Hacettepe Journal of Mathematics & Statistics

Hacet. J. Math. Stat.
Volume 54 (5) (2025), 1976 – 2006
DOI : 10.15672/hujms.1696033

Research Article

Particle swarm optimization of a single server retrial queue with balking and immediate feedback under Bernoulli working vacation

J. Kalaiselvi, M. C. Saravanarajan*

Department of Mathematics, School of Advanced Sciences, Vellore Institute of Technology, Vellore - 632 014, Tamil Nadu, India.

Abstract

In this study, we investigate a single-server retrial queueing system that incorporates balking, immediate feedback, and a Bernoulli working vacation policy. Customers arriving to find the server busy, under repair, or on working vacation may balk; otherwise, they either join the orbit or receive immediate service if the server is available. Upon completion of the service, the customer can request a finite number of immediate feedback services. When the orbit is empty after the completion of a service, the server initiates a working vacation, serving at a reduced rate. If customers are present in the orbit at the end of the vacation, the server resumes normal operation. If the system is empty, the server remains idle or continues the vacation. We analyze the ergodicity conditions to ensure system stability and derive the stationary distribution of the underlying Markov process. Several key performance measures are computed. Furthermore, a comprehensive cost function is developed and optimized using metaheuristic approaches, including particle swarm optimization and the genetic algorithm. The convergence behavior and optimization results are illustrated through graphical analysis, offering insight into improving the efficiency of complex retrial queueing systems.

Mathematics Subject Classification (2020). 68M20, 90B22, 60K25.

Keywords. Balking, Bernoulli working vacation, immediate feedback, particle swarm optimization, retrial queue.

1. Introduction

Queueing systems are a crucial part of daily life, managing customer flow in various settings such as banks, hospitals, and supermarkets. Retrial queues (RQ) are particularly valuable in scenarios where customers, unable to receive service immediately, may leave and return later, improving system performance and customer retention. This study focuses on a single server retrial queue model incorporating balking, where customers decide not to wait based on queue conditions, and immediate feedback, allowing retrial customers

*Corresponding Author.
Email addresses: muthukalai2015@gmail.com (J. Kalaiselvi), mcsaravanarajan@vit.ac.in (M. C. Saravanarajan).
Received: 09.05.2025; Accepted: 07.08.2025

to make informed decisions. Additionally, the model integrates a Bernoulli working vacation policy, in which the server continues to operate at a reduced rate during vacation periods, providing a more flexible service management approach. Server failures and vacations can disrupt operations, increasing wait times and retries, necessitating optimization. Using Particle Swarm Optimization (PSO) and a genetic algorithm (GA), this paper aims to improve key performance metrics, offering a comprehensive framework for predicting and resolving potential bottlenecks, ultimately improving system efficiency and customer satisfaction.

A comprehensive and systematic evaluation of the previous studies has been provided to understand the research efforts made and the research gaps that exist. In recent years, RQ has become a vital topic of research. On top of that, several papers have been written about retrial QS with the possibility of single phases of service provided by the server. This kind of queuing is prevalent. Rajadurai [1] study explores an M/G/1 preemptive priority retrial queue incorporating Bernoulli working vacations and vacation interruptions. Li et al. [2] studied an M/G/1 retrial queue with balking customers and Bernoulli working vacation interruptions. Keerthiga and Indhira [3] analyzed a single-server feedback retrial queue incorporating Bernoulli working vacations and starting failure. Recently, GnanaSekar and Kandaiyan [4] examined a retrial queue M/G/1 with delayed repair and feedback, incorporating a work-vacation policy and customer impatience. Sundarapandiyan and Nandhini [5] investigated a non-Markovian feedback retrial queue with two types of customers and delayed repair under a Bernoulli working vacation policy. In addition, Sivasubramaniam and Jagannathan [6] analyzed a large arrival queue with an unreliable server, fumbling behavior, and a modified Bernoulli vacation policy. Dehamnia et al. [7] conducted a performance and economic analysis of an unreliable single-server retrial queue with general retrial times and varying levels of customer patience. Dehamnia et al. [8] analyzed an unreliable retrial queue with two types of customer arrivals and a service orbit.

Customers who choose not to join a queue because it seems too lengthy or slow are said to be engaging in balking behavior in QS. Recently, a lot of research has been done on the behavior of balkers. Arivudainambi and Godhandaraman [9] addressed a bulk arrival feedback RQ with dual service phases, balking, and k discretionary vacations. Ke et al. [10] examined a single server queue that was exposed to balking. Nithya and Haridass [11] addressed the maximum entropy of a $M^{[X]}/G/1$ QS under failure, startup, and vacation interruption. An unstable retrial model with a single server, customer resistance, and server failures and fixes was examined by [12]. Recently, Bouchentouf et al. [13] conducted a mathematical analysis of a Markovian multi-server feedback queue with multiple vacation variants, incorporating balking and reneging behaviors. Bouchentouf et al. [14] explored a variant vacation queueing system with Bernoulli feedback, balking, and server state-dependent reneging.

This concept of immediate feedback differs significantly from instantaneous feedback. Here, after completing their initial service, if a customer notices any issues or desires additional service, they can receive it immediately without rejoining the queue. A practical example of this is an ATM, where after completing a transaction, such as withdrawing money, a customer can immediately perform another transaction, such as requesting a mini statement or changing their PIN, without waiting in line again. Several authors, including Madan et al. [15], Baruah et al. [16], and Rajadurai et al. [17], have explored the concept of re-service or immediate feedback, highlighting scenarios where customers can immediately access the server for another round of service. Recently, Bouchentouf et al. [18] conducted a comprehensive performance and economic analysis of a single server feedback queuing model that incorporates vacation periods and impatient customers. Saravanan et.al. [19] examine a retrial queuing system characterized by optional service, an unreliable server, balking behavior, and feedback mechanisms. Boualem et al. [20] studied a single-server retrial queue incorporating Bernoulli feedback and negative customers.

During a working vacation (WV) period, the server continues to provide service, but at a reduced rate. In contrast, during a regular vacation period, the server completely ceases to provide service. Lately, Bouchentouf et al. [21] analyze a multi-station unreliable machine model with a working vacation policy and customer impatience. Bouchentouf et al. [22] model and simulate a Bernoulli feedback queue with general customer impatience under a variant vacation policy. Chettouf et al. [23] present a Markovian queueing model for telecommunications support centers, which incorporates breakdowns and vacation periods. Dehimi et al. [24] explore the analytical and computational aspects of a multi-server queue with impatience under a differentiated working vacation policy. Shanmugam and Saravanarajan [25] investigate an unreliable retrial queueing system with a working vacation. Mathavavisakan and Indhira [26] focus on nonlinear metaheuristic cost optimization and ANFIS computing for a feedback retrial queue with two dependent phases of service under Bernoulli working vacation. Abir et al. [27] investigated a finite-capacity $M/M/2$ machine repair model that incorporates impatient customers, triadic service discipline, and dual working vacation policies.

There is not much of work on RQ that addresses cost optimization and evaluates appropriate control parameters for the queueing model using optimization techniques. Lately, Vaishnawi et al. [28] investigated the accuracy of a $Geo^{[}X]/Geo/1$ recurrent model in discrete time using various optimization techniques. Similarly, Kumar and Jain [29] formulated a cost function and analyzed a Markovian queueing system to identify optimal service strategies. Harini et al. [30] studied a batch arrival retrial queue with optional re-service and M-optional vacations using advanced metaheuristic methods. Kumar et al. [31] explored machine interference problems with standby and vacation interruptions, applying sophisticated optimization techniques. Amit Kumar et al. [32] addressed cost optimization in a feedback queueing system with breakdowns and threshold policy, while Dehmi et al. [33] analyzed an M/M/c/M queue with feedback, balking, and a hybrid hiatus policy using ANFIS and cost minimization. Jain and Jain [34] investigated a retrial queue with server breakdown and caller intolerance, applying the Genetic Algorithm to optimize system performance under voluntary service. Similarly, Malik et al. [35] performed a cost analysis of a Geo/G/1 retrial model using both Particle Swarm Optimization and the genetic algorithm, highlighting the effectiveness of hybrid evolutionary techniques.

Although previous studies have independently investigated working vacation, immediate feedback, and Bernoulli work vacation policies, the interplay between immediate feedback mechanisms and the transition from Bernoulli working vacation has not been thoroughly examined. To bridge this research gap, this work focuses on a single-arrival retrial queue model that incorporates immediate balking feedback and a Bernoulli working vacation policy.

1.1. Research objective

This study aims to utilize particle swarm optimization to investigate and optimize the performance of a single-server retrial queue system, incorporating balking, immediate feedback, and the transition from Bernoulli working vacation policy, with a focus on minimizing costs and improving system efficiency.

The specific focus on an $M/G/1$ retrial queue with immediate feedback, along with the consideration of server vacations and customer balk behavior, reflects the complexity of real-world scenarios. Understanding and optimizing such intricate systems can lead to improvements in customer satisfaction, resource utilization, and operational efficiency. By addressing the ergodicity requirement for system stability and deriving analytical findings for the stationary distribution, this study aims to provide a solid theoretical foundation for understanding the behavior of the queueing system under different conditions. In addition, the establishment of various performance metrics enables a comprehensive evaluation of the

efficiency and effectiveness of the system. The formulation of a comprehensive cost function and its optimization using PSO and GA further contribute to the practical applicability of the study. By minimizing the cost function, decision-makers can make informed choices to enhance system performance while considering various trade-offs. In general, this study seeks to offer valuable information and methodologies for optimizing the performance of complex retrial queueing systems, thus addressing the needs and challenges of modern-day applications in communication networks, manufacturing processes, service operations, and beyond.

1.2. Research contribution

A review of the literature reveals a significant gap in studies that address single-arrival retrial queues under these conditions, although research on retrial queues with immediate feedback and working vacation has been conducted. Motivated by this observation, the primary contributions of this study are as follows: To achieve this objective, the study employs the supplementary variable technique (SVT) and the generating function method to evaluate the behavior of the single-server retrial queue. These techniques enable a thorough analysis of system dynamics, including the impact of immediate feedback clients and balking behavior on system performance. Furthermore, the study goes beyond theoretical analysis by proposing a novel structure for the queueing system and presenting numerical instances to illustrate its practical implications. Through the use of graphs and tables, the study provides insight into the behavior of the proposed system under different scenarios and parameter settings. In addition to theoretical and numerical analysis, the study introduces a comprehensive cost function to quantify system performance and proposes its optimization using PSO and GA.

The convergence analysis of the optimization process, supported by illustrative figures, further enhances the practical relevance of the study. In general, the novelty of this work lies in its holistic approach to addressing a previously unexplored research gap in queueing theory. By combining analytical, numerical, and optimization techniques, the study offers valuable insight into optimizing the performance of intricate queueing systems with immediate feedback consumers under BWV, thus contributing to both theoretical advancements and practical applications in various domains.

The primary goal of this investigation is to establish the queue distribution length and orbit size with the intention to assist in the design and evaluation of new metrics to monitor the behavior of the system.

The contents of our study are as follows: Upon meeting the prerequisites, Section 2 offers an in-depth discussion of the suggested system, covering its practical implications as well. Section 3 discusses the steady-state (SS) characteristics of the system and the probability generating function (PGF) for queue size. Section 4 provides assessments for different system behavior metrics. Section 5 includes notable exception cases. Section 6 presents numerical and graphic representations of desired outcomes. Ultimately, Section 7 leverages PSO to conduct the cost evaluation. Section 8 summarizes the primary concepts of the paper.

2. Overview and analysis of the proposed framework

This paper considers the $M/G/1$ retrial queueing system with balking and immediate feedback under Bernoulli working vacation policy. The detailed description of the model is given as follows:

- **The arrival process:** The positive customers arrive into the system according to a Poisson process with rate η.
- **Retrial process:** When a customer arrives and finds the server available, it begins service immediately. If the server is occupied, the customer has two possible

actions: either join a group of blocked customer, known as the "orbit," with probability $(1-\alpha)$, or exit the system (balk) with probability (α). The inter-retrial times are governed by an arbitrary distribution $(H_1(x))$, with the corresponding Laplace-Stieltjes Transform (LST) represented as $(H_1^*(t))$.

- **The regular service process:**If a new positive customer or a retrying customer arrives when the server is idle, the server immediately commences regular service for the arrival. The service time follows a general distribution, described by the random variable (H_b), with its distribution function given as $(H_b(x))$ and its Laplace-Stieltjes Transform (LST) expressed as $(H_b^*(t))$.
- **Immediate feedback:** After completing the initial round of service, a customer either immediately re-enters the system for the first phase of service with a probability $(1-\beta_1)$ or exits the system permanently with a probability (β_1). Once the feedback service is finished, the customer has a probability $(1-\beta_2)$ of entering the system again for a third round of service, or may leave with a probability (β_2). This cycle can continue until the customer has undergone (k) rounds of service, after which they must exit the system. The next customer can only enter the system once all feedback rounds of the preceding customer are fully completed.
- **Bernoulli Working Vacation process:**: The server initiates a working vacation whenever the orbit becomes empty, with the vacation duration being exponentially distributed with rate(ξ). During this working vacation, if a customer arrives, the server continues to operate but at a reduced service rate. This period is thus a lower-speed operational phase. If, at the moment of completing a service during this vacation, there are customers waiting in the orbit, the server immediately interrupts the vacation and resumes normal service, leading to a vacation interruption. However, if there are no customers present when the vacation ends, the server has two choices: with probability (γ) it remains in the system idle, ready to serve new arrivals at the regular service rate (single working vacation scenario), or with probability $(1-\gamma)$. it starts another working vacation (multiple working vacation scenario). Whenever a vacation concludes and customers are found in the orbit, the server switches back to its normal service mode. During the vacation period, the service time is modeled by a random variable (H_ϑ), with a distribution function $(H_\vartheta(x))$ and Laplace-Stieltjes transform $(H_\vartheta^*(t))$.

An illustration of the model under consideration is given below and its pictorial depiction is presented in Figure 1.

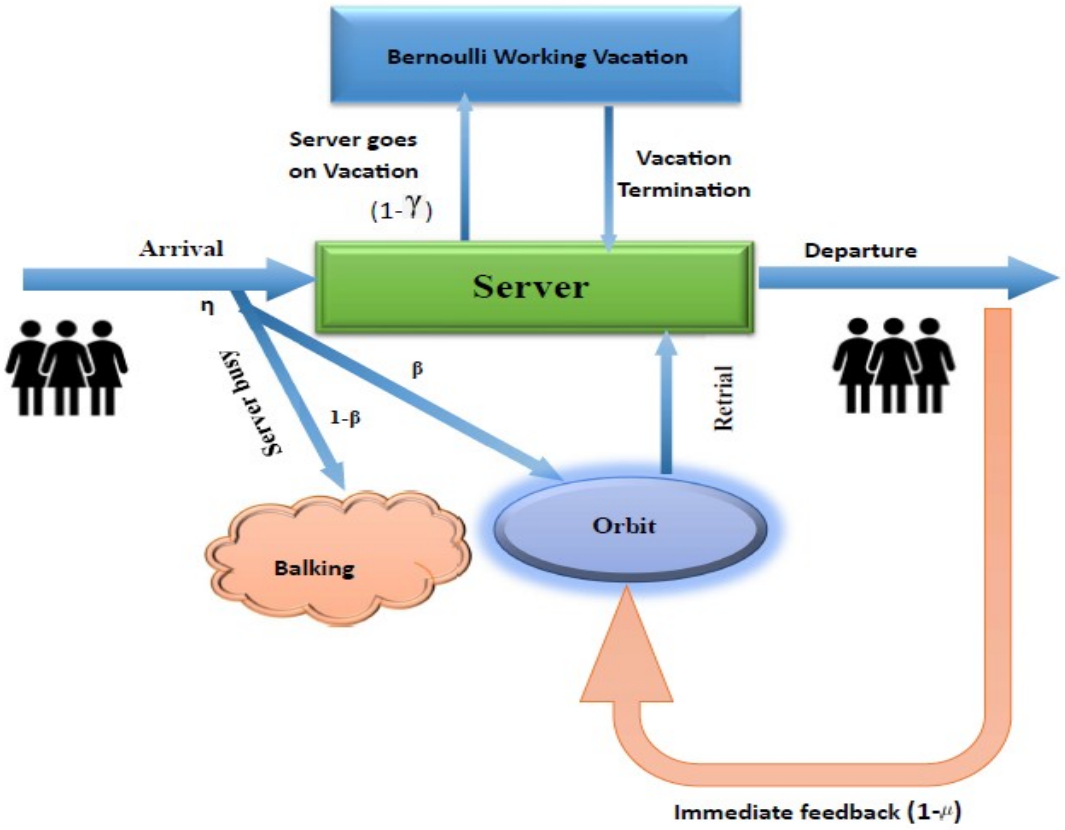


Figure 1. Schematic diagram

2.1. Real-world implementation of the model

A real-life example of a single-server retrial queue with balking and immediate feedback under a Bernoulli working vacation can be found in a hospital's diagnostic lab. Patients (customers) arrive to consult with a doctor (server). If the doctor is available, the patient is served immediately; otherwise, the patient either waits in a virtual queue (retrial orbit) or leaves without receiving service (balking), depending on their urgency and willingness to wait. After completing their consultation, some patients may require follow-up tests or (feedback) and re-enter the system. During low-demand periods, the doctor may go into a (working vacation), performing less critical tasks (e.g., paperwork or administrative duties) but still attending to patients at a slower pace if they arrive. If demand increases during vacation, the doctor interrupts their vacation and resumes full service immediately. This setup models both patient behavior and the physician's adaptive workflow under varying demand.

Our model can be illustrated in the context of a customer support center. Customers call the support line (server) for assistance. If an agent (server) is available, the call is handled immediately. If the agent is busy, the customer either retries later (retrial orbit) or hangs up (balking). After resolving the issue, some customers may require further clarification or follow-up (immediate feedback) and re-enter the system. During low-call periods, the agent might switch to a (working vacation) mode, such as handling back-office tasks or

responding to emails, where they operate at a slower pace but are still available to answer calls. If call volume increases, the agent ends the vacation and resumes normal operations. This system reflects dynamic customer behavior and agent workload adjustments.

3. Examination of the probability in a steady state

In this part, we discussed the steady-state equations for retrial queues by dealing the lapsed retrial, lapsed service and lapsed working vacation time as supplementary variable Technique. Then, we develop the PGF for the assistance condition and the number of arrivals in the retrial group and in the system

3.1. Formation of steady state equations

In steady state, we presume that $H_1(0) = 0$, $H_1(\infty) = 1$, $H_b(0) = 0$, $H_b(\infty) = 1$ and $H_\vartheta(0) = 0$, $H_\vartheta(\infty) = 1$, are continuous at $x = 0$. So that the function $\alpha_1(x)$, $\alpha_b(x)$ and $\alpha_\vartheta(x)$ are the conditional hazard rate for retrial, regular service, and working vacation.

$$\alpha_1(x)dx = \frac{dH_1(x)}{1 - H_1(x)}; \alpha_b(x)dx = \frac{dH_b(x)}{1 - H_b(x)}; \alpha_\vartheta(x)dx = \frac{dH_\vartheta(x)}{1 - H_\vartheta(x)}$$

Furthermore, let $H^0(\varphi)$, $H_b^0(\varphi)$ and $H_\vartheta^0(\varphi)$, be the passed retrial, the passed regular service, the passed working vacation, the period at time φ. Additionally, generate the random variable,

$$\Lambda(\varphi) = \begin{cases} 0, & \text{during when the server is empty and in vacation period} \\ 1, & \text{where the server is empty and during normal service hours} \\ 2, & \text{where the server is engaged and during normal service hours} \\ 3, & \text{where server is engaged and during vacation period} \end{cases}$$

The bi-variate Markov process $\{\Lambda(\varphi), N(\varphi); \varphi \geq 0\}$ describes the system's states at time φ. $\Lambda(\varphi)$ represents the state of the server (0,1,2,3), indicating idle, busy, or vacation. $N(\varphi)$ represents the number of consumers in the orbit. If $\Lambda(\varphi) = 0$ and $N(\varphi) \geq 0$, then $H^0(\varphi)$ represents the elapsed retrial time; if $\Lambda(\varphi) = 1$ and $N(\varphi) \geq 0$, then $H_b^0(\varphi)$ corresponds to the elapsed time of the customer serviced. If $\Lambda(\varphi) = 2$ and $N(\varphi) \geq 0$, $H_\vartheta^0(\varphi)$ represents the elapsed vacation time.

Theorem 3.1. *The embedded Markov chain $\{Z_n/n \in N\}$ is ergodic if and only if $\rho < H_1^*(\eta)$ for this system to be steady, where* $\rho = (-\eta\theta)\Big\{\beta_1 E(H_b) + \beta_1\beta_2(E(H_b))^2 + \ldots + (\beta_1\beta_2\ldots\beta_{m-1})(E(H_b))^{k-1}\Big\} - \eta\theta - \eta\theta E(H_b)$.

Proof. It is easy to verify the necessary condition of ergodicity using Foster's criteria [36], which asserts that the chain $\{F_n; n \in N\}$ is an irreducible and aperiodic chain. Assuming a non-negative measure $e(\varepsilon)$, $\varepsilon \in N$ and $\epsilon > 0$,the MC is ergodic, and mean value $\delta_\varepsilon = E[e(v_{n+1}) - e(v_n)/v_n = \varepsilon]$ with the limited exception ε's, $\varepsilon \in N$ and $\delta_\varepsilon \leq -\in$ for all $\varepsilon \in N$. In this case, considering $e(\varepsilon) = \varepsilon$. we get

$$\delta_\varepsilon = \begin{cases} \rho - 1, & \text{if } \varepsilon = 0 \\ \rho - H_1^*(\eta) & \text{if } \varepsilon = 1, 2, \ldots \end{cases}$$

However, it is obvious that ergodicity is required by $\rho < 1$.

As said by [37], if the MC $\{F_n; n\epsilon N\}$ matches Kaplan's status, generally $\delta_\varepsilon < \infty \; \forall \; \varepsilon \geq 0$ and $\exists \; \varepsilon_0 \in N$ s.t $\delta_\varepsilon \geq 0$ for $\varepsilon \geq \varepsilon_0$, the prerequisite is satisfactory. $W = (w_{k\varepsilon})$ is the unit-step transition matrix (UTM) of $\{F_n; n \in N\}$ for $\varepsilon < k - i$ and $k > 0$, where $W = (w_{k\varepsilon})$ is the UTM of $\{F_n; n \in N\}$. The MC's non-ergodicity of is provided by $\rho \geq H_1^*(\eta)$.

Let $\{\varphi_n; n = 1, 2, \dots\}$ represent the sequence of epochs that either result in the end of the service period or in a shorter service term. then we have to generate a random vector sequence $F_n = \{\Lambda(\varphi_n+), N(\varphi_n+)\}$. As a result of Theorem 3.1 $\{F_n; n \in N\}$ is ergodic iff $\rho < H_1^*(\eta)$ which means that for our system to be stable. □

For the method $\{N(\varphi), \varphi \geq 0\}$, we specify the probabilities,
$\Upsilon_0(\varphi) = P\{\Lambda(\varphi) = 0, N(\varphi) = 0\}$, $\Omega_0(\varphi) = P\{\Lambda(\varphi) = 0, N(\varphi) = 1\}$ and the probability densities are as follows:
$\Omega_1(x, \varphi)dx = P\{\Lambda(\varphi) = 1, N(\varphi) = n, x \leq H_1^0(\varphi) < x + dx\}$, for $\varphi \geq 0$, $x \geq 0$ and $n \geq 1$.
$\Omega_{i_b}(x, \varphi)dx = P\{\Lambda(\varphi) = 2, N(\varphi) = n, x \leq H_{b_i}^0(\varphi) < x + dx\}$, for $\varphi \geq 0$, $x \geq 0$, $i = 1, 2$ and $n \geq 0$.
$\Omega_\vartheta(x, \varphi)dx = P\{\Lambda(\varphi) = 3, N(\varphi) = n, x \leq H_\vartheta^0(\varphi) < x + dx\}$,for $\varphi \geq 0$, $x \geq 0$, $n \geq 0$ and $0 \leq i \leq k - 1$.
We assume that the stability requirement is satisfied in the sequel, so we may assign $\Omega_0 = lim_{\varphi\to\infty}\Omega_0(t)$, $\Upsilon_0 = lim_{t\to\infty}\Upsilon_0(t)$ and the limiting densities are
$\Omega_{1,n}(x) = lim_{t\to\infty}\Omega_{1,n}(x, t)$; $\Omega_{i_b,n}(x) = lim_{t\to\infty}\Omega_{i_b,n}(x, t)$; $\Omega_{\vartheta,n}(x) = lim_{t\to\infty}\Omega_{\vartheta,n}(x, t)$, $0 \leq i \leq k - 1$.
Using the supplemental variable strategy, we create the equation system shown below.

$$\eta\Omega_0 = \xi\gamma\Upsilon_0 \tag{3.1}$$

$$(\eta + \xi)\Upsilon_0 = \sum_{i=0}^{m-2} \bar{\beta}_{i+1} \int_0^\infty \Omega_{i_b,0}\, \alpha_b(x)dx + \int_0^\infty \Omega_{k-1_b,0}\,(x)\alpha_b(x)dx + \xi\bar{\gamma}\Upsilon_0 + \int_0^\infty \Omega_{\vartheta,0}(x)\alpha_\vartheta(x)dx \tag{3.2}$$

$$\frac{d}{dx}\Omega_{1,n}(x) + (\eta + \alpha_1(x))\Omega_{1,n}(x) = 0,\ n \geq 1 \tag{3.3}$$

$$\frac{d}{dx}\Omega_{i_b,n}(x) + (\eta + \alpha_b(x)) = \eta\bar{\theta}\Omega_{i_b,n}(x) + \eta\theta\Omega_{i_b,n-1}(x) \tag{3.4}$$

$$\frac{d}{dx}\Omega_\vartheta(x) + (\eta + \alpha_\vartheta(x))\Omega_{\vartheta,n}(x) = \eta\bar{\theta}\Omega_{i_b,n}(x) + \eta\theta\Omega i_b, n - 1(x) \tag{3.5}$$

The boundary conditions at $x = 0$, where the system is in steady state, are as follows:

$$\Omega_{1,n}(0) = \sum_{i=0}^{m-2} \bar{\beta}_{i+1} \int_0^\infty \Omega_{i_b,n}\alpha_b(x)dx + \int_0^\infty \Omega_{k-1_b,n}(x)\alpha_b(x)dx + \int_0^\infty \Omega_{\vartheta,n}(x)\alpha_\vartheta(x)dx \tag{3.6}$$

$$\Omega_{0_b}, n(0) = \int_0^\infty \Omega_{n+1}(x)\alpha_1(x)dx + \eta\int_0^\infty \Omega_n(x)dx + \xi\int_0^\infty \Omega_{\vartheta,n}(x)dx \tag{3.7}$$

$$\Omega_{i_b,n}(0) = \beta_i \int_0^\infty \Omega_{i-1_b,n}(x)\alpha_b(x)dx, j = 1, 2, 3, \dots, m - 1 \tag{3.8}$$

$$\Omega_{\vartheta,n}\,(0) = \begin{cases} \eta\gamma_0,\ n = 0 \\ 0,\ n \geq 1 \end{cases} \tag{3.9}$$

The normalization condition is

$$\Omega_0 + \gamma_0 + \sum_{n=1}^{\infty}\int_0^\infty \Omega_{1,n}(x)dx + \sum_{n=0}^{\infty}\left\{\sum_{i=0}^{k-1}\left[\int_0^\infty \Omega_{i_b,n}(x)dx\right] + \int_0^\infty \Omega_{\vartheta,n}(x)dx\right\} = 1 \tag{3.10}$$

3.2. The steady state solution

The equilibrium state result of the retrial queues is acquired by applying the generating function method. Defining the generating function for $|z| \leq 1$, concluding the above equations defined above (3.3) to (3.9), multiplying by z^n and summation $n = 0, 1, 2, 3 \dots$

$$\frac{\partial}{\partial x}\Big[\Omega_1(x,z)\Big] + \Big[\eta + \alpha_1(x)\Big]\Omega_1(x,z) = 0 \tag{3.11}$$

$$\frac{\partial}{\partial x}\Big[\Omega_{i_b}(x,z)\Big] + \Big[\eta\theta(1-z) + \alpha_b(x)\Big]\Omega_{i_b}(x,z) = 0 \tag{3.12}$$

$$\frac{\partial}{\partial x}\Big[\Omega_\vartheta(x,z)\Big] + \Big[\xi + \eta\theta(1-z) + \alpha_\vartheta(x)\Big]\Omega_\vartheta(x,z) = 0 \tag{3.13}$$

$$\Omega_1(0,z) = \sum_{i=0}^{k-2}\bar{\beta}_{i+1}\int_0^\infty \Omega_{i_b}(x,z)\alpha_b(x)dx + \int_0^\infty \Omega_{k-1_b}(x,z)\alpha_b(x)dx \tag{3.14}$$
$$+ \int_0^\infty \Omega_\vartheta(x,z)\alpha_\vartheta(x)dx - \Big[(\eta+\xi) - \xi\bar{\gamma}\Big]\Upsilon_0$$

$$\Omega_{0_b}(0,z) = \frac{1}{z}\int_0^\infty \Omega_1(x,z)\alpha_1(x)dx + \eta\int_0^\infty \Omega_1(x,z)dx\eta\Omega_0 + \xi\int_0^\infty \Omega_\vartheta(x,z)dx \tag{3.15}$$

$$\Omega_{i_b}(0,z) = \bar{\beta}_i\int_0^\infty \Omega_{i-1}\alpha_b(x)dx \tag{3.16}$$

$$\Omega_\vartheta(0,z) = \eta\Upsilon_0 \tag{3.17}$$

After some mathematical calculations, we attain partial differential equations

$$\Omega_1(x,z) = \Omega_1(0,z)\Big[1 - H_1(x)\Big]e^{-\eta x} \tag{3.18}$$

$$\Omega_{i_b}(x,z) = \Omega_{i_b}(0,z)\Big[1 - H_b(x)\Big]e^{-G(z)x} \tag{3.19}$$

$$\Omega_\vartheta(x,z) = \Omega_\vartheta(0,z)\Big[1 - H_\vartheta(x)\Big]e^{-G_\vartheta(z)x} \tag{3.20}$$

$$\Omega_{0_b}(0,z) = \frac{H(z)}{z}\Omega_1(0,z) + \eta\Upsilon_0 W(z) + \eta\Omega_0 \tag{3.21}$$

$$\Omega_{i_b}(0,z) = \Big[\prod_{i=0}^{k-1}\beta_i\Big]H_b^*(G(z))^k\left\{\frac{H(z)}{z}\Omega_1(0,z) + \eta W(z)\Upsilon_0 + \eta\Omega_0\right\} \tag{3.22}$$

$$\Omega_\vartheta(0,z) = \eta\Upsilon_0 \tag{3.23}$$

where, $G(z) = \eta\theta(1-z)$; $G_\vartheta(z) = \xi + G(z)$

$$H(z) = H_1^*(\eta) + z(1 - H_1^*(\eta)); \; W(z) = \frac{\xi(1 - H_\vartheta^*(G_\vartheta(z)))}{G_\vartheta(z)}$$

After some mathematical progress, we attain the following result.

$$\Omega_1(0,z) = \Upsilon_0\frac{Nr_1(z)}{Dr(z)} \tag{3.24}$$

$$Nr_1(z) = z\left\{\big[G(z)\chi(z) + \chi_1(z)\big]\big[\eta H_b^*(G(z))W(z)\xi\gamma\big] + \eta H_v^*(G_\vartheta(z))\big[(\eta+\xi) - \xi\bar{\gamma}\big]\right\}$$
$$Dr(z) = z - H(z)H_b^*(G(z))\big[G(z)(\chi_z) + \chi_1(z)\big]$$

$$\Omega_{i_b}(0,z) = \Upsilon_0\frac{Nr_2(z)}{Dr(z)} \tag{3.25}$$

$$Nr_2(z) = \{[\prod_{i=0}^{k-1}\beta_i]H_b^*(G(z))^k\{H(z)Nr_1(z) + \eta Dr(z)W(z)\} + \xi\gamma Dr(z)\} \tag{3.26}$$

where,

$$\chi(z) = \bar{\beta}_1 H_b^*(G(z)) + \sum_{i=1}^{k-1} \bar{\beta}_{i+1}(\beta_1\beta_2 \ldots \beta_i)[H_b^* G(z)]^{i+1}$$

$$\chi_1(z) = 1 + \beta_1 H_b^*(G(z)) + \beta_1\beta_2(H_b^*(G(z)))^2 + \cdots + (\beta_1\beta_2 \ldots \beta_{m-1})(H_b^*(G(z)))^{k-1}$$

We get $\Omega_1(x,z), \Omega_{i_b}(x,z), \Omega_\vartheta(x,z)$ by substituting the Equations (3.24) to (3.26) in (3.18) to (3.20). Next we focus on analyzing the marginal retrial group capacity expected to station condition of the server as per the following theorem.

Theorem 3.2. *The stationary distribution of the number of consumers in the orbit while the server is free, regular busy, reduced speed service, and the prob. that the server is free is provided by* $\rho < H_1^*(\eta)$ *under the stability condition*

$$\Omega_1(z) = \frac{\Upsilon_0[1 - H_1^*(\eta)]}{\eta} \frac{Nr_1(z)}{Dr_1(z)} \tag{3.27}$$

$$\Omega_{i_b}(z) = \frac{\Upsilon_0[1 - H_b^*(G(z))]}{G(z)} \frac{Nr_2(z)}{Dr(z)} \tag{3.28}$$

$$\Omega_\vartheta(z) = \frac{\eta\Upsilon_0[1 - H_\vartheta^*(G_\vartheta(z))]}{G_\vartheta(z)} \tag{3.29}$$

$$\Omega_0 = \frac{\xi\gamma\Upsilon_0}{\eta} \tag{3.30}$$

$$\Upsilon_0 = \frac{\begin{array}{r} H_1^*(\eta) + \eta\theta E(H_b) - \Big\{\Big(-\eta\theta\Big)\Big\{\beta_1 E(H_b) + \beta_1\beta_2(E(H_b))^2 + \ldots \\ + (\beta_1\beta_2 \ldots \beta_{m-1})(E(H_b))^{k-1}\Big\} - \eta\theta\Big\} \end{array}}{Denom[\Upsilon_0]} \tag{3.31}$$

where,

$$\begin{aligned} Denom(\Upsilon_0) = {} & \Big\{\frac{\xi\gamma + \eta}{\eta} + \frac{\eta(1 - H_\vartheta^*(\xi))}{\xi}\Big\}H_1^*(\eta) + \eta\theta E(H_b) - \Big\{(-\eta\theta)\Big\{\beta_1 E(H_b) \\ & + \beta_1\beta_2(E(H_b))^2 + \cdots + +(\beta_1\beta_2 \ldots \beta_{m-1})(E(H_b))^{k-1}\Big\} - \eta\theta\Big\} + \frac{(1 - H^*(\eta))}{\eta} \\ & \Big\{\frac{\eta(1 - H_\vartheta^*(\xi))}{\xi}[1 - \eta\theta E(H_b)] + \Big\{(-\eta\theta)\Big\{\beta_1 E(H_b) + \beta_1\beta_2(E(H_b))^2 + \ldots \\ & + (\beta_1\beta_2 \ldots \beta_{m-1})(E(H_b))^{k-1}\Big\} - \eta\theta[\eta E(H_\vartheta) + 1] + \frac{\eta}{\xi}\{\xi E(G_\vartheta) - G_\vartheta^*(\xi) + 1\} \\ & + \xi\gamma\Big\} - \eta\theta E(H_b)\Big\{\Big[\prod_{i=0}^{k-1}\beta_i\Big]\Big\{\frac{\eta(1 - H_\vartheta^*(\xi))}{\xi}[1 - \eta\theta E(H_b)] + \Big\{(-\eta\theta) \\ & \Big\{\beta_1 E(H_b) + \beta_1\beta_2(E(H_b))^2 + \cdots + (\beta_1\beta_2 \ldots \beta_{m-1})(E(H_b))^{k-1}\Big\} - \eta\theta[\eta E(H_\vartheta) \end{aligned}$$

$$+1]+\frac{\eta}{\xi}\Big\{\xi E(G_\vartheta)+1-G^*_\vartheta(\xi)\Big\}+\xi\gamma+\eta(1-H^*_\vartheta(\xi))H^*_1(\eta)+\eta\theta E(H_b)$$
$$-\Big\{(-\eta\theta)\Big\{\beta_1E(H_b)+\beta_1\beta_2(E(H_b))^2+\cdots+(\beta_1\beta_2\ldots\beta_{m-1})(E(H_b))^{k-1}\Big\}$$
$$-\eta\theta\Big\}\Big\}+\xi\gamma H^*_1(\eta)+\eta\theta E(H_b)-\Big\{(-\eta\theta)\Big\{\beta_1E(H_b)+\beta_1\beta_2(E(H_b))^2+\ldots$$
$$+(\beta_1\beta_2\ldots\beta_{m-1})(E(H_b))^{k-1}\Big\}-\eta\theta\Big\}$$

Proof. By integrating $\Omega_1(x,z),\Omega_{i_b}(x,z),\Omega_\vartheta(x,z)$ with respect to x and determining the partial PGF

$$\omega_1(z)=\int_0^\infty\Omega_1(x,z)dx,\omega_{i_b}(z)=\int_0^\infty\Omega_{i_b}(x,z)dx,\Omega_\vartheta(z)=\int_0^\infty\Omega_\vartheta(x,z)dx$$

we obtain the required equations. Then, the normalization condition given below, we can estimate the probability that the server is free (Ψ_0) when there is no consumer in the orbit by giving $z=1$ in (3.27)-(3.29), and utilizing the rule of l'Hospital anytime required.

$$\Omega_0+\Upsilon_0+\Omega_1(1)+\Omega_{i_b}(1)+\Omega_\vartheta(1)=1.$$

□

Theorem 3.3. *The stability constraint is used to compute the PGF for the number of consumers in the system and the orbit size distance at a stationary point of time.* $\rho<H^*_1(\eta)$,

$$A_s(z)=\frac{\Upsilon_0\Big\{z-H(z)H^*_b(G(z))\big[G(z)\chi(z)+\chi_1(z)\big]\big((\eta\theta(1-z))\Big\{(\frac{\xi\gamma+\eta}{\eta})+\frac{Z\eta\big[1-H^*_\vartheta(G^*_\vartheta(z))\big]}{G_{\vartheta(z)}}\Big\}+G(z)\Big(z\Big\{\big[G(z)\chi(z)+\chi_1(z)\big]\big[\eta H^*_b(G(z))W(z)\xi\gamma\big]+\eta H^*_v(G_\vartheta(z))\Big[(\eta+\xi)-\xi\bar{\gamma}\Big]\Big\}\Big)+Z(1-H^*_b(G(z)))z\Big\{\big[G(z)\chi(z)+\chi_1(z)\big]\big[\eta H^*_b(G(z))W(z)\xi\gamma\big]+\eta H^*_v(G_\vartheta(z))\big[(\eta+\xi)-\xi\bar{\gamma}\big]\Big\}\Big\}}{\eta\theta(1-z)\Big(z-H(z)H^*_b(G(z))\big[G(z)\chi(z)+\chi_1(z)\big]\Big)} \tag{3.32}$$

$$A_0(z)=\frac{\Upsilon_0\Big\{\eta\theta(1-z)\Big(Z-H(z)H^*_b(G(z))\big[G(z)\chi(z)+\chi_1(z)\big]\Big)\Big\{(\frac{\xi\gamma+\eta}{\eta})+\frac{\eta[1-H^*_\vartheta(G_\vartheta(z))]}{G_{\vartheta(z)}}\Big\}+(\eta\theta(1-z))\Big(z\Big\{\big[G(z)\chi(z)+\chi_1(z)\big]\big[\eta H^*_b(G(z))W(z)\xi\gamma\big]+\eta H^*_v(G_\vartheta(z))\big[(\eta+\xi)-\xi\bar{\gamma}\big]\Big\}\Big)+(1-H^*_b(G(z)))z\{\big[G(z)\chi(z)+\chi_1(z)\big]\big[\eta H^*_b(G(z))W(z)\xi\gamma\big]+\eta H^*_v(G_\vartheta(z))\big[(\eta+\xi)-\xi\bar{\gamma}\big]\}\Big\}}{\eta\theta(1-z)\Big(z-H(z)H^*_b(G(z))\Big[G(z)\chi(z)+\chi_1(z)\Big]\Big)} \tag{3.33}$$

where Ω_0 *is denoted by Equation* (3.30).

Proof. The PGF of the number.of consumer in system $(A_s(z))$ and in orbit $(A_0(z))$ is determined by using $A_s(z) = \Omega_0 + \Upsilon_0 + \Omega_1(z) + z\{\Omega_{i_b}(z) + \Omega_\vartheta(z)\}$ and $A_o(z) = \Omega_0 + \Upsilon_0 + \Omega_1(z) + \Omega_{i_b}(z) + \Omega_\vartheta(z)$. When the eqns. (3.27)-(3.29) are substituted in the previous findings, the Equations (3.32) and (3.33) can be calculated directly. □

4. Metrics of system performance

This section obtains many key system probabilities, system efficiency metrics, and the model's mean busy time and mean busy cycle while the system is in different states.

4.1. System state probabilities

We get the following results using Equations (3.27)–(3.29) by giving $z \to 1$ and using the L'Hospital's rule unless possible.
(i) Let $\Omega_1(1)$ be the steady-state probability that the server being free for the duration of the retrial.

$$\Omega_1(1) = \Upsilon_0 \frac{1 - H^*(\eta)}{\eta} \times \frac{Nr_1(1)}{Dr(1)} \tag{4.1}$$

where

$$\begin{aligned} Nr_1(1) = {} & \frac{\eta(1 - H^*_\vartheta(\xi))}{\xi}[1 - \eta\theta E(H_b)] \\ & + \Big\{(-\eta\theta)\Big[\beta_1 E(H_b) + \beta_1\beta_2(E(H_b))^2 + \cdots + (\beta_1\beta_2 \ldots \beta_{m-1})(E(H_b))^{k-1}\Big]\Big\} \\ & - \eta\theta[\eta E(H_\vartheta) + 1] \\ & + \frac{\eta}{\xi}\Big[\xi E(G_\vartheta) + 1 - G^*_\vartheta(\xi)\Big] + \xi\gamma, \\ Dr(1) = {} & H^*_1(\eta) + \eta\theta E(H_b) \\ & - \Big\{(-\eta\theta)\Big[\beta_1 E(H_b) + \beta_1\beta_2(E(H_b))^2 + \cdots + (\beta_1\beta_2 \ldots \beta_{m-1})(E(H_b))^{k-1}\Big] - \eta\theta\Big\}. \end{aligned}$$

(ii) Let $\Omega_{i_b}(1)$ be the steady-state probability that the server is busy.

$$\Omega_{i_b}(1) = \big(-\eta\theta E(H_b)\big) \times \frac{Nr_2(1)}{Dr(1)} \tag{4.2}$$

where

$$\begin{aligned} Nr_2(1) = {} & \Big[\prod_{i=0}^{k-1} \beta_i\Big]\Bigg\{ \\ & \frac{\eta(1 - H^*_\vartheta(\xi))}{\xi}[1 - \eta\theta E(H_b)](-\eta\theta) \\ & + (-\eta\theta)\Big[\beta_1 E(H_b) + \beta_1\beta_2(E(H_b))^2 + \cdots + (\beta_1\beta_2 \ldots \beta_{m-1})(E(H_b))^{k-1}\Big] \\ & - \eta\theta[\eta E(H_\vartheta) + 1] \\ & + \frac{\eta}{\xi}\Big[\xi E(G_\vartheta) + (1 - G^*_\vartheta(\xi))\Big] \\ & + \xi\gamma + \eta(1 - H^*_\vartheta(\xi))H^*_1(\eta) + \eta\theta E(H_b) \\ & - \Big\{(-\eta\theta)\Big[\beta_1 E(H_b) + \beta_1\beta_2(E(H_b))^2 + \cdots + (\beta_1\beta_2 \ldots \beta_{m-1})(E(H_b))^{k-1}\Big] - \eta\theta\Big\} \\ & + \xi\gamma H^*_1(\eta)\Bigg\}. \end{aligned}$$

(iii) Let $\Omega_{\vartheta}(1)$ be the steady-state probability that the server is on vacation.

$$\Omega_{\vartheta}(1) = \frac{\eta\,\Upsilon_0[1 - H^*_{\vartheta}(\xi)]}{\xi} \tag{4.3}$$

4.2. The mean length of a system and its orbit

When the system is in a steady state:
(i) Differentiating Equation (3.33) with respect to z and setting $z = 1$, we obtain the expected number of consumers in the orbit (L_q).

$$L_q = A_o^{'}(1) = \lim_{z\to 1}\frac{d}{dz}A_o(z) = \Upsilon_o\left[\frac{Ne_q^{'''}(1)De_q^{''}(1) - De_q^{'''}(1)Ne_q^{''}(1)}{3\,(De_q^{''}(1))^2}\right] \tag{4.4}$$

$$\begin{aligned}
Ne_q^{''}(1) = &-2\eta\theta H_1^*(\eta) + \eta\theta E(H_b) - \Big\{(-\eta\theta)\big[\beta_1 E(H_b) + \beta_1\beta_2(E(H_b))^2 + \dots \\
&+ (\beta_1\beta_2\dots\beta_{m-1})(E(H_b))^{k-1}\big] - \eta\theta\Big\}\Big\{\tfrac{\xi\gamma+\eta}{\eta} + \tfrac{\eta}{\xi}(1 - H^*_{\vartheta}(\xi))\Big\} - 2\eta\theta\Big\{\cdots\Big\},\\
De_q^{''}(1) = &-2\eta\theta H_1^*(\eta) + \eta\theta E(H_b) - \Big\{(-\eta\theta)\big[\beta_1 E(H_b) + \beta_1\beta_2(E(H_b))^2 + \dots \\
&+ (\beta_1\beta_2\dots\beta_{m-1})(E(H_b))^{k-1}\big] - \eta\theta\Big\},\\
De_q^{'''}(1) = &\,3\eta\theta\Big[\,2(1 - H_1^*(\eta))\{\cdots\} - 2\eta\theta E(H_b)\{\cdots\} + (\eta\theta)^2E(H_b)^2 + \cdots\Big],
\end{aligned}$$

$$\begin{aligned}
Ne_q^{'''}(1) = &-6\eta\theta H_1^*(\eta) + \eta\theta E(H_b)\\
&- \Big\{(-\eta\theta)\big[\beta_1 E(H_b) + \beta_1\beta_2(E(H_b))^2 + \cdots + (\beta_1\beta_2\dots\beta_{m-1})(E(H_b))^{k-1}\big] - \eta\theta\Big\}\{\cdots\}\\
&+ \Big(\tfrac{\xi\gamma+\eta}{\eta} + \tfrac{\eta}{\xi}(1 - H^*_{\vartheta}(\xi))\Big) + \cdots
\end{aligned}$$

(ii) By differentiating (3.32) with respect to z and setting $z = 1$, we obtain the expected number of consumers in the system (L_s).

$$L_s = A_s^{'}(1) = \lim_{z\to 1}\frac{d}{dz}A_s(z) = \Upsilon_0\left[\frac{Ne_s^{'''}(1)De_q^{''}(1) - De_q^{'''}(1)Ne_q^{''}(1)}{3\,(De_q^{''}(1))^2}\right] \tag{4.5}$$

$$\begin{aligned}
Ne_s^{'''}(1) = Nr_q^{'''}(1) &- 6\eta\theta H_1^*(\eta) + \eta\theta E(H_b) - \Big\{(-\eta\theta)\big[\beta_1 E(H_b) + \beta_1\beta_2(E(H_b))^2 + \dots\\
&+ (\beta_1\beta_2\dots\beta_{m-1})(E(H_b))^{k-1}\big] - \eta\theta\Big\}\Big\{\tfrac{\eta}{\xi}(1 - H^*_{\vartheta}(\xi))\Big\} + 6\eta\theta E(H_b)\Big[\prod_{i=0}^{k-1}\beta_i\Big]\{\cdots\}
\end{aligned}$$

(iii) The expected time that a consumer is in the system (W_s) and in the queue (W_q) is, according to Little's law, $W_s = L_s/\eta$ and $W_q = L_q/\eta$, respectively.

5. Special cases

This section deals with a few special cases of our approach which lead to concrete applications.
Case (i): No balking, no immediate feedback. Then we get a single-server retrial queue with Bernoulli vacation; the results coincide with [38].
Case (ii): No balking, no immediate feedback. Then we obtain an $M/G/1$ retrial queue with Bernoulli working vacation; the results coincide with [39].

6. Numerical results

MATLAB has been utilized in this section to demonstrate the range of possible outcomes for the dynamic behavior of the system. In addition, exponentially distributed retrials have been considered during regular service and in WV. To maintain stability, the numerical measurements are randomly selected. The computed values for various characteristics of the framework, such as the probabilities that the server is idle (Ω_0 and Υ_0), the average queue size L_q, the average system size (L_s), the probability that the server is serving regular service (Ω_b) and the probability that the server is in WV, are summarized in Tables 1 to 3 and illustrated using 2D graphs in Figure 2

As the retrial rate $\alpha_1(x)$ increases, Υ_0, L_q, L_s, Ω_v and W_q decreases. for the values of $\eta = 2$, β=0.5, θ=0.5, $\xi = 3$ and $\gamma = 0.9$ is shown in Table 1.

As the service rate $\alpha_b(x)$ increases, Υ_0, Ω_b, Ω_v, L_q, L_s and W_q decrease. for the values of $\eta = 2$, β=0.5, θ=0.5, $\xi = 3$ and $\gamma = 0.9$ is shown in Table 2.

As the lowest service rate ξ increases, L_q, L_s and W_q increases, Υ_0, Ω_b decreases. for the values of $\eta = 2$, β=0.5, θ=0.5 and $\gamma = 0.9$ is depicted in Table 3.

Table 1. The effect ot retrial rate $\alpha_1(x)$ on Υ_0, L_q, L_s, Ω_v and W_q

Retrial rate $\alpha_1(x)$	Υ_0	L_q	L_s	Ω_v	W_q
3.1	0.0765	5.5123	5.5376	0.0255	2.7561
3.2	0.0715	4.9533	4.9771	0.0238	2.4766
3.3	0.0669	4.4667	4.4892	0.0223	2.2333
3.4	0.0629	4.0415	4.0627	0.0209	2.0208
3.5	0.0591	3.6683	3.6883	0.0197	1.8342
3.6	0.0556	3.3394	3.3583	0.0185	1.6697
3.7	0.0525	3.0485	3.0664	0.0175	1.5242

Table 2. The effect of service rate $\alpha_b(x)$, on Υ_0, L_s, Ω_b, Ω_v, and W_s

Service rate $\alpha_b(x)$	Υ_0	L_s	Ω_b	Ω_v	W_s
5.1	0.0764	5.5316	0.6958	0.0254	2.7658
5.2	0.0763	5.5259	0.6830	0.0253	2.7630
5.3	0.0762	5.5205	0.6706	0.0252	2.7602
5.4	0.0761	5.5155	0.6586	0.0251	2.7577
5.5	0.0760	5.5106	0.6471	0.0250	2.7553
5.6	0.0759	5.5059	0.6360	0.0249	2.7410
5.7	0.0758	5.5015	0.6252	0.0248	2.7370

Table 3. The effect of lower service rate ξ on Υ_0, L_q, L_s,Ω_v, W_q

lower service rate (ξ)	Υ_0	L_q	L_s	Ω_v	W_q
3.1	0.0748	5.8139	5.8374	0.0241	2.9069
3.2	0.0732	6.1193	6.1409	0.0229	3.0596
3.3	0.0718	6.4282	6.4481	0.0217	3.2141
3.4	0.0703	6.7406	6.7588	0.0207	3.3703
3.5	0.0689	7.0561	7.0728	0.0197	3.5280
3.6	0.0676	7.3748	7.3899	0.0187	3.6874
3.7	0.0663	7.6965	7.7102	0.0179	3.8483

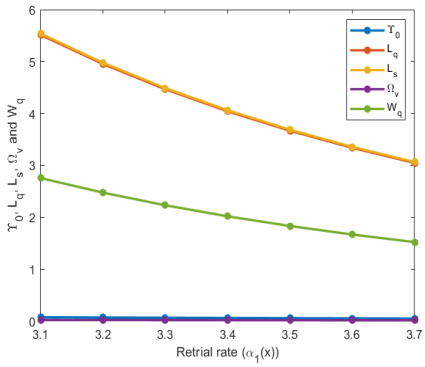
Figures 2 and 3 illustrate the impact of various system parameters using 2D and 3D graphs. In Figure 2(a), the surface displays that as the retrial rate $\alpha_1(x)$ escalates, Υ_0, L_q, L_s, Ω_v and W_q decreases. In Figure 2(b), we find that as the normal service rate $\alpha_b(x)$ increases, Υ_0, Ω_b, L_s, Ω_v and W_s decrease. In Figure 2(c), we find that as the lower service rate ξ, L_q, L_s, W_q increases, then Υ_0 and Ω_v decreases.

In Figure 3(a), the surface displays that as the retrial rate $\alpha_1(x)$ escalates, L_q, W_q decreases. In Figure 3(b), we find that as the normal service rate α_b, increases, Υ_0, L_s decreases. In Figure 3(c), we find that as the lower service rate ξ increases, then L_s, W_s, increases.

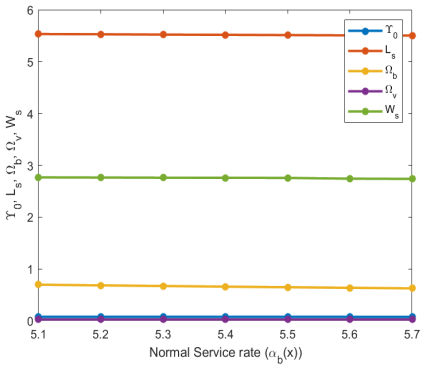
We use the numerical findings mentioned above to identify factors that impact the system's evaluation criteria and we confidently affirm that the results accurately reflect real-world scenarios.

Furthermore, the proposed queueing model is applicable to systems such as warehouse order processing or hospital pharmacy inventory management, offering meaningful insights into the system's performance and behavior. The key numerical results can be interpreted as follows:

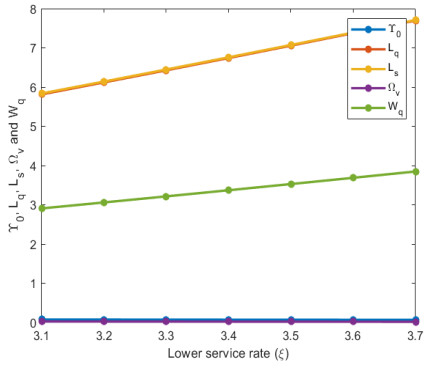
- Queue Length: The queue length indicates the number of customers (orders or patients) waiting in line at any moment. Tracking this metric helps identify times of peak demand or congestion, enabling effective resource planning and management.
- Server Utilization: This metric reflects the percentage of time the server (e.g., warehouse staff or pharmacist) is actively engaged in serving customers relative to the total available time. High utilization signifies efficient use of resources, while low utilization could point to underuse or excessive staffing.



(a) $\Upsilon_0, L_q, L_s, \Omega_v$ and W_q verses retrial rate $\alpha_1(x)$

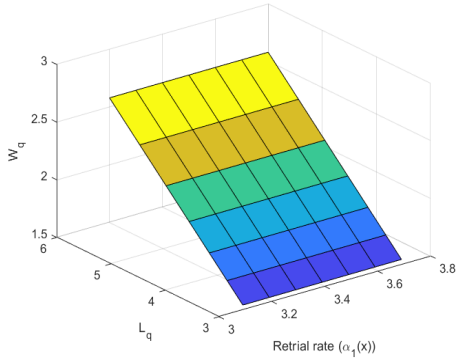


(b) $\Upsilon_0, L_q, \Omega_b, \Omega_v$ and W_q verses normal service rate $\alpha_b(x)$

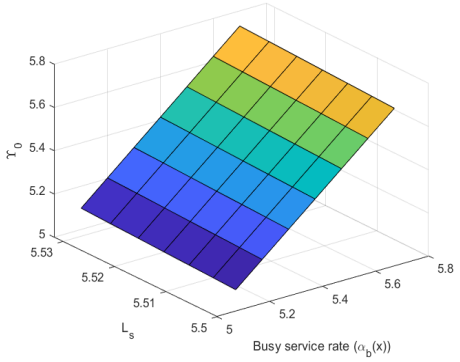


(c) $\Upsilon_0, L_q, L_s, \Omega_v$ and W_q verses lower service rate ξ

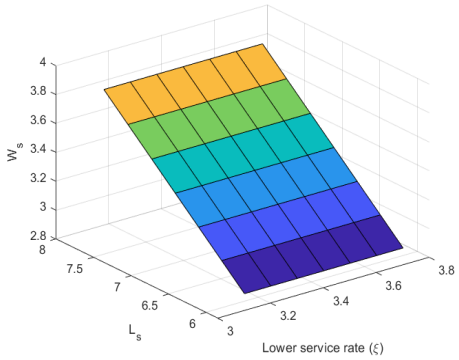
Figure 2. Two dimensional graph



(a) L_q, W_q verses retrial rate $\alpha_1(x)$



(b) Υ_0, L_s verses normal service rate $\alpha_b(x)$



(c) L_s, W_s verses lower service rate ξ

Figure 3. Three dimensional graph

7. Cost optimization

Optimization is the process of choosing the set of inputs to an objective function that yields the highest or lowest result. Cost optimization is the process of continuously concentrating on a business's operations to reduce expenditures and costs while also raising the firm's worth. It requires obtaining the best terms and prices for all business transactions in addition to standardizing, reducing, and rationing platforms, apps, processes, and services. The link between the system's profit and operating costs is rather close in a scenario that more closely reflects real life. This means that to maximize the profitability of the system, the main duty of system developers or administrators is to reduce the amount of money spent on operations for each unit of time. Here, our main goal is to identify the characteristics that enable us to calculate the optimal average cost per unit of time (CPUT). We want to accomplish this aim by including cost functionality into this section of our established model to make it more cost-effective.

Find the optimal values for the parameters, including the service and vacation rate (μ_b, μ_v) employing a cost optimization technique. In the projected cost function, it is assumed that the various activities of the system and the various cost components related to those activities have a linear connection. To obtain the predicted total cost function

(TC) for per unit time, the cost element variables are specified as follows:

$C_h \implies$ Holding cost of each client in the system for a predetermined amount of time
$C_b \implies$ CPUT when the server is normally active
$C_v \implies$ CPUT when the server is normally active
$C_1 \implies$ CPUT during busy period
$C_2 \implies$ CPUT spent during vacation period

In the light of this, the cost function is expressed as follows,

$$TC(\mu_b, \mu_v) = C_h L_q + C_b\{\Omega_{i_b}\} + C_v\Omega_\vartheta + C_1\mu_b + C_2\mu_v \tag{7.1}$$

Due to its substantial non-linearity, the cost function shown in Eqn. (7.1) is difficult to optimize analytically. Thus, we use a heuristic technique to minimize the overall cost, which depends on the service and vacation rates μ_b and μ_v.
Our main objective is to minimize the total cost function while determining the best service rate (μ_b^*) during the busy mode and the optimal service rate (μ_v^*) during the vacation mode. As a result, the cost minimization is expressed mathematically as follows:

$$TC(\mu_b^*, \mu_v^*) = \min_{\mu_b^*, \mu_v^*} TC(\mu_b, \mu_v)$$

In addition, the cost components mentioned in Table 4 are employed to provide a graphical representation of the cost function's sensitivity analysis.

Table 4. Cost sets for the purpose of cost analysis

Cost set	C_h	C_b	C_v	C_1	C_2
Set 1	80	70	85	115	40
Set 2	90	80	65	110	50
Set 3	100	60	50	70	70

7.1. Particle swarm optimization (PSO)

PSO is a computational method inspired by the collective behavior of bird flocks and fish schools. Introduced by [40], it was designed to optimize inherently non-linear functions. PSO is widely recognized for its simplicity and effectiveness in solving optimization problems within complex, multi-dimensional spaces. It is easy to implement, requires minimal parameter adjustment, and achieves rapid convergence to optimal or near-optimal solutions. Moreover, it performs effectively in both constrained and unconstrained optimization scenarios across various domains.
In cost optimization, PSO initializes a swarm of particles, where each particle represents a potential solution within the specified search space. The particles iteratively adjust their positions, influenced by their individual best-known positions and the swarm's global best-known position, progressively moving toward regions with lower cost values. This iterative process continues until a predefined stopping criterion is met, such as a set number of iterations or a target cost threshold. PSO's ability to efficiently explore complex, non-linear, and multi-modal cost landscapes makes it an effective solution for cost optimization across various fields.
Upadhyaya [41] extended this strategy to optimize costs in a discrete-time retrial queue (RQ) with Bernoulli feedback and initial failure. Zhang et al. [42] proposed computational and cost-effective solutions for a single-server recurrent system with state-dependent service using a PSO-based algorithm. For further insights into the functioning of PSO, the

study by [43] has been referenced. Harini [44] addressed the dynamical behavior and meta-heuristic optimization of a hospital management software system in her research.

7.2. PSO implementation

PSO is utilized in queueing models by treating system parameters, such as arrival rates and service rates, as particles. A fitness function is used to assess performance metrics such as waiting time or cost. The particles adjust their positions on the basis of their own best solutions and the best solution found by the swarm, iteratively converging toward the optimal configuration.

7.2.1. Steps involved

- Problem Representation: Model queue parameters (e.g., arrival rates, service rates) as particles in the search space.
- Cost Function: Combine factors such as operational costs and customer dissatisfaction into a unified objective.
- Initialization: Initialize the positions and velocities of the particles randomly within the defined range.
- Fitness Evaluation: Evaluate the cost of each particle based on the performance of the queueing model.
- Particle Update: Update velocities and positions using inertia, individual best positions, and global best position.
- Stopping Criteria: Stop the process when a set number of iterations is completed or when there is no significant improvement.
- Optimal Solution: Select the particle with the lowest cost as the best solution.

The default specifications were set to $\gamma = 0.8$, $\alpha_1(x) = 1.5$, $\alpha_b(x) = 3.5$, $\alpha_v(x) = 2$, and $\gamma = 0.9$ to optimize the cost. The values assigned to μ_v, namely 1, and 10, accordingly, indicate the lower and upper bounds. The parameters that correspond to the numberăof repetitions, the size of the population, inertial weight, and dual acceleration factors tend to be 100, 100, 1, and 2. The influence of various cost factors, notably C_h, C_b, C_v, C_1, and C_2, on the optimal service rates and the optimal total cost for each of the three cost sets is illustrated in Table 5.

Table 5. Effect of η, θ, ξ on (TC^*, μ_b^*, μ_v^*) using PSO

Parameters		(TC^*, μ_b^*, μ_v^*)		
		Cost set 1	Cost set 2	Cost set 3
η	1.1	(77.8471,5.4397,2.3410)	(59.6224,4.9484,2.9300)	(108.7397,4.4072,2.2914)
	1.2	(61.2975,5.7264,2.0383)	(47.2829,5.3306,2.1037)	(88.2092,5.3968,2.0981)
	1.3	(36.1398,5.9807,1.4709)	(45.3090,2.9853,2.0940)	(68.9083,8.3439,2.2292)
θ	0.5	(80.9873,5.2985,2.1041)	(72.3988,4.1950,1.0972)	(112.7906,4.6099,1.2166)
	0.6	(77.8791,5.0297,2.3210)	(56.2914,4.8091,1.2034)	(120.0997,4.0292,2.1094)
	0.7	(69.0103,4.0943,2.1099)	(45.0921,5.3011,2.8397)	(85.0981,5.0484,2.6480)
ξ	1.5	(103.2968,5.2017,2.1023)	(76.1097,5.0973,2.0029)	(129.1090,5.2095,2.2029)
	1.6	(80.0299,5.1022,2.1034)	(90.1092,5.1092,2.1082)	(136.0925,5.0911,2.9028)
	1.7	(105.0198,5.1088,2.9059)	(102.2038,5.1094,2.1987)	(108.1094,5.1025,2.2973)

In addition, the pseudocode for the PSO algorithm is provided in Alg-1. In addition, the PSO method is mathematically described by the iterative adjustment of particles (potential solutions) in a search space to minimize a given cost function is given as follows:

Mathematical model

(1) Initialization

- A swarm of n particles is initialized randomly, with each particle representing a potential solution to the optimization problem.
- Each particle i has a position vector $X_i(t)$ and velocity vector $V_i(t)$.

(2) Update Rules
- The velocity $V_i(t+1)$ is updated using
  $V_i(t+1) = wV_i(t) + c_1r_1(P_i - X_i(t) + c_2r_2(G - X_i(t)$
  where
  w= the inertia weight
  c_1 and c_2 = cognitive and social coefficients
  r_1 and r_2 = random numbers between 0 and 1
  P_i = the personal best position of particle i
  G = the global best position among all particles.
- The position $X_i(t+1)$ is updated as
  $X_i(t+1) = X_i(t) + V_i(t+1)$

(3) Objective Function and Termination:
- The cost function given in eqn. (7.1) is further minimized.
- The algorithm continues iterating until a stopping criterion is met, such as the maximum number of iterations or convergence of the cost function.

(4) Output:
Optimal service rates μ_b^* and μ_v^* and total cost $TC(\mu_b^*, \mu_v^*)$ are finally obtained.

Algorithm 1 Pseudo Code of PSO Algorithm

INPUT: Objective function = $TC(\mu_b, \mu_v)$, acceleration factors, inertia weight and Maximum number of iterations
OUTPUT: The cost function's value $TC(\mu_b^*, \mu_v^*)$
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 TC $\{F_1, \ldots, F_n\}$
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 TC^*

7.3. Genetic algorithm

The genetic algorithm (GA) is a population-based optimization technique inspired by the principles of natural selection and genetics. First introduced by Holland in 1975 [45], GA is designed to solve complex and non-linear optimization problems by mimicking biological evolutionary processes. It has gained widespread recognition for its robustness, adaptability, and global search capability across various application areas. GA is relatively easy to implement, highly parallelizable, and effective in navigating large, multi-dimensional search spacesparticularly where the objective function is discontinuous, noisy, or lacks gradient information.
In the context of cost optimization, GA begins with an initial population of chromosomes, where each chromosome encodes a potential solution. Through successive generations, the

population evolves using three primary genetic operators: selection, crossover, and mutation. Selection identifies fitter individuals for reproduction; crossover recombines pairs of chromosomes to explore new regions of the search space; and mutation introduces random variations to maintain diversity. This evolutionary process continues until a stopping criterionsuch as a maximum number of generations or the convergence of the cost functionis satisfied. The strength of GA in balancing exploration and exploitation makes it a powerful tool to minimize complex cost functions in various constrained and unconstrained optimization scenarios.
Mathavavisakan and Indhira [46] present a feedback retrial queue under a Bernoulli working vacation model with two dependent service phases, in which a nonlinear cost function is minimized using PSO, ABC, and GA. Jain and Dhibar [47] investigated ANFIS and metaheuristic optimization for a strategic joining policy with re-attempt and vacation.

Steps involved

- Problem Representation: Encode queue parameters (e.g., arrival rates, service rates) as chromosomes in the population.
- Cost Function: Define an objective function that combines operational cost, waiting time, and customer dissatisfaction.
- Initialization: Generate an initial population of chromosomes randomly within feasible parameter bounds.
- Fitness Evaluation: Evaluate each chromosome based on the cost function derived from the queueing model's performance.
- Selection: Choose parent chromosomes based on fitness, using techniques such as roulette wheel or tournament selection.
- Crossover: Perform crossover operations to generate new offspring by combining genes from selected parents.
- Mutation: Apply mutation with a small probability to maintain genetic diversity and avoid local optima.
- Stopping Criteria: Terminate the algorithm after a fixed number of generations or when fitness improvement plateaus.
- Optimal Solution: Identify the chromosome with the best (lowest) cost as the optimal solution.

The default specifications were set to $\gamma = 0.8$, $\alpha_1(x) = 1.5$, $\alpha_b(x) = 3.5$, $\alpha_v(x) = 2$, and $\gamma = 0.9$ to optimize the cost. The values assigned to μ_v, namely 1, and 10, accordingly, indicate the lower and upper bounds. The parameters that correspond to the numberăof repetitions, the size of the population, inertial weight, and dual acceleration factors tend to be 100, 100, 1, and 2. The influence of various cost factors, notably C_h, C_b, C_v, C_1, and C_2, on the optimal service rates and the optimal total cost for each of the three cost sets, is illustrated in Table 6.

In addition, the pseudocode for the GA algorithm is provided in Alg-2. In addition, the GA method is mathematically described by the evolutionary process of a population of candidate solutions through selection, crossover, and mutation operators to minimize a given cost function, as outlined below:

Mathematical model

(1) Initialization
   - An initial population of n chromosomes is generated randomly, where each chromosome encodes a potential solution to the optimization problem.
   - Each chromosome represents a vector of decision variables (e.g., service rates).

(2) Genetic Operators

Table 6. Effect of η, θ, ξ on (TC^*, μ_b^*, μ_v^*) using GA

Parameters		(TC^*, μ_b^*, μ_v^*)		
		Cost set 1	Cost set 2	Cost set 3
η	1.1	(97.5078,5.0942,2.4262)	(70.8299,4.4751,2.5914)	(109.6058,4.4645,2.4760)
	2.2	(89.3843,5.1118,2.3843)	(80.5206,4.4537,2.5194)	(104.0871,5.4351,2.3720)
	1.3	(105.0329,5.1380,1.3317)	(89.5969,4.4381,2.4393)	(112.0871,2.4351,2.3720)
θ	0.5	(93.2282,5.1069,2.4558)	(90.1998,4.4679,1.6137)	(120.9716,2.4505,1.4870)
	0.6	(84.7519,5.1234,2.4655)	(83.5852,5.4808,1.6259)	(85.4975,5.4671,2.4954)
	0.7	(92.2746,4.1397,2.4750)	(94.9698,5.4983,2.6334)	(108.0225,5.4837,2.5036)
ξ	1.5	(104.0133,4.1316,2.4703)	(102.9014,5.5045,2.6536)	(127.7601,4.4754,2.4995)
	1.6	(105.3345,5.1346,2.4856)	(90.0104,5.5183,2.6788)	(138.0993,5.4868,2.5168)
	1.7	(102.6385,5.1379,2.5007)	(104.1056,5.5322,2.7044)	(158.4194,5.4981,2.5341)

- **Selection:** Parent chromosomes are selected based on fitness using methods such as roulette wheel or tournament selection.
- **Crossover:** New offspring are generated by combining pairs of parents using crossover operators (e.g., single-point, two-point, or uniform crossover).
- **Mutation:** With a small probability, mutation is applied to offspring genes to maintain genetic diversity and avoid premature convergence.

(3) Objective Function and Termination:
- The cost function defined in Equation (7.1) is minimized by evaluating the fitness of each chromosome.
- The algorithm continues through generations until a termination condition is met, such as a maximum number of generations or convergence in cost value.

(4) Output:
The optimal service rates μ_b^* and μ_v^* and the corresponding minimum total cost $TC(\mu_b^*, \mu_v^*)$ are obtained from the best-fit chromosome.

Algorithm 2 Pseudo Code for GA Algorithm

INPUT: Objective function $TC(\mu_b, \mu_v)$
OUTPUT: Optimal cost value $TC(\mu_b^*, \mu_v^*)$

Calculate the objective function (OF)
Set the generation counter $t = 0$
Generate an initial population of users randomly: $P(t)$
Evaluate the population $P(t)$ using the objective function
while termination criterion is not satisfied **do**
    $t = t + 1$
    Select users from $P(t-1)$ to form a new population $P(t)$
    Apply crossover and mutation to the selected users
    Evaluate the new population $P(t)$ using the objective function
end while
return the best user (solution) found during the evaluation

7.4. Advantages of PSO and GA

PSO offers several benefits that make it ideal for complex optimization tasks. It is easy to implement and does not require gradient information, making it suitable for non-linear, non-differentiable, or multi-modal cost functions. PSO efficiently balances global exploration and local exploitation by using the collective behavior of particles based on their individual and global best experiences. Its ability to remember previous best positions

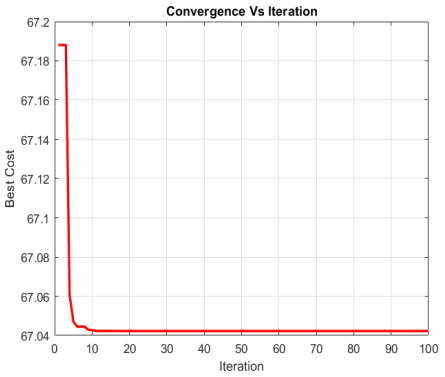
improves convergence speed and its low computational cost makes it especially valuable in time-sensitive optimization problems.

GA is a flexible and powerful optimization technique that mimics the principles of natural evolution. It can effectively handle large, complex, and discontinuous search spaces. By applying selection, crossover, and mutation, GA maintains population diversity and avoids getting stuck in local optima. It is particularly well-suited for discrete or combinatorial optimization problems. Unlike gradient-based methods, GA works without requiring differentiability or continuity, making it adaptable to a wide range of real-world scenarios.

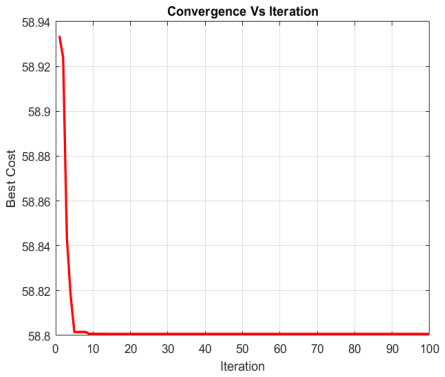
7.5. Convergence in PSO and GA

In cost optimization within queueing models, convergence refers to the process by which an algorithm iteratively adjusts system parameterssuch as service rates or scheduling policiesto minimize a predefined cost function involving factors such as customer wait times and operational costs. Although PSO and GA achieve convergence, PSO typically converges faster and more smoothly because of its cooperative learning strategy, where particles adjust their positions using both individual and collective experience. In contrast, GA relies on random evolutionary processes like crossover and mutation, which may require more iterations to stabilize. Thus, PSO offers superior convergence behavior, making it particularly effective and efficient for complex, time-sensitive cost optimization in queueing systems.

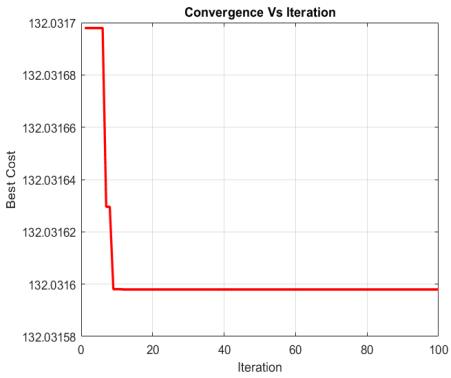
Figures 4 and 6 depict the convergence behavior of the cost function for PSO and GA, respectively, in three key parameters. In Figure 4, the PSO algorithm exhibits a rapid decrease in the best cost during initial iterations, highlighting its efficiency in quickly locating near-optimal solutions. The cost stabilizes after a certain number of iterations, indicating convergence to a global or near-global minimum. The smoothness of the curve beyond this point reflects the fine-tuning capability of PSO's, underscoring its robustness and reliability in optimization. In contrast, Figure 6 shows that the GA achieves a gradual but steady reduction in the best cost between generations, demonstrating its effectiveness in exploring the solution space and achieving convergence. As with PSO, the cost stabilizes after several generations, with minimal fluctuations, affirming the robustness of GA. Moreover, the convexity and optimality of the cost function are illustrated in Figures 5 and 7 for PSO and GA, respectively, validating both algorithms' ability to identify optimal parameter configurations across different cost sets.



(a) Convergence vs Iteration on η

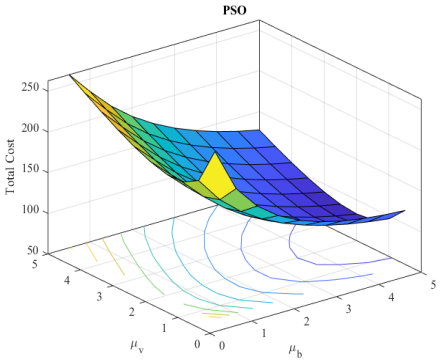


(b) Convergence vs Iteration on θ

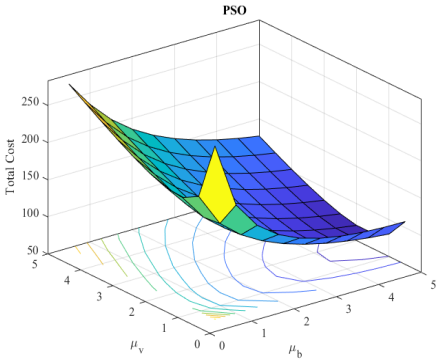


(c) Convergence vs Iteration on ξ

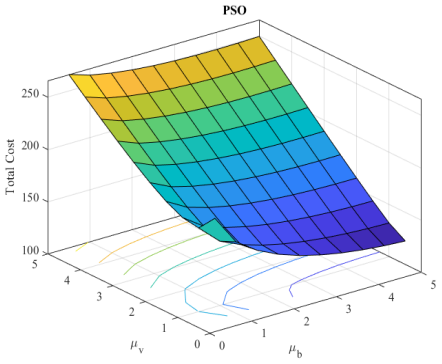
Figure 4. Convergence of the cost function using PSO



(a) TC vs μ_b, μ_v on η

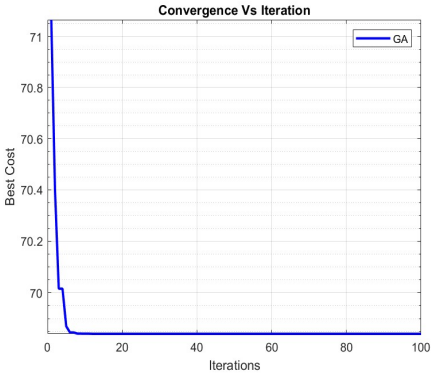


(b) TC vs μ_b, μ_v on θ

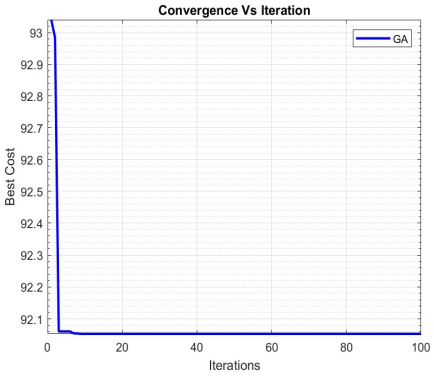


(c) TC vs μ_b, μ_v on ξ

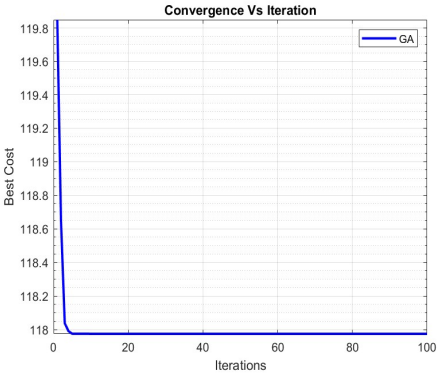
Figure 5. Optimality of the cost function using PSO



(a) Convergence vs Iteration on η

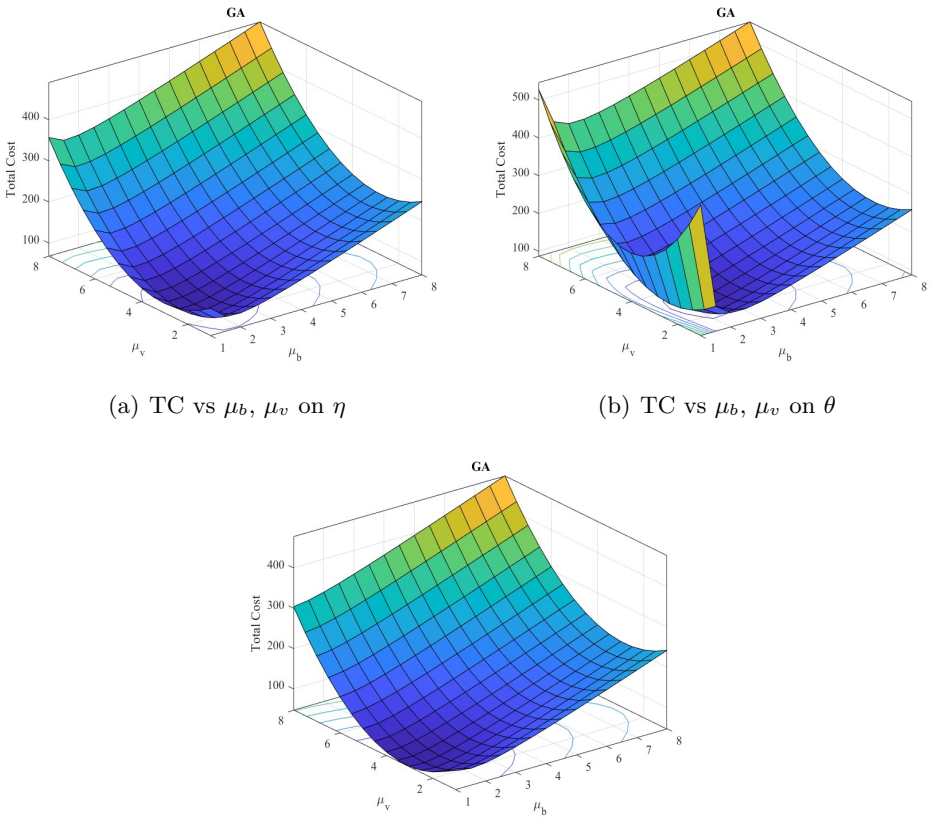


(b) Convergence vs Iteration on θ



(c) Convergence vs Iteration on ξ

Figure 6. Convergence of the cost function using GA



(a) TC vs μ_b, μ_v on η

(b) TC vs μ_b, μ_v on θ

(c) TC vs μ_b, μ_v on ξ

Figure 7. Optimality of the cost function using GA

8. Limitation of the study

- Single Server Assumption: The model considers only one server, which may not reflect real-world multi-server systems.
- Bernoulli Vacation Policy: The working vacation policy is restricted to a Bernoulli scheme, limiting applicability to other vacation disciplines.
- Static Parameters: System parameters (arrival rate, service rate, etc.) are assumed constant, ignoring time-varying behavior.
- Simplified Feedback and Balking: Feedback and balking behaviors are modeled in a simplified, probabilistic way, which may not capture complex customer psychology.
- Heuristic Optimization: PSO provides near-optimal solutions but does not guarantee global optimality.

9. Conclusion

This study presents a detailed investigation of a single-server retrial queueing system incorporating balking behavior, immediate feedback, and a Bernoulli working vacation strategy. A key assumption in the analysis is that the server initiates a working vacation immediately after completing a service. Through the application of the Supplementary Variable Technique (SVT), valuable insights have been obtained into the behavior and performance of the system. By exploring different configurations involving arrival rates, service durations, and other parameters, we have gained a clearer understanding of their impact on overall system efficiency. Furthermore, by applying optimization techniques tailored to the models structure, we achieved a cost reduction along with an improved service quality. This research deepens our understanding of complex single-arrival retrial queueing models and underscores the importance of optimization and empirical validation to enhance the practical relevance of theoretical frameworks.

9.1. Future research directions

Based on the present findings using GA and PSO for optimization, future research could focus on analyzing the transient behavior of retrial queueing systems to better understand their time-dependent dynamics. Expanding the model to incorporate features such as batch arrivals, multiple servers, or advanced vacation schemes would enhance its relevance to practical service scenarios. Additionally, the optimization framework can be strengthened by integrating other metaheuristic approaches such as Simulated Annealing (SA), Differential Evolution (DE), or Ant Colony Optimization (ACO), which may offer improved convergence rates and solution quality. These enhancements would contribute to more robust and efficient cost optimization in complex queuing environments.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of this paper.

Author contributions. All the authors have comtributed equally in all aspects of the preparation of this submission.

Conflict of interest statement. The author declare that they have no conflict of interest or personal relationship that could have appeared to influence the work reported in this paper.

Funding. This research received no external funding.

Data availability. No data was used for the research described in the article.

References

[1] P. Rajadurai, *A study on M/G/1 preemptive priority retrial queue with Bernoulli working vacations and vacation interruption* , Int. J. Process Manag. Benchmark. **9** (2), 193-215, 2019.

[2] T. Li, L. Zhang and S. Gao, *An M/G/1 retrial queue with balking customers and Bernoulli working vacation interruption*, Qual. Technol. Quant. Manag. **16** (5), 511-530, 2019.

[3] S. Keerthiga and K. Indhira, *Cost optimization for system stability and orbital search under working vacation and starting failure by using the ANFIS soft computing*, Ain Shams Eng. J. **15**, 102-847, 2024.

[4] M. 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** (10), 2022.

[5] S. Sundarapandiyan and S. Nandhini, *Non-Markovian Feedback Retrial Queue with Two Types of Customers and Delayed Repair Under Bernoulli Working Vacation* , Contemp. Math. **5** (2), 2093-2122, 2024.

[6] N. Sivasubramaniam and B. Jagannathan, *Bulk Arrival queue with Unreliable Server, Balking and Modified Bernoulli Vacation*, Hacet. J. Math. Stat. **53** (1), 289 304, 2024.

[7] N. Dehamnia, M. Boualem and D. Aïssani, *Performance and economic analysis of an unreliable single-server queue with general retrial times and varied customer patience levels*, Hacet. J. Math. Stat. **54** (2), 128, 2025.

[8] N. Dehamnia, M. Boualem and D. Aïssani, *Performance of an unreliable retrial queue with two types of customer arrivals and service orbit*, Yugosl. J. Oper. Res. **00**, 16-16, 2025.

[9] D. Arivudainambi and P. Godhandaraman, *A batch arrival retrialqueue with two phases of service, feedback and k-optional vacations*, Appl. Math. Sci. **6** (22), 10711087, 2012.

[10] J. C. Ke, T. H. Liu, S. Su and Z. G. Zhang, *On retrial queue with customer balking and feedback subject to server breakdowns*, Commun. Stat. Theory Methods. **51** (17), 6049-6063, 2022.

[11] R. P. Nithya and M. Haridass, *Stochastic modelling and analysis of maximum entropy of MX/G/1 queuing system with balking, startup and vacation interruption*, Int. J. Serv. Oper. **37** (3), 343-371, 2020.

[12] M. Boualem, Stochastic *analysis of a single server unreliable queue with balking and general retrial time*, Discrete Contin. Models Appl. Comput. Sci.**28** (4), 319-326, 2020.

[13] A. A. Bouchentouf, L. Medjahri, M. Boualem and A.Kumar, *Mathematical analysis of a Markovian multi-server feedback queue with a variant of multiple vacations, balking and reneging*, Discrete Contin. Models Appl. Comput. Sci. **30** (1), 21-38, 2022.

[14] A. A. Bouchentouf, M. Boualem, M. Cherfaoui and L. Medjahri, *Variant vacation queueing system with Bernoulli feedback, balking and server's states-dependent reneging* , YUJOR **31** (4), 557-575, 2021.

[15] K. C. Madan, D. Al-Nasser Amjad and A.Q. Al-Masri, *On M[x]/(G1,G2)/1 queue with optional re-service*, Appl. Math Comput. **152** (1), 7188, 2024.

[16] M. Baruah, K. C. Madan and T. Eldabi, *Balking and re-service in a vacation queue with batch arrival and two types of heterogeneous service,* J. Math. Res. **4** (4), 114124, 2012.

[17] P. Rajadurai, M. C. Saravanarajan and V. M. Chandrasekaran, *Analysis of an M[X]/(G1,G2)/1 retrial queueing system with balking, optional re-service under modified vacation policy and service interruption*, Ain Shams Eng. J. **5** (3), 935950, 2014.

[18] A.A Bouchentouf, M. Cherfaoui and M. Boualem, *Performance and economic analysis of a single server feedback queueing model with vacation and impatient customers*, Opsearch **56** (1), 300-323, 2019.

[19] V. Saravanan, V. Poongothai and P. Godhandaraman, *Performance analysis of a retrial queueing system with optional service, unreliable server, balking and feedback*, Int. J. Math. Eng. Manag. Sci. **8** (4), 769, 2023.

[20] M. Boualem, A. A Bouchentouf, A. Bareche and M. Cherfaoui, *Stochastic interpretation for a single server retrial queue with Bernoulli feedback and negative customers.* Applied Mathematics-A Journal of Chinese Universities **40** (1), 1-19, 2025

[21] A. A Bouchentouf, M. Boualem, L. Yahiaoui and H. Ahmad, *A multi-station unreliable machine model with working vacation policy and customers impatience*, Qual. Technol. Quant. Manag. **19** (6), 766796, 2022.

[22] M. Cherfaoui, A. A. Bouchentouf and M. Boualem, *Modelling and simulation of Bernoulli feedback queue with general customers' impatience under variant vacation policy*, Int. J. Oper. Res. **46** (4), 451-480, 2023.

[23] A. Chettouf, A. A. Bouchentouf and M. Boualem, *A Markovian Queueing Model for Telecommunications Support Center with Breakdowns and Vacation Periods*, In Oper. Res. **5** (1), 22, 2024.

[24] A. Dehimi, M. Boualem, A. A Bouchentouf, S. Ziani and L. Berdjoudj, *Analytical and Computational Aspects of a Multi-Server Queue With Impatience Under Differentiated Working Vacations Policy.* Reliab. Theory Appl. **19** (79), 393-407, 2024.

[25] B. Shanmugam and M. C. Saravanarajan, *Unreliable retrial queueing system with working vacation*, AIMS Math. **8** (10), 24196-24224, 2023.

[26] N. M. Mathavavisakan and K. Indhira, *Nonlinear metaheuristic cost optimization and ANFIS computing of feedback retrial queue with two dependent phases of service under Bernoulli working vacation*, Int. J. Mod. Phys. B. **38** (30), 2440004, 2024.

[27] K. Abir, T. Nassim, A. A. Bouchentouf and B. Mohamed, *Finite-capacity $M/M/2$ machine repair model with impatient customers, triadic discipline, and two working vacation policies.* Journal of Mathematical Modeling **13** (1), 2025.

[28] M. Vaishnawi, S. Upadhyaya and R. Kulshrestha *Optimal cost analysis for discrete-time recurrent queue with bernoulli feedback and emergency vacation.* Int. J .Appl. Comput Math. **8** (5), 254, 2022.

[29] A. Kumar and M. Jain, *Cost optimization of an unreliable server queue with two stage service process under hybrid vacation policy*, Math. Comput. Simul. **204** 259281, 2023.

[30] R. Harini and K. Indhira, *Meta heuristic optimization of a batch arrival retrial queue with optional re-service and M-optional vacations* , Int. J. Syst. Assur. Eng. Manag. **15** (9), 4252-4282, 2024.

[31] A. Kumar, M. Boualem and A. A. Bouchentouf, *Optimal analysis of machine interference problem with standby, random switching failure, vacation interruption and synchronized reneging*, Appl. Adv. Optim. Tech. Ind. Eng. 155-168, 2022.

[32] A. Kumar, A. A. Bouchentouf and M. Boualem; *Cost Optimisation Analysis for a Markovian Feedback Queueing System with Discouragement, Breakdown, and Threshold based Recovery Policy*, Optim. Tech. Decis. Mak. Inf. Secur. Comput. Intell. Data Anal. **3** (1), 1–17, 2024.

[33] A.Dehmi, M.Boualem, S.Kahla and L. Berdjoudj, *ANFIS computing and cost optimization of an M/M/c/M queue with feedback and balking customers under a hybrid hiatus policy.* Croatian Operational Research Review **15** (2), 159-170. 2024.

[34] M. Jain and A. Jain, *Genetic algorithm in retrial queueing system with server breakdown and caller intolerance with voluntary service.* Int. J. Syst. Assur. Eng. Manag. **13**(2), 582-598, 2022.
[35] G. Malik, S. Upadhyaya and R. Sharma, Cost inspection of a Geo/G/1 retrial model using particle swarm optimization and genetic algorithm. *Ain Shams Eng. J.* **12**(2): 2241-2254, (2021).
[36] A.G. Pakes, *Some conditions for ergodicity and recurrence of Markov chains.* Oper. Res. **17**, 1058-1061, 1969.
[37] L. I. Sennott, P. A. Humblet and R. L. Tweedie, *Mean drifts and the non-ergodicity of Markov chains,* Oper. Res. **31** (4), 783-789, 1983.
[38] P. Rajadurai, M. Saravanarajan and V. Chandrasekaran, *A single server retrial queue with bernoulli working vacation and vacation interruption.* Int. J. Appl. Eng. Res. **11** (1), 2016.
[39] D. Arivudainambi, P. Godhandaraman and P.Rajadurai, *Performance analysis of a single server retrial queue with working vacation.* Opsearch **51(3)**, 434-462, 2014.
[40] J.Kennedy and R. Eberhart, *Particle swarm optimization*, ieee. **4**, 1942–1948, 1995.
[41] S. Upadhyaya, *Cost optimisation of a discrete-time retrial queue with Bernoulli feedback and starting failure.* Int. J. Ind. Syst. Eng. **36** (2), 165-196, 2020.
[42] X. Zhang, J. Wang and Q. Ma, *Optimal design for a retrial queueing system with state-dependent service rate* J. Syst. Sci. Complex. **30** (4), 883-900, 2017.
[43] G. Malik, S. Upadhyaya and R. Sharma, *Cost inspection of a Geo/G/1 retrial model using particle swarm optimization and Genetic algorithm* , Ain Shams Eng. J. **12** (2), 2241-2254, 2021.
[44] R. Harini, and K. Indhira, *Dynamical behaviour and meta heuristic optimization of a hospital management software system,* Heliyon. **10** (16), 2024.
[45] J. H. Holland, *Adaptation in natural and artificial systems*, Univ. of Mich. Press **2**, 29-41, 1975.
[46] N. M. Mathavavisakan and K. Indhira, *Nonlinear metaheuristic cost optimization and ANFIS computing of feedback retrial queue with two dependent phases of service under Bernoulli working vacation*, Int. J. Mod. Phys. B. **38**(30), 2440004, 2024.
[47] M. Jain and S. Dhibar, *ANFIS and metaheuristic optimization for strategic joining policy with re-attempt and vacation.* Math. Comput. Simul. **211**, 57-84, 2023.