Contemporary Mathematics
http://ojs.wiserpub.com/index.php/CM/

UNIVERSAL WISER PUBLISHER

Research Article

Particle Swarm Optimization of a Single Server Retrial Queue with Delayed Repair Under Working Vacation From Optional Service to Re-Service

Kalaiselvi J, Saravanarajan MC*

Department of Mathematics, School of Advanced Sciences, Vellore Institute of Technology, Vellore, 632014, Tamil Nadu, India
E-mail: mcsaravanarajan@vit.ac.in

Received: 29 October 2024; **Revised:** 23 December 2024; **Accepted:** 1 January 2025

Abstract: Queueing systems (QS) play a critical role in modeling and optimizing various real-world processes by managing the flow of entities within systems involving queues. To address this, the proposed QS integrates features such as single arrivals, a retrial mechanism, optional re-service, working vacations, and delay repair scenarios. The system's dynamics are comprehensively analyzed using the supplementary variable technique, which provides deeper insights into its behavior and performance metrics. Furthermore, to enhance operational efficiency, an advanced cost optimization technique is employed to identify the optimal cost structure for the system. This holistic approach not only sheds light on the operational intricacies of the QS but also offers practical strategies for cost reduction and validation of analytical findings, thereby advancing queueing theory and broadening its practical applications across multiple domains.

Keywords: retrial queue, delayed repair, optional re-services, working vacation, particle swarm optimization

MSC: 68M20, 90B22, 60K25

1. Introduction

Waiting in queues is a common aspect of daily life, from long checkout lines at stores to traffic jams on highways. Retrial queues (RQ) are particularly valuable in scenarios where customers may return after leaving due to extended waits or inadequate service, thus improving system performance and customer retention. In a similar fashion, single-arrival retrial queues manage scenarios where entities arrive one by one and may attempt to access the service again after an initial failure, while single-phase service models provide a unified level of service to these customers. Optional re-service adds another layer of complexity, enabling customers to seek additional assistance, which is applicable in settings such as Automated Teller Machine (ATMs), banks, supermarkets, and hospitals. Extensive research has also explored RQ systems with server vacations, including working vacations, which allow servers to work even during vacation but at a slower rate, thus offering a more adaptive approach to service management. In addition, server failures in RQ systems can significantly disrupt operations, leading to longer wait times and increased customer retries, further complicating queue dynamics. Finally, cost optimization plays a critical role in designing efficient systems, focusing on balancing costs and performance. By carefully evaluating cost factors and constraints, organizations can improve operational efficiency, minimize waste,

DOI: https://doi.org/10.37256/cm.6120255992

and streamline processes. By integrating all the previously discussed scenarios, we present an innovative retrial queueing framework designed to predict and resolve potential bottlenecks, thus improving overall system performance and customer satisfaction.

1.1 *Literature survey*

To understand the research efforts undertaken and identify existing research gaps, a comprehensive and systematic review of previous studies is provided below.

In recent years, retrial queues (RQ) have emerged as a significant area of research. In addition, numerous studies have been published on retrial queueing systems. Rajadurai et al. [1] have analyzed a $M/G/1$ retrial QS with impatient customer, discretionary re-service subjected to a working vacation policy, and server interruption. Analysis of $M/G/1$ retrial queues with a second optional service and customer balking under two types of Bernoulli vacation schedule was analyzed by Madheshwari and Suganthi [2]. Santhi [3] analyzed an RQ with a second discretionary service and multiple working vacations. Arivudainambi and Godhandaraman [4] have discussed the retrial queueing system with balking, discretionary service, and vacation policy. Baskar and Saravanarajan [5] have discussed a feedback $M/G/1$ double retrial orbit queue with optional two-phase service and repair under the working vacation policy. Further, Rajam [6] has also studied more about the concept of re-service as well.

Vacation queues in queueing systems involve servers temporarily leaving for a short duration. Similarly, in queueing systems, servers may take working vacations, during which they continue to provide service at a reduced rate, based on predefined conditions. Gnanasekar [7] has discussed the analysis of an $M/G/1$ retrial queue with delayed repair and feedback under a working vacation policy with impatient customers. Rajadurai et al. [8] investigated the $M/G/1$ retrial queue with the priority customer under the multiple working vacation policy. Bharathi Shanmugam and Saravanarajan [9] studied an unreliable retrial queueing system with a working vacation policy. A steady-state analysis of single-server QS with multiple working vacations was explored by Seeniraj et al. [10]. Further, sensitivity analysis of a non-Markovian feedback retrial queue, reneging, delayed repair, with working vacation subject to server breakdown was scrutinized by Sundarapandiyan and Nandhini [11].

Many authors have investigated the delayed repair behavior. Recently, Revathi [12] addressed a single server RQ with delayed repair and optional re-service subjected to modified Bernoulli vacation. Gao et al. [13] analyzed a retrial queue with two types of breakdowns and delayed repairs. Singh et al. [14] studied an unreliable retrial G-queue with bulk arrival, optional additional service, and delayed repair. Particle swarm optimization and maximum entropy results for a $M^{[X]}/G/1$ retrial G-queue with delayed repair were investigated by Malik and Upadhyaya [15]. Jain and kumar [16] addressed the dynamical behavior of an unreliable $M^{[X]}/G/1$ queue with working vacation and multi-phase repair with delay in verification.

Limited research has been conducted on retrial queues that focus on cost optimization and the assessment of suitable control parameters for queueing models using various optimization techniques. Recently, Vaishnawi et al. [17] examined the accuracy of a $Geo^{[X]}/Geo/1$ recurrent model in discrete time, employing various optimization approaches to minimize system costs. Similarly, Kumar and Jain [18] analyzed a Markovian QS by formulating a cost function to identify optimal service system solutions. For further insights into cost optimization, readers can refer to the works of Deora et al. [19], Malik et al. [20], Jain and Jain [21], Jain and Kumar [22], and Vijaya Laxmi and Jyothsna [23]. Recently, Harini [24] employed distinct optimization techniques to analyze a batch arrival retrial queue with optional re-service and M-optional vacations.

1.2 *Research gap*

Section 1.1 provides an overview of the existing literature, highlighting gaps or inconsistencies that require further exploration. While Rajadurai et al. [1] analyzed a retrial queueing system with impatient customers, discretionary re-service under a working vacation policy. Revathi [12] studied a single-server retrial queue with delayed repair and optional re-service under a modified Bernoulli vacation policy. Harini [24] investigated the meta-heuristic optimization of a batch arrival retrial queue with optional re-service and M-optional vacations. There remains a gap in addressing the complexity of delayed repair with optional service transitioning to re-service under working vacation along with the cost optimization for such systems in light of this, this paper explores the performance of such model under varying operational conditions with the aid of an optimization technique.

1.3 *Research novelty*

Although previous studies have investigated discretionary re-service, delayed repair, and working vacation policies independently, the interplay between delayed repair mechanisms and the transition from optional service to re-service has not been thoroughly examined. To bridge this research gap, this work focus on a single-arrival retrial queue model incorporating optional re-service, working vacations, and delayed repair processes.

1.4 *Research objective*

This study aims to utilize particle swarm optimization to investigate and optimize the performance of a single-server retrial queue system, incorporating delayed repair, working vacation policies, and the transition from optional service to re-service, with a focus on minimizing costs and improving system efficiency.

1.5 *Research contribution*

Although research on retrial queues with optional re-service and working vacation has been conducted, a review of the literature reveals a significant gap in studies addressing single-arrival retrial queues under these conditions. Motivated by this observation, the primary contributions of this study are as follows:

• This study presents an innovative framework by developing a retrial queueing system as a single-arrival model featuring discretionary re-service, implemented within a working vacation setting, and further incorporating delayed repair, thereby bridging a critical gap in the existing literature.

• The probability-generating functions and steady-state probabilities for different server states were determined using the supplementary variable technique (SVT).

• The use of cost optimization techniques simplifies the process of identifying the most efficient cost structure, enhancing resource allocation and management strategies. This comprehensive approach ensures a thorough evaluation of cost-effectiveness, supporting decision-making processes and improving overall system performance.

2. Overview and analysis of the proposed framework

The $M/G/1$ queuing system with retrials is a popular model in queuing theory that includes retrial, normal service, lower rate service, delayed repair, and repair. This system involves a Poisson arrival process, and exponential service times with a finite capacity. An illustration of the model under consideration and its pictorial depiction (Figure 1) have been presented below. Let $R(t)$ represent the number of customers in the retrial state, $S(t)$ in normal service, $L(t)$ in lower rate service, $D(t)$ in delayed repair, and $P(t)$ in the repair state.

The arrival process: Positive customers enter the system following a Poisson process with an arrival rate of λ.

The retrial process: If an arriving customer finds the server available, they immediately begin service. Otherwise, the customer joins the orbit of blocked customers and continues to request service until the server becomes free. The retrial times follow a general distribution $R(x)$ with a Laplace-Stieltjes Transform (LST) denoted as $R^*(t)$.

The service process: During the regular busy period, a single server delivers standard service with an option for re-service. After completing the initial service, a positive customer can either opt for a repeat of the same service with probability P_r or exit the system with probability $1 - P_r$. It is assumed that re-service can occur only once. The service time is represented by a general random variable S with a distribution function $S(x)$ and a LST denoted as $S^*(t)$.

The working vacation process: When the orbit becomes empty, the server automatically begins a working vacation, characterized by an exponential distribution with rate μ. If customers arrive during this period, the server continues to operate but at a reduced speed. If an orbiting customer completes service during the vacation, the server interrupts the vacation and transitions back to the normal busy period, resulting in a vacation interruption. Otherwise, the vacation continues. If customers remain in the orbit when the vacation ends, the server resumes normal operations; otherwise, a new vacation begins. During the working vacation, the service time is modeled by a general random variable L, with a distribution function $L(x)$ and LST $L^*(t)$.

Breakdown process: The server may malfunction at any point when providing any kind of service or re-service, and the service channel will briefly stop functioning, meaning the server will be unavailable. Exogenous Poisson processes with rates b_d produce the breakdowns, or server life lifetimes. Governed by a general distribution $D(x)$ with its LST represented as $D^*(t)$.

Repair process: The server is sent for repair as soon as a breakdown happens, and while it waits for the repair to begin, it ceases serving incoming customers. This is known as the server's waiting period. Waiting time is defined as delay time. The duration of the repair is governed by a probability distribution $P(x)$, with its LST denoted as $P^*(t)$.

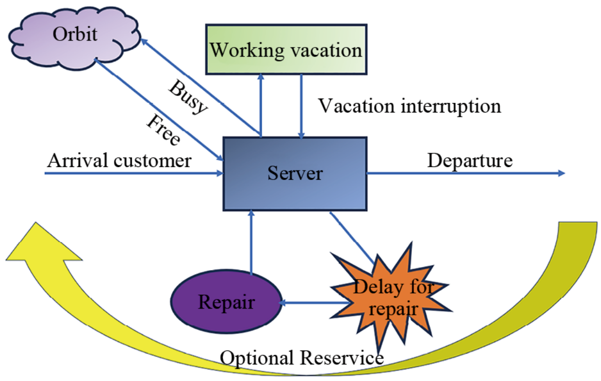


Figure 1. Schematic diagram

2.1 *Real-world implementation of the model*

In a healthcare setting, the concept of a single-server retrial queue with delayed repair under working vacation and optional re-service can be effectively applied to manage diagnostic facilities, such as patients arrive (arrival) at diagnostic facilities such as MRI, CT scan, or X-ray services, either scheduled or unscheduled, based on their medical needs or referrals. Upon arrival, they may join a queue if the facility is busy or retry (orbit) later if the system is inaccessible. The service process involves conducting diagnostic tests, with patients being attended to immediately or after waiting, depending on resource availability. The (optional re-service) feature accounts for situations where patients may require additional tests or re-evaluations after the initial scan, based on medical recommendations, before being discharged. In case of equipment breakdown, the (delayed repair) mechanism models the time needed to restore functionality, affecting patient scheduling and service availability. During periods of low patient inflow, the (working vacation) policy ensures efficient resource utilization by allowing machines or operators to perform at reduced efficiency, focusing on tasks like preventive maintenance, calibration, or simpler diagnostic services.

Our model finds application in airline ticket reservation processing systems, efficiently managing customer booking processes. Customers arrive (arrival) to book tickets either online, through mobile apps, or at physical counters. Arrivals can be scheduled, such as for group bookings, or unscheduled, depending on immediate travel needs or inquiries. Upon arrival, customers may join a queue if the system is busy or retry (orbit) later if access is unavailable due to high demand. The (service process) includes booking tickets, selecting seats, and managing travel preferences, with customers being attended to immediately or after waiting based on system availability. The (optional re-service) feature captures scenarios where customers may require additional services, such as seat upgrades, flight changes, or meal preferences, after booking, requiring them to re-enter the system. If the reservation system encounters a technical issue, the (delayed repair) mechanism models the time required to resolve the issue, temporarily halting new bookings. During periods of low customer demand, the (working vacation) policy allows the system to operate at reduced capacity, handling basic requests like flight inquiries or ticket modifications, while preparing for high-demand periods.

3. Examination of the probability in a steady state

The steady-state equations for the retrial system are initially formulated in this section by incorporating supplementary variables to represent the elapsed retrial times, normal service times, lower-speed service times, delay periods, and repair times. Subsequently, generating functions (GFs) for the orbit size corresponding to different server states are derived, along with the probability generating function (PGF) for the number of consumers present in the system and the orbit.

3.1 *The steady state equations*

In steady state, we presume that $A_n(0) = 0$, $A_n(\infty) = 1$, $S_{1,\, n}(0) = 0$, $S_{1,\, n}(\infty) = 1$ and $S_{2,\, n}(0) = 0$, $S_{2,\, n}(\infty) = 1$ are continuous at $a_p = 0$ and $W_{1,\, n}(0) = 0$, $W_{1,\, n}(\infty) = 1$, and $R_{1,\, n}(0) = 0$, $R_{1,\, n}(\infty) = 1$, $V_n(0) = 0$, $V_n(\infty) = 1$ are continuous at $a_p = 0$. Therefore, we define the hazard rate functions $A_n(a_p)$, $S_{1,\, n}(a_p)$, $S_{2,\, n}(a_p)$, $V_n(a_p)$, $W_{1,\, n}(a_p)$, $R_{1,\, n}(a_p)$ for retrial, normal service, optional re-service, lower rate service, delayed repair and repair, respectively.

$$A_n(a_p)da_p = \frac{dA_n(a_p)}{1-A_n(a_p)};\ S_{1,\, n}(a_p)da_p = \frac{dS_{1,\, n}(a_p)}{1-S_{1,\, n}(a_p)};$$

$$S_{2,\, n}(a_p)da_p = \frac{dS_{2,\, n}(a_p)}{1-S_{2,\, n}(a_p)};\ V_n(a_p)da_p = \frac{V_n(a_p)}{1-V_n(a_p)};$$

$$W_{1,\, n}(a_p)da_p = \frac{dW_{1,\, n}(a_p)}{1-W_{1,\, n}(a_p)};\ R_{1,\, n}(a_p)da_p = \frac{R_{1,\, n}(a_p)}{1-R_{1,\, n}(a_p)}.$$

Further, let $A_n^0(t)$, $S_{1,\, n}^0(t)$, $S_{2,\, n}^0(t)$, $V_n^0(t)$, $W_{1,\, n}^0(t)$, $R_{1,\, n}^0(t)$ be the elapsed retrial, elapsed regular service, elapsed optional re-service, elapsed working vacation, elapsed delay repair and elapsed repair time period at time t. Additionally, generate the random variable,

$$P(t) = \begin{cases} 0, & \text{when the server is idle} \\ 1, & \text{server is busy (providing service and re-service)} \\ 2, & \text{server is on vacation} \\ 3, & \text{server is delayed repair and providing normal service and optional re-service} \\ 4, & \text{server is repair and providing normal service and optional re-service.} \end{cases}$$

We also note that the states of the system at time t can be described by the bi-variate Markov process $\{\Delta(t),\ P(t);\ t \geq 0\}$ where $P(t)$ denotes the server state (0, 1, 2, 3, 4) depending on whether the server is idle, busy, optional re-service, on vacation, on delay repair, or repair. $P(t)$ denotes the number of customers in the orbit. If $\Delta(t) = 0$ and $P(t) \geq 0$, then $A_n^0(t)$ represents the elapsed retrial time. If $\Delta(t) = 1$ and $P(t) \geq 0$ then $S_{1,\, n}^0(t)(\ S_{2,\, n}^0(t))$ corresponds to the elapsed time of the customer being served(re-served). If $\Delta(t) = 2$ and $P(t) \geq 0$, then it corresponds to the elapsed vacation time. If $\Delta(t) = 3$ and $P(t) \geq 0$, then $W_{1,\, n}^0(t)$ is corresponding to the elapsed delay repair time. If $\Delta(t) = 4$ and $P(t) \geq 0$, then $R_{1,\, n}^0(t)$ is corresponds to the elapsed repair time.

Theorem 3.1 The embedded Markov Chain $\{F_n/n \in N\}$ is ergodic if and only if $\rho < 1$ for this system to be steady, where

$$\rho = \left(X^*(\lambda) - \lambda\left[E(Y_1)\left[1+\psi_1\left(E(Z_1)\right)\right] + hE(Y_2)\left[1+\psi_2\left(E(Z_2)\right)\right] + xE(Z)\right]\right)$$

Proof. It is easy to verify the necessary condition of ergodicity by using Foster's criteria [25], 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 $\varepsilon > 0$, 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, were considering $e(\varepsilon) = \varepsilon$. then 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 Humblett et al. [26], if the MC $\{F_n;\ n \in N\}$ 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 $\{t_n;\ n = 1,\ 2,\ \ldots\}$ 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 $F_n = \{\Delta(t_n+),\ P(t_n+)\}$. As a result of Theorem 3.1 $\{F_n;\ n \in N\}$ is ergodic iff $\tau < H_1^*(\eta)$ which means that for our system to be stable.

For the method $\{P(t),\ t \geq 0\}$, we specify the probabilities $P_0(t)$=P$\{\Delta(t) = 0,\ P(t) = 0\}$ and the prob. densities are

$A_n(a_p,\ t)da_p = P\{\Delta(t) = 0,\ P(t) = n,\ a_p \leq A_n(t) < a_p + da_p\}$, for $t \geq 0,\ a_p \geq 0$ and $n \geq 1$.

$S_{i,\ n}(a_p,\ t)da_p = P\{\Delta(t) = 1,\ P(t) = n,\ a_p \leq S_{i,\ n}(t) < a_p + da_p\}$, for $t \geq 0,\ a_p \geq 0,\ i = 1,\ 2$ and $n \geq 0$.

$V_n(a_p,\ t)da_p = P\{\Delta(t) = 2,\ P(t) = n,\ a_p \leq V_n(t) < a_p + da_p\}$, for $t \geq 0,\ a_p \geq 0,$ and $n \geq 0$.

$W_{1,\ n}(a_p,\ t)da_p = P\{\Delta(t) = 3,\ P(t) = n,\ a_p \leq W_{1,\ n}(t) < a_p + da_p\}$, for $t \geq 0,\ a_p \geq 0,$ and $n \geq 0$.

$R_{1,\ n}(a_p,\ t)da_p = P\{\Delta(t) = 4,\ P(t) = n,\ a_p \leq R_{1,\ n}(t) < a_p + da_p\}$, for $t \geq 0,\ a_p \geq 0,$ 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

$$A_n(a_p) = \lim_{t\to\infty} A_n(a_p,\ t);\ S_i(a_p) = \lim_{t\to\infty} S_i(a_p,\ t)(i = 1,\ 2);\ V_n(a_p) = \lim_{t\to\infty} V_n(a_p,\ t);$$

$$W_1(a_p) = \lim_{t\to\infty} W_1(a_p,\ t);\ R_1(a_p) = \lim_{t\to\infty} R_1(a_p,\ t).$$

4. Probability metrics

The following are various probability metrics for the server (idle, busy, repair, delayed repair and vacation).

$p_i(0, t)$ is the probability that the system is idle.

$A_n(a_p, t)$ is the number of customers in the orbit with the elapsed retrial time.

$S_{1,n}(a_p, t)$ is the probability number of customer in the orbit and getting services.

$S_{2,n}(a_p, t)$ is the probability of number of customer in the orbit and getting additional services.

$W_{1,n}(a_p, b_d, t)$ is the probability that the waiting time and optional re-service time of the customer when the server is breakdown (server under repair).

$R_{1,n}(a_p, b_d, t)$ is the probability of getting chance that n clients in the orbit with the elapsed service time during the delayed repair time of the server.

$V_n(a_p, t)$ is the probability of getting chance that number of customer in the orbit with the working vacation.

In general, the delay is denoted by b_d and time is denoted as t. The following stability conditions satisfied the conditional probabilities of:

$$P_0 = \lim_{n\to\infty} P_0(t), \quad \text{for } t \geq 0 \tag{1}$$

$$S_{1,n}(a_p, t) = \lim_{n\to\infty} S_{1,n}(a_p, t), \quad \text{for } t \geq 0,\ a_p \geq 0,\ n \geq 0 \tag{2}$$

$$W_{1,n}(a_p, b_d, t) = \lim_{n\to\infty} W_{1,n}(a_p, b_d, t), \quad \text{for } t \geq 0,\ a_p \geq 0,\ b_d \geq 0 \tag{3}$$

$$R_{1,n}(a_p, b_d, t) = \lim_{n\to\infty} R_{1,n}(a_p, b_d, t), \quad \text{for } t \geq 0,\ a_p \geq 0,\ b_d \geq 0 \tag{4}$$

$$V_n(a_p, t) = \lim_{n\to\infty} V_n(a_p, t), \quad \text{for } t \geq 0,\ a_p \geq 0 \text{ exist.} \tag{5}$$

Construct the stability conditions of equations (1) to (5) are

$$\lambda P_0 = (1 - P_l)\left[\overline{u}\int_0^\infty S_{1,n}(a_p, t)\ \mu_1(a_p)da_p + \int_0^\infty S_{2,n}(a_p, t)\ da_p\right] + \int_0^\infty V_n(a_p, t)a_p(p_l)\,da_p,$$

$$0 \leq l \leq n,\ \ l \leq \theta \leq n \tag{6}$$

$$\frac{dP_l(a_p)}{da} + (\lambda + P_l(a_p))P_l(a_p) = 0. \tag{7}$$

Consider the equations (2) to (5) , we have

$$\frac{d\,S_{1,\,n}(a_p,\,t)}{da_p}+\Big[\lambda+\psi_0(1)+\mu_1(a_p)\Big]S_{1,\,n}(a_p,\,t)=\int_0^\infty \gamma_1 S_{1,\,0}(a_p,\,b_d)db_d \tag{8}$$

$$\frac{d\,S_{2,\,n}(a_p,\,t)}{da_p}+\Big[\lambda+\psi_1+\mu_1(a)\Big]S_{1,\,n}(a_p,\,t)=\lambda\sum_{\theta=1}^{n}(\lambda+\psi_1)S_{1,\,n}(a_p,\,t)+\int_0^\infty \gamma S_{1,\,l}(1,\,b_d)db_d. \tag{9}$$

In case of break down we have the following equations

$$\frac{dW_{1,\,n}(a_p,\,b_d,\,t)}{db_d}+(\lambda+\tau_{ns}(t))W_{1,\,n}(a_p,\,b_d,\,t)=0. \tag{10}$$

Substitute the arrival rate and characteristic function in equation (10), we get

$$\frac{dW_{1,\,n}(a_p,\,b_d,\,t)}{db_d}+(\lambda+\tau_{ns}(t))W_{1,\,n}(a_p,\,b_d,\,t)=\lambda\sum_{\theta=1}^{n}(\lambda+\psi_1)W_{1,\,n-1}(a_p,\,b_d,\,t) \tag{11}$$

$$\frac{dW_{2,\,n}(a_p,\,b_d,\,t)}{db_d}+(\lambda+\tau_{ds}(t))W_{2,\,n}(a_p,\,b_d,\,t)=0. \tag{12}$$

By substituting arrival rate and characteristic function in equation (11), we get

$$\frac{dW_{2,\,n}(a_p,\,b_d,\,t)}{db_d}+(\lambda+\tau_{ds}(t))W_{2,\,n}(a_p,\,b_d,\,t)=\lambda\sum_{t=1}^{n}(\lambda+\psi_1)W_{2,\,n}(a_p,\,b_d,\,t) \tag{13}$$

$$\frac{dR_{1,\,0}(a_p,\,b_d)}{db_d}+(\lambda+\tau_{ns}(t))R_{1,\,0}(a_p,\,b_d)=0. \tag{14}$$

By substituting arrival rate and characteristic function in equation (14), we get

$$\frac{dR_{1,\,l}(a_p,\,b_d)}{db_d}+(\lambda+\tau_{ns}(t))R_{1,\,l}(a_p,\,b_d)=\lambda\sum_{\theta=1}^{n}(\lambda+\psi_1)R_{1,\,l-\theta}(a_p,\,b_d),\quad 0\le l\le n,\;\; l\le\theta\le n. \tag{15}$$

$$\frac{dR_{2,\,0}(a_p,\,b_d)}{db_d}+(\lambda+\tau_{ds}(t))R_{2,\,0}(a_p,\,b_d)=0. \tag{16}$$

By substituting arrival rate and characteristic function in equation (16), we get

$$\frac{dR_{2,\,l}(a_p,\,b_d)}{db_d}+(\lambda+\tau_{ds}(t))R_{2,\,l}(a_p,\,b_d)=\lambda\sum_{\theta=1}^{n}(\lambda+\psi_1)R_{2,\,l-\theta}(a_p,\,b_d),\quad l\leq\theta\leq n. \tag{17}$$

$$\frac{dR_{2,\,0}(a_p,\,b_d)}{db_d}+(\lambda+\tau_{os}(t))R_{2,\,0}(a_p,\,b_d)=0. \tag{18}$$

By substituting arrival rate and characteristic function in equation (18), we get

$$\frac{dR_{2,\,l}(a_p,\,b_d)}{db_d}+(\lambda+\tau_{os}(t))R_{2,\,l}(a_p,\,b_d)=\lambda\sum_{\theta=1}^{n}(\lambda+\psi_1)R_{2,\,l-\theta}(a_p,\,b_d) \tag{19}$$

$$\frac{dR_{2,\,0}(a_p,\,b_d)}{db_d}+(\lambda+\tau_{ors}(t))R_{2,\,0}(a_p,\,b_d)=0 \tag{20}$$

$$\frac{dR_{2,\,l}(a_p,\,b_d)}{db_d}+(\lambda+\tau_{ors}(t))R_{2,\,0}(a_p,\,b_d)=\lambda\sum_{\theta=1}^{n}(\lambda+\psi_1)R_{2,\,l-\theta}(a_p,\,b_d) \tag{21}$$

$$\frac{dV_n(a_p,\,t)}{da_p}+(\lambda+u(a_p))V_n(a_p,\,t)=0 \tag{22}$$

$$\frac{dV_n(a_p,\,t)}{da_p}+(\lambda+a_p(p_l)\,da_p)V_n(a_p,\,t)=0 \tag{23}$$

$$\frac{dV_n(a_p,\,t)}{da_p}+(\lambda+a_p(p_l)V_n(a_p,\,t)=\lambda\sum_{\theta=1}^{n}(\lambda+\psi_1)V_{n-1}(a_p,\,t). \tag{24}$$

The boundary conditions for the steady state are

$$P_0(0,\ t) = \int_0^\infty V_n(a_p,\ c)a_p\,(p_l)\,da_p + (1-x)\left[\overline{u}\int_0^\infty S_{1,\ n}(a_p,\ t)\mu_1(a_p)da_p \right. \tag{25}$$

$$\left. + (1-x)\int_0^\infty S_{2,\ n}(a_p,\ t)\mu_2(a_p)da_p - \lambda P_0\right] \tag{26}$$

$$S_{1,\ n}(a_p,\ t) = \int_0^\infty P_{l+1}\,(a_p)\,x\,(a_p)\,da_p + \lambda D_{l+1}P_\circ + \lambda\sum_{\theta=1}^{n} D_\theta\int_0^\infty P_{l-(\theta-1)}(a)da_p,\ T_1 \le D_\theta \le T_2 \tag{27}$$

$$S_{2,\ n}(a_p,\ t) = u\int_0^\infty S_{1,\ n}(a_p,\ t)\mu_1(a_p)da_p \tag{28}$$

$$W_{1,\ n}(a_p,\ b_d,\ t) = \psi_1 S_{1,\ n}(a_p,\ t) \tag{29}$$

$$W_{2,\ n}(a_p,\ b_d,\ t) = \psi_2 S_{2,\ n}(a_p,\ t) \tag{30}$$

$$W_{1,\ n}(a_p,\ b_d,\ T_1) = \psi_1 S_{1,\ n}(a_p,\ t) \tag{31}$$

$$W_{2,\ n}(a_p,\ b_d,\ T_2) = \psi_2 S_{2,\ n}(a_p,\ t) \tag{32}$$

$$R_1(a_p,\ 0,\ t) = \int_0^\infty S_{1,\ n}(a_p,\ t)a_p\,(p_l)\,(b_d)db_d \tag{33}$$

$$R_2(a_p,\ 0,\ t)=\int_0^{\infty} S_{2,\ n}(a_p,\ t)a_p\,(p_l)\,(b_d)db_d \tag{34}$$

$$R_1(a_p,\ 0,\ T_1)=\int_0^{\infty} S_{1,\ n}(a_p,\ t)a_p\,(p_l)\,(b_d)db_d \tag{35}$$

$$R_2(a_p,\ 0,\ T_2)=\int_0^{\infty} S_{2,\ n}(a_p,\ t)a_p\,(p_l)\,(b_d)db_d \tag{36}$$

$$V_n(a_p,\ tb_d)=\overline{u}\int_0^{\infty} S_{1,\ n}(a_p,\ b_d)\mu_1(a_p)da_p+\int_0^{\infty} S_{2,\ n}(a_p,\ b_d)\mu_2(a_p)da_p \tag{37}$$

$$V_n(a_p,\ t)=x\overline{u}\int_0^{\infty} S_{1,\ n}(a_p,\ t)\mu_1(a_p)da_p+x\int_0^{\infty} S_{2,\ n}(a_p,\ t)\mu_2(a_p)da_p \tag{38}$$

$$V_n(a_p,\ T_1)=\overline{u}\int_0^{\infty} S_{1,\ n}(a_p,\ b_d)\mu_1(a_p)da_p+\int_0^{\infty} S_{2,\ n}(a_p,\ b_d)\mu_2(a_p)da_p \tag{39}$$

$$V_n(a_p,\ T_1)=x\overline{u}\int_0^{\infty} S_{1,\ n}(a_p,\ t)\mu_1(a_p)da_p+x\int_0^{\infty} S_{2,\ n}(a_p,\ t)\mu_2(a_p)da_p. \tag{40}$$

Finally the normalizing condition is

$$P_o+\int_0^{\infty} P_l(a_p)da_p+\int_0^{\infty} S_1,\ n(ap,\ t)da_p+\int_0^{\infty} S_2,\ n(ap,\ t)da_p \tag{41}$$

$$+\int_0^{\infty} V_n(ap)da_p+\int_0^{\infty}\int_0^{\infty} W_{1,\ n}(ap,\ bd,\ t)da_pdb_d \tag{42}$$

$$+\int_0^{\infty}\int_0^{\infty} W2,\ n(a_p,\ b_d,\ t)da_pdb_d+\int_0^{\infty}\int_0^{\infty} R_1,\ l(a_p,\ b)da_pdb_d+\int_0^{\infty}\int_0^{\infty} R_2,\ l(a_p,\ b)da_pdb_d=1 \tag{43}$$

$$P\,(a_p,\ T,\ \ t)=e^{-\int(1+x(a_p))da_p}\quad,\ T_1\leq T\leq T_2. \tag{44}$$

5. Steady state solutions

Solving the above the differential equations (7) to (24), we get

$$P(a_p,\ t)=P(0,\ t)\left[1-X(a_p)\right]e^{-\lambda a_p} \tag{45}$$

$$P(b_d,\ t)=P(0,\ t)\left[1-T(b_d)\right]e^{-\lambda b_d} \tag{46}$$

$$M_1(a_p,\ t) = M_1(0,\ t)\left[1 - T_1(a_p)\right]e^{-C_1(t)a_p} \tag{47}$$

$$M_2(a_p,\ t) = M_2(0,\ t)\left[1 - T_2(a_p)\right]e^{-C_2(t)a_p} \tag{48}$$

$$M_1(b_d,\ t) = M_1(0,\ t)\left[1 - T_1(b_d)\right]e^{-C_1(t)b_d} \tag{49}$$

$$M_2(b_d,\ t) = M_2(0,\ t)\left[1 - T_2(b_d)\right]e^{-C_2(t)b_d} \tag{50}$$

$$W_{1,\ n}(a_p,\ t)\ = W_{1,\ n}(a_p,\ t)\left[\ 1 - W_{ns}(b_d)\right]e^{-r(t)a_p} \tag{51}$$

$$W_{2,\ n}(a_p,\ t) = W_{2,\ n}(a_p,\ t)\left[1 - W_{os}(a_p)\right]e^{-r(t)a_p} \tag{52}$$

$$W_{1,\ n}(b_d,\ t)\ = W_{1,\ n}(b_d,\ t)\left[1 - W_{ns}(b_d)\right]e^{-r(c)b_d} \tag{53}$$

$$W_{2,\ n}(b_d,\ t) = W_{2,\ n}(b_d,\ t)\left[1 - W_{os}(b_d)\right]e^{-r(t)b_d} \tag{54}$$

$$R_{1,\ n}(a_p,\ t) = R_{1,\ n}(a_p,\ t)\ \left[1 - S_{rs}(a_p)\right]e^{-r(t)a_p} \tag{55}$$

$$R_{2,\ n}(a_p,\ t) = R_{1,\ n}(a_p,\ t)\ \left[1 - S_{os}(a_p)\right]e^{-r(t)a_p} \tag{56}$$

$$R_{1,\ n}(b_d,\ t) = R_{1,\ n}(b_d,\ t)\ \left[1 - S_{rs}(b_d)\right]e^{-r(t)b_d} \tag{57}$$

$$R_{2,\ n}(b_d,\ t) = R_{1,\ n}(b_d,\ t)\ \left[1 - S_{os}(b_d)\right]e^{-r(t)b_d}. \tag{58}$$

In general,

$$Z(a_p,\ t) = Z(0,\ t)\left[1 - Z(a_p)\right]e^{-r(t)a_p} \tag{59}$$

$$Z(b_d,\, t) = Z(0,\, t)[1 - Z(b_d)]e^{-r(t)b_d}.$$

The sequence of random values $F_i^*(h(Z))$ along with Markov chain is ergodic and hence

$$A_1(a_p,\, t) = h(Z) + \psi_1 \left[1 - C_1^*(h(Z))F_1^*(h(Z))\right]$$

$$A_2(a_p,\, t) = h(Z) + \psi_2 \left[1 - C_2^*(h(Z))F_2^*(h(Z))\right]$$

and

$$h(Z) = \lambda(1 - X(Z)).$$

From(27), we have

$$S_{1,\, n}(a_p,\, t) = \frac{P(0,\, t)}{t}\left[X^*(\lambda) + (1 - X^*(\lambda))A(t)\right] + \frac{\lambda A(t)}{t}P_0. \tag{60}$$

From (28)

$$S_{2,\, n}(a_p,\, t) = uM_1(0,\, t)Y_1^*(X_1(t)). \tag{61}$$

Simplifying (29) to (32) and (50) to (53), we get

$$W_{1,\, n}(a_p,\, b_d,\, t) \ = \psi_1(M_1(0,\, t))(1 - Y_1(a_p))e^{-X_1(t)a_p} \tag{62}$$

$$W_{1,\, n}(a_p,\, b_d,\, t) \ = \psi_1 M_2(0,\, t)(1 - Y_2(a_p))e^{-X_2(t)a_p}. \tag{63}$$

Simplifying (33) to (36) and (54) to (57), we get

$$R_1(a_p,\, 0,\, t) = W_{1,\, n}(a_p,\, b_d,\, t)\; Z_1^*(r(t)) \tag{64}$$

$$R_2(a_p,\, 0,\, t) = W_{1,\, n}(a_p,\, b_d,\, t)\; Z_2^*(r(t)). \tag{65}$$

From (37) to (40), we get

$$V_n(0,\ t) = x\overline{u}M_1(0,\ t)Y_1^*(X_1(t)) + xM_2(0,\ t)Y_2^*(M_2(t)). \tag{66}$$

Hence,

$$P(0,\ t) = \frac{\pi_1 u(t)}{\pi_2 u(t)} \tag{67}$$

$$\pi_1 u(t) = \lambda P_0 \left\{ A(t)Y_1^*(X_1(t)) \left[\left(xhZ^*(\lambda_0(t)) + (1-x)h \right) Y_2^*(X_2(t)) + (x\overline{u}z^*(\lambda_0(t)) + (1-x)\overline{u}) \right] - t \right\}$$

and

$$\pi_2 u(t) = \left\{ t - \{A(t) + (1-A(t))X^*(\lambda)\} Y_1^*(X_1(t)) \left[(xhZ^*(\lambda_\circ(t)) + (1-x)h)Y_2^*(X_2(t)) \right.\right.$$

$$\left.\left. + (x\overline{u}Z^*(\lambda_\circ(t)) + (1-x)\overline{u}) \right] \right\}.$$

From (47) to (50), we have,

$$M_1(0,\ t) = \frac{\lambda A(t)}{t} P_0 + \frac{P(0,\ t)}{t} [(A(t) - (1-A(t))X_1{}^*(\lambda)] \tag{68}$$

$$M_2(0,\ t) = u\lambda P_0 + \left[\frac{(A(t) - (1-A(t))X_1{}^*(\lambda)}{Gu(t)} \right] y_1^* X_1(t). \tag{69}$$

From (62) and (63), we obtain

$$W_{1,\ n}(a_p,\ b_d,\ t)\ = \psi_1 \left[\frac{\lambda P_0(A(u)-1)X^*(\lambda)}{\pi_2 u(t)} \right] (1 - y_1(a))e^{-x_1(t)a_p} \tag{70}$$

$$W_{2,\ n}(a_p,\ b_d,\ t)\ = \psi_2 u\lambda P_0 \left[\frac{(A(u)-1)X^*(\lambda)}{\pi_2 u(t)} \right] (1 - y_2(a))e^{-x_2(t)a_p}. \tag{71}$$

From (64) and (65),

$$R_1(a_p,\ 0,\ t) = \lambda P_0 \psi_1 \left[\frac{(A(t-1)X^*(\lambda)}{\pi_2 u(t)} \right] (1 - y_1(a_p))e^{-x_1(c)a} w_1{}^*(r(t)) \tag{72}$$

$$R_2(a_p,\ 0,\ t) = \lambda P_0 \psi_1 \left[\frac{(A(t-1)X^*(\lambda)}{\pi_2 u(t)}\right](1-y_2(a_p))e^{-x_2(t)a}w_2{}^*(r(t)). \tag{73}$$

From (66)

$$V_n(0,\ t) = \frac{\lambda P_0}{Gu(t)}\bigg((A(t)-1)X^*(\lambda)\bigg)(x\overline{u}+xuY_2^*(X_2(t))Y_1^*(X_1(t)).$$

The limiting probability generating functions of the following

$$P(a_p,\ t) = \lambda P_\circ \left[\frac{\pi_1 u(t)}{\pi_2 u(t)}\right](1-X(a_p))e^{-\lambda a_p} \tag{74}$$

$$\pi_1 u(t) = A(t)Y_1^*(X_1(t))\bigg[(x\overline{u}Z^*(r(t))+(1-x)\overline{u})Y_2^*(X_2(t))+(\overline{u}Z^*(r(t))+(1-x)\overline{u})\bigg]-t$$

$$\pi_2 h(t) = \bigg\{t-\bigg\{A(t)+(1-A(t))X^*(\lambda)\bigg\}Y_1^*(X_1(t))\bigg[(x\overline{u}Z^*(r(t))+(1-x)u)Y_2^*(X_2(t))$$

$$+(xuZ^*(r(t))+(1-x)\overline{u})\bigg]\bigg\}$$

$$M_1(a_p,\ t) = \lambda P_0 + \left[\frac{(A(t)-1)X_1{}^*(\lambda)}{\pi_2 u(t)}\right](1-y_1(a_p))e^{-X_1(t)a_p} \tag{75}$$

$$M_2(a_p,\ t) = \lambda P_0 + \left[\frac{(A(t)-1)X_1{}^*(\lambda)}{\pi_2 u(t)}\right]y_1^*x_1(t)(1-y_2(a))e^{-X_2(t)a_p} \tag{76}$$

$$W_{1,\ n}(a_p,\ b_d,\ t)\ = \psi_1\left[\frac{\lambda P_0(A(u)-1)X^*(\lambda)}{\pi_2 u(t)}\right](1-y_1(a_p))e^{-x_1(t)a_p}(1-w_1(b_d))e^{-r(t)b_d} \tag{77}$$

$$W_{1,\ n}(a_p,\ b_d,\ t)\ = \psi_2 u\lambda P_0\left[\frac{(A(u)-1)X^*(\lambda)}{\pi_2 u(t)}\right]y_1^*x_1(t)(1-y_2(a_p))e^{-x_2(t)a_p}(1-w_2(b_d))e^{-r(t)b_d} \tag{78}$$

$$R_1(a_p,\ b_d,\ t) = \lambda P_0\psi_1\left[\frac{(A(t)-1)X^*(\lambda)}{\pi_2 u(t)}\right](1-y_1(a_p)e^{-x_1(t)a_p}w_1{}^*(r(t))e^{-r(t)b_d} \tag{79}$$

$$R_2(a_p,\ b_d,\ t) = \lambda P_0\psi_2\left[\frac{(A(t)-1)X^*(\lambda)}{\pi_2 u(t)}\right]y_1^*x_1(t)(1-y_2(a_p))e^{-x_2(t)a_p}w_2{}^*(r(t)))e^{-r(t)a_p} \tag{80}$$

$$V_n(a_p,\ t) = \frac{\lambda P_0}{\pi_2 u(t)}\left((A(t)-1)X^*(\lambda)\right)(x\bar{u} + xuY_2^*(X_2(t))Y_1^*(X_1(t))(1-z(a_p))e^{-r(t)a_p}. \tag{81}$$

The partial probability generating functions are defined as

$$P(t) = \int_0^\infty P(a_p,\ t)da_p,\ M_1(t) = \int_0^\infty M_1(a_p,\ t)da_p,\ M_2(t) = \int_0^\infty M_2(a_p,\ t)da,\ G_1(a_p,\ t)$$

$$= \int_0^\infty G_1(a_p,\ b,\ t)da_p,\ G_1(t) = \int_0^\infty G_1(a_p,\ t)da_p,$$

$$W_{1,\ n}(a_p,\ t) = \int_0^\infty W_{1,\ n}(a_p,\ b_d,\ t),\ W_{1,\ n}(t) = \int_0^\infty W_{1,\ n}(a_p,\ t)\,da,\ R_1(t) = \int_0^\infty R_1(a_p,\ t)da_p,$$

$$R_2(a_p,\ t) = \int_0^\infty R_2(a_p,\ b_d,\ t)\,da_p,\ R_2(t) = \int_0^\infty R_2(a_p,\ t)\,da_p,\quad V_n(t) = \int_0^\infty V_n(t)(a_p,\ t)da_p.$$

Note that $P(t),\ M_1(t),\ M_2(t),\ G_1(t),\ G_2(t),\ R_1(t),\ R_2(t),\ V(t)$ are the probability function of orbit size when the server is idle, busy, re-service, repair on normal service, repair on re-service, delaying repair on normal service, delaying repair on re-service, on vacation respectively.

$$\pi_1 u(t) = P_0(1-X^*(\lambda))\Big\{A(t)Y_1^*(X_1(t))\Big[(xuZ^*(r(t)) + (1-x)u)Y_2^*(X_2(t))$$

$$+ (x\bar{u}Z^*(r(t)) + (1-x)\bar{u})\Big] - t\Big\}$$

$$\pi_2 u(t) = \Big\{t - \Big\{A(t) + (1-A(t))X^*(\lambda)\Big\}Y_1^*(X_1(t))\Big[(xuZ^*h(t)) + (1-x)uY_2^*(X_2(t))$$

$$+ (x\bar{u}Z^*(r(t)) + (1-x)\bar{u})\Big]\Big\}$$

$$S_1(a_p,\ t) = \lambda P_0\left[\frac{(A(t)-1)X_1{}^*(\lambda)}{\pi_2 u(t)}\right]\left[\frac{(1-y_1^*(x_1(t)))}{x_1(t)}\right] \tag{82}$$

$$S_2(a_p,\ t) = \lambda P_0\left[\frac{(A(t)-1)X_1{}^*(\lambda)}{\pi_2 u(t)}\right]y_1^* x_1(t)\left[\frac{(1-y_1^*(x_1(t)))}{x_1(t)}\right] \tag{83}$$

$$W_1(a_p,\ t) = \psi_1\left[\frac{P_0(w_1^* r(u)-1)}{\pi_2 u(t)}\right](1-y_1(a_p))e^{-x_1(t)a_p} \tag{84}$$

$$W_2(a_p,\, t) = \psi_2 \left[\frac{P_0(w_2^* r(u) - 1)}{\pi_2 u(t)} \right] (1 - y_2(a_p)) e^{-x_2(t) a_p} \tag{85}$$

$$R_1(t) = P_0 \psi_1 \left[\frac{(X^*(\lambda) z_1^* \mu(t) - 1)}{\pi_2 u(t)} \right] \left[\frac{1 - y_1^* x_1(t)}{x_1(t)} \right] \tag{86}$$

$$R_2(t) = P_0 \psi_1 u \left[\frac{(X^*(\lambda) z_2^* \mu(t) - 1)}{\pi_2 u(t)} \right] \left[\frac{1 - y_2^* x_1(t)}{y_1(t)} \right] \tag{87}$$

$$V_n(c) = \frac{P_0 x^*(\lambda)(y_1^* x_1(t) - 1) \left\{ (x\overline{u} + xuY_2^*(X_2(t))) \right\} (z^* \mu_1(t) - 1)}{\pi_2 u(t)}. \tag{88}$$

Since P_0 is the probability that the server is idle when the server is idle and no customer in orbit, it is determined by the following normalizing condition $P_0 + P(1) + \pi_1(1) + \pi_2(1) + M_1(1) + M_2(1) + R_1(1) + R_2(1) + Z(1) = 1$.

Hence the probability generating functions of the number of customers in the system is

$$H(t) = P_0 + P(t) + V_n(t) + t \Big[M_1(t) + M_2(t) + G_1(t) + G_2(t) + R_1(t) + R_2(t) \Big]$$

$$H(t) = \frac{P_0 X^*(\lambda) Y_1^*(X_1(t))\,(t-1)\,(\overline{u} + Y_2^*(X_2(t))}{t - (A(t) + (1 - A(t)) X^*(\lambda))\,\big(Y_1^*(X_1(t))(xZ^*(r(t)) + 1 - x)(\overline{u} + hY_2^*(X_2(t))\big)}.$$

Probability generating functions of the number of customers in the orbit is

$$H(0) = P_0 + P(t) + V(t) + M1(t) + M2(t) + G1(t) + G2(t) + R1(t) + R2(t)$$

$$H(0) = \frac{P_0 X^*(\lambda)\{1 - t\}}{t - (t + (1-t)^*(\lambda))(Y_1^*(X_1(t))(xZ^*(\lambda_0(t)) + (1-x))(\overline{u} + hY_2^*(X_2(t))}.$$

The average number of customers in the system L_s under steady state condition is

$$L_s = \lim_{t \to 1} H^{'}(t)$$

$$L_s = \frac{(\lambda E(A))^2 \left[\begin{array}{c} E(Y_1^2)(1+\psi_1(E(Z_1))^2 + hE(Y_2^2)(1+\psi_2(E(Y_2))^2 + xE(Z^2)+ \\ 2hE(Y_1)(1+\psi_1(E(Z_1)) + 2xE(Y_1)(1+\psi_1(E(Z_1)+E(Z)) + 2xhE(Y_2)(1+\psi_2 E(Z_2)) \end{array} \right] + 2\lambda E(A) \left\{ \begin{array}{c} [E(Y_1)(1+\psi_1 E(Z_1)) + hE(Y_2)(1+\psi_2 E(Z_2)) + xE(Z)] \\ \left[\begin{array}{c} \lambda E(Y_1)(1+\psi_1(E(Z_1)) \\ +hE(Y_2)(1+\psi_2(E(Z_2)) \end{array} \right] + X^*(\lambda) \end{array} \right\}}{2X^*(\lambda) - E(A)\{hE(Y_2)(1+\psi_2 E(Z_2) + xE(Z)\}\lambda \left(\begin{array}{c} E(Y_1)(1+\psi_1(E(Z_1)) \\ +hE(Y_2)(1+\psi_2 E(Z_2)) + xE(Z) \end{array} \right)}.$$

Average number of customers in the orbit L_q under steady state condition is obtained by

$$L_q = \underset{t \to 1}{lim} H^{'}(0)$$

$$L_q = \frac{(1-X^*(\lambda))E(A)\{[E(Y_1)(1+\psi_1(E(Z_1) + hE(Y_2)(1+\psi_2(E(Z_2) + xE(Z)]\} - (\lambda E(A))^2 \left[\begin{array}{c} E(Y_1^2)(1+\psi_1(E(Z_1))^2 + tE(Y_2^2)(1+\psi_2(E(Z_2))^2 + xE(Z^2)+ \\ 2hE(Y_1)(1+\psi_1(E(Z_1))E(Y_2)(1+\psi_2 E(Z_2)))+ \\ 2xE(Y_1)(1+\psi(E(Z_1))E(Z) + 2xhE(Y_2)(1+\psi_2 E(Z_2))E(Z) - 1 \end{array} \right]}{2\{\lambda E(A)(E(Y_1)(1+\psi_1(E(Z_1))) + hE(Y_2)(1+\psi(E(Z_2)) + xE(Z)) - X^*(\lambda)\}}$$

and probability that the server is idle during vacation and system as follows,

$$V_n(t) = \frac{P_\circ X^*(\lambda)\lambda E(A)[E(Y_1)[1+\psi_1(E(Z_1)] + hE(Y_2)[1+\psi_2(E(Z_2)]]}{X^*(\lambda) - \lambda E(A)[E(Y_1)[1+\psi_1(E(Z_1))] + hE(Y_2)[1+\psi_2(E(Z_2)] + xE(Z)]}$$

$$P_0(t) = P_0(1) = \frac{E(A)\left(X^*(\lambda) - \lambda\left[E(Y_1)\left[1+\psi_1(E(Z_1))\right] + hE(Y_2)\left[1+\psi_2(E(Z_2))\right] + xE(Z)\right]\right)}{X^*(\lambda)}. \tag{89}$$

6. System state probabilities and system performance metrics

In this section, various system states like server busy, when the server is idle, server under repair, delayed repair and working vacation is analyzed.

By substituting $t = 1$ in (82) to (88) and applying L-Hospital's rule whenever necessary

(i) Let S_n be the steady-state probability when the server is busy

$$S_n = \lambda P_0 \left[\frac{(A(x)-1)X_1{}^*(\lambda)}{\pi_2 u(t)}\right]\left[\frac{(1-x_1^*(E(Y_1))E(A)}{x_1(t)}\right] \tag{90}$$

$$+\lambda P_0 \left[\frac{(A(y)-1)X_1{}^*(\lambda)}{\pi_2 u(t)}\right] y_1^* x_1(t)\left[\frac{(1-y_1^*(E(X_1))E(A))}{x_1(t)}\right]. \tag{91}$$

(ii) Let W_n be the steady-state probability of the server is in working vacation

$$W_n = \psi_1(1-y_1(a_p))e^{-x_1(t)a_p} + \psi_2(1-y_2(a_p))e^{-x_2(t)a_p}. \tag{92}$$

(iii) Let R_n be the steady-state probability that the server is under delaying repair

$$R_n(1) = P_0\psi_1\left[\frac{X^*(\lambda)z_1^*E(X_1)E(A)-1)}{\pi_2}\right]\left[\frac{1-y_1^*E(X_1)E(A)}{\pi_1}\right] \tag{93}$$

$$+P_0\psi_1\left[\frac{X^*(\lambda)z_1^*E(Y_1))E(A)-1)}{\pi_2}\right]\left[\frac{1-y_1^*E(X_1)E(A)}{\pi_1}\right]. \tag{94}$$

(iv) Let V_n be the steady-state probability that the server is under repair,

$$V_n = \frac{P_0 x^*(\lambda)(y_1^*E(X_1)E(A)-1)\left\{(x\bar{u}+xuY_2^*(X_2(t))\right\}(z^*E(X_1)E(A)-1)}{\pi_2}. \tag{95}$$

7. Special cases

This section deals with a few special cases of our approach which leads to some concrete applications.

Case (i): No Repair, No working vacation, and No retrial then this model can be reduced to a non-Markovian queue with optional service whose results coincide with Sundari and Maragatha [27].

Case (ii): No Repair, No optional re-service, then this model can be reduced to the unreliable retrial queueing system with working vacation whose results coincide with Bharathi shanmugam [5].

8. Numerical results

In this section, we show the various settings on system behavior measurements using MATLAB. We investigate exponentially distributed retrial, service, and optional re-service times. The stability criteria are satisfied by selecting the numerical measurements at random.

As the arrival rate λ, $S_1(1)$, $V(1)$ and W_q escalates, P_0 and L_q diminish for the value of $\psi_1 = 0.6$, $a_p = 5$, $\tau_{ns} = 3$, $\tau_{ds} = 4$, $\tau_{os} = 0.2$, and $p_r = 0.8$ which is depicted in Table 1.

As the lower service rate $\psi_1(1)$ increases L_q, W_q also increases and P_0, $W(1)$ and $R(1)$ diminish for the values of $\lambda = 2$, $a_p = 5$, $\tau_{ns} = 3$, $\tau_{ds} = 4$, $\tau_{os} = 0.2$, and $p_r = 0.8$ which is presented in Table 2.

Table 3 displays that as repair rate (a_p) rises P_0, L_q, $S_1(1)$ and W_q also escalates yet, $R(1)$ diminish for the values of $\psi_1 = 0.8$, $\lambda = 2$, $\tau_{ns} = 3$, $\tau_{ds} = 4$, $\tau_{os} = 0.2$, and $p_r = 0.8$.

Table 1. The effect of arrival rate (λ) on $S_1(1)$, $V(1)$, W_q, P_0, L_q

Arrival rate (λ)	P_0	L_q	$S_1(1)$	$V(1)$	W_q
2.1	0.0075	0.5186	0.0208	0.0198	0.2468
2.2	0.0067	0.5713	0.0234	0.0213	0.2596
2.3	0.0061	0.6341	0.0264	0.0229	0.2748
2.4	0.0055	0.7105	0.0301	0.0251	0.2960
2.5	0.0052	0.8051	0.0346	0.0277	0.3224

Table 2. The effect of lower service rate (ψ_1) on L_q, W_q, P_0, $W(1)$, $R(1)$

Lower service rate (ψ_1)	P_0	L_q	$W(1)$	$R(1)$	W_q
0.6	0.0354	5.4383	0.1365	0.0090	5.4382
0.7	0.0344	6.2446	0.1162	0.0084	6.2446
0.8	0.0334	6.7839	0.1004	0.0079	6.7840
0.9	0.0325	7.1439	0.0867	0.0075	7.1439
1.0	0.0315	7.3796	0.0773	0.0070	7.3796

Table 3. The effect of repaie rate (a_p) on P_0, L_q, $S_1(1)$, W_q, $R(1)$

Breakdown rate (a_p)	P_0	L_q	$S_1(1)$	$R(1)$	W_q
3.1	0.0118	1.1824	0.0165	0.0127	11.8241
3.2	0.0119	1.1966	0.0167	0.0125	11.9662
3.3	0.0121	1.2107	0.0169	0.0124	12.1074
3.4	0.0122	1.2248	0.0170	0.0123	12.2480
3.5	0.0124	1.2388	0.0172	0.0121	12.3883

In Figure 2, for the values of $\psi_1 = 0.8$, $\lambda = 2$, $\tau_{ns} = 3$, $\tau_{ds} = 4$, $\tau_{os} = 0.2$, and $p_r = 0.8$, the influence of the distinct system parameters are displayed via 2-D graph. Figure 2 (a), displays that as arrival rate λ rises $S_1(1)$, $V(1)$ and W_q also escalates, yet P_0 and L_q diminish. In Figure 2 (b), we found that as the lower service rate $\psi_1(1)$ rises L_q, W_q also increases, however P_0, $W(1)$ and $R(1)$ diminish. In Figure 2 (c), we found that as the breakdown rate (a_p) elevates P_0, L_q, $S_1(1)$, W_q escalates, yet $R(1)$ diminish.

Figure 3 shows the three-dimensional graph that depicts the system performance measures. In Figure 3 (a), the surface displays that as the arrival rate λ escalates, L_q, W_q also increase. In Figure 3 (b), we found that as the lower service rate ψ_1, escalates, L_q, and W_q also increase. In Figure 3 (c), we found that as the breakdown rate a_p increases, then P_0, L_q, also increases.

The aforementioned numerical findings allow us to determine the impact of features on the system's assessment criteria, and we can be sure that the results are appropriate for real-world situations.

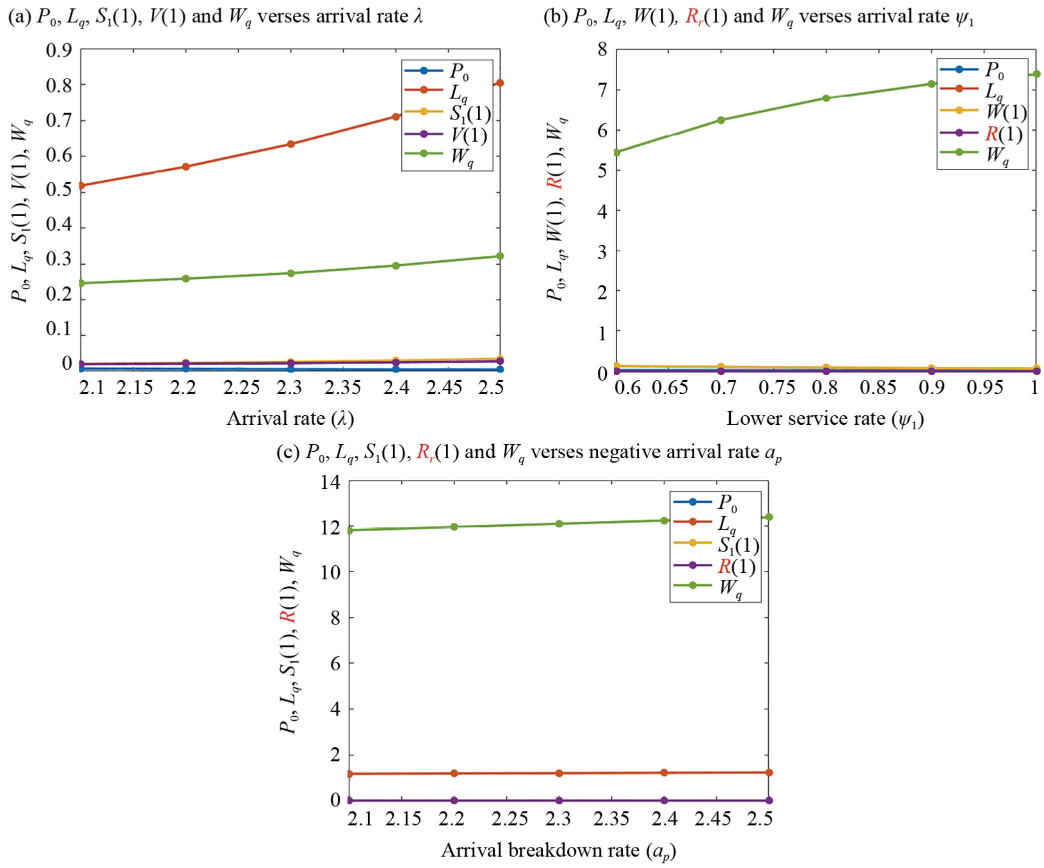
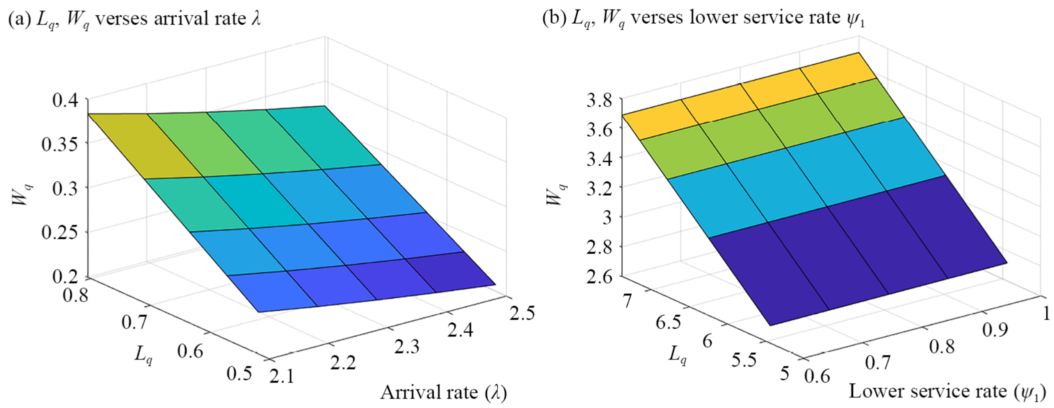


Figure 2. Two dimensional graph



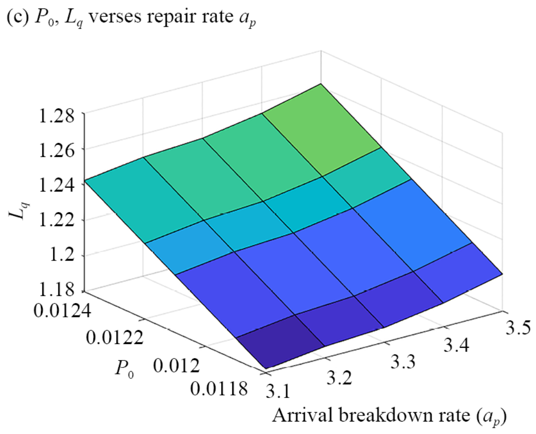


Figure 3. Three dimensional graph

9. 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 necessitates getting the best terms and prices on all business transactions in addition to standardizing, streamlining, and rationalizing 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 in order 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 in this section of our established model to make it more cost-effective.

To find the optimal values for the parameters, which include the service and vacation rate $(\xi_b,\ \xi_w)$ by employing a cost optimization technique. In the projected cost function, it is assumed that the various system activities and the various cost components related to those activities have a linear connection.

The following is a definition of each of the cost element variables that are included in finding the expected The following defines each cost element variable included in determining the expected total cost function (TC) $(\xi_b,\ \xi_w)$ per unit of time.

$G_h \Longrightarrow$ Holding costs in the system for a predetermined amount of time

$G_b \Longrightarrow$ CPUT during regular service operation

$G_w \Longrightarrow$ CPUT unit for normal re-service operation

$G_d \Longrightarrow$ CPUT while the server is in lower service operation

$G_r \Longrightarrow$ Cost of acquiring customers while performing server repair

$$G_1 \implies \text{Cost per customer during busy hours}$$

$$G_2 \implies \text{Cost per customer while on vacation.}$$

The cost function's predictions are stated as

$$TC(\xi_b,\ \xi_w) = G_h L_q + G_b S_n + G_w W_n + G_d R_n + G_r V_n + C_1 \xi_b + G_2 \xi_w. \tag{96}$$

Due to its substantial non-linearity, the cost function shown in 96 is difficult to optimize analytically. Thus, we use a heuristic technique to minimize the overall cost, which is dependent on the service and vacation rates ξ_b and ξ_w.

Our main objective is to minimize the total cost function while determining the best service rate (ξ_b^*) during busy mode and the optimal service rate (ξ_w^*) during vacation mode.

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

$$TC(\xi_b^*,\ \xi_w^*) = \min_{\xi_b^*,\ \xi_w^*} TC(\xi_b,\ \xi_w).$$

Further, 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	G_h	G_b	G_w	G_d	G_r	C_1	C_2
Set 1	80	78	92	100	89	90	75
Set 2	50	69	90	86	75	85	80
Set 3	60	80	85	90	75	90	105

9.1 *Particle swarm optimization (PSO)*

Particle Swarm optimization (PSO) is a computational algorithm inspired by the social behavior of birds flocking or fish schooling. It was introduced in 1995 by Kennedy [28]. The objective of its development was to optimize a function that is fundamentally non-linear. It is widely used for optimization tasks due to its simplicity and effectiveness in finding optimal solutions in complex, multi-dimensional spaces. It is easy to implement and requires minimal parameter tuning. PSO quickly converges to optimal or near-optimal solutions. It works well in both constrained and unconstrained optimization problems across diverse fields.

In the context of cost optimization, PSO operates by initializing a swarm of particles, each representing a potential solution within the defined search space. Each particle adjusts its position based on its own best-known position and the best-known position of the entire swarm, iteratively moving toward regions of the search space that offer lower cost values. This process continues until a predefined stopping criterion is met, such as a maximum number of iterations or a satisfactory cost threshold. PSO's ability to efficiently navigate complex, non-linear, and multi-modal cost landscapes makes it a powerful tool for cost optimization across various applications.

Upadhyaya [29] extended this strategy to optimize costs in a discrete-time retrial queue (RQ) with Bernoulli feedback and initial failure. Zhang et al. [30] 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 Malik et al. [31] has been referenced. Harini [32] addressed the dynamical behavior and meta-heuristic optimization of a hospital management software system in her research.

9.2 *PSO implementation*

PSO is applied to queueing models by representing system parameters (e.g., arrival rates, service rates) as particles. The fitness function evaluates performance metrics like cost or waiting time. Particles update their positions based on individual and group best solutions, iterating until the optimal configuration is found.

9.2.1 *Steps involved*

• Problem representation: Encode queue parameters as particles.
• Cost function: Combine metrics like operating costs and customer dissatisfaction.
• Initialization: Randomly initialize particle positions and velocities.
• Fitness evaluation: Compute the cost for each particle using the queue model.
• Update particles: Adjust velocities/positions using inertia, personal best, and global best.
• Stopping criteria: End when iterations reach a limit or improvement stagnates.
• Optimal solution: Extract the best particle (minimum cost).

Thus, by establishing the following settings for the default parameters, $\psi_1 = 0.6$, $\tau_{ns} = 3$, $\tau_{ds} = 4$ and $\tau_{os} = 0.2$ we were able to optimize the cost. It is assumed that the lower bound is 1 and the upper bound is 7. The equivalent values of 50, 100, 2, and 5 have been specified for the number of repetitions, population size, inertial weight, and both acceleration factors respectively. The impact of λ, p_r, a_p including G_h, G_b, G_w, G_d, G_r, G_1, and G_2, on the ideal service rates and optimal total cost for each of the three cost sets is shown in Table 5.

Further, the pseudo-code 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

$$V_i(t+1) = wV_i(t) + c_1 r_1 (P_i - X_i(t) + c_2 r_2 (G - X_i(t)$$

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 is given in eqn. 99 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:

The optimal service rates ξ_b^* and ξ_w^* and the total cost $TC(\xi_b^*, \xi_w^*)$ are finally obtained.

Algorithm 1 Pseudo Code of PSO Algorithm

INPUT: Objective function = $TC(\xi_b, \xi_w)$, acceleration factors, inertia weight and Maximum number of iterations.

OUTPUT: The cost function's value $TC(\xi_b^*, \xi_w^*)$.

Step 1: Finding initial locations X_i for the n particles in a population.

Step 2: Determine G(g-best) using best(min) as the TC $\{P_1, \ldots, P_n\}$.

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 $V_i(t+1)$.

Step 5: Obtain the new locations for i^{th} particle $X_i(t+1) = X_i(t) + V_i(t+1)$.

Step 6: Check the objective function at new locations $X_i(t+1)$.

Step 7: Discover the current best (p-best) for each particle P_i.

end for

Step 8: Upgrade global best G.

end while.

Step 9: Deliver the optimal value of the objective function TC.

Table 5. Effect of λ, p_r, a_p on (TC^*, ξ_b^*, ξ_w^*) using PSO

Parameters		(TC^*, ξ_b^*, ξ_w^*)		
		Cost set 1	Cost set 2	Cost set 3
λ	2.1	(118.5891, 2.9482, 1.2210)	(82.9358, 2.2012, 1.2019)	(65.2104, 2.1022, 1.3391)
	2.2	(112.3049, 2.1039, 1.2039)	(84.3049, 2.2039, 1.1029)	(67.2093, 2.2034, 1.5559)
	2.3	(115.2031, 2.4530, 1.0214)	(86.0981, 2.2092, 1.0291)	(68.2029, 2.2094, 1.3029)
p_r	0.7	(110.0937, 2.0388, 1.1021)	(82.0921, 2.2919, 1.9292)	(66.9482, 2.2029, 1.1021)
	0.8	(110.0932, 2.4852, 1.0847)	(82.4821, 2.9409, 1.2091)	(66.4974, 2.0298, 1.0928)
	0.9	(111.2659, 2.9485, 1.0912)	(83.9021, 2.8868, 1.2837)	(66.0821, 2.0019, 1.2837)
a_p	3.1	(110.1093, 2.1029, 1.3891)	(82.3873, 2.8491, 1.3078)	(66.9830, 2.3810, 1.2074)
	3.2	(110.8748, 2.9739, 1.3012)	(82.0741, 2.8379, 1.9286)	(66.0181, 2.0381, 1.3901)
	3.3	(110.0491, 2.3901, 1.1093)	(82.9041, 2.2938, 1.9085)	(66.9301, 2.2883, 1.0984)

The analysis of PSO optimization results in Table 5 demonstrates that parameters such as re-service probability (p_r) and breakdown rate (a_p) is highly sensitive, causing significant cost variations, while the arrival rate (λ) shows moderate sensitivity. To enhance long-term operational efficiency, managing (p_r) through improved first-service quality and reducing (a_p) via preventive maintenance and rapid repairs is crucial. The optimization highlights the need for dynamic resource allocation to balance service and vacation rates under changing conditions. Managers should prioritize monitoring (p_r) and (a_p), invest in infrastructure reliability, and adopt flexible strategies to mitigate cost impacts, ensuring sustainable and efficient system operations.

9.3 *Advantages of PSO*

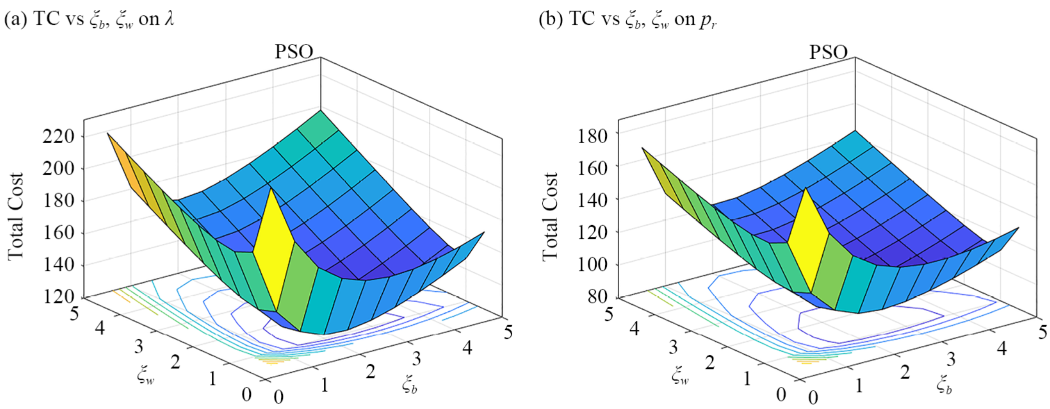
PSO offers several advantages over traditional optimization methods, making it particularly effective for cost optimization. Its simplicity and ease of implementation allow for quick deployment and adaptation to various problems. PSO's robustness and flexibility enable it to handle a wide range of optimization problems, including those with non-linear, non-differentiable, and multi-modal cost functions. The algorithm's global search capability effectively balances exploration and exploitation, enabling it to search the entire solution space and avoid local optima. Additionally, PSO's computational efficiency allows for faster convergence to optimal solutions, which is beneficial in time-sensitive cost optimization applications. These advantages make PSO a powerful and effective tool for cost optimization and are often superior to traditional optimization methods.

Moreover, PSO stands out by not requiring gradient information, making it ideal for discontinuous or non-differentiable objective functions. Unlike genetic algorithms that rely on evolution-based operations like crossover and mutation, PSO uses collaborative interactions between particles, leveraging personal and global best solutions. Additionally, PSO retains the memory of past solutions, enabling informed updates, unlike techniques like simulated annealing that reset search strategies. These features make PSO a versatile and efficient tool for solving complex optimization challenges.

9.4 *Convergence in PSO*

In cost optimization within queueing models, convergence refers to the process where an optimization algorithm iteratively adjusts system parameters to minimize a defined cost function. This function typically encompasses various factors, including operational expenses, customer waiting times, and service quality metrics. The goal is to achieve a stable state where further adjustments do not yield significant improvements, indicating that the optimal or near-optimal cost configuration has been reached. Achieving convergence is crucial for effective decision-making in managing queueing systems, as it ensures that the implemented strategies lead to sustainable and efficient operations.

The convergence plots for the cost function across the three parameters displayed in Figure 5 show a rapid decrease in the best cost within the first few iterations, indicating the efficiency of the PSO algorithm in quickly finding near-optimal solutions. After some pre-defined iterations, the cost stabilizes, suggesting convergence to a global or near-global minimum. The smoothness of the curves beyond this point reflects the algorithm's ability to refine solutions without significant fluctuations, demonstrating its reliability and robustness in optimization. Furthermore, Figure 4 illustrates the cost function's convexity and optimality with regard to the three cost sets taken into account in the optimization research.



(c) TC vs ξ_b, ξ_w on a_p

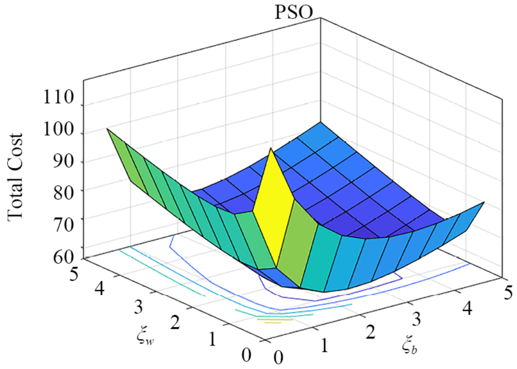
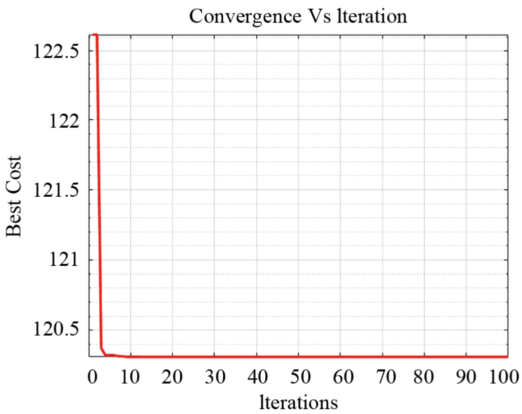
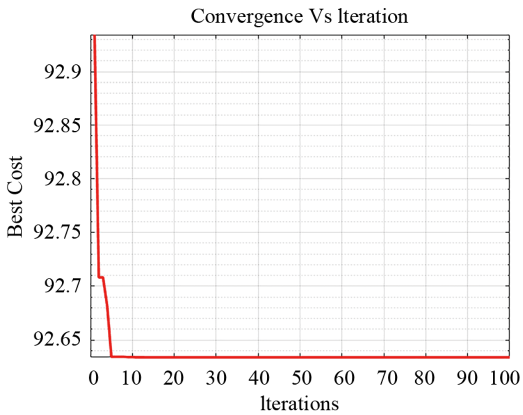


Figure 4. Optimality of the cost function using PSO

(a) Convergence vs Iteration on λ



(b) Convergence vs Iteration on p_r



(c) Convergence vs Iteration on a_p

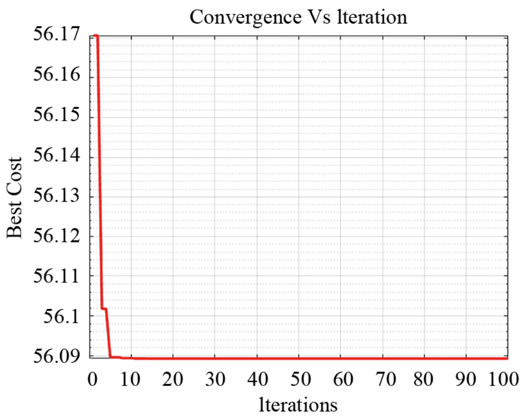


Figure 5. Convergence of the cost function using PSO

10. Conclusion

In this study, we present a comprehensive queueing model that integrates several advanced features, including single arrivals, a retrial mechanism, optional re-service, working vacations, and delayed repair scenarios. To analyze the dynamics of the system, we employ the supplementary variable technique, providing deeper insights into its behavior and performance metrics. Additionally, we incorporate Particle swarm optimization (PSO) to optimize the system's cost structure, identifying the optimal balance between operational costs and efficiency. The results from the PSO optimization highlight key cost-reduction strategies and demonstrate how the model can be fine-tuned to improve operational performance. Finally, these findings contribute to advancing queueing theory and offer practical implications for optimizing real-world systems. Future research could further refine the model by exploring multi-server scenarios or investigating alternative optimization techniques for broader applications.

Conflict of interest

The authors declare no competing financial interest.

References

[1] Rajadurai P, Saravanarajan MC, Chandrasekaran VM. An $M/G/1$ retrial G-queue with optional re-service, impatient customers, multiple working vacations and vacation interruption. *International Journal of Operational Research.* 2017; 30(1): 33-64.

[2] Madheswari SP, Kumar BK, Suganthi P. Analysis of $M/G/1$ retrial queues with second optional service and customer balking under two types of Bernoulli vacation schedule. *RAIRO-Operations Research*. 2019; 53(2): 415-443.

[3] Santhi K. An $M/G/1$ retrial queue with second optional service and multiple working vacation. *Advances in Applied Mathematics*. 2021; 20(6): 1129-1146.

[4] Arivudainambi D, Godhandaraman P. A batch arrival retrial queue with two phases of service, feedback and k-optional vacations. *Applied Mathematical Sciences*. 2012; 6(22): 1071-1087

[5] Baskar A, Saravanarajan MC. Analyzing feedback $M/G/1$ double retrial orbit with two-phase optional service and repair under working vacation policy. *Contemporary Mathematics*. 2024; 5(4): 5795-5827.

[6] Rajam V. Retrial queue $M^{[X]}/G/C$ with bernoulli's vacation, repair, service and re-service in orbit. *Turkish Journal of Computer and Mathematics Education*. 2021; 12(12): 4535-4540.

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

[8] Rajadurai P, Janani H, Venkatesh S, Parameswari K. N-Policy for $M/G/1$ retrial queue with priority customers under multiple working vacations. *Mathematics and Its Applications in Technology*. 2024; 3180(1): 020004.

[9] Shanmugam B, Saravanarajan MC. Unreliable retrial queueing system with working vacation. *AIMS Mathematics.* 2023; 8(10): 24196-24224.

[10] Seeniraj G, Moganraj D, Jayaraman R. A steady state analysis of single server $M/G/1$ queueing system with multiple working vacations. In: *2nd International Conference on Recent Trends in Applied and Computational Mathematics: Icrtacm-2021*. 2023; 2649(1): 030026.

[11] Sundarapandiyan S, Nandhini S. Sensitivity analysis of a non-Markovian feedback retrial queue, reneging, delayed repair with working vacation subject to server breakdown. *AIMS Mathematics*. 2024; 9(8): 21025-21052.

[12] Revathi C. Search of arrivals of an $M/G/1$ retrial queueing system with delayed repair and optional re-service using modified bernoulli vacation. *Journal of Computational Mathematica*. 2022; 6(1): 2002-2009.

[13] Gao S, Zhang J, Wang X. Analysis of a retrial queue with two-type breakdowns and delayed repairs. *IEEE Access.* 2020; 8: 172428-172442. Available from: https://doi.org/10.1109/ACCESS.2020.3023191.

[14] Singh CJ, Kaur S, Jain M. Unreliable server retrial G-queue with bulk arrival, optional additional service and delayed repair. *International Journal of Operational Research.* 2020; 38(1): 82-111.

[15] Malik G, Upadhyaya S, Sharma R. Particle swarm optimization and maximum entropy results for $M^{[X]}/G/1$ retrial G-queue with delayed repair. *International Journal of Mathematical, Engineering and Management Sciences.* 2021; 6(2): 541.

[16] Jain M, Kumar A. Unreliable server $M/G/1$ queue with working vacation and multi-phase repair with delay in verification. *International Journal of Applied and Computational Mathematics.* 2022; 8(4): 210.

[17] Vaishnawi M, Upadhyaya S, Kulshrestha R. Optimal cost analysis for discrete-time recurrent queue with bernoulli feedback and emergency vacation. *International Journal of Applied and Computational Mathematics.* 2022; 8(5): 254.

[18] Kumar A, Jain M. Cost optimization of an unreliable server queue with two stage service process under hybrid vacation policy. *Mathematics and Computers in Simulation.* 2023; 204(2): 259-281.

[19] Deora P, Kumari U, Sharma D. Cost analysis and optimization of machine repair model with working vacation and feedback policy. *International Journal of Applied and Computational Mathematics*. 2021; 7(6): 1-14.

[20] Malik G, Upadhyaya S, Sharma R. Particle swarm optimization and maximum entropy results for $M^{[X]}/G/1$ retrial G-queue with delayed repair. *International Journal of Mathematical Engineering and Management Sciences.* 2021; 6(2): 541-563.

[21] Jain M, Jain A. Genetic algorithm in retrial queueing system with server breakdown and caller intolerance with voluntary service. *International Journal of Systems Assurance Engineering and Management.* 2022; 13(1): 1-17.

[22] Jain M, Kumar A. Unreliable server $M^{[X]}/G/1$ queue with working vacation and multi-phase repair with delay in verification. *International Journal of Applied and Computational Mathematics.* 2022; 8(4): 210.

[23] Vijaya Laxmi P, Jyothsna K. Cost and revenue analysis of an impatient customer queue with second optional service and working vacations. *Communication in Statistics-Simulation and Computation*. 2022; 51(8): 4799-4814.

[24] Harini R. Indhira K. Meta heuristic optimization of a batch arrival retrial queue with optional re-service and M-optional vacations. *International Journal of System Assurance Engineering and Management*. 2024; 15(9), 4252-4282.

[25] Pakes AG. Some conditions for ergodicity and recurrence of Markov chains. *Operations Research.* 1969; 17(6): 1058-1061.

[26] Sennott LI, Humblet PA, Tweedie RL. Mean drifts and the non-ergodicity of Markov chains. *Operations Research* 1983; 31(4): 783-789.

[27] Sundari S, Maragatha, Srinivasan S. Optional services in a non-Markovian queue. *International Journal of Operational Research*. 2017; 28(1): 1-17.

[28] Kennedy J, Eberhart R. Particle swarm optimization. *Proceedings of ICNN'95-International Conference on Neural Networks*. Perth, WA, Australia: IEEE; 1995.

[29] Upadhyaya S. Cost optimization of a discrete-time retrial queue with Bernoulli feedback and starting failure. *International Journal of Industrial and Systems Engineering.* 2020; 36 (2): 165-196.

[30] Zhang XL, Wang JT, Ma Q. Optimal design for a retrial queueing system with state dependent service rate. *Journal of Systems Science and Complexity.* 2017; 30(2): 883-900.

[31] Malik G, Upadhyaya S, Sharma R. Cost inspection of a $M/G/1$ retrial model using particle swarm optimization and genetic algorithm. *Ain Shams Engineering Journal.* 2021; 12(4): 2241-2254.

[32] Harini R, Indhira K. Dynamical behavior and meta heuristic optimization of a hospital management software system. *Heliyon.* 2024; 10(16): e35937.