

and $x_{0,0,1} = \left(1 + \frac{\lambda_1}{\mu_1} + \frac{(\lambda_2+\alpha)}{\mu_2}\right)^{-1}$.

Substituting **x** into equation (2) gives

$$\frac{\beta\lambda_2}{\alpha}\left(\frac{\lambda_1}{\mu_1} + \frac{\lambda_2+\alpha}{\mu_2}\right) < 1. \tag{4}$$

According to the system's stability conditions under the above special cases, we can draw the following conclusions:

- When the mean service rate for class $P_1$ customers approaches infinity, the service time of class $P_1$ customers can be ignored and the number of $P_1$ customers will decrease sharply in a short time. From the left side of formula (4), we can get

$$\lim_{\mu_1 \to \infty} \frac{\beta\lambda_2}{\alpha}\left(\frac{\lambda_1}{\mu_1}+\frac{\lambda_2+\alpha}{\mu_2}\right)=\frac{\beta\lambda_2}{\alpha}\frac{\lambda_2+\alpha}{\mu_2}<1.$$

  The reduced stable requirement matches that of the M/M/1 retrial queue with a constant retrial rate and vacation interruption.
- If the mean service rate for class $P_1$ customers approaches 0, then the system is prone to instability when class $P_1$ customers arrive. The following results can be obtained from formula (4):

$$\lim_{\mu_1 \to 0} \frac{\beta\lambda_2}{\alpha}\left(\frac{\lambda_1}{\mu_1}+\frac{\lambda_2+\alpha}{\mu_2}\right)=\infty.$$

  The system's stability condition cannot hold when $\mu_1 \to 0$.
- When the mean service rate for class $P_2$ customers approaches infinity, the service time of class $P_2$ customers can be ignored and the number of $P_2$ customers will decrease sharply in a short time. From the left side of formula (4), we can get

$$\lim_{\mu_2 \to \infty} \frac{\beta\lambda_2}{\alpha}\left(\frac{\lambda_1}{\mu_1}+\frac{\lambda_2+\alpha}{\mu_2}\right)=\frac{\beta\lambda_2}{\alpha}\frac{\lambda_1}{\mu_1}<1.$$

  If the ratio of the expected rate of retrials to the expected rate of entering in orbit is less than the ratio of mean service rate of class $P_1$ customers to the mean arrival rate of class $P_1$ customers, the system is stable.
- If the mean service rate for class $P_2$ customers approaches 0, then the system is prone to instability when class $P_2$ customers arrive. The following results can be obtained from formula (4):

$$\lim_{\mu_2 \to 0} \frac{\beta\lambda_2}{\alpha}\left(\frac{\lambda_1}{\mu_1}+\frac{\lambda_2+\alpha}{\mu_2}\right)=\infty.$$

  Under this condition, the system cannot reach a stable state.
- The case where $\lambda_1 \to 0$ is like the case where $\mu_1 \to \infty$. The following result constitutes a valid conclusion equivalent to the stability requirement of the M/M/1 retrial queue with a constant retrial rate and vacation interruption.

$$\lim_{\mu_1 \to \infty} \frac{\beta\lambda_2}{\alpha}\left(\frac{\lambda_1}{\mu_1}+\frac{\lambda_2+\alpha}{\mu_2}\right)=\frac{\beta\lambda_2}{\alpha}\frac{\lambda_2+\alpha}{\mu_2}<1.$$

### *3.2. Rate matrix*

An important matrix for evaluating system characteristics to analyze this process is the matrix **G**, known as the rate matrix of the Markov chain. It is the minimal nonnegative

matrix solution to $\mathbf{G}^2\mathbf{L} + \mathbf{G}\mathbf{D} + \mathbf{U} = \mathbf{0}$, whose spectral radius is less than 1. Since the rate matrix is very difficult to derive an analytic solution, we use the following iteration procedure to approximate the rate matrix in the numerical illustration. Starting with the initial value $\mathbf{G}_0 = \mathbf{0}$, the iteration procedure terminates when the entry-wise norm of the residual matrix falls below the tolerance $\epsilon = 10^{-5}$, i.e., $\mathbf{e}^T|\mathbf{G}_n^2\mathbf{L} + \mathbf{G}_n\mathbf{D} + \mathbf{U}|\mathbf{e} < 10^{-5}$. This ensures that the approximation $\mathbf{G}_n$ satisfies the balancing equations with high precision.

$$\mathbf{G}_{n+1} = -\left(\mathbf{U}\mathbf{D}^{-1} + \mathbf{G}_n^2\mathbf{L}\mathbf{D}^{-1}\right) \text{ for } n \geq 0 .$$

**Iteration procedure**

**Input** matrices $\mathbf{U}$, $\mathbf{D}$, $\mathbf{L}$ and tolerance $\epsilon$
**Output** approximate rate matrix $\mathbf{G}$
Step 1 set the initial solution $\mathbf{G}_0 = \mathbf{0}$ (a zero matrix of appropriate dimension).
Step 2 computing the residual: $\mathbf{e}^T|\mathbf{G}_n^2\mathbf{L} + \mathbf{G}_n\mathbf{D} + \mathbf{U}|\mathbf{e}$, which is the summation of the absolute value of all elements of the matrix $\mathbf{G}_n^2\mathbf{L} + \mathbf{G}_n\mathbf{D} + \mathbf{U}$
Step 3 while residual $> \epsilon$ do steps 4 and 5
Step 4 Set $\mathbf{G}_{n+1} = -(\mathbf{G}_n^2\mathbf{L}\mathbf{D}^{-1} + \mathbf{U}\mathbf{D}^{-1})$
Step 5 and residual: $\mathbf{e}^T|\mathbf{G}_n^2\mathbf{L} + \mathbf{G}_n\mathbf{D} + \mathbf{U}|\mathbf{e}$
Step 6 Output the approximate rate matrix $\mathbf{G}$

It noted that each iteration (Step 4) involves two matrix multiplications and two matrix inversions (or solving linear systems). The time complexity per iteration is $O(S^3)$, where $S = m + 2(m - m_1) + 1$ is the dimension of the square submatrices $\mathbf{U}$, $\mathbf{D}$ and $\mathbf{L}$. Further, the algorithm requires storing the current iteration $\mathbf{G}_n$, the constant matrices $\mathbf{U}$, $\mathbf{D}$, $\mathbf{L}$ and an auxiliary matrix for the residual calculation. The memory complexity is $O(S^2)$.

To illustrate the above procedure, a numerical example is presented. Given the system parameters $m = 2$, $m_1 = 1$, $q = 1$, $\lambda_1 = 1.8$, $\lambda_2 = 1.8$, $\mu_1 = 1.5$, $\mu_2 = 1.5$, $\eta_1 = 0.05$, $\eta_2 = 0.05$, $\alpha = 2$, $\theta = 0.5$, $\beta = 0.7$ and $\delta = 0.7$, we have the matrices

$$\mathbf{U} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.26 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.26 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.26 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.26 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.26 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.26 \end{bmatrix},$$

$$\mathbf{L} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{D} = \begin{bmatrix} -6.1 & 1.8 & 0.5 & 0 & 0 & 1.8 & 0 & 0 \\ 0 & -1.81 & 0.05 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & -5.6 & 1.8 & 0 & 0 & 1.8 & 0 \\ 0 & 0 & 1.5 & -4.56 & 1.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & -4.26 & 0 & 0 & 0 \\ 0 & 0 & 0.05 & 0 & 0 & -1.81 & 0.5 & 0 \\ 0 & 0 & 1.5 & 0 & 0 & 0 & -4.56 & 1.8 \\ 0 & 0 & 0 & 1.5 & 0 & 0 & 1.5 & -4.26 \end{bmatrix}.$$

**Iteration 0**: $\mathbf{G}_0 = \mathbf{O}_{(m+2)(m-m_1+1)}$ and therefore, $\mathbf{e}^T|\mathbf{U}|\mathbf{e} = 7.56$. The next approximated rate matrix is

$$\mathbf{G}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.6961 & 0.0488 & 0.1367 & 0.0578 & 0 & 0.0224 & 0.0095 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1450 & 0.4748 & 0.2006 & 0 & 0.0665 & 0.0281 \\ 0 & 0 & 0.1021 & 0.3343 & 0.4370 & 0 & 0.0468 & 0.0198 \\ 0 & 0 & 0.0488 & 0.0481 & 0.0203 & 0.6961 & 0.1110 & 0.0469 \\ 0 & 0 & 0.1450 & 0.1538 & 0.0650 & 0 & 0.3874 & 0.1637 \\ 0 & 0 & 0.1021 & 0.2213 & 0.0935 & 0 & 0.1598 & 0.3633 \end{bmatrix}.$$

**Iteration 1**: The error can be computed as $\mathbf{e}^T|\mathbf{G}_1^2\mathbf{L} + \mathbf{G}_1\mathbf{D} + \mathbf{U}|\mathbf{e} = 1.0705$. The next approximated rate matrix is

$$\mathbf{G}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.6961 & 0.0635 & 0.1523 & 0.0644 & 0 & 0.0617 & 0.0261 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1684 & 0.4996 & 0.2111 & 0 & 0.1291 & 0.0545 \\ 0 & 0 & 0.1255 & 0.3592 & 0.4476 & 0 & 0.1095 & 0.0462 \\ 0 & 0 & 0.0635 & 0.0637 & 0.0269 & 0.6961 & 0.1503 & 0.0635 \\ 0 & 0 & 0.1684 & 0.1787 & 0.0755 & 0 & 0.4500 & 0.1901 \\ 0 & 0 & 0.1255 & 0.2462 & 0.1040 & 0 & 0.2225 & 0.3898 \end{bmatrix}.$$

**Iteration 2**: The error can be computed as $\mathbf{e}^T|\mathbf{G}_2^2\mathbf{L} + \mathbf{G}_2\mathbf{D} + \mathbf{U}|\mathbf{e} = 0.4165$. The next approximated rate matrix is

$$\mathbf{G}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.6961 & 0.0699 & 0.1591 & 0.0672 & 0 & 0.0787 & 0.0333 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1770 & 0.5087 & 0.2150 & 0 & 0.1521 & 0.0643 \\ 0 & 0 & 0.1345 & 0.3687 & 0.4516 & 0 & 0.1334 & 0.0564 \\ 0 & 0 & 0.0699 & 0.0704 & 0.0298 & 0.6961 & 0.1674 & 0.0707 \\ 0 & 0 & 0.1770 & 0.1878 & 0.0794 & 0 & 0.4730 & 0.1999 \\ 0 & 0 & 0.1345 & 0.2557 & 0.1081 & 0 & 0.2464 & 0.3999 \end{bmatrix}.$$

**Iteration 3**: The error can be computed as $\mathbf{e}^T|\mathbf{G}_3^2\mathbf{L} + \mathbf{G}_3\mathbf{D} + \mathbf{U}|\mathbf{e} = 0.1771$. The next approximated rate matrix is

$$\mathbf{G}_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.6961 & 0.0728 & 0.1622 & 0.0685 & 0 & 0.0865 & 0.0366 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1805 & 0.5125 & 0.2165 & 0 & 0.1615 & 0.0682 \\ 0 & 0 & 0.1383 & 0.3727 & 0.4533 & 0 & 0.1435 & 0.0606 \\ 0 & 0 & 0.0728 & 0.0735 & 0.0311 & 0.6961 & 0.1752 & 0.0740 \\ 0 & 0 & 0.1805 & 0.1916 & 0.0809 & 0 & 0.4824 & 0.2038 \\ 0 & 0 & 0.1383 & 0.2597 & 0.1097 & 0 & 0.2565 & 0.4041 \end{bmatrix}.$$

Table 1 below lists the error values for different number of iterations. When the number of iterations exceeds 45, the $\mathbf{e}^T|\mathbf{G}_n^2\mathbf{L} + \mathbf{G}_n\mathbf{D} + \mathbf{U}|\mathbf{e}$ value is less than $2.71 \times 10^{-15}$. The steps of procedure were iterated for 200 times; the output approximate rate matrix can be expressed as

$$\mathbf{G}_{200}=\begin{bmatrix}0&0&0&0&0&0&0&0\\0&0.6961&0.0753&0.1648&0.0696&0&0.0931&0.0394\\0&0&0&0&0&0&0&0\\0&0&0.1832&0.5153&0.2177&0&0.1686&0.0712\\0&0&0.1411&0.3758&0.4545&0&0.1511&0.0639\\0&0&0.0753&0.0762&0.0322&0.6961&0.1818&0.0768\\0&0&0.1832&0.1944&0.0821&0&0.4895&0.2068\\0&0&0.1411&0.2628&0.1110&0&0.2641&0.4074\end{bmatrix}.$$

Table 1. The approximation error $\mathbf{e}^T|\mathbf{G}_n^2\mathbf{L}+\mathbf{G}_n\mathbf{D}+\mathbf{U}|\mathbf{e}$ under different iteration numbers

| Iteration | $\mathbf{e}^T\|\mathbf{G}_n^2\mathbf{L}+\mathbf{G}_n\mathbf{D}+\mathbf{U}\|\mathbf{e}$ |
|---|---|
| 1 | 1.0705 |
| 2 | 0.4165 |
| 3 | 0.1771 |
| 4 | 0.0775 |
| 5 | 0.0343 |
| 10 | $5.8279\times10^{-4}$ |
| 20 | $1.5857\times10^{-7}$ |
| 30 | $4.2087\times10^{-11}$ |
| 40 | $1.3861\times10^{-14}$ |
| 50 | $2.7074\times10^{-15}$ |

### *3.3. Stationary distribution*

Under the stability condition, the stationary limit of the process $\{C(t),I_1(t),I_2(t),Q(t)\}$ is denoted as $\{C,I_1,I_2,Q\}$. We define the steady state probabilities as follows:

$$\pi_{i,j,k}(n)=\lim_{t\to\infty}P\{C(t)=k,I_1(t)=i,I_2(t)=j,Q(t)=n\},(i,j,k,n)\in\Omega.$$

$$\mathbf{\Pi}_j(n)=\left[\pi_{0,j,0}(n),...,\pi_{m-m_1-j,j,0}(n),\pi_{0,j,1}(n),...,\pi_{m-j,j,1}(n)\right],\ 0\le j\le m-m_1,$$

$$\mathbf{\Pi}(n)=\left[\mathbf{\Pi}_0(n),\mathbf{\Pi}_1(n),...,\mathbf{\Pi}_{m-m_1}(n)\right].$$

According to the matrix geometric method, the steady-state probabilities are calculated by

$$\mathbf{\Pi}(0)\mathbf{D}_0+\mathbf{\Pi}(1)\mathbf{L}=\mathbf{0},\tag{5}$$

$$\boldsymbol{\Pi}(k-1)\mathbf{U}+\boldsymbol{\Pi}(k)\mathbf{D}_1+\boldsymbol{\Pi}(k+1)\mathbf{L}=\mathbf{0},1\le k\le q-1, \tag{6}$$

$$\boldsymbol{\Pi}(q-1)\mathbf{U}+\boldsymbol{\Pi}(q)(\mathbf{D}+\mathbf{GL})=\mathbf{0}, \tag{7}$$

$$\boldsymbol{\Pi}(k)=\boldsymbol{\Pi}(q)\mathbf{G}^{k-q},k\ge q+1, \tag{8}$$

and the normalizing equation

$$\sum_{k=0}^{\infty}\boldsymbol{\Pi}(k)\mathbf{e}=1. \tag{9}$$

Solving equations (5) and (6), we finally get

$$\boldsymbol{\Pi}(0)=\boldsymbol{\Pi}(1)\mathbf{L}(-\mathbf{D}_0)^{-1}=\boldsymbol{\Pi}(1)\boldsymbol{\varphi}_1, \tag{10}$$

$$\boldsymbol{\Pi}(k-1)=\boldsymbol{\Pi}(k)\mathbf{L}(-\boldsymbol{\varphi}_{k-1}\mathbf{U}-\mathbf{D}_1)^{-1}=\boldsymbol{\Pi}(k)\boldsymbol{\varphi}_k,2\le k\le q, \tag{11}$$

where

$$\boldsymbol{\varphi}_k=\begin{cases}\mathbf{L}(-\mathbf{D}_0)^{-1}, & k=1,\\ \mathbf{L}(-\boldsymbol{\varphi}_{k-1}\mathbf{U}-\mathbf{D}_1)^{-1}, & k=2,3,...,q.\end{cases}$$

Substituting equations (10) and (11) into equations (7) and (9) yields

$$\boldsymbol{\Pi}(q)(\boldsymbol{\varphi}_q\mathbf{U}+\mathbf{D}+\mathbf{GL})=\mathbf{0}, \tag{12}$$

$$\sum_{k=0}^{\infty}\boldsymbol{\Pi}(k)\mathbf{e}=\boldsymbol{\Pi}(q)\left(\sum_{k=1}^{q}\boldsymbol{\Psi}_k+(\mathbf{I}-\mathbf{G})^{-1}\right)\mathbf{e}=1, \tag{13}$$

where

$$\boldsymbol{\Psi}_k=\boldsymbol{\varphi}_q\boldsymbol{\varphi}_{q-1}\cdots\boldsymbol{\varphi}_{k+1}.$$

The vector $\boldsymbol{\Pi}(q)$ can be determined by equations (12) and (13). Once $\boldsymbol{\Pi}(q)$ is obtained, $\boldsymbol{\Pi}(k)$ $(k\ge q+1)$ can be obtained using equation (8) and $\boldsymbol{\Pi}(k)$ $(0\le k\le q-1)$ can be determined by equations (10)-(11).

### *3.4. System characteristics*

Some system characteristics are developed using the stationary probabilities obtained in the previous subsection in this section.

- The mean number of servers on vacation is

$$E[V]=m\sum_{n=0}^{\infty}\sum_{j=0}^{m-m_1}\sum_{i=0}^{m-m_1-j}\pi_{i,j,0}(n)=\mathbf{\Pi}(q)\left(\sum_{n=1}^{q}\mathbf{\Psi}_n+(\mathbf{I}-\mathbf{G})^{-1}\right)\mathbf{v},$$

where $\mathbf{v}=[\mathbf{v}_0,\mathbf{v}_1,\cdots,\mathbf{v}_{m-m_1}]^T$ and $\mathbf{v}_j=\left[\underbrace{m,\cdots,m}_{\#=m-m_1-j+1},\ \underbrace{0,\cdots,0}_{\#=m-j+1}\right]$.

- The mean number of busy servers during normal busy period is

$$E[B]=\sum_{n=0}^{\infty}\sum_{j=0}^{m-m_1}\sum_{i=0}^{m-j}(i+j)\pi_{i,j,1}(n)=\mathbf{\Pi}(q)\left(\sum_{n=1}^{q}\mathbf{\Psi}_n+(\mathbf{I}-\mathbf{G})^{-1}\right)\mathbf{b},$$

where $\mathbf{b}=[\mathbf{b}_0,\mathbf{b}_1,\cdots,\mathbf{b}_{m-m_1}]^T$ and $\mathbf{b}_j=\left[\underbrace{0,\cdots,0,}_{\#=m-m_1-j+1}\ \underbrace{j,j+1,\cdots,m}_{\#=m-j+1}\right]$.

- The mean number of idle servers during normal busy period is

$$E[I]=\sum_{n=0}^{\infty}\sum_{j=0}^{m-m_1}\sum_{i=0}^{m-j}(m-i-j)\pi_{i,j,1}(n)=\mathbf{\Pi}(q)\left(\sum_{n=1}^{q}\mathbf{\Psi}_n+(\mathbf{I}-\mathbf{G})^{-1}\right)(m\mathbf{e}-\mathbf{v}-\mathbf{b})$$

- The mean number of $P_1$ customers lost in the system is

$$\begin{aligned}E[Loss]&=\lambda_1\sum_{n=0}^{\infty}\sum_{j=0}^{m-m_1}\pi_{m-m_1-j,j,0}(n)+\lambda_1\sum_{n=0}^{\infty}\sum_{j=0}^{m-m_1}\pi_{m-j,j,1}(n)\\&=\mathbf{\Pi}(q)\left(\sum_{n=1}^{q}\mathbf{\Psi}_n+(\mathbf{I}-\mathbf{G})^{-1}\right)\boldsymbol{\rho}\end{aligned}$$

where $\boldsymbol{\rho}=[\boldsymbol{\rho}_0,\boldsymbol{\rho}_1,\cdots,\boldsymbol{\rho}_{m-m_1}]^T$ and $\boldsymbol{\rho}_j=\left[\underbrace{0,\cdots,0,}_{\#=m-m_1-j}\ \lambda_1,\ \underbrace{0,\cdots,0}_{\#=m-j},\lambda_1\right]$.

- The mean number of $P_1$ customers being served is

$$\begin{aligned}E[S_1]&=\sum_{n=0}^{\infty}\sum_{j=0}^{m-m_1}\sum_{i=0}^{m-m_1-j}i\pi_{i,j,0}(n)+\sum_{n=0}^{\infty}\sum_{j=0}^{m-m_1}\sum_{i=0}^{m-j}i\pi_{i,j,1}(n)\\&=\mathbf{\Pi}(q)\left(\sum_{n=1}^{q}\mathbf{\Psi}_n+(\mathbf{I}-\mathbf{G})^{-1}\right)\mathbf{s}_1\end{aligned}$$

where $\mathbf{s}_1=[\mathbf{s}_{1,0},\mathbf{s}_{1,1},\cdots,\mathbf{s}_{1,m-m_1}]^T$ and

$$\mathbf{s}_{1,j}=\left[\underbrace{0,1,\ \cdots,m-m_1-j,}_{\#=m-m_1-j+1}\ \underbrace{0,1,\ \cdots,m-j}_{\#=m-j+1}\right].$$

- The mean number of $P_2$ customers being served is

$$E[S_2]=\sum_{n=0}^{\infty}\sum_{j=0}^{m-m_1}\sum_{i=0}^{m-m_1-j} j\pi_{i,j,0}(n)+\sum_{n=0}^{\infty}\sum_{j=0}^{m-m_1}\sum_{i=0}^{m-j} j\pi_{i,j,1}(n)$$
$$=\mathbf{\Pi}(q)\left(\sum_{n=1}^{q}\mathbf{\Psi}_n+(\mathbf{I}-\mathbf{G})^{-1}\right)\mathbf{s}_2$$

where $\mathbf{s}_2=[\mathbf{s}_{2,0},\mathbf{s}_{2,1},\cdots,\mathbf{s}_{2,m-m_1}]^T$ and $\mathbf{s}_{2,j}=\left[\underbrace{j,\ \cdots,j,}_{\#=m-m_1-j+1}\ \underbrace{j,\cdots,j}_{\#=m-j+1}\right]$.

- The mean number of $P_2$ customers in orbit during the working vacation period is

$$E[N_0]=\sum_{n=0}^{\infty}\sum_{j=0}^{m-m_1}\sum_{i=0}^{m-m_1-j} n\pi_{i,j,0}(n)$$
$$=\mathbf{\Pi}(q)\left(\sum_{n=1}^{q-1}n\mathbf{\Psi}_{n+1}+q(\mathbf{I}-\mathbf{G})^{-1}+\mathbf{G}(\mathbf{I}-\mathbf{G})^{-2}\right)\mathbf{h}$$

where $\mathbf{h}=[\mathbf{h}_0,\mathbf{h}_1,\cdots,\mathbf{h}_{m-m_1}]^T$ and $\mathbf{h}_j=\left[\underbrace{1,\ \cdots,1,}_{\#=m-m_1-j+1}\ \underbrace{0,\cdots,0}_{\#=m-j+1}\right]$.

- The mean number of $P_2$ customers in orbit during a regular busy period is

$$E[N_1]=\sum_{n=0}^{\infty}\sum_{j=0}^{m-m_1}\sum_{i=0}^{m-j} n\pi_{i,j,1}(n)$$
$$=\mathbf{\Pi}(q)\left(\sum_{n=1}^{q-1}n\mathbf{\Psi}_{n+1}+q(\mathbf{I}-\mathbf{G})^{-1}+\mathbf{G}(\mathbf{I}-\mathbf{G})^{-2}\right)(\mathbf{e}-\mathbf{h})^{\cdot}$$

- The mean orbit length of $P_2$ customers is

$$E[N]=E[N_0]+E[N_1].$$

- The mean waiting time of a $P_2$ customer in orbit is

$$E[W]=\frac{E[N_0]}{\lambda_2\delta}+\frac{E[N_1]}{\lambda_2\beta}.$$

## 4. Numerical analysis

Here, we firstly present the results of two numerical experiments, which aim to discuss the feasibility of the proposed model, focusing on the system characteristics and profit optimization via the genetic algorithm. Finally, a comparison between the system with one-class customers and the system with two classes of customers is made.

### *4.1. Sensitivity analysis*

We investigate the influence of various parameters on system performance. The default system parameters are $m = 6,\ m_1 = 2,\ q = 15,\ \lambda_1 = 1.8,\ \lambda_2 = 1.8,\ \mu_1 = 1.5,\ \mu_2 = 1.5,\ \eta_1 = 0.05,\ \eta_2 = 0.05,\ \alpha = 2,\ \theta = 0.5,\ \beta = 0.7$ and $\delta = 0.7$.

To ensure the validity of the numerical results, we first follow a non-recursive three-step procedure:

Step 1 (Construction of the limit generator): Before compute the rate matrix $\mathbf{G}$, we construct the matrix $\mathbf{U} + \mathbf{D} + \mathbf{L}$, which represents the transition rates between phases when the orbit size $n$ is sufficiently large $n \geq q$.

Step 2 (Solving for the invariant vector): Solve the finite system of linear equations $\mathbf{x}(\mathbf{U} + \mathbf{D} + \mathbf{L}) = 0$ subject to $\mathbf{xe} = 1$. Since $\mathbf{U} + \mathbf{D} + \mathbf{L}$ is a finite generator matrix of the internal phases, $\mathbf{x}$ is determined solely by the arrival and service parameters $(\lambda_1, \lambda_2, \mu_1, \mu_2, \eta_1, \eta_2, \theta)$ and is independent of the level $n$ or the matrix $\mathbf{G}$.

Step 3 (Drift evaluation): Check if the stability condition $\mathbf{xUe} < \mathbf{xLe}$ is met. If the condition is met, the system is guaranteed to be positive recurrent (stable), and we proceed to Step 4.

**Numerical procedure for computation:**

Step 4 (Matrix iteration): Compute the approximate rate matrix $\mathbf{G}$ using the iteration procedure from subsection 3.2.

Step 5 (Vector computation): Compute $\mathbf{\Pi}(q)$ and subsequent vectors $\mathbf{\Pi}(k)$ $(k \geq q + 1)$ using equations (8) - (13).

Step 6 (Results): Compute the system characteristics such as mean orbit length $E[N]$, loss rate $E[Loss]$ and busy servers $E[B]$.

Table 2. Variations be tween various system characteristics and $\lambda_1$.

| $\lambda_1$ | $E[N]$ | $E[Loss]$ | $E[B]$ | $E[V]$ |
|---|---|---|---|---|
| 0.5 | 2.9942 | 0.2220 | 1.1343 | 3.1197 |
| 0.7 | 3.0233 | 0.3039 | 1.2122 | 3.0159 |
| 0.9 | 3.0565 | 0.3808 | 1.2942 | 2.9082 |
| 1.1 | 3.0942 | 0.4521 | 1.3805 | 2.7963 |
| 1.3 | 3.1369 | 0.5173 | 1.4713 | 2.6799 |
| 1.5 | 3.1856 | 0.5758 | 1.5667 | 2.5589 |
| 1.7 | 3.2418 | 0.6274 | 1.6671 | 2.4329 |
| 1.9 | 3.3071 | 0.6716 | 1.7725 | 2.3019 |
| 2.1 | 3.3841 | 0.7082 | 1.8832 | 2.1659 |
| 2.3 | 3.4761 | 0.7369 | 1.9992 | 2.0248 |
| 2.5 | 3.5877 | 0.7576 | 2.1206 | 1.8786 |

Table 3. Variations between various system characteristics and $\lambda_2$.

| $\lambda_2$ | $E[N]$ | $E[Loss]$ | $E[B]$ | $E[V]$ |
|---|---|---|---|---|
| 0.5 | 1.5042 | 1.3806 | 0.4259 | 4.9628 |
| 0.7 | 1.7677 | 1.2362 | 0.6429 | 4.4749 |
| 0.9 | 2.0267 | 1.1080 | 0.8508 | 4.0302 |
| 1.1 | 2.2859 | 0.9921 | 1.0518 | 3.6194 |
| 1.3 | 2.5500 | 0.8857 | 1.2475 | 3.2360 |
| 1.5 | 2.8247 | 0.7870 | 1.4388 | 2.8750 |
| 1.7 | 3.1172 | 0.6946 | 1.6265 | 2.5329 |
| 1.9 | 3.4378 | 0.6074 | 1.8110 | 2.2070 |
| 2.1 | 3.8030 | 0.5246 | 1.9927 | 1.8952 |
| 2.3 | 4.2403 | 0.4456 | 2.1718 | 1.5958 |
| 2.5 | 4.8022 | 0.3700 | 2.3484 | 1.3076 |

Table 4. Variations between various system characteristics and $\mu$. ($\mu_1 = \mu_2 = \mu$)

| $\mu$ | $E[N]$ | $E[Loss]$ | $E[B]$ | $E[V]$ |
|---|---|---|---|---|
| 1.0 | 5.8768 | 0.2769 | 2.9878 | 0.8052 |
| 1.2 | 3.8048 | 0.4644 | 2.3214 | 1.6109 |
| 1.4 | 3.3763 | 0.5985 | 1.8836 | 2.1601 |
| 1.6 | 3.2013 | 0.6945 | 1.5806 | 2.5431 |
| 1.8 | 3.1089 | 0.7647 | 1.3607 | 2.8182 |
| 2.0 | 3.0527 | 0.8171 | 1.1946 | 3.0213 |
| 2.2 | 3.0154 | 0.8570 | 1.0652 | 3.1748 |
| 2.4 | 2.9891 | 0.8880 | 0.9615 | 3.2932 |
| 2.6 | 2.9697 | 0.9124 | 0.8767 | 3.3861 |
| 2.8 | 2.9548 | 0.9319 | 0.8060 | 3.4602 |
| 3.0 | 2.9432 | 0.9477 | 0.7461 | 3.5199 |

Table 5. Variations between various system characteristics and $\eta$. ($\eta_1 = \eta_2 = \eta$)

| $\eta$ | $E[N]$ | $E[Loss]$ | $E[B]$ | $E[V]$ |
|---|---|---|---|---|
| 0.1 | 2.8589 | 0.6630 | 1.6564 | 2.4976 |
| 0.2 | 2.3663 | 0.6950 | 1.5076 | 2.8089 |
| 0.3 | 2.0459 | 0.7362 | 1.3212 | 3.2014 |
| 0.4 | 1.7823 | 0.7878 | 1.0858 | 3.6991 |
| 0.5 | 1.5309 | 0.8488 | 0.7898 | 4.3259 |
| 0.6 | 1.2879 | 0.9009 | 0.4518 | 5.0422 |
| 0.7 | 1.1073 | 0.8938 | 0.1756 | 5.6276 |
| 0.8 | 1.0277 | 0.8127 | 0.0470 | 5.9003 |
| 0.9 | 1.0060 | 0.7058 | 0.0104 | 5.9779 |
| 1.0 | 1.0012 | 0.6055 | 0.0022 | 5.9953 |

Table 6. Variations between various system characteristics and $\theta$.

| $\theta$ | $E[N]$ | $E[Loss]$ | $E[B]$ | $E[V]$ |
|---|---|---|---|---|
| 0.1 | 5.1555 | 0.6649 | 1.7057 | 2.3927 |
| 0.3 | 3.8263 | 0.6551 | 1.7148 | 2.3760 |
| 0.5 | 3.2731 | 0.6504 | 1.7191 | 2.3681 |
| 0.7 | 2.9706 | 0.6477 | 1.7217 | 2.3634 |
| 0.9 | 2.7799 | 0.6459 | 1.7234 | 2.3604 |
| 1.1 | 2.6488 | 0.6446 | 1.7245 | 2.3582 |
| 1.3 | 2.5531 | 0.6437 | 1.7254 | 2.3566 |
| 1.5 | 2.4802 | 0.6429 | 1.7261 | 2.3554 |
| 1.7 | 2.4228 | 0.6424 | 1.7266 | 2.3544 |
| 1.9 | 2.3764 | 0.6419 | 1.7271 | 2.3536 |

Table 7. Variations between various system characteristics and $m_1$.

| $m_1$ | $E[N]$ | $E[Loss]$ | $E[B]$ | $E[V]$ |
|---|---|---|---|---|
| 0 | 2.0731 | 0.8434 | 1.4867 | 6.0157 |
| 1 | 2.1643 | 0.8411 | 1.5074 | 5.8968 |
| 2 | 2.2670 | 0.8401 | 1.5266 | 5.7580 |
| 3 | 2.3937 | 0.8328 | 1.5497 | 5.5715 |
| 4 | 2.5628 | 0.8105 | 1.5825 | 5.2885 |
| 5 | 2.8191 | 0.7565 | 1.6361 | 4.8114 |
| 6 | 3.3150 | 0.6335 | 1.7359 | 3.9245 |
| 7 | 5.3403 | 0.3431 | 1.9466 | 2.0695 |
| 8 | NaN | NaN | NaN | NaN |
| 9 | NaN | NaN | NaN | NaN |
| 10 | NaN | NaN | NaN | NaN |

NaN indicates that the stability condition is not met.

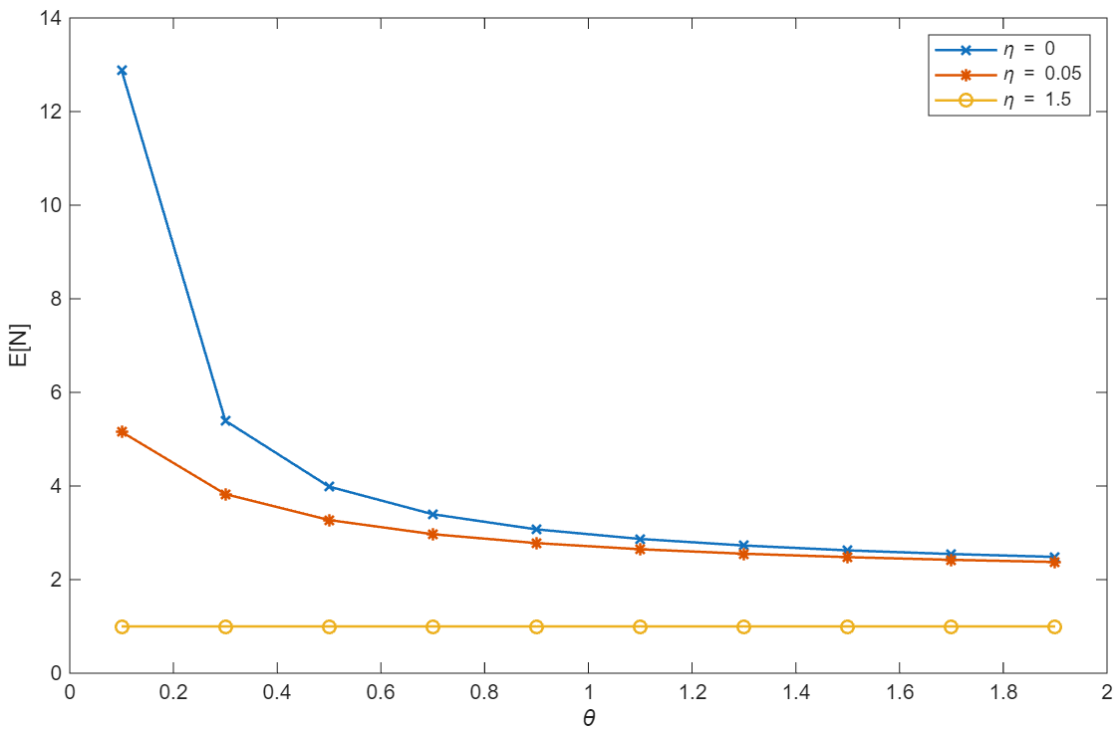


Figure 3. The impact of mean orbit length when $\theta$ vs. $\eta$.

From these results, we have the following observations and insights:

- **Impact of arrival rates:** As $\lambda_1$ increases, $E[N]$ and $E[Loss]$ increase while vacation servers decrease, as shown in Table 2. This occurs because class $P_1$

customers have non-preemptive priority, meaning they occupy servers immediately if available. However, since special servers $m_1$ are fixed, a higher $\lambda_1$ creates a "priority bottleneck" where class $P_1$ customers arrivals exhaust their dedicated capacity faster. Similarly, Table 3 illustrates that an increase in $\lambda_2$ directly inflates the mean orbit length $E[N]$, as more class $P_2$ customers are forced to join the orbit when servers are occupied.

- **Impact of service and vacation parameters:** The mean orbit length $E[N]$ is sensitive to the service rates and vacation durations. As observed in Tables 4 and 5, higher $\mu$ and $\eta$ values effectively reduce the congestion in both the service facility and the orbit. Furthermore, Table 6 indicates that a higher $\theta$ (shorter vacation duration) reduces both $E[N]$ and $E[Loss]$. This highlights the "recovery lag" of the system; increasing $\theta$ accelerates the transition back to the regular busy state, restoring the full-service capacity and clearing the orbital bottleneck more effectively.
- **Impact of special servers $m_1$:** As illustrated in Table 7, the number of special servers $m_1$ plays a critical role in balancing the service quality between the two priority classes. As $m_1$ increases, the mean number of lost class $P_1$ customers ($E[Loss]$) shows a significant downward trend. This is because a larger $m_1$ expands the reserved capacity exclusively available for high-priority traffic, effectively buffering $P_1$ customers against system congestion. Conversely, we observe that the mean orbit length $E[N]$ for $P_2$ customers tends to increase with $m_1$. This phenomenon is driven by the threshold $k = m_1 + 1$; as $m_1$ grows, the entry barrier for $P_2$. Customers becomes more restrictive, forcing more low-priority arrivals into the orbit to avoid interfering with the expanded $P_1$ service zone. These results suggest that $m_1$ is a key tuning parameter for administrators to meet specific service level agreements.

### *4.2. Profit-optimized analysis*

Next, we perform the second experiment to study the expected profit of the proposed parametric model. To evaluate the economic viability of the system, we construct an expected profit function as below. Due to the large number of financial variables involved, their definitions and the specific values used in our numerical experiments are summarized in Table 8. This function accounts for revenues from both customer classes, operational costs of servers in various states (busy, idle or vacation), and specific overheads associated with the priority mechanisms and vacation rates.

$$EP(m_1,\theta) = r_1E[S_1] + r_2E[S_2] - \left(c_hE[N] - c_vE[V] + c_lE[Loss] + c_bE[B] + c_iE[I] + c_1m_1 + c_t\left[m/m_1\right] + c_\theta\theta\right)$$

Table 8. Definition of profit and cost parameters

| Parameter | Description | Default value (Sec. 4.2 / 4.3) |
|---|---|---|
| $r_1$ | Revenue per unit time for serving class $P_1$ customers | 680 |
| $r_2$ | Revenue per unit time for serving class $P_2$ customers | 550 |
| $c_h$ | Holding cost per customer in orbit | 125 |
| $c_l$ | Cost of a lost class $P_1$ customer | 65 |
| $c_b$ | Cost per unit time for a busy server | 75 |
| $c_i$ | Cost per unit time for an idle server | 30 |
| $c_v$ | Cost per unit time for a vacation server | 45 |
| $c_1$ | Cost per unit time to maintain for special servers | 50 |
| $c_t$ | Administrative or overhead cost per unit time for each "team group". The system is divided into $[m/m_1]$ groups | 90 |
| $c_\theta$ | Fixed cost per unit time during a working vacation period | 5 |

The expected profit function $EP(m_1, \theta)$ incorporates a hierarchical operational cost structure. To clarify the terms $c_1 m_1$ and $c_t[m/m_1]$:

- The term $c_1$ represents the fixed cost per unit time for maintaining each special server. Since there are $m_1$ such servers, the total cost is $c_1 m_1$.
- In this model, we assume the $m$ total servers are organized into independent "team group" to ensure manageable supervision. Each group is structured around the capacity of special servers. The total number of such groups is defined as $[m/m_1]$, where $[\cdot]$ denotes the ceiling function, ensuring that even a fractional group is accounted for in administrative overhead. The parameter $c_t$ denotes the administrative or overhead cost per unit time for each group. Therefore, $c_t[m/m_1]$ represents the total organizational cost.

This grouping logic reflects real-world scenarios in call centers or hospital wards, where a large pool of $m$ resources is divided into smaller, manageable units $(m/m_1)$ to ensure that priority tasks are distributed evenly across the facility.

It is not easy to get the optimal parameter values due to the highly nonlinear and complex of the expected profit function. Therefore, we will seek the numerical optimal solution through the genetic algorithm, which is briefly described below, and then present the numerical analysis.

Genetic algorithm, proposed by Holland in 1992, is a search heuristic inspired by the process of natural selection. A major advantage of genetic algorithms is that they do not require derivatives of the error function, which makes them well suited for both continuous and discrete optimization problems. If the criteria are not met, questions are asked to create a cycle to improve the answer. The genetic algorithm initiates a series of operations divided into three main steps: selection, crossover and mutation.

*Crossover*: This operator swaps part of two solutions to produce a new solution.

*Mutation*: This operator flips part of the new solution to produce a new solution and prevents it from converging to a local optimum.

*Selection*: This operator evaluates new solutions according to the fitness function and selects the best candidates.

To solve the non-linear profit function, we utilize a genetic algorithm with the following settings for reproducibility:

- Population size: 100
- Selection: Roulette Wheel Selection
- Crossover rate: 0.8 (swaps parts of two solutions)
- Mutation rate: 0.05 (prevents local optimum convergence)
- Stopping criteria: 500 generations or stall tolerance of $10^{-6}$.

The curve of the profit function for different values of $m_1$ and $\theta$ is shown in Figure 4. The other parameters are $r_1 = 680,\ r_2 = 550,\ c_h = 125,\ c_l = 65,\ c_b = 75,\ c_i = 30,\ c_v = 45,\ c_1 = 50,\ c_t = 90,\ c_\theta = 5,\ m = 12,\ q = 36,\ \lambda_1 = 6,\ \lambda_2 = 8,\ \mu_1 = 10,\ \mu_2 = 15,\ \eta_1 = 0.1,\ \eta_2 = 0.15,\ \alpha = 10,\ \beta = 0.95$ and $\delta = 0.8$. We develop approximations by MATLAB programs to find the optimal parameter values. From Figure 4, there exists an optimal vacation rate and the optimal number of special servers to maximize profit. Implementing the computer software MATLAB by genetic algorithm, we find the solutions $m_1^* = 2$ and $\theta^* = 10.1657$ with $EP(m_1^*, \theta^*) = 3169.5154$. System administrators can control the system parameter values to achieve the desired maximum benefit based on this observation.

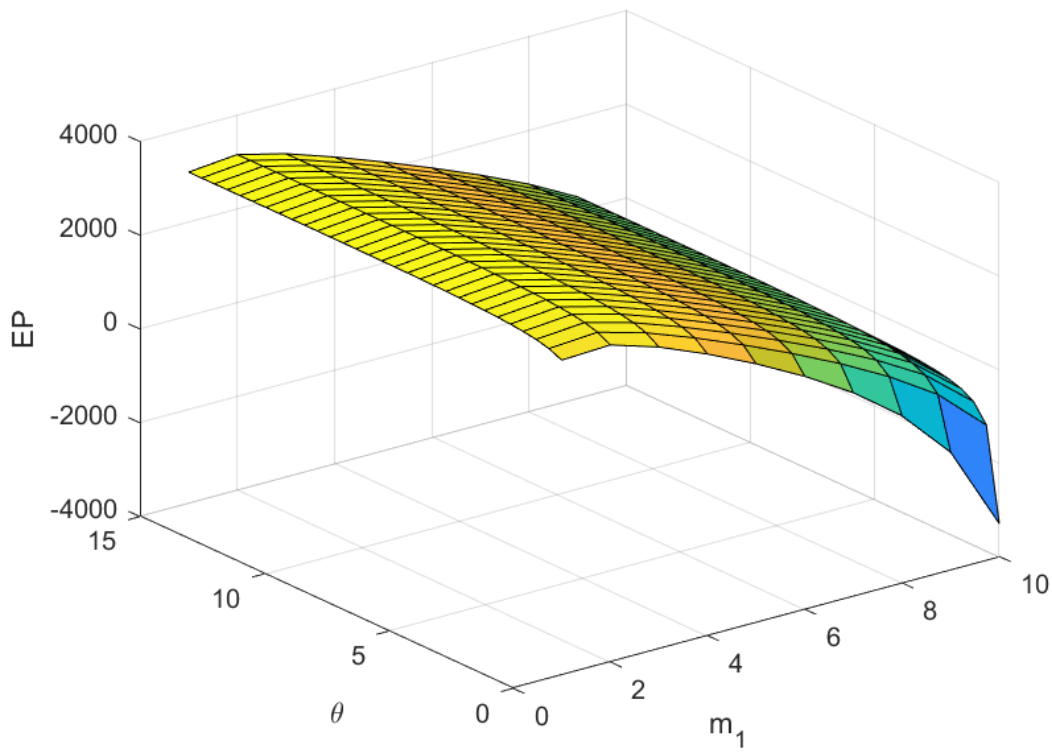


Figure 4. The impact of $\theta$ and $m_1$ on the expected profit.

### *4.3. Comparison analysis*

In practice, treating customers equally, or dividing customers into two classes and formulating different service policies are challenges that managers often face. To demonstrate the economic advantages of the proposed model, we compare system A (one-class, no priority, $m_1 = 0$) and system B (two-class, priority for class $P_1$ customers with special servers). In system A, because there are no class $P_1$ customers, the fixed costs associated with maintaining priority infrastructure ($c_1$) and the administrative overhead of managing divided team groups ($c_t$) do not apply. Further, since system A treats all arrivals as a single class with no loss discipline (all arrivals enter the orbit if servers are busy), the loss rate is zero by definition.

By omitting these costs in system A, we provide a conservative "baseline". This ensures that system B is only considered superior if its increased revenue from class $P_1$ customers significantly outweighs both the operational costs and the additional fixed costs of the priority infrastructure. To make it even clearer for the reader, we add this small comparison table (shown in Table 9).

Table 9. Summary of profit components

| Cost component | System A ($m_1 = 0$) | System B ($m_1 > 0$) |
|---|---|---|
| Orbit holding $c_h E[N]$ | Included | Included |
| $P_1$ Loss cost $c_l E[Loss]$ | N/A | Included |
| Operational (busy/idle/vacation) | Included | Included |
| Priority Infrastructure $c_1 m_1$ | N/A | Included |
| Team management $c_t(m/m_1)$ | N/A | Included |

For the two systems discussed, the effects of parameters such as the revenue per unit time when serving class $P_1$ customers, the arrival rate of class $P_1$ customers and the service rate of class $P_1$ customers on the expected profit function are studied. In addition, the following parameters are considered in the numerical experiments:

$r_1 = 680$, $r_2 = 550$, $c_h = 125$, $c_l = 65$, $c_b = 75$, $c_i = 30$, $c_v = 45$, $c_1 = 50$, $c_t = 90$, $c_\theta = 5$, $m = 10$, $m_1 = 2$, $q = 1$, $\lambda_1 = 4.5$, $\lambda_2 = 6.5$, $\mu_1 = 2$, $\mu_2 = 2$, $\eta_1 = 0.2$, $\eta_2 = 0.2$, $\alpha = 15$, $\theta = 1.5$, $\beta = 0.75$, and $\delta = 0.75$.

From these figures, we have the following key findings:

- System B is more profitable when $r_1 > 642.0528$.
- System B outperforms system A when $\lambda_1 > 4.0995$.
- If the service rate of class $P_1$ customers is less than 1.6703, system B is preferred.

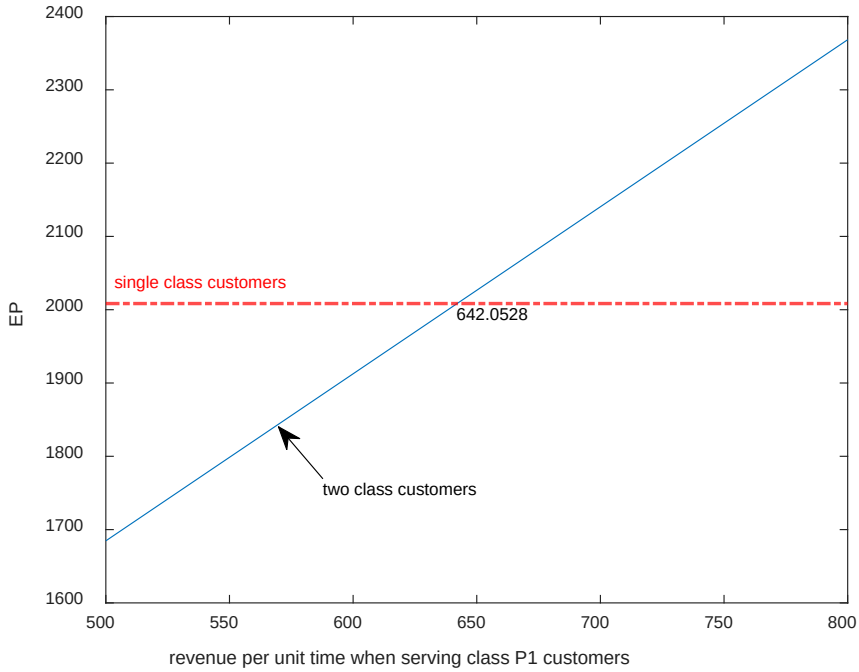


Figure 5. The effect of the revenue per unit time when serving class $P_1$ customers on the expected profit.

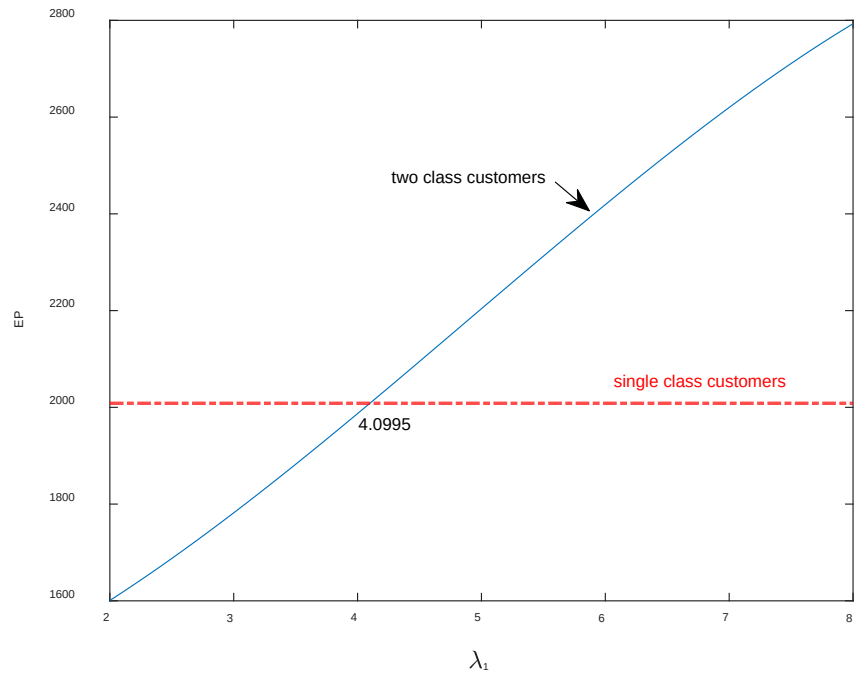


Figure 6. The effect of the arrival rate of class $P_1$ customers on the expected profit.

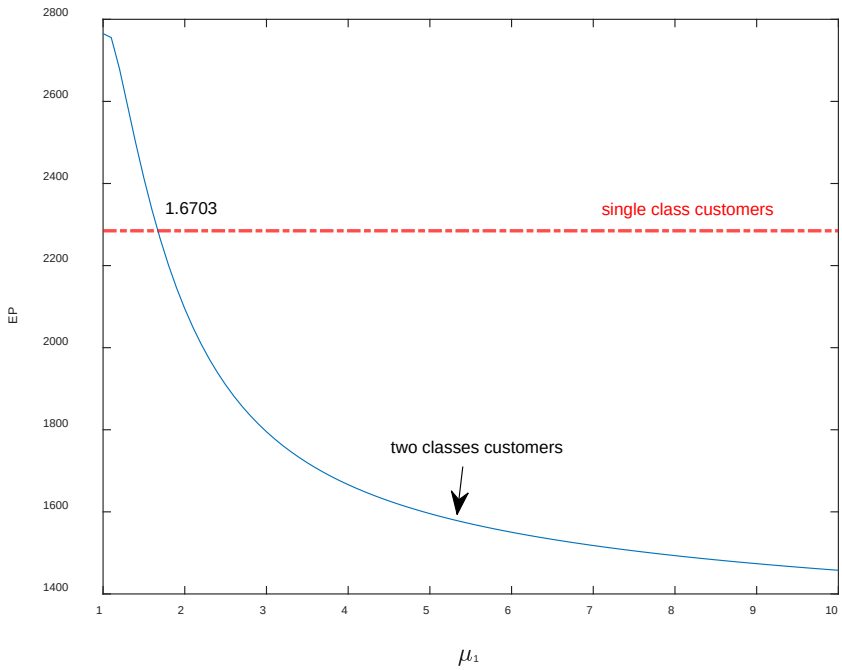


Figure 7. The effect of the service rate of class $P_1$ customers on the expected profit.

## 5. Conclusions

This paper investigates a multi-server non-preemptive priority retrial queue system featuring balking customers, synchronous working vacations, and vacation interruption. Using the matrix-geometric approach, we derived the stationary distribution and key system characteristics, such as mean orbit length, the loss rate of class $P_1$ customers, and various server states.

Based on our numerical and optimization analysis, the following insights are provided for real-world system management:

- Managers should adopt a two-class service policy (system B) when the revenue from high-priority (class $P_1$) customers exceeds a specific threshold (e.g., $r_1 > 642.0528$ in our case study) to maximize expected profit.
- While increasing the number of special servers ($m_1$) effectively reduces the loss of class $P_1$ customers, it simultaneously increases the orbit length of other customers due to reduced general capacity. Managers must use the derived profit function to find the "sweet spot" between customer satisfaction and operational costs.
- The study demonstrates that the vacation rate ($\theta$) significantly impacts system stability. We recommend using the genetic algorithm approach described in this paper to determine the optimal vacation duration, especially in environments with highly non-linear cost structures.
- Managers should monitor arrival rates ($\lambda_1$) and service rates ($\mu_1$) closely; if the arrival rate of priority customers exceeds identified thresholds (e.g., 4.0995), transitioning to a prioritized system is economically essential.

While this model provides a robust tool for decision-making, it assumes ideal server reliability. Future research could extend this work by:

- **Incorporating server breakdowns:** Accounting for service interruptions caused by technical failures or incorrect handling.
- **Diverse distributions:** Exploring the impact of arbitrarily distributed service or retrial times to better reflect stochastic real-world environments.
- **Disaster impact:** Studying how external system shocks or "disasters" influence the stability of the retrial queue or recovery times.

## References

[1] Ammar, S. I., & Rajadurai, P. (2019). Performance analysis of preemptive priority retrial queueing system with disaster under working breakdown services. *Symmetry*, 11(3), 419.

[2] Bouchentouf, A. A., Boualem, M., Yahiaoui, L., & Ahmad, H. (2022). A multi-station unreliable machine model with working vacation policy and customers' impatience. *Quality Technology & Quantitative Management*, 19(6), 766-796.

[3] Bouchentouf, A. A., Yahiaoui, L., & Ziad, I. (2024). Modeling and optimizing an AMS with DV policy, waiting servers, impatient customers, and failures: A queueing analysis. *Results in Control and Optimization*, 15, 100427.

[4] Chakravarthy, S. R., & Kulshrestha, R. (2020). A queueing model with server breakdowns, repairs, vacations, and backup server. *Operations Research Perspectives*, 7, 100131.

[5] Chang, F. M., Liu, T. H., & Ke, J. C. (2018). On an unreliable-server retrial queue with customer feedback and impatience. *Applied Mathematical Modelling*, 55, 171-182.

[6] Damodaran, S., Subramanian, A., & Sekar, G. (2021). Time dependent retrial queueing model with orbital search under non-preemptive priority services. *Journal of Mathematical and Computational Science*, 11, 3276-3299.

[7] Devos, A., Walraevens, J., Phung-Duc, T., & Bruneel, H. (2020). Analysis of the queue lengths in a priority retrial queue with constant retrial policy. *Journal of Industrial and Management Optimization*, 16, 2813-2842.

[8] D'arienzo, M. P., Dudin, A. N., Dudin, S. A., & Manzo, R. (2020). Analysis of a retrial queue with group service of impatient customers. *Journal of Ambient Intelligence and Humanized Computing*, 11, 2591–2599.

[9] Dhibar, S., & Jain, M. (2025). Metaheuristics and strategic behavior of Markovian retrial queue under breakdown, vacation and Bernoulli feedback. *Applied Intelligence*, 55(4), 273.

[10] GnanaSekar, M. M. N., & Kandaiyan, I. (2022). Analysis of an M/G/1 retrial queue with delayed repair and feedback under working vacation policy with impatient customers. *Symmetry*, 14(10), 2024.

[11] Gupta, P., & Kumar, N. (2021). Performance analysis of retrial queueing model with working vacation, interruption, waiting server, breakdown and repair. *Journal of Scientific Research*, 13(3), 833-844.

[12] Gupta, P., & Kumar, N. (2021). Cost optimization of single server retrial queueing model with Bernoulli schedule working vacation, vacation interruption and balking. *Journal of Mathematics and Computer Science*, 11, 2508-2523.

[13] Kalita, P., Choudhury, G., & Selvamuthu, D. (2020). Analysis of single server queue with modified vacation policy. *Methodology and Computing in Applied Probability*, 22, 511-553.

[14] Kannadasan, G., & Sathiyamoorth, N. (2018). The analysis of M/M/1 queue with working vacation in fuzzy environment. *Applications and Applied Mathematics: An International Journal*, 13, 566-577.

[15] Kim, J., & Kim, B. (2016). A survey of retrial queueing systems. *Annals of Operations Research*, 247, 3-36.

[16] Ke, J. C., Chang, F. M., & Liu, T. H. (2019). M/M/c balking retrial queue with vacation. *Quality Technology & Quantitative Management*, 16, 54-66.

[17] Ke, J. C., Wu, C. H., & Zhang, Z. G. (2010). Recent developments in vacation queueing models: a short survey. *International Journal of Operations Research*, 7, 3-8.

[18] Khan, I. E., & Paramasivam, R. (2023). Performance study of an M/M/1 retrial queueing system with balking, dissatisfied customers, and server vacations. *Contemporary Mathematics*, 4, 467-483.

[19] Kumar, B. K., Sankar, R., Krishnan, R. N., & Rukmani, R. (2023). Multiserver call center retrial queue under Bernoulli vacation schedule with two-way communication and orbital search. *Telecommunication Systems*, 84, 23-51.

[20] Krishna Kumar, B., Navaneetha Krishnan, R., Sankar, R., & Rukmani, R. (2023). Performance analysis of cognitive wireless retrial queueing networks with admission control for secondary users. *Quality Technology & Quantitative Management*, 20, 633-670.

[21] Li, T., Zhang, L., & Gao, S. (2019). An M/G/1 retrial queue with balking customers and Bernoulli working vacation interruption. *Quality Technology & Quantitative Management*, 16, 511-530.

[22] Li, J. H., Zhang, Z. G., & Chen, X. (2022). A note on customer joining strategy in a discrete-time Geo/G/1 queue with server vacations – a simple mean value analysis. *Queueing Models and Service Management*, 5(2), 45-61.

[23] Liu, T. H., Zhang, Z. G., & Ke, J. C. (2020). Retrial system with three retrial policies subject to repairable starting failures. *Queueing Models and Service Management*, 3(1), 89-109.

[24] Liu, T. H., Chiou, K. C., Chen, C. M., & Chang, F. M. (2024). Multiserver retrial queue with two-way communication and synchronous working vacation. *Mathematics*, 12, Article 1163.

[25] Melikov, A., Chakravarthy, S. R., & Aliyeva, S. (2023). A retrial queueing model with feedback. *Queueing Models and Service Management*, 6(1), 63-95.

[26] Muthusamy, S., Devadoss, N., & Ammar, S. I. (2022). Reliability and optimization measures of retrial queue with different classes of customers under a working vacation schedule. *Discrete Dynamics in Nature and Society*, 2022, Article 6806104.

[27] Neuts, M. F. (1981). Matrix-geometric Solutions in Stochastic Models: An Algorithmic Approach. Baltimore, The John Hopkins University Press.

[28] Peng, Y., & Wu, J. (2021). Analysis of a batch arrival retrial queue with impatient customers subject to the server disasters. *Journal of Industrial and Management Optimization*, 17, 2243-2264.

[29] Rajadurai, P. (2019). A study on M/G/1 preemptive priority retrial queue with Bernoulli working vacations and vacation interruption. *International Journal of Process Management and Benchmarking*, 9, 193-215.

[30] Shi, X. & Liu, J. (2023). Equilibrium joining strategies in the retrial queue with two classes of customers and delayed vacations. *Methodology and Computing in Applied Probability*, 25, Article 52.

[31] Sundarapandiyan, S., & Nandhini, S. (2024). Sensitivity analysis of a non-Markovian feedback retrial queue, reneging, delayed repair with working vacation subject to server breakdown. *AIMS Mathematics*, 9, 21025-21052.

[32] Tian, N., & Zhang, Z. G. (2006). Vacation queueing models: Theory and Applications, Springer, Boston, MA.

[33] Walraevens, J., Claeys, D., & Phung-Duc, T. (2018). Asymptotics of queue length distribution in priority retrial queues. *Performance Evaluation*, 127-128, 235-252.

[34] Xu, J., Liu, L., & Wu, K. (2023). Analysis of a retrial queueing system with priority service and modified multiple vacations. *Communications in Statistics - Theory and Methods*, 52, 6207-6231.

[35] Yang, D. Y., & Wu, C. H. (2019). Performance analysis and optimization of a retrial queue with working vacations and starting failures. *Mathematical and Computer Modelling of Dynamical Systems*, 25, 463-481.

[36] Zhang, Y. (2020). Strategic behavior in the constant retrial queue with a single vacation. *RAIRO-Operations Research*, 54, 569-583.

[37] Zhang, Y., & Wang, J. (2021). Strategic joining and information disclosing in Markovian queues with an unreliable server and working vacations. *Quality Technology & Quantitative Management*, 18, 298-325.

[38] Zirem, D., Boualem, M., Adel-Aissanou, K., & Aïssani, D. (2019). Analysis of a single server batch arrival unreliable queue with balking and general retrial time. *Quality Technology & Quantitative Management*, 16, 672-695