Palestine Journal of Mathematics

Vol 14(1)(2025) , 606–627

Neuro fuzzy system and metaheuristic cost optimization of a batch arrival retrial queue with variable server model under working vacation

N. Micheal Mathavavisakan and K. Indhira

Communicated by Jaganmohan J

MSC 2010 Classifications: Primary 60K25; Secondary 90B22, 60K30.

Keywords: Retrial queue, feedback customers, working vacation, ANFIS, heuristic algorithms.

The authors would like to thank the reviewers and editor for their constructive comments and valuable suggestions that improved the quality of our paper.

Corresponding Author: K. Indhira

Abstract This article delves into the analysis of a queueing system that operates within the framework of a working vacation policy, incorporating variable server capacity. In this system, the server is capable of serving a batch of customers ranging from one to a maximum of Q customers, based on the variable server capacity bulk service rule. The study employs the supplementary variables approach (SVT) to derive the steady-state probability generating function for both system size and orbit size. The discourse extends to various aspects including system performance metrics, notable special instances, and the examination of system parameter impacts through cost optimization and numerical examples. To facilitate this, a cost function is developed, and its minimization is tackled using optimization techniques such as particle swarm optimization (PSO), artificial bee colony (ABC), and genetic algorithm (GA). The convergence of these optimization methods is explored through illustrative figures. To validate the analytical findings, the study also compares them with results obtained through neuro-fuzzy analysis utilizing an adaptive neuro-fuzzy inference system (ANFIS) grounded in soft computing technology.

1 Introduction

Research into vacation queues (VQs) and retrial queues (RQs) has been ongoing in queueing theory for some time. When a customer arrives and finds the server occupied, they're directed to leave the service area and enter a retry line called the "orbit." In an RQ system, this is known as an RQ with repeated tries. Customers in the orbit can attempt their service request again after some time has elapsed. Importantly, these orbiting customers can repeatedly request the same service without impacting others. Artalejo and Gomez Corral have proposed modified models for RQs [1], while Ke et al. have explored similar modifications for VQs [2]. These queues find distinct applications in computer and communication systems.

In a VQ system, the server operates at a reduced speed during the working vacation (WV) period and completely halts service during regular vacation periods. This queueing system finds various applications including network services, online services, file transfers, and mail services. Researchers such as Gautam Choudhury [3] investigated bulk arrival queues with vacation periods employing a single vacation strategy. Shan Gao [4] explored batch arrival queues with delayed single WV. Chandrasekaran et al. [5] provided a concise overview of WV queueing systems in recent years. Rajadurai [6] developed a unique RQ model incorporating WV and breaks. Exponentially distributed multiple WV and bulk arrival RQs with feedback were studied by Pazhani Bala Murugan and Vijaykrishnaraj [7]. Additionally, Madhu Jain and Anshul Kumar [8] analyzed the $M^{[X]}/G/1$ model with WV, balking, and an unreliable server.

Further, feedback plays a crucial role in communication systems, allowing consumers to retry services if they are dissatisfied. Maragathasundari and Balamurugan [9] investigated the

$M^{[X]}/G/1$ feedback queue, considering two stages of repair times and general delay times. Madhu Jain and Anshul Kumar [10] explored a bulk arrival general service RQ incorporating balking, feedback, and vacation interruptions under a multiple WV policy. GnanaSekar and Kandaiyan [11] analyzed the $M/G/1$ RQ with delayed repair and feedback, considering impatient consumers under a WV policy.

Moreover, bulk queueing systems find wide application across various real-world scenarios such as elevators, freight handling, industrial processes, communication networks, and tourism. Bailey [12] introduced batch service queueing techniques, which have been further investigated by researchers like Sasikala and Indhira [13]. Jaiswal [14] conducted original research on variable server capacity bulk service rules, while Banerjee et al. [15] explored queueing models incorporating variable server capacity and bulk service rules. Recently, Sasikala et al. [16] investigated bulk RQ systems with a Bernoulli vacation schedule and variable server capacity. In the context of working vacation queues for bulk arrival feedback, no previous work has been done. Therefore, our focus lies on batch arrival scenarios, specifically on batch service feedback RQ systems with variable server capacity during working vacations.

“Optimization" refers to the process of finding the best solution for a given fitness function. Cost optimization (CO) is a continuous business-oriented process aimed at reducing expenses while enhancing the overall value of a company. This involves securing the most cost-effective prices and terms for all business transactions, as well as streamlining and standardizing platforms, applications, procedures, and services. In practical terms, the operating costs of a system are directly linked to its profits. Therefore, system developers or administrators strive to minimize the operational expenses per unit of time in order to maximize the financial success of the system.

Nair and Jose [17] investigated solutions for production inventory systems incorporating orbit, buffer, and varying service rates. Jouilik et al. [18] analyzed a numerical optimization algorithm, employing genetic algorithms, to solve an inverse problem related to reconstructing the Robin coefficient in a boundary value problem. Upadhyaya et al. [19] examined an unreliable multi-server retrial queue system with feedback, where clients may resist joining the queue and the service provider's performance is unpredictable. Agarwal et al. [20] delved into the cost optimality of a discrete-time retrial queue with erratic behavior under a J-vacation scheme, using nature-inspired algorithms. Additionally, Agarwal et al. [21] discussed the optimization of a stochastic model featuring an erratic server with immediate or delayed repair. Tajani and Fakhouri [22] conducted a comparative study of various ant colony optimization variants for solving the probabilistic traveling salesman problem.

This research aims to determine the distributions of queue length and orbit length, which are essential for understanding other behavioral metrics of the system. The structure of our article is outlined as follows: We provide a detailed description of the queueing model in Section 2, after meeting the necessary prerequisites. In Section 3, we precisely analyze the system's behavior under steady-state conditions and derive the probability generating function (PGF) of the queue size at a random epoch. Section 4 discusses various crucial system behavior indicators. There are both numerical and pictorial findings in section 5. Results from the neuro-fuzzy analysis of the system and the results of varying the system's parameters are graphically analyzed and discussed in Section 6. Finally, in Section 7, numerical findings and cost analyses are conducted using PSO, ABC, and GA, while Section 8 summarizes the key ideas of the paper.

2 Description of the model and its implementation in real world

Under the working vacation (WV) policy, we implement a $M^{[X]}/G^{Q}/1$ feedback retrial queue (RQ). The detailed rationale for our model is as follows:

The arrival approach: In accordance with the Poisson process, consumers arrive for service at a rate of λ. Additionally, we denote A as the random variable representing batch size, characterized by its probability mass function $P\{A = n\} = a_n$, $n = 1, 2, 3, ...$ probability generating function (PGF) $A(z) = \sum_{n=0}^{\infty} z^n a_n$ and mean batch size $E(I)$.

The retrial approach: We presume there is no waiting space; therefore if a consumer arrives and finds the server empty, the consumer immediately begins his service. However, if a consumer arrives and the server busy, on vacation, then the consumer has two options: they can either depart

the service area with probability $\bar{b} = (1-b)$ and enroll in a group of blocked consumers who have been blocked, known as a “orbit” or balk the system with probability b. Inter retrial times have a random distribution, $R(x)$ with corresponding “Laplace-Stieltjes Transform" (LST) $R^*(\delta)$.
The regular service approach: Following the variable server capacity bulk service policy, the server handles consumer transmissions. Under the variable server capacity batch service rule, the server's action depends on the situation: it either serves a fixed size, denoted as "Q," or serves all consumers from the orbit, whichever is fewer. If there are "Q" or more consumers in the orbit after transmitting a group, the server sends "Q" packets in a batch. Conversely, if fewer than "Q" packets remain in the orbit after transmission, the server sends all remaining consumers in a batch. Once service begins, latecomers cannot join the ongoing service, even if the batch size is less than "Q." The service duration follows a general distribution, represented by the arbitrary variable H with distribution function $H(x)$ and Laplace-Stieltjes transform $H^*(\delta)$.
Feedback rule: Unsatisfied consumers have the option to re-enter the orbit as feedback consumers once their normal service is complete in order to maybe receive another service with probability, α $(0 \leq \alpha \leq 1)$ will exit the system with probability, $\bar{\alpha} = (1-\alpha)$.
The working vacation policy: When the orbit is free, the server periodically takes a WV. The vacation period takes an exponential distribution with variable ϑ. If a consumer enters during a vacation time, the server keeps running at a reduced rate. During the WV time, tasks are carried out at a slower pace. If any consumers are in the orbit at a slower service completion moment during the vacation period, the server will end the vacation and return to the normal busy time, interrupting the vacation. If not, the vacation, keeps going. When the vacation gets over, the server restores normal operations if there are still customers in the orbit. During the WV period, the service period is assessed by a random variable H_v with distribution function $H_v(x)$ and LST $H_v^*(\delta)$.
The system's stochastic processes are considered to be independent of one another.

2.1 Practical application of the model

The proposed paradigm is applicable to established airline booking processes. In order to sell seats on their flights, airlines use systems called airline reservation systems (ARS). It includes a database of reservations (or passenger name records) and tickets issued, as well as information on schedules and fares (if applicable). Passenger service systems (PSS) include ARSs, which are used to facilitate communication with passengers. In time, ARS developed into what is currently known as the “electronic reservation system" (ERS). One airline's computer reservation system interfaces with a global distribution system (GDS) that allows travel agencies and other distribution channels to make reservations with the world's major airlines through a single system. Airline reservation systems incorporate airline schedules, fare tariffs, passenger reservations, and ticket records. Direct distribution allows an airline to distribute information both internally and externally (via the GDS). Customers who book directly through a website or mobile app constitute the second type of direct distribution channel.
We consider the ARS system as well as the booking portal (server). The customer tries to book tickets through the website or mobile apps, but there will be a restriction on the number of seats for a journey (variable server capacity). If a group of people is trying to book tickets, the customer who enters the booking portal first will book the tickets and leave the portal. While booking the tickets, if the server is busy, the customer will wait on the website or homepage and try again after some time (retrial). If the system is affected by a virus or the internet source is poor, then the system will give the service at a slower rate (working vacation). If the problem is rectified, then the server will be back to its normal busy mode. Furthermore, the unsatisfied customer (person who got the service and wants to book an additional ticket) may re-enter the orbit when other customers' bookings are completed. This is done to reduce the booking portal's idle time.

3 Overview of steady state probabilities

This division first develops the steady-state (SS) equations for the RQ system by considering the elapsed retrial period, the elapsed service time and the elapsed lower-speed service times as supplementary variable (SV). The PGF of the number of consumers in the orbit and system, as well as the orbit length generating functions for numerous server states, are computed.

3.1 Probabilities and Notations

It is assumed in SS that $R(0) = 0$, $R(\infty) = 1$, $H(0) = 0$, $H(\infty) = 1$ and $H_v(0) = 0$, $H_v(\infty) = 1$ are continuous at $x = 0$. So that the functions $\eta(x)$, $\gamma_b(x)$, $\gamma_v(x)$ are the hazard rates for retrial, service and slower pace service respectively.

Apart from it, let $R^0(t)$, $H^0(t)$ and $H_v^0(t)$ be the elapsed retrial period, the elapsed period of normal service and the elapsed slower-rate service period respectively at time t. Additionally, generate the random variable,

$$\Pi(t) = \begin{cases} 0, & \text{if the server is available and in WV time} \\ 1, & \text{if the server is available and in normal service time} \\ 2, & \text{if the server is unavailable and in normal service at time } t \\ 3, & \text{if the server is unavailable and in lower speed rate at time } t \end{cases}$$

Here, we highlight the usage of bivariate Markov process to describe the system's state at time $\{\Pi(t), \Gamma(t);\, t \geq 0\}$, where $\Pi(t)$ signifies the server state $(0, 1, 2, 3)$ depending on whether the server is free or busy on both normal service and WV periods. $\Gamma(t)$ denotes the number of consumers in the orbit. If $\Pi(t) = 1$ and $\Gamma(t) > 0$, then $R^0(t)$ is equivalent to the elapsed retrial time. If $\Pi(t) = 2$ and $\Gamma(t) \geq 0$, then $H^0(t)$ is equivalent to the elapsed time of the consumer served in normal busy period. If $\Pi(t) = 3$ and $\Gamma(t) \geq 0$, then $H_v^0(t)$ is equivalent to the elapsed time of the consumer being served in lower rate service period.

3.2 Ergodicity analysis of the model

We examine the embedded Markov chain's ergodicity during the departure and vacation epochs. Let $\{t_n; n = 1, 2, ...\}$ be the series of epochs where either a service period completion or a shorter service period happens. $G_n = \{\Pi(t_n+), \Gamma(t_n+)\}$ sequence of random vectors. The Markov chain formed by embedded in the RQ system. It follows from Appendix A that is the embedded Markov chain $\{G_n; n \in N\}$ is ergodic iff $\Lambda < Q$ for our system will be stable.
For the method $\{\Gamma(t), t \geq 0\}$, we specify the probabilities $P_0(t) = P\{\Pi(t) = 0, \Gamma(t) = 0\}$ and the probability densities are
$P_n(x,t)dx = P\{\Pi(t) = 1, \Gamma(t) = n, x \leq R^0(t) < x + dx\}$,
for $t \geq 0$, $x \geq 0$ and $n \geq 1$.
$S_n(x,t)dx = P\{\Pi(t) = 2, \Gamma(t) = n, x \leq H^0(t) < x + dx\}$,
for $t \geq 0$, $x \geq 0$ and $n \geq 0$.
$V_n(x,t)dx = P\{\Pi(t) = 3, \Gamma(t) = n, x \leq H_v^0(t) < x + dx\}$,
for $t \geq 0$, $x \geq 0$ and $n \geq 0$.
We presume that the stability requirement is satisfied in the sequel, so we may assign $P_0 = lim_{t\to\infty} P_0(t)$ and limiting densities are
$P_n(x) = lim_{t\to\infty} P_n(x,t)$; $S_n(x) = lim_{t\to\infty} S_n(x,t)$;
$V_n(x) = lim_{t\to\infty} V_n(x,t)$;
Using the supplementary variable method, we build the following system of equations.

$$\lambda P_0 = \bar{\alpha} \int_0^\infty S_0(x)\gamma_b(x)dx + \bar{\alpha} \int_0^\infty V_0(x)\gamma_v(x)dx + \lambda \int_0^\infty S_n(x)dx, n \geq 0 \tag{3.1}$$

$$\frac{d}{dx}P_n(x) + (\lambda + \eta(x))P_n(x) = 0, n \geq 1 \tag{3.2}$$

$$\frac{d}{dx}S_0(x) + (\lambda + \gamma_b(x))S_0(x) = \lambda\bar{b}S_0 f_k(x), n = 0 \tag{3.3}$$

$$\frac{d}{dx}S_n(x) + (\lambda + \gamma_b(x))S_n(x) = \lambda b\sum_{k=1}^{n} S_{n-k}f_k(x) + \lambda\bar{b}S_n f_k(x), n \geq 1 \tag{3.4}$$

$$\frac{d}{dx}V_0(x) + (\lambda + \vartheta + \gamma_v(x))V_0(x) = \lambda\bar{b}V_0 f_k(x), n = 0 \tag{3.5}$$

$$\frac{d}{dx}V_n(x) + (\lambda + \vartheta + \gamma_v(x))V_n(x) = \lambda b\sum_{k=1}^{n} V_{n-k}f_k(x) + \lambda\bar{b}V_n f_k(x), n \geq 0 \tag{3.6}$$

At $x = 0$ the SS boundary criteria are as follows:

$$\begin{aligned} P_n(0) &= \alpha\int_0^\infty S_n(x)\gamma_b(x)dx + \bar{\alpha}\int_0^\infty S_{n-1}(x)\gamma_b(x)dx \\ &+ \alpha\int_0^\infty V_n(x)\gamma_v(x)dx + \bar{\alpha}\int_0^\infty V_{n-1}(x)\gamma_v(x)dx, n \geq 1 \end{aligned} \tag{3.7}$$

$$\begin{aligned} S_n(0) &= \int_0^\infty P_{n+Q}(x)\eta(x)dx + \lambda\int_0^\infty \sum_{k=1}^{\infty} a_k P_{n-k+Q}(x)dx \\ &+ \vartheta\int_0^\infty V_n(x)dx, n \geq 1 \end{aligned} \tag{3.8}$$

$$S_0(0) = \int_0^\infty \sum_{n=1}^{Q} P_n(x)\eta(x)dx + \lambda\sum_{k=1}^{Q} a_k P_0 + \vartheta\int_0^\infty V_0(x)dx, n = 0 \tag{3.9}$$

$$V_n(0) = \begin{cases} \lambda P_0, & n = 0 \\ 0, & n \geq 1 \end{cases} \tag{3.10}$$

The normalizing criteria is

$$P_0 + \sum_{n=1}^{\infty}\int_0^\infty P_n(x)dx + \sum_{n=0}^{\infty}\left(\int_0^\infty S_n(x)dx + \int_0^\infty V_n(x)dx\right) = 1 \tag{3.11}$$

3.3 The steady state solution

The generating functions for $|\ z\ |< 1$ in order to solve the aforementioned equations, are expressed in the form.

$$\begin{aligned} P(x,z) &= \sum_{n=1}^{\infty} P_n(x)z^n; P(0,z) = \sum_{n=1}^{\infty} P_n(0)z^n; \\ S(x,z) &= \sum_{n=0}^{\infty} S_n(x)z^n; S(0,z) = \sum_{n=0}^{\infty} S_n(0)z^n; \\ V(x,z) &= \sum_{n=0}^{\infty} V_n(x)z^n; V(0,z) = \sum_{n=0}^{\infty} V_n(0)z^n. \end{aligned}$$

Now multiply the SS equation and SS boundary criteria from (3.2) to (3.10) by z^n and summing over n, $(n = 0, 1, 2, ...)$

$$\frac{\partial}{\partial x}P(x,z) + (\lambda + \eta(x))P(x,z) = 0 \tag{3.12}$$

$$\frac{\partial}{\partial x}S(x,z) + (\lambda b(1 - A(z)) + \gamma_b(x))S(x,z) = 0 \tag{3.13}$$

$$\frac{\partial}{\partial x}V(x,z) + (\lambda b(1 - A(z)) + \vartheta + \gamma_v(x))V(x,z) = 0 \tag{3.14}$$

$$P(0,z) = (\alpha + \bar{\alpha}z)\int_0^\infty S(x,z)\gamma_b(x)dx + (\alpha + \bar{\alpha}z)\int_0^\infty V(x,z)\gamma_v(x)dx - \lambda P_0 \tag{3.15}$$

$$S(0,z) = \frac{1}{z^Q}\int_0^\infty P(x,z)\eta(x)dx + \frac{\lambda A(z)}{z^Q}\int_0^\infty P(x,z)dx + \vartheta\int_0^\infty V(x,z)dx \tag{3.16}$$

$$V(0,z) = \lambda P_0 \tag{3.17}$$

Solving the partial differential eqns. (3.12) to (3.14), we obtain

$$P(x,z) = P(0,z)[1 - R(x)]e^{-\lambda x} \tag{3.18}$$

$$S(x,z) = S(0,z)[1 - H(x)]e^{-S(z)x} \tag{3.19}$$

$$V(x,z) = V(0,z)[1 - H_v(x)]e^{-S_v(z)x} \tag{3.20}$$

where $S(z) = \lambda b(1 - A(z))$, and $S_v(z) = \vartheta + \lambda b(1 - A(z))$
Inserting the eqns. (3.17) to (3.20) in (3.8) after some computation, we eventually arrive to,

$$S(0,z) = \frac{P(0,z)}{z^Q}\{R^*(\lambda) + A(z)[1 - R^*(\lambda)]\} + \lambda P_0 W(z) \tag{3.21}$$

where $W(z) = \frac{\vartheta}{\vartheta+\lambda b(1-A(z))}\,(1 - H_v^*(S_v(z)))$,

$$P(0,z) = (\alpha + \bar{\alpha}z)S(0,z)H^*(S(z)) + (\alpha + \bar{\alpha}z)V(0,z)H_v^*(S_v(z)) - \lambda P_0 \tag{3.22}$$

Combining (3.10) and (3.21) in (3.22), we get

$$\begin{aligned} &S(0,z)\{z^Q - (\alpha + \bar{\alpha}z)[R^*(\lambda) + A(z)(1 - R^*(\lambda))]H^*(S(z))\} \\ &= \lambda P_0\{z^Q W(z) + [(\alpha + \bar{\alpha}z)H_v^*(S_v(z)) - 1][R^*(\lambda) + A(z)(1 - R^*(\lambda))]\} \end{aligned} \tag{3.23}$$

In the following theorem, we are willing to exploring the marginal orbit size distributions caused by the server's system state.

Theorem 3.1. *Under the stability requirement, $\Lambda < Q$ provides the stationary distribution, of the number of customers in the orbit when the server is available, busy, reduced rate service, and the probability, that the server is available given by,*

$$P(z) = \frac{Ne(z)}{De(z)} \tag{3.24}$$

$$\begin{aligned} Ne(z) =& z^Q P_0(1 - R^*(\lambda))\{(\alpha + \bar{\alpha}z)[H^*(S(z))W(z) + H_v^*(S_v(z))] - 1\} \\ De(z) =& z^Q - (\alpha + \bar{\alpha}z)\{R^*(\lambda) + A(z)[1 - R^*(\lambda)]\}H^*(S(z)) \end{aligned}$$

$$S(z) = \frac{\lambda P_0(1 - H^*(S(z)))}{S(z)De(z)}\{z^Q W(z) + [(\alpha + \bar{\alpha}z)H_v^*(S_v(z)) - 1][R^*(\lambda) + A(z)[1 - R^*(\lambda)]]\} \tag{3.25}$$

$$V(z) = \frac{\lambda P_0}{\vartheta}W(z) \tag{3.26}$$

where

$$P_0 = \frac{Q - \{\bar{\alpha} - \lambda b E(I)E(H) + E(I)(1 - R^*(\lambda))\}}{\begin{aligned}&\lambda b E(I)E(H)\{\tfrac{2\lambda}{\vartheta}(1 - H_v^*(\vartheta)) - 2H_v^*(\vartheta)R^*(\lambda) + H_v^*(\vartheta) + R^*(\lambda) + 1\} \\ &- E(I)(1 - R^*(\lambda))[1 + \lambda E(H_v)] - \lambda E(H)[1 + B(1 - H_v^*(\vartheta))] + B(1 + \tfrac{\lambda}{\vartheta}(1 - H_v^*(\vartheta)))\end{aligned}} \tag{3.27}$$

Proof. Taking the eqns. (3.18)-(3.20) and integrate with respect to x and compute the probability generating function $P(z) = \int_0^\infty P(x,z)dx$, $S(z) = \int_0^\infty S(x,z)dx$, $V(z) = \int_0^\infty V(x,z)dx$. We calculate the probability that the server is empty using the normalization condition (P_0) by establishing functions as, when there is no consumer in the orbit $z = 1$ in (3.24)-(3.26) and whenever the condition of L'Hospital is needed, we get $P_0 + P(1) + S(1) + V(1) = 1$. □

Theorem 3.2. *Utilizing the PGF function, the number of consumers in the system and the orbit size distribution at a stationary point of period are calculated under the stability constraint* $\Lambda < Q$,

$$K_s(z) = \frac{Ne_s(z)}{De_s(z)} \tag{3.28}$$

$$\begin{aligned}
Ne_s(z) =& P_0\{S(z)\{z^Q - (\alpha + \bar{\alpha}z)\{R^*(\lambda) + A(z)[1 - R^*(\lambda)]\}H^*(S(z))\} \\
& [1 + \frac{\lambda}{\vartheta} zW(z)]\} + z^Q S(z)(1 - R^*(\lambda))\{(\alpha + \bar{\alpha}z)[H^*(S(z))W(z) + H_v^*(S_v(z))] - 1\} \\
& + z\lambda(1 - H^*(\lambda(1 - A(z))))\{z^Q W(z) + [(\alpha + \bar{\alpha}z)H_v^*(S_v(z)) - 1] \\
& [R^*(\lambda) + A(z)[1 - R^*(\lambda)]]\}\} \\
De_s(z) =& S(z)\{z^Q - (\alpha + \bar{\alpha}z)\{R^*(\lambda) + A(z)[1 - R^*(\lambda)]\}H^*(S(z))\}
\end{aligned}$$

$$K_o(z) = \frac{Ne_o(z)}{De_s(z)} \tag{3.29}$$

$$\begin{aligned}
Ne_o(z) =& P_0\{S(z)\{z^Q - (\alpha + \bar{\alpha}z)\{R^*(\lambda) + A(z)[1 - R^*(\lambda)]\}H^*(S(z))\} \\
& [1 + \frac{\lambda}{\vartheta} W(z)]\} + z^Q S(z)(1 - R^*(\lambda))\{(\alpha + \bar{\alpha}z)[H^*(S(z))W(z) + H_v^*(S_v(z))] - 1\} \\
& + \lambda(1 - H^*(\lambda(1 - A(z))))\{z^Q W(z) + [(\alpha + \bar{\alpha}z)H_v^*(S_v(z)) - 1] \\
& [R^*(\lambda) + A(z)[1 - R^*(\lambda)]]\}
\end{aligned}$$

where P_0 *is denoted by eqn.* (3.27).

Proof. The PGF of the number of consumer in the system $(K_s(z))$ and in the orbit $(K_o(z))$ is calculated by using $K_s(z) = P_0 + P(z) + S(z) + V(z)$. The eqns. (3.28) and (3.29) may be derived directly when the eqns. (3.24)-(3.27) are substituted in the earlier results. □

4 System performance measures

In this section, different system states are used to derive a number of pertinent system probabilities, system efficiency metrics, and the model's mean busy time and mean busy cycle.

4.1 Probabilities of system states

Utilizing eqns, (3.24)-(3.26) we obtain the findings shown below, giving $z \to 1$ and, if feasible, using L'Hospital's rule.

(i) Let P be the SS probability of the server is available during the retrial,

$$P = P(1) = P_0(1 - R^*(\lambda)) \left\{ \frac{\bar{\alpha} + \lambda b E(I)[E(H)H_v^*(\vartheta) + \frac{1}{\vartheta}(1 - H_v^*(\vartheta)) - E(D_v)]}{Q - \{\bar{\alpha} - \lambda b E(I)E(H) + E(I)(1 - R^*(\lambda))\}} \right\} \tag{4.1}$$

(ii) Let S be the SS probability that the server is full,

$$S = S(1) = \lambda E(H) P_0 \left\{ \frac{E(I)(1 - H_v^*(\vartheta))[R^*(\lambda) + \frac{\lambda}{\vartheta}] + (\bar{\alpha} - Q)H_v^*(\vartheta) + Q - 1}{Q - \{\bar{\alpha} - \lambda b E(I)E(H) + E(I)(1 - R^*(\lambda))\}} \right\} \tag{4.2}$$

(iii) Let V be the SS probability that the server is on WV,

$$V = V(1) = \frac{\lambda P_0}{\vartheta}[1 - H_v^*(\vartheta)] \tag{4.3}$$

4.2 Mean size of a system and orbit

When the system is in a steady state,
(i) With respect to z, (3.29) and providing $z = 1$ yields the mean number of consumers in the orbit (L_q)

$$L_q = K_0'(1) = \lim_{z\to 1} \frac{d}{dz} K_o(z) = P_0 \left[\frac{N_q'''(1)D_q''(1) - D_q'''(1)N_q''(1)}{3(D_q''(1))^2} \right] \tag{4.4}$$

$$\begin{aligned}
N_q''(1) =& -2\lambda bE(I)\{[1+\frac{\lambda}{\vartheta}(1-H_v^*(\vartheta))][Q-\bar{\alpha}+\lambda bE(I)E(H)-E(I)(1-R^*(\lambda))]\\
&+(1-R^*(\lambda))\{\bar{\alpha}+\lambda bE(I)[E(H)H_v^*(\vartheta)+\frac{1}{\vartheta}(1-H_v^*(\vartheta))-E(H_v)]\}\\
&-\lambda E(H)\{E(I)(1-H_v^*(\vartheta))[R^*(\lambda)+\frac{\lambda}{\vartheta}]+(\bar{\alpha}-Q)H_v^*(\vartheta)+Q-1\}\}\\
D_q''(1) =& -2\lambda bE(I)\{Q+\bar{\alpha}-E(I)(1-R^*(\lambda))+\lambda bE(I)E(H)\}\\
N_q'''(1) =& -6\lambda bE(I)[Q-\bar{\alpha}+\lambda bE(I)E(H)-E(I)(1-R^*(\lambda))]\{\frac{\lambda}{\vartheta}E(I)(1+\vartheta E(H)\\
&-H_v^*(\vartheta))\}+D_q'''(1)[1+\frac{\lambda}{\vartheta}(1-H_v^*(\vartheta))]-3\lambda bE(I)(1-R^*(\lambda))\\
&\{\bar{\alpha}+\lambda bE(I)(1+Q)[\frac{1}{\vartheta}(1-H_v^*(\vartheta))+E(H)H_v^*(\vartheta)-E(H_v)]\\
&+2\bar{\alpha}\{[\frac{\lambda}{\vartheta}E(I)(1+\vartheta E(H_v)-H_v^*(\vartheta))]-\lambda bE(I)E(H)(1-H_v^*(\vartheta))\\
&-\lambda bE(I)E(H_v)\}-2\lambda bE(I)E(H)[\frac{\lambda}{\vartheta}E(I)(1+\vartheta E(H)-H_v^*(\vartheta))]+(1-H_v^*(\vartheta))\\
&[\lambda^2E(I)E^2(H)-\lambda E(I(I-1))E(H)]-\lambda E(I(I-1))E(H_v)-\lambda^2E(I)E^2(D_v)\\
&+W''(1)+3\lambda\{\lambda bE(I)E(H)\{Q(Q-1)(1-H_v^*(\vartheta))+(Q+1)[\frac{\lambda}{\vartheta}E(I)\\
&(1+\vartheta E(H)-H_v^*(\vartheta))]+2\lambda bE(I)(1-R^*(\lambda))[\bar{\alpha}-\lambda bE(I)E(H_v)]\\
&-2\bar{\alpha}\lambda bE(I)E(H_v)+\lambda^2E(I)E^2(H_v)-\lambda E(I(I-1))E(H_v)+W''(1)\}\\
&+\lambda[E(I(I-1))E(H)-\lambda bE(I)E^2(H)]\{\bar{\alpha}+\lambda bE(I)[E(H)H_v^*(\vartheta)\\
&+\frac{1}{\vartheta}(1-H_v^*(\vartheta))-E(H_v)]\}\}\}\\
D_q'''(1) =& -3\lambda bE(I)\{Q(Q-1)-E(I(I-1))(1-R^*(\lambda))+2\{\lambda\bar{\alpha}E(I)E(H)\\
&+\lambda E(I(I-1))E(H)+E(I)(1-R^*(\lambda))[\lambda bE(I)E(H)+\bar{\alpha}]\}\}
\end{aligned}$$

where $W''(1) = \frac{\lambda}{\vartheta}E(I(I-1))[1+\vartheta E(H_v)-H_v^*(\vartheta)] + \frac{E(I)}{\vartheta^3}\{\vartheta^2E^2(H_v)-2\lambda\vartheta E(I)E(H_v)+\lambda bE(I)E(H_v)\}+\lambda bE(I)(1-H_v^*(\vartheta))$
(ii) With regard to z, (3.28) and providing $z = 1$ yields the mean number of consumers in the system (L_s)

$$L_s = K_s'(1) = \lim_{z\to 1} \frac{d}{dz} K_s(z) = P_0 \left[\frac{N_s'''(1)D_q''(1) - D_q'''(1)N_q''(1)}{3(D_q''(1))^2} \right] \tag{4.5}$$

$$\begin{aligned}
N_s'''(1) =& N_q'''(1)+6\lambda bE(I)\{E(H)\{E(I)(1-H_v^*(\vartheta))[R^*(\lambda)+\frac{\lambda}{\vartheta}]+(\bar{\alpha}-Q)H_v^*(\vartheta)\\
&+Q-1\}-\frac{\lambda}{\vartheta}[1-H_v^*(\vartheta)]\{Q-\bar{\alpha}+\lambda bE(I)E(H)-(1-R^*(\lambda))\}\}
\end{aligned}$$

(iii) The mean period of the consumers in the system (W_s) and the mean period of the consumers in the queue (W_q) are estimated utilizing Little's method. $W_s = \frac{L_s}{\lambda b E(I)}$ and $W_q = \frac{L_q}{\lambda b E(I)}$, respectively.

4.3 Special cases

In this section, we examine a few real-world examples of our strategy that are consistent with recent literature.

Case (i):

Let $Pr[A = 1] = 1$, $Q = 1$, $\vartheta, \bar{\alpha} = 0$, $b = 0$ and $R^*(\lambda) \to 1$. Our model can be simplified to a $M/G/1$ queue. The results agree with Takagi [26].

$$K_s(z) = P_0 \left\{ \frac{Ne_s(z)}{De_s(z)} \right\} \tag{4.6}$$

$$\begin{aligned} Ne_s(z) =& (1-z)\{z - H^*(\lambda(1-z))\} + z(1 - H^*(\lambda(1-z)))\{H_v^*(\lambda(1-z))\} \\ De_s(z) =& (1-z)\{z - H^*(\lambda(1-z))\} \\ \text{where,} \quad P_0 =& \frac{1 + \lambda E(I)E(H)}{\lambda b E(I)E(H) - \lambda E(H)} \end{aligned}$$

Case (ii):

Let $Pr[A = 1] = 1$, $Q = 1$, $b = 0$ and $\vartheta, \bar{\alpha} = 0$. Our model simplified to an $M/G/1$ RQ. Here are the results agree with Gao and Wang [23].

$$K_s(z) = P_0 \left\{ \frac{Ne_s(z)}{De_s(z)} \right\} \tag{4.7}$$

$$\begin{aligned} Ne_s(z) =& (1-z)\{z - [R^*(\lambda) + z(1 - R^*(\lambda))]H^*(\lambda(1-z))\} + z\lambda(1-z)[1 - R^*(\lambda)] \\ & (H_v^*(\lambda(1-z) - 1)) + \lambda z[1 - H^*(\lambda(1-z))]\{H_v^*(\lambda(1-z) - 1) \\ & [R^*(\lambda) + z(1 - R^*(\lambda))]\} \\ De_s(z) =& \lambda(1-z)\{z - [R^*(\lambda) + z(1 - R^*(\lambda))]H^*(\lambda(1-z))\} \\ \text{where,} \quad P_0 =& \frac{1 + \lambda b E(I)E(H) + E(I)(1 - R^*(\lambda))}{\lambda b E(I)E(H)\{1 - R^*(\lambda)\} - \lambda E(H) - E(I)(1 - R^*(\lambda))[1 + \lambda E(H_v)] + Q} \end{aligned}$$

Case (iii):

Let $Pr[A = 1] = 1$, $Q = 1$, $b = 0$ and $\bar{\alpha} = 0$. our model simplified to an $M/G/1$ queue with WVs. Here are the results agree with Zhang and Hou [28].

$$K_s(z) = P_0 \left\{ \frac{Ne_s(z)}{De_s(z)} \right\} \tag{4.8}$$

$$\begin{aligned} Ne_s(z) =& (1-z)\{z - [R^*(\lambda) + z(1 - R^*(\lambda))]H^*(\lambda(1-z))\} + z\lambda(1-z)[1 - R^*(\lambda)] \\ & (H_v^*(\lambda(1-z) - 1)) + \lambda z[1 - H^*(\lambda(1-z))]\{H_v^*(\lambda(1-z) - 1) \\ & [R^*(\lambda) + z(1 - R^*(\lambda))]\} \\ De_s(z) =& \lambda(1-z)\{z - [R^*(\lambda) + z(1 - R^*(\lambda))]H^*(\lambda(1-z))\} \end{aligned}$$

5 Numerical results

The various effects on system performance measurements are demonstrated using MATLAB in this section. We examine exponentially distributed retrial times, service times, and slower service times. The numerical measurements that satisfy the stability condition are chosen at random. Table 1 clearly displays that arrival rate (λ) escalates, L_q, L_s, $V(1)$ are increases. Table 2

Table 1. P_0 and L_q for different arrival rate (λ) for the values of $Q = 30$, $\alpha = 0.5$, $\vartheta = 2$, $b = 0.2$, $\eta(x) = 6$, $\gamma_b(x) = 0.6$, $\gamma_v(x) = 0.7$

Arrival rate (λ)	P_0	L_q	L_s	$V(1)$	W_q
1.5	0.7495	0.0020	0.0007	0.2274	0.0040
2.5	0.7606	0.0031	0.0018	0.3582	0.0059
3.5	0.7704	0.0042	0.0029	0.4917	0.0061
4.5	0.7789	0.0061	0.0036	0.6273	0.0072
5.5	0.7859	0.0079	0.0047	0.7644	0.0085
6.5	0.7914	0.0091	0.0058	0.9023	0.0090
7.5	0.7954	0.0097	0.0068	0.9782	0.0142

Table 2. P_0 and L_q for different feedback rate (λ) for the values of $Q = 30$, $\lambda = 5$, $b = 0.2$, $\vartheta = 4$, $\eta(x) = 6$, $\gamma_b(x) = 0.6$, $\gamma_v(x) = 0.5$

Feedback rate (α)	P_0	L_q	L_s	$V(1)$	W_q
2.1	3.8675	0.0305	0.0182	4.0421	0.0386
3.1	3.6988	0.0422	0.0289	3.9367	0.0550
4.1	3.5301	0.0531	0.0387	3.8313	0.0703
5.1	3.3614	0.0631	0.0477	3.7259	0.0743
6.1	3.1928	0.0823	0.0507	3.5205	0.0872
7.1	3.0209	0.0906	0.0546	3.4150	0.0989
8.1	2.8554	0.0994	0.0593	3.3096	0.0109

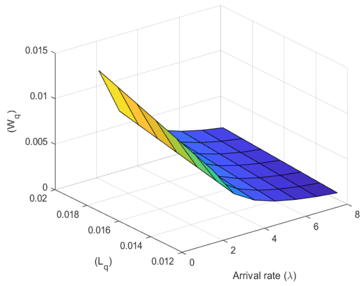
Table 3. P_0 and L_q for different lower service rate (γ_v) for the values of $Q = 30$, $\lambda = 1$, $b = 0.2$, $\eta(x) = 4$, $\gamma_b(x) = 0.6$, $\alpha = 0.7$

Lower service rate (ϑ)	P_0	L_q	L_s	$V(1)$	W_q
0.2	2.0937	0.0330	0.0040	0.2321	0.0898
0.4	2.0457	0.0238	0.0036	0.2191	0.0868
0.6	2.0018	0.0158	0.0031	0.1853	0.0841
0.8	1.9614	0.0149	0.0028	0.1542	0.0816
1.0	1.9242	0.0136	0.0025	0.1055	0.0793
1.2	1.8897	0.0121	0.0022	0.0789	0.0771
1.4	1.8577	0.0114	0.0019	0.0443	0.0751

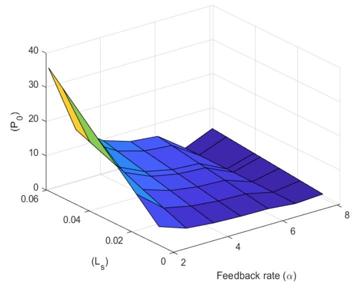
displays that feedback rate α escalates, L_q, L_s, are increases and P_0 decreases. Table 3 displays that lower service rate γ_v escalates, L_q, L_s, $V(1)$ and P_0 decreases.

With the impact of the parameters Q, λ, b, α, ϑ, $\eta(x)$, $\gamma_b(x)$, $\gamma_v(x)$, Fig. 1(a) shows that (L_q) and (W_q) increases while increasing the arrival rate λ. Fig. 1(b), we found that (W_q) diminishes while increasing the feedback rate (α), (L_s). In Fig. 1(c), we found that (P_0) and $(V(1))$ diminishes while increasing the lower service rate γ_v.

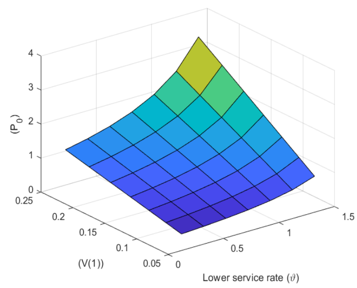
The numerical findings above may be used to determine the impact of attributes on the system's assessment criteria, and we can be sure that the results are representative of actual conditions.



(a) L_q, W_q verses arrival rate λ



(b) P_0, L_s verses feedback rate α



(c) P_0, $V(1)$ verses lower service rate γ_v

Figure 1. 3D visualization of λ, α, and ϑ

6 Computing of ANFIS

Adaptive neuro-fuzzy inference systems (ANFIS) are a type of artificial neural network that borrows heavily from the evolutionary channel fuzzy inference system. The method was developed by Jang [29] in the early 1990s. As a result of its incorporation of both neural networks and fuzzy logic principles, it is able to leverage the advantages of both in a unified framework. It is generally agreed that the ANFIS is a reliable estimator of anything. For practical, everyday congestion, the ANFIS soft computing approach is a powerful instrument for achieving meaningful results.

In addition, the nodes and arrow links make up the adaptive network's structure. The parameters associated with the nodes determine both the nodes' outputs and the means by which their parameters can be adjusted to reduce a specified error measure. To achieve the desired mapping between input and output, the parameters are updated with the aid of the trained data and the gradient-based methodology. ANFIS can be used to generate an input-output mapping based on humans' understanding of if-then rules and the convention of storing input-output data in pairs. The direct search method is inconvenient to employ because of the time constraints caused by the iterative repetition of the process to find the best achievable solution. There is potential for ANFIS to perform flexible data processing. Fig. 2 shows the overall layout of the ANFIS. When exact answers are hard to come by for certain performance indices, ANFIS can be used to find good approximations. In this section, we examine the differences in neuro-fuzzy results between SVT or PGF analysis using ANFIS technology.

Some factors are noticed as linguistic terms and are considered as inputs to associate a fuzzy approach with ANFIS networks. Each of these input variables is assumed to have a membership function that is a Gaussian. Membership functions are shown in Fig. 3 in their corresponding forms. Table 4 provides the membership count function along with its corresponding parameter values and their respective languages.

For analytically determining ANFIS values, we employ the MATLAB software. We calculate the ranges of L_S, L_q, and $V(1)$ from 0 to 0.26 in relation to the values of λ, α, and ϑ, respectively. Fig. 4 demonstrates that the exponential function is depicted by the continuous lines and the ANFIS result by the discrete lines. ANFIS is used to display the λ, α, and ϑ in three dimensions, as shown in Fig. 5. In the end, we discovered that the results of the ANFIS and the

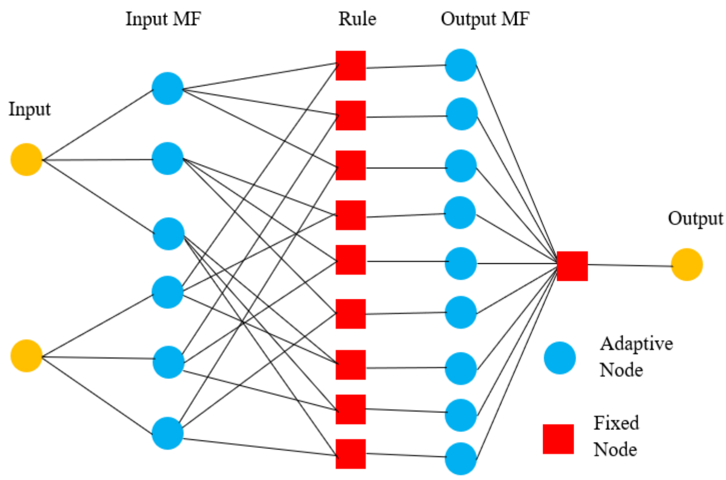
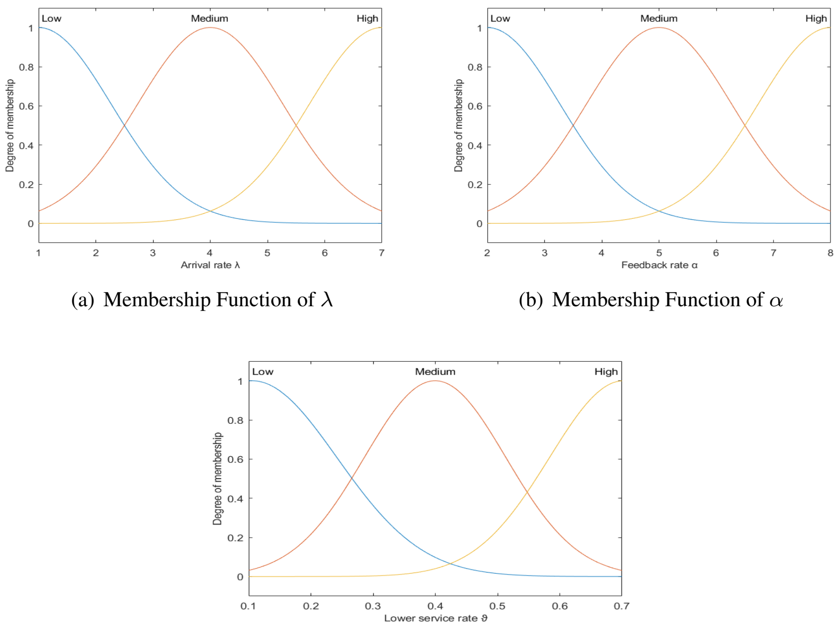


Figure 2. Pictorial representation of ANFIS

Table 4. Values of the membership functions based on the language of the input parameters

Input parameters	Number of membership functions	Linguistic Values
λ, P_0	3	Low, Medium, High
α, P_0	3	Low, Medium, High
ϑ, P_0	3	Low, Medium, High



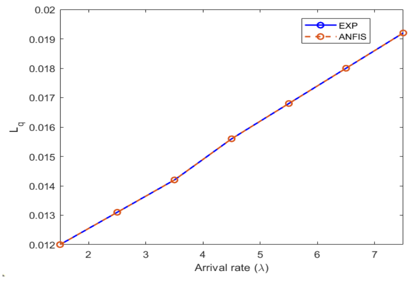
(a) Membership Function of λ

(b) Membership Function of α

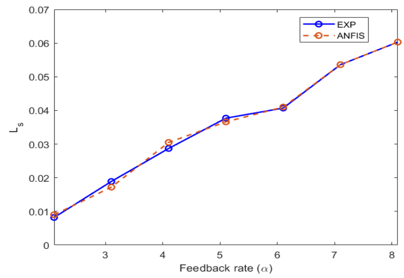
(c) Membership Function of ϑ

Figure 3. Membership Function

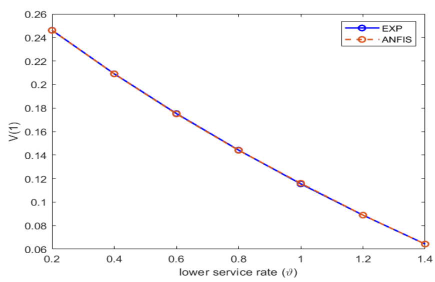
exponential function were similar.



(a) L_q verses λ

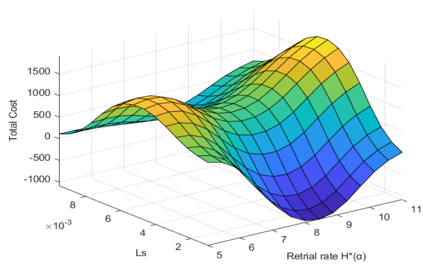


(b) L_S verses α

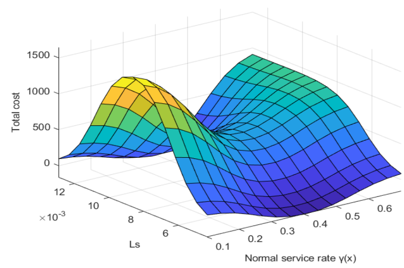


(c) $V(1)$ versus γ_v

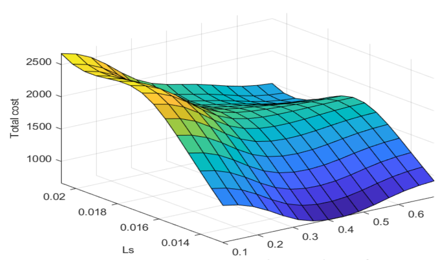
Figure 4. Relation between exponential function and ANFIS



(a) L_s and TEC verses $R^*(\lambda)$



(b) L_s and TEC verses $\gamma_b(x)$

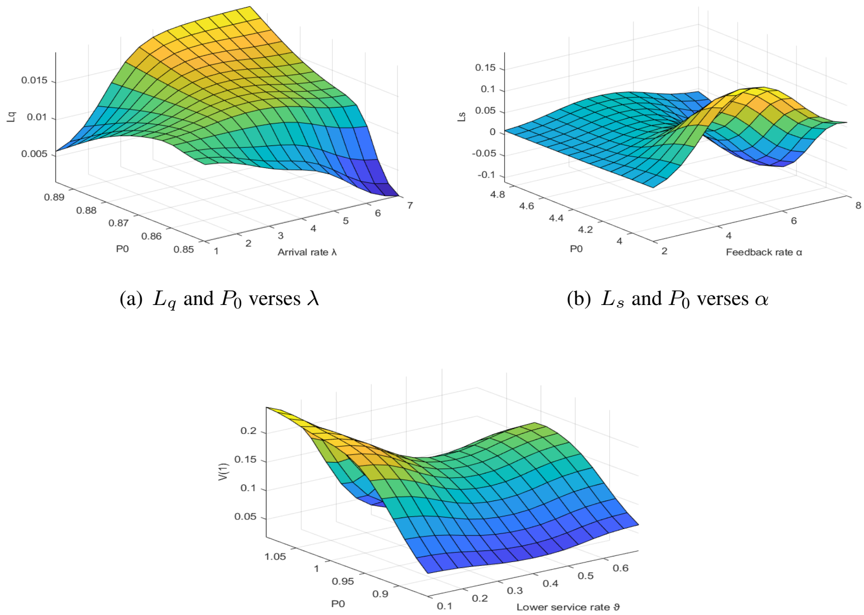


(c) L_s and TEC versus ϑ

Figure 5. ANFIS-based 3D visualization of the TEC

7 Cost Optimization

The process of selecting the set of inputs to an objective function that delivers the maximum or minimum output is known as “optimization." The practise of continually focusing on a company’s operations in order to minimize expenses and costs while simultaneously increasing the value of the firm is known as cost optimization. It means acquiring the most competitive prices and conditions on all company transactions, and it also involves reaching a point where plat-



(a) L_q and P_0 verses λ (b) L_s and P_0 verses α

(c) $V(1)$ and P_0 versus ϑ

Figure 6. ANFIS-based 3D visualization of λ, α and ϑ

forms, applications, procedures, and services will be standardized, streamlined, and rationalized. In a situation that more closely resembles actual life, the relationship between the profit and the operational expenses of the system is quite close. As a consequence of this, the primary responsibility of system developers or administrators is to cut down on the amount of money spent on operations for each unit of time in order to increase the system's profitability. Finding the parameters that allow us to compute the best average cost per unit of time (TEC) is our primary objective here. In order to achieve this goal, we will implement cost functionality in this part of the model that we have developed in order to make it more cost-effective.
We take a cost optimization strategy in order to get the ideal values for the parameters, which include the service rates (γ_b, γ_v). It is presumed that there is a linear relationship between the different system activities and the various cost components associated with those activities in the projected cost function.
The following is a definition of each of the cost element variables that are included in the expected total cost function TEC (γ_b, γ_v) for per unit time:

$S_h \implies$ Cost of holding each consumer in the system for a given period of time.

$S_b \implies$ Cost per unit time (CPUT) when the server is normally active

$S_v \implies$ CPUT when the server is vacation mode

$S_1 \implies$ First-stage service cost per consumer during busy times

$S_2 \implies$ Server's cost per consumer serviced while in Working Vacation mode

Predictions of the cost function expressed as

$$TEC(\gamma_b, \gamma_v) = S_h L_q + S_b\, S(1) + S_v\, V(1) + S_1\gamma_b + S_2\gamma_v \tag{7.1}$$

The cost function that is provided in 7.1 is not simple to optimize in an analytical method because of the substantial non-linearity that it possesses. So, in order to optimize the overall cost, which is supposed to be a function of the service rates γ_b and γ_v, we make use of the heuristic technique. Our primary goal is to discover the ideal service rate (γ_b^*) on busy mode and the optimal service rate (γ_v^*) on vacation mode while reducing the total cost function.

In mathematical terms, the problem of minimizing costs is stated as follows:

$$TEC(\gamma_b^*, \gamma_v^*) = \min_{\gamma_b^*, \gamma_v^*} TEC(\gamma_b, \gamma_v)$$

In order to produce a graphical representation of the sensitivity analysis of the cost function, we arranged the cost components according to Table 5.

Since the beginning of the 1960s, a great deal of research and development has gone into the creation of various optimization strategies. These algorithms have each demonstrated that they are capable of addressing a broad variety of optimization problems. When we don't know much about the structure of the objective function (like a response surface), or when we know the function has local optima, we should utilize a global optimization procedure. On the other hand, when we know we are near the global optima or our objective function has a single optima, such as unimodal, then we should apply a local optimization approach. The application of a local search algorithm to a problem that calls for a global search algorithm would lead to unsatisfactory results since the local search will be fooled by local optima. Particle swarm optimization (PSO), artificial bee colony (ABC), and genetic algorithms (GA) are the global search optimization algorithms that we used to carry out this study. Each of these three algorithms is independently described in one of the five separate subsections that are a part of this section. Keeping in mind the necessity and significance of cost optimization, we used these algorithms. Local search approaches typically decrease the computational complexity involved with identifying the global optimal solution, provided that the assumptions set by the algorithm continue to be valid.

Table 5. Several cost sets for the purpose of cost analysis

Cost set	S_h	S_b	S_v	S_1	S_2
Set 1	10	40	15	20	15
Set 2	5	35	10	15	10
Set 3	20	30	15	9	5

7.1 Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a stochastic optimization technique that was first developed by Kennedy and Eberhart [30]. It was developed with the goal of optimizing a function that was substantially non-linear in its underlying nature. PSO is an effective method for solving problems involving highly non-linear optimization. PSO works by taking into account a set of particles (solutions) and locating the optimal solution to the problem in accordance with the prescribed formulae [31]. Each particle has a fitness value assigned to it, which contributes to the process of getting both the personal best and the global best solution. If a user's best particle yields a better answer than the current global best, the user's best value is used instead of the global best's, and so on, until some maximum number of iterations has been reached. When dealing with queueing scenarios, the PSO method is typically employed to optimize the proposed model's cost function. We optimized the cost by setting the default parameters to have the following values: $\lambda = 3$, $\alpha = 7$, $Q = 30$, $\eta(x) = 6$, $\gamma_b(x) = 0.6$, $\gamma_v(x) = 0.5$ and $R^*(\lambda) = 4$. The lower bound of γ_b, which is taken to be 1, and the upper bound of γ_v, which is taken to be 5, are taken respectively. The number of repetitions, population size, inertial weight, and both acceleration factors have been set to corresponding values of 100, 50, 1, and 2. Table 6 displays the influence of several cost elements, such as S_h, S_b, S_v, S_f, $Q = 30$, $b = 0.2$ and $R^*(\lambda) = 4$ on the optimal service rates and optimal total cost for all three cost sets. Within algorithm 1, the pseudo code for the PSO algorithm is presented.

7.2 Artificial Bee Colony optimization (ABC)

Karaboga and Basturk [32] introduced the Artificial Bee Colony (ABC) approach, which is a swarm-based solution for optimization issues and was inspired by the intelligent behaviour of honey bees when they were foraging. The foraging and food source components are the two most important parts of the algorithm. Foraging bees are categorized as either employed, onlookers, or

Algorithm 1 Pseudo Code of PSO Algorithm

INPUT: Objective function =$TEC(\gamma_b, \gamma_v)$, acceleration factors, inertia weight and Maximum number of iterations
OUTPUT: The cost function's value $TEC(\gamma_b^*, \gamma_v^*)$
Step 1: Finding initial locations F_i for the n particles in a population.
Step 2: Determine H^*(g-best) using best(min) as the TEC $\{F_1, ..., F_n\}$
Step 3: **While** ($t <$ Maximum Generation)
for loop over all n particles and all d dimensions
Step 4: Obtain the new velocity for i^{th} particle $U_i(t+1)$
Step 5: Obtain the new locations for i^{th} particle $R_i(t+1) = R_i(t) + U_i(t+1)$
Step 6: Check the objective function at new locations $R_i(t+1)$
Step 7: Discover the current best (p-best) for each particle R_i^*.
end for
Step 8: Upgrade global best H^*.
end while.
Step 9: Deliver the optimal value of the objective function TEC^*

Table 6. Effect of $\lambda, \alpha, \vartheta$ on $(TEC^*, \gamma_b^*, \gamma_v^*)$ using PSO

Parameters		$(TEC^*, \gamma_b^*, \gamma_v^*)$		
		Cost set 1	Cost set 2	Cost set 3
	3.1	(119.5660,2.2490,1.1958)	(92.0495,2.3562,1.3100)	(55.7764,2.4524,1.4109)
λ	3.2	(122.6688,2.2967,1.2472)	(94.4166,2.4336,1.3474)	(57.0913,2.4177,1.4801)
	3.3	(125.7827,2.3999,1.2621)	(96.7775,2.5079,1.3582)	(58.4047,2.5137,1.4853)
	7.1	(120.1449,2.2580,1.1913)	(92.4986,2.4027,1.2926)	(56.0363,2.4018,1.4628)
α	7.2	(120.7113,2.2602,1.2485)	(92.9499,2.4298,1.2802)	(56.2992,2.4245,1.4479)
	7.3	(121.2724,2.3043,1.2403)	(93.3822,2.4252,1.2896)	(56.5654,2.4209,1.4354)
	5.1	(120.2041,2.3014,1.2116)	(92.5444,2.3597,1.2991)	(56.0596,2.4642,1.4300)
ϑ	5.2	(120.2549,2.2633,1.2170)	(92.5907,2.3932,1.3126)	(56.0770,2.3821,1.4309)
	5.3	(120.3189,2.2832,1.2370)	(92.6351,2.3917,1.3129)	(56.0907,2.4162,1.4621)

scouts according to the roles that they play in the circumstance at hand, depending on the types of jobs that they perform. Bees that are involved in foraging as well as bees that are not employed in foraging look for rich food sources. In this method, a colony of artificial forager bees searches for sources of artificial food that are particularly rich (a good solution). In order to employ this optimization strategy and get the objective function optimized, you are going to need to have the ideal parameter vector. After that, artificial bees will search in a random manner for a population of initial solution vectors. The technique of searching for the nearest neighbour provides the basis for the iterative processes that are used to refine the response. The algorithmic sequence of steps known as ABC is described in algorithm 2 along with its corresponding pseudo code. Table 7 demonstrates the influence of various cost factors, including S_h, S_b, S_v, S_f, $Q = 30$, $b = 0.2$, $\eta(x) = 6$, $\gamma_b(x) = 0.6$, $\gamma_v(x) = 0.5$ and $R^*(\lambda) = 4$.

7.3 Genetic Algorithm (GA)

In the 1960s and 1970s, Bremermann [33] and his coworkers came up with the idea for what is now known as the genetic algorithm. This algorithm is a method for solving problems associated with restricted and uncontrolled optimization that is derived from natural selection. Natural selection is the mechanism that drives the progression of biological development. They are frequently used for the purpose of providing high-quality solutions to problems involving stochastic search. The entirety of the algorithm depicts the selection criteria that are used to identify the individuals who are the healthiest and most suited for procreation in order to generate offspring for the next generation. Genetic algorithms are a replication of a natural selection criterion that looks for organisms that are able to live, reproduce, and pass on their genes to subsequent generations while also being able to adapt to the conditions of their environment. To put it another way, the process of finding a solution to a problem involves simulating the "sustainability of the most suitable" among people from successive generations. The method described above is appealing

Algorithm 2 Pseudo Code of ABC Algorithm

INPUT: Objective function $=TEC(\gamma_b, \gamma_v)$,
OUTPUT: The cost function's value $TEC(\gamma_b^*, \gamma_v^*)$
Step 1: Create the populace of outcome $M_i, i = 1$
Step 2: Check out the populace, period 1, $h = 0$
Step 3: Choose the best outcome, Mbest and fix Mbest1 = Mbest
Step 4: Redo
Step 5: Obtain a new way out Mnew = M_i for the worker bees and to get them.
Step 6: Use the greedy selection method for the worker bees.
Step 7: Give each result a grade, and then choose the best one. Step 8: Find the probability P_i of the solution M_i.
Step 9: With P_1 as a starting point, generate a fresh outcome M_i for the onlookers.
Step 10: Use the greedy selection method to the onlookers.
Step 11: If the scout's result has been cancelled, then proceed with a newly created result M_i.
Step 12: Keep in mind the M new conclusion you've reached thus far.
Step 13: Put $h = h + 1$ period = period + 1.
Step 14: Until (the end condition has been met, i.e., period = MCN)

Table 7. Effect of $\lambda, \alpha, \vartheta$ on $(TEC^*, \gamma_b^*, \gamma_v^*)$ using ABC

Parameters		$(TEC^*, \gamma_b^*, \gamma_v^*)$		
		Cost set 1	Cost set 2	Cost set 3
	3.1	(119.5598,2.2427,1.2223)	(92.0488,2.3593,1.3014)	(55.7679,2.4160,1.4387)
λ	3.2	(122.6683,2.3045,1.2499)	(94.4192,2.3996,1.3604)	(57.0901,2.4828,1.4659)
	3.3	(125.8635,2.3858,1.3600)	(96.7861,2.4375,1.3481)	(58.4615,2.4317,1.6318)
	7.1	(120.1313,2.3167,1.2099)	(92.4937,2.3913,1.3366)	(56.0339,2.4163,1.4378)
α	7.2	(120.7002,2.2739,1.2231)	(93.0356,2.4732,1.3801)	(56.3505,2.4116,1.3436)
	7.3	(121.2659,2.3815,1.2623)	(93.3759,2.4001,1.3121)	(56.5631,2.4527,1.4477)
	5.1	(120.1940,2.2635,1.2215)	(92.5422,2.3782,1.3066)	(56.0530,2.4173,1.4412)
ϑ	5.2	(120.2545,2.2691,1.4707)	(92.5889,2.3722,1.3167)	(56.0715,2.4102,1.4529)
	5.3	(120.3128,2.0273,1.0807)	(92.6340,2.3811,1.3066)	(56.0893,2.4221,1.4428)

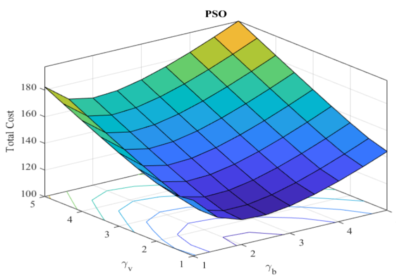
to a large number of research analysts since it has a wide range of applications, including those in the fields of code breaking, data centres, and electrical circuit design, amongst others. They have made use of GA in order to cut down on the costs associated with an interruptive strategy by introducing the concept of state-based bulk service, in addition to varying vacations and shifting work hours. In order to attain the optimal values of all of the cost elements described before, this methodology is implemented in research studies. In algorithm 3, the pseudo code for the algorithmic sequence of steps that constitutes GA is presented. For performing GA optimization, Table 8 show the effect of cost elements as S_h, S_b, S_v, S_f, $Q = 30$, $b = 0.2$, $\eta(x) = 6$, $\gamma_b(x) = 0.6$, $\gamma_v(x) = 0.5$ and $R^*(\lambda) = 4$.

Algorithm 3 Pseudo Code of GA Algorithm

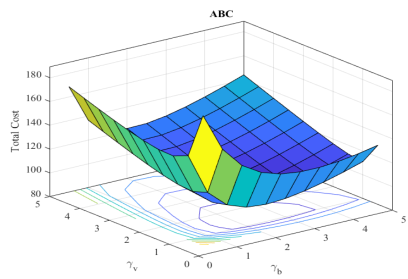
INPUT: Objective function $=TEC(\gamma_b, \gamma_v)$,
OUTPUT: The cost function's value $TEC(\gamma_b^*, \gamma_v^*)$
Step 1: Establishing a base population
Step 2: **for** population size **do**
Step 3: execute phases
Step 4: **if** elitism **then**
Step 5: population[0] =fittest
Step 6: **end**
Step 7: **end**
Step 8: **return** fittest

Table 8. Effect of $\lambda, \alpha, \vartheta$ on $(TEC^*, \gamma_b^*, \gamma_v^*)$ using GA

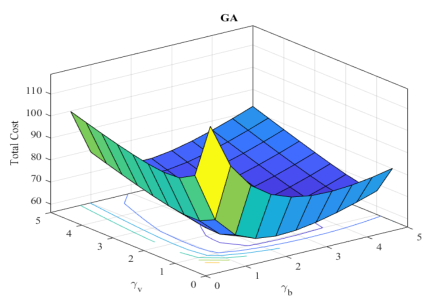
Parameters		$(TEC^*, \gamma_b^*, \gamma_v^*)$		
		Cost set 1	Cost set 2	Cost set 3
	3.1	(119.5591,2.2480,1.2162)	(92.0488,2.3595,1.3020)	(55.7672,2.4010,1.4367)
λ	3.2	(122.6683,2.3030,1.2499)	(94.4139,2.4197,1.3379)	(57.0875,2.4534,1.4749)
	3.3	(125.7669,2.3578,1.2836)	(96.7708,2.4375,1.3737)	(58.4015,2.5054,1.5129)
	7.1	(120.1313,2.2616,1.2191)	(92.4937,2.3736,1.3052)	(56.0339,2.4154,1.4401)
α	7.2	(120.7002,2.2751,1.2221)	(92.9360,2.3877,1.3083)	(56.2991,2.4296,1.4435)
	7.3	(121.2659,2.2885,1.2250)	(93.3759,2.4019,1.3117)	(56.5628,2.4437,1.4413)
	5.1	(120.1940,2.2635,1.2203)	(92.5422,2.3755,1.3064)	(56.0530,2.4173,1.4413)
ϑ	5.2	(120.2547,2.2603,1.2226)	(92.5889,2.3774,1.3077)	(56.0715,2.4192,1.4425)
	5.3	(120.3128,2.2671,1.2226)	(92.6340,2.3792,1.3088)	(56.0893,2.4209,1.4436)



(a) TEC vs γ_b, γ_v using PSO



(b) TEC vs γ_b, γ_v using ABC



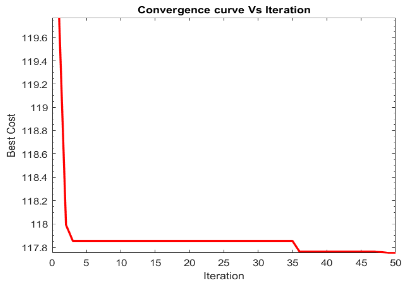
(c) TEC vs γ_b, γ_v using GA

Figure 7. TEC vs γ_b, γ_v

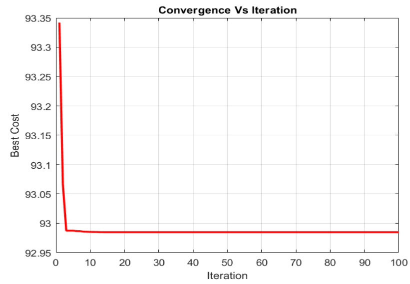
7.4 Comparative analysis of PSO, ABC, and GA

Here we evaluate the MATLAB implementations of three different cost-finding algorithms: the genetic algorithm (GA), the artificial bee colony (ABC), and the particle swarm optimization (PSO) to see which yields the best results. We consider three distinct cost sets in Table 5 and three distinct pairs of optimum parameters $(\lambda, \alpha, \vartheta)$. We next iteratively execute the MATLAB code corresponding to each of the aforementioned algorithms. Therefore, we have continued this procedure and created Table 6-8. We discovered that, the results from all three programmes were very similar to one another. As a result, the best solutions and associated costs for these three techniques are quite close to one another. This proves that the aforementioned heuristics provide dependable (local) optimal solutions.

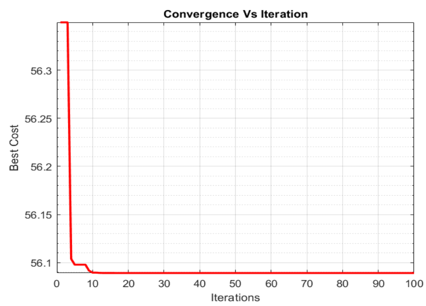
As can be seen in Tables 6-8, the maximum number of iterations required by GA is significantly lower than that required by other methods. We are able to determine the ideal cost using any approach; however, if we compare these techniques for our model, GA is the strategy that is most suited to determining the optimal cost. We discovered that the GA technique is the best approach out of all of these strategies since it offers a lot of benefits to its users. It is simple to configure, needs only a few parameters, performs well in global searches, and is unaffected by the scaling of design variables. GA has a propensity to result in a rapid and early convergence



(a) Convergence vs Iteration using PSO



(b) Convergence vs Iteration using ABC



(c) Convergence vs Iteration using GA

Figure 8. Convergence vs Iteration

in mid-optimal spots (having poor local search ability), in addition to a slow convergence in a region where the search has been improved.

7.5 Convergence analysis of PSO, ABC and GA

When we employ a technique for optimization, whether it be PSO, ABC, or GA, the particles are not in a stable state at the beginning of the process. It is essential to determine whether or not the particle returns to its normal state and whether or not it will continue to search for a more optimal solution. As a result, convergence is an essential component of cost analysis. In Fig. 7 presents the convexity and optimality of the cost function with regard to the three cost sets that are taken into consideration in the optimization analysis. This figure was generated using the PSO, ABC, and GA optimization methods. In addition, we make use of an optimization methodology, and whether we use PSO, ABC, or GA, the particles do not begin the process in a state of stability. Because of this, it is essential to determine whether or not the particle returns to its normal state and whether or not it will search for a more optimal solution. As a result, convergence is an essential component of cost analysis. According to Fig. 8, the elements are converge to the optimal cost in GA the fastest, with less convergence in PSO and ABC. This occurs over the course of the same amount of time.
From the convergent nature of these optimization techniques, which was previously described, we can derive the following inferences:

- The total costs of the parameters that were derived by PSO, ABC, and GA, as well as the optimal values, are the same.
- According to the findings of the research, the model that we supplied is consistent with the actual situations. The cost of the analyzers brings this system up to its total cost, which will help fix some of their economic concerns to some amount.
- The cost benefit analysis that was produced can be relied upon to a significant level under the current circumstances, which not only demonstrate the logic of our model but also assist network administrators and specialists in reducing the difficulty of the challenge that blocking poses to specific telecommunication services.

8 Conclusion

This article explores a batch arrival feedback retrial queue system with balking and variable server capacity, operating under a working vacation scenario. Under certain conditions, the system can achieve stability. The PGF approach and supplementary variable technique are used to compute the PGF of the system's consumer size and its behavior under different service states. The numerical findings provide insights into how various system parameters impact its performance. The model, designed for real-time systems with staged service, is validated using the soft computing approach ANFIS to evaluate performance and cost functions. Incorporating fuzzy parameters through ANFIS enhances the realism of the model for queueing systems. Furthermore, optimization algorithms like PSO, ABC, and GA are employed to compute costs, allowing for a comparison of their effectiveness. The study's results can inform the design of various systems such as computer networks, packet switching networks, manufacturing lines, and postal systems. Future research could include studying bulk service queueing systems with prioritized consumers under a working vacation setup, as well as exploring transient solutions for such systems.

References

[1] J.R. Artalejo and A. Gómez-Corral, Limiting Distribution of the System State, *In Retrial Queueing Systems: A Computational Approach*, Berlin, Heidelberg, (2008). https://doi.org/10.1007/978-3-540-78725-9

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

[3] Gautam Choudhury, *A batch arrival queue with a vacation time under single vacation policy,* Computers and Operations Research, **29** (2002), 1941–1955. https://doi.org/10.1016/S0305-0548(01)00059-4

[4] Shan Gao, *An $M^{[X]}/G/1$ Queue System with Delayed Single Working Vacation,* Energy Procedia, **13** (2011), 1493–1498. https://doi.org/10.1016/j.egypro.2011.11.210

[5] V. Chandrasekaran, K. Indhira, M. Saravanarajan and P. Rajadurai, *A survey on working vacation queueing models,* International Journal of Pure and Applied Mathematics, **106** (2016), 33–41. https://doi:10.12732/ijpam.v106i6.5

[6] P. Rajadurai, *Sensitivity analysis of an $M/G/1$ retrial queueing system withd disaster under working vacations and working breakdowns*, RAIRO Operation Research, **52** (2018), 35–54. https://doi.org/10.1051/ro/2017091

[7] S. Pazhani Bala Murugan and R. Vijaykrishnaraj, *A bulk arrival retrial queue with feedback and exponentially distributed multiple working vacation*, International Journal of Computing Sciences Research, **10** (2019), 81–91.

[8] Madhu Jain and Anshul Kumar, *Unreliable server $M^{[X]}/G/1$ queue with working vacation and multiphase repair with delay in verification,* International journal of applied engineering research, **8** (2022), 1–17. https://doi.org/10.1007/s40819-022-01383-5

[9] S. Maragathasundari and B. Balamurugan, *Analysis of an $M^{[X]}/G/1$ feedback Queue with Two stages of Repair times, General Delay Time*, International journal of applied engineering research, **10** (2015), 25165–25174.

[10] Madhu Jain and Anshul Kumar, *Unreliable Server $M^{[X]}/G/1$ Retrial Feedback Queue with Balking, Working Vacation and Vacation Interruption*, Proceedings of the National Academy of Sciences, India Section A: Physical Sciences, 2022 https://doi.org/10.1007/s40010-022-00777-w

[11] M.M.N. GnanaSekar and I. Kandaiyan, *Analysis of an $M/G/1$ Retrial Queue with Delayed Repair and Feedback under Working Vacation policy with Impatient Customers,* Symmetry, **14** (2022). https://doi.org/10.3390/sym14102024

[12] N.T. Bailey, *On queueing processes with bulk service*, Journal of the Royal Statistical Society, **16** (1954), 80–87. https://doi.org/10.1111/j.2517-6161.1954.tb00149.x

[13] S. Sasikala and K. Indhira, *Bulk service queueing models-a survey*, International Journal of Pure Applied Mathematics, **106** (2016), 43–56. https://doi.org/10.12732/ijpam.v106i6.6

[14] N.K. Jaiswal, *A bulk service queueing problem with variable capacity,* Journal of the Royal Statistical Society, **106** (1961), 143–148. https://doi.org/10.1111/j.2517-6161.1961.tb00397.x

[15] A. Banerjee, U.C. Gupta and Sikdar, *Analysis of finite-buffer bulk-arrival bulk service queue with variable service capacity and batch-size-dependent service,* International Journal of Mathematics in Operational Research, **5** (2013), 358–386. https://doi.org/10.1504/IJMOR.2013.053629

[16] S. Sasikala, K. Indhira and V.M. Chandrasekaran, *A study on $M^X/G^B/1$ retrial queueing system with Bernoulli vacation schedule and variable server capacity*, International Journal of Knowledge Management in Tourism and Hospitality, **1** (2017), 263–277. https://doi.org/10.1287/opre.17.6.1058

[17] S.S. Nair and K.P. Jose, *Solution to production inventory systems with orbit, buffer and different service rates*, Palestine Journal of Mathematics, **8** (2019), 328–347.

[18] B. Jouilik, J. Daoudi, C. Tajani and J. Abouchabaka, *Optimization approach based on genetic algorithm for a robin coefficient problem*, Palestine Journal of Mathematics, **11** (2022), 708–718.

[19] S. Upadhyaya, D. Agarwal, R. Agarwal, I. Ahmad, A.H. Ibrahim and S. Al-Homidan, *Queueing and reliability analysis of unreliable multi-server retrial queue with Bernoulli feedback*, Carpathiyan Journal of Mathematics, **39** (2023), 745–758.

[20] R. Agarwal, S. Upadhyaya, D. Agarwal and S. Kumar, *Cost optimality of an erratic GeoX/G/1 retrial queue under J-vacation scheme using nature inspired algorithms*, International Journal of Industrial and Systems Engineering, **44** (2023), 1–33.

[21] R. Agarwal, D. Agarwal, S. Upadhyaya and I. Ahmad, *Optimization of a stochastic model having erratic server with immediate or delayed repair*, Annals of Operations Research, **331** (2023), 605–628.

[22] C. Tajani and H. Fakhouri, *A comparative study of ant colony optimization variants for the probabilistic travelling salesman problem*, Palestine Journal of Mathematics, **13** (2024), 104–112.

[23] S. Gao, J. Wang and W. Li, *An $M/G/1$ retrial queue with general retrial times, working vacations and vacation interruption*, Asia Pacific Journal of Operational Research. **31** (2014), 6–31. https://doi.org/10.1142/S0217595914400065

[24] Humblet, Sennott and Tweedi, *Average drifts and the non-Ergodicity of Markov chains*, Operation Research, **31** (1983), 783–789. https://doi.org/10.1287/opre.31.4.783

[25] A.G. Pakes, *Some conditions for Ergodicity and recurrence of Markov chains*, Operation Research, **17** (1969), 1058–1061. https://doi.org/10.1287/opre.17.6.1058

[26] H. Takagi, *Vacation and priority systems. Part I. Queueing analysis: A foundation of performance evaluation*, North-Holland, Amsterdam, New York, 1991.

[27] D.Y. Yang and C.H. Wu, *Cost-minimization analysis of a working vacation queue with N-policy and server breakdowns,* Computers and Industrial Engineering, **82** (2015), 151–158. https://doi.org/10.1016/j.cie.2015.01.017

[28] M. Zhang and Z. Hou, *$M/G/1$ queue with single working vacation*, Journal of Applied Mathematics and Computing, **39** (2012), 221–234. https://doi.org/10.1007/s12190-011-0532-x

[29] J.S.R. Jang, *ANFIS: adaptive-network-based fuzzy inference system,* IEEE Transactions on Systems, Man, and Cybernetics: System, **23** (1993), 665–685.

[30] J. Kennedy and R. Eberhart, *Particle swarm optimization. Proceedings of ICNN'95-international conference on neural networks,* **4** (1995), 1942–1948.

[31] R. Poli, J. Kennedy and T. Blackwell, *Particle Swarm Optimization*, Swarm Intelligence, **1** (2007), 33–57.

[32] D. Karaboga and B. Basturk, *A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm*, Journal of Global Optimization, **3** (2007), 459–471.

[33] H.J. Bremermann, *The evolution of intelligence: the nervous system as a model of its environment*, 1958.

Appendix A

Theorem 8.1. *The embedded Markov chain $\{G_n; n \in N\}$ is ergodic iff $\Lambda < Q$ for our system will be stable, where $\Lambda = \bar{\alpha} - \lambda b E(I)E(H) + E(I)(1 - R^*(\lambda))$.*

Proof. Foster's [25] criteria, which claim that the chain $\{G_n; n \in N\}$ is an irreducible and aperiodic chain, may be used to easily confirm the required condition of ergodicity. Assuming a non-negative measure $e(r)$, $r \in M$ and $\delta > 0$, the Markov chain is ergodic, and mean drift $\nu_r = E[e(u_{m+1}) - e(u_m)/v_m = r]$ with a limited exception r's, $r \in M$ and $\nu_r \leq -\delta \, \forall \, r \in M,$. In this case, we're focusing on the function $e(r) = r$. Next, we obtain

$$\nu_r = \begin{cases} \bar{\alpha} - \lambda b E(I)E(H) - Q, & \text{if } r\text{=0} \\ \bar{\alpha} - \lambda b E(I)E(H) + E(I)(1 - R^*(\lambda)) - Q, & \text{if } r\text{=1,2,...} \end{cases}$$

In this case, $\bar{\alpha} - \lambda b E(I)E(H) + E(I)(1 - R^*(\lambda)) < Q$ is undoubtedly a prerequisite for ergodicity.
As said by Humblett et al. [24], if the Markov chain $\{G_n; n \in N\}$ matches Kaplan's status,

specifically $\nu_r < \infty \; \forall \; r \geq 0$ and $\exists \; r_0 \in M$ such that $\nu_r \geq 0$ for $r \geq r_0$, the necessary condition is satisfied. $U = (v_{qr})$ is the the unit-step transition matrix of $\{G_n; n \in N\}$ for $r < q - j$ and $q > 0$. The Markov chain's non-ergodicity is suggested by $\Lambda \geq Q$. □

Author information

N. Micheal Mathavavisakan, Department of Mathematics, Vellore Institute of Technology, Vellore, 632014, Tamilnadu, India.
E-mail: michealmathava.visakan2020@vitstudent.ac.in

K. Indhira, Department of Mathematics, Vellore Institute of Technology, Vellore, 632014, Tamilnadu, India.
E-mail: kindhira@vit.ac.in (Corresponding author)

Received: 2024-03-16
Accepted: 2024-06-15