J SYST SCI SYST ENG
Vol. 0, No. 0, 0, pp. 1–31
DOI: https://doi.org/10.1007/s11518-024-5612-1

ISSN: 1004-3756 (paper), 1861-9576 (online)
CN 11-2983/N

Finite Capacity Service System with Partial Server Breakdown and Recovery Policy: An Economic Perspective

Shreekant Varshney,[a] Suman Kaswan,[b] Mahendra Devanda,[c] Chandra Shekhar[b]

[a]Department of Mathematics, School of Technology, Pandit Deendayal Energy University, Gandhinagar, Gujarat 382426, India
skvarshney91@gmail.com

[b]Department of Mathematics, Birla Institute of Technology and Science Pilani, Pilani Campus, Pilani, Rajasthan 333031, India
sumanchaudharymh@gmail.com, chandrashekhar@pilani.bits-pilani.ac.in (✉)

[c]Department of Mathematics, Maharaja Surajmal Brij University, Bharatpur, Rajasthan 321201, India
mahendradevandamaths@gmail.com

Abstract. Developing a comprehensive service strategy to optimize customer satisfaction presents an ongoing challenge for effective facility provider. The essence of comprehensive systems is selecting the suitable service design, establishing an effective service delivery process, and building continuous improvement. This research analyzes a finite capacity service system incorporating several realistic customer-server dynamics: customer impatience, server's partial breakdown, and threshold recovery policy. When the number of customers is more, the server is under pressure to increase the service rate to mitigate the service system's load. Motivating from this fact, the concept of service pressure condition is also incorporated. For characterization, we evaluate state probabilities derived using the matrix-analytic method and henceforth several performance measures. To address the cost optimization problem involving the developed Chapman-Kolmogorov forward differential-difference equations and determine optimal operational parameters, we employ the recently devised cuckoo search (CS) optimization approach. A comparative analysis is performed with the semi-classical optimizer: quasi-Newton (QN) method, and metaheuristics technique: particle swarm optimization (PSO), to validate the efficacy of results. Lastly, several numerical illustrations are depicted in different tables and graphs to understand essential characteristics quickly.

Keywords: Customer impatience, service pressure condition, partial server breakdown, threshold-based recovery policy, Cuckoo search, particle swarm optimization, quasi-Newton

1. Introduction

Within the constraints of socio-economic-technological factors, an efficient service system is indispensable for progressive and sustainable development in the fast-growing competitive world. Effective service includes customer satisfaction and uninterrupted, quality, cost-effective, time-prompt service. In congestion, we have to experience the cognition of waiting in a queue and waiting for one's our turn to seek the hassle-free service. Waiting is ubiquitous and reinforces strategic research on critical areas of the service facility. The pursuit of optimal service design ranges from optimal capacity to uninterrupted availability, optimal service rate to optimal cost, quality to prompt service, etc. Decision-makers must explore the existing technological innovation and the acceptance of the customers to design a robust service system.

In today's technological era, the study of queueing-based service systems has gained significant prominence due to the growing importance and complexity. This research paper highlights essential reflections of a queueing-based service system using several realistic queueing notions such as balking, service pressure coefficient, threshold-based recovery policy, and partial server breakdown. A comprehensive survey of the literature on queueing systems featuring the aforementioned queueing notions shows that these queueing notions have been rarely studied in conjunction with different theoretical concepts. The paramount importance for organizations, be it in the realm of services or production, lies in the quality of service provision and operational efficiency. Over the past few decades, there has been an increased interest among researchers, system analysts, and decision-makers/policymakers in congestion problems, including work related to server breakdown, threshold-based recovery policies, and service pressure coefficients.

In congestion, impatience is prevalent among the customers. In general, at the epoch of arrival, if the server is unavailable due to busy in serving waiting customers, breaks down, or finite capacity, customers may show a reluctance attitude to join the queue and therefore may be uncertain whether to enter the service system. The longer the waiting queue, the higher the likelihood of customers balking. Haight (1957) was the first researcher who introduced the notion of customer balking in the queueing literature for a Markovian environment. Later, Haight (1959) again envisaged a single server Markovian queue that characterized the customers' continuous abandonment. Abou El Ata and Hariri (1992) investigated Markovian overflow queue with balking behavior of customers. Abou El Ata and Hariri (1992) extended the analytical solution for the multi-server Markovian queue with customer impatience. Drekic and Woolford (2005) investigated a priority queue assigning low priority to impatient customers. Lozano and Moreno (2008) studied the abandonment behavior of arrived customers in a single-server service system in a discrete-time environment with an infinite/finite buffer. Sun et al. (2017) explored the customer impatience (balking) in a single server Markovian environment with the double-adaptive working vacation (WV) policy. Since impatience attributes directly affect the quality of service (QoS), queueing problems with the attribute of impatience customers have motivated many scholars to investigate a distinguished service environment (Shekhar et al. 2020ab). We analyze how service availability and system capacity impact customer choices to join or avoid a queue, vital for understanding behavior and optimizing resources.

The efficient service system is dynamic with a load of customers, seeks to improve customer service to impact customer retention levels. Under the pressure of increased congestion, the server may tempt to increase the service efficiency. This paper also incorporates the concept of service pressure coefficient to model real-time strategic policies. The pressure coefficient an absolute constant, defines as the amount to which the server increases the service capacity (rate) to diminish the over waiting load of the service system. For the higher backlog of waiting, there is a high chance that the servers may start operating intensely until the backlog becomes small or nonexistent. Wang and Lin (2011) were the first to introduce the concept of pressure conditions for the service systems for the first time in the queueing literature. Wang et al. (2013) examined the warm-standby provisioning machine interference problem with multiple-imperfect coverage and multiple-server with the pressure condition for improving the repair rate. More recently, Shekhar et al. (2021a) conceptualized service pressure conditions for retaining the reneged customers in the multi-server Bernoulli's vacation queueing problem.

The literature on queue-based service systems is rich with assumptions about reliable servers, which is seldom. The server is subject to breakdowns randomly at any instant in practice. Most research findings on queueing-based service systems with server breakdown consider that the server terminates working completely when the breakdown occurs. Nevertheless, in practice, some real-time systems exist in which the service provider still works at a lesser rate in the breakdown state, which is referred to as working breakdown or partial breakdown in the queueing literature (Sridharan and Jayashree 1996, Kalidass and Kasturi 2012, Li et al. 2013, Liu and Song 2014) studied the single-server Markovian queue with working breakdown. A detailed survey addressing queueing-based service systems with the breakdown of the server is provided by Krishnamoorthy et al. (2014). Liou (2015) explored a single server queue with customer impatience and servers' working breakdown using the matrix method. Yang and Chen (2018) analyzed a single server service system with the working breakdown and optional service policies. Rajadurai (2018) employed the supplementary variable technique to analyze the general retrial queue with the catastrophic conditions and working breakdown under multiple working vacation policies. Recently, Yen et al. (2022) dealt with a retrial MRP with the working breakdown & exponentially start-up time and implemented the PSO algorithm to establish the optimal management policy with optimal joint values of the faster and slower service rates simultaneously at the minimum mean cost of the system.

The breakdown of the server leads in pronounced congestion or high impatience attributes among the customers, which increases the economic losses, customer dissatisfaction, etc. The breakdown of the service facility necessitates strategic recovery. The present study focuses on employing strategic corrective measures: threshold recovery policy. According to these economic corrective measures, when the active server is broken down, the recovery can be performed if there exists a pre-specified T $(1 \leq T \leq K)$ number of customers in the service system. The concept of threshold recovery policy was firstly introduced by Efrosinin and Semenova (2010). Jain and Bhagat (2012) envisaged a finite capacity retrial queueing-based service system with a threshold recovery policy for unreliable servers. Yang et al. (2013) formulated a cost optimization problem for a threshold-based recovery policy for repairable $M/M/1/N$ system. Yang and Chiang (2014) incorporated the concept of threshold recovery policy for a machine interference problem and employed the metaheuristics and PSO algorithm to obtain the converging results along with the mean cost of the machine interference problem.

The cost optimization problems are systematically developed to infer the strategic policies integral to achieving the optimal design. For better understanding of the converging results and utilization of several nature-inspired optimization techniques, one can refer the research works (Shekhar et al. 2020b 2021ab) and references therein.

To the best of our knowledge, no research within the queueing literature has comprehensively addressed threshold-based recovery policy, servers' working breakdown, customer impatience, and service pressure conditions in a research article. This notable research gap in the literature motivates us for the present study. Moreover, motivated by the results of the nature-inspired algorithms: particle swarm optimization (PSO) and cuckoo search (CS) algorithm, we employ these techniques to optimize the system parameters ($i.e.$, decision variables) and the mean cost of the developed model. A comparative study among CS algorith & PSO algorithm, and QN method has also been conferred to prove the excellence of the metaheuristics approaches. The significant contribution of the present study is to implement the optimization algorithms and to develop MATLAB codes for comparing the findings of the CS algorithm, PSO algorithm, and the QN method in terms of statistical parameters, computation time, and operating policies in optimal conditions, among others.

The proposed model has many real-life applications across a spectrum of real-life service systems like computer and communication systems, supply chain management, production systems, inventory control, and machine repair problems. The hardware unit consisting of routers, computers, switches, etc., processes the data packets in several communication systems. When a data packet arrives and finds a long latency, it may lose the information. As the number of data packets load increases in a hardware unit, it extends its built-in standby power to a faster processing rate thereby mitigating latency. The processing slows down due to technical issues in the hardware unit or associated software. The persistent technical issues are recovered following some state-dependent strategic policies. This model's adaptability lends itself to numerous real-world scenarios, underscoring its relevance in optimizing various service systems and operational settings.

The remaining content of this article is framed as follows. Section 2 introduces the proposed queueing modeling and defines its states with several assumptions and notations. The matrix analytic method and corresponding solution algorithm to compute the steady-state probability distribution are discussed in section 3. Section 4 showcases how the system performance indicators are defined and formulated in vector form. Section 5 confers the cost function as a constrained optimization problem. Besides this, some of the special cases are provided in section 6. Next, the QN method, PSO & CS algorithms are discussed in detail along with their pseudo-codes in subsections 7.1, 7.2, and 7.3, respectively. In section 8, several numerical illustrations with the help of numerous graphs and tables are explained. Lastly, in section 9 some of the concluding remarks and future prospects are provided.

2. Proposed Model and State Description

Notations

λ: Mean arrival rate of customers in the system

μ_b: Service rate during the normal busy state of the server

μ_d: Service rate during the partial breakdown period of the server

α: Breakdown rate of the server

β: Repair rate when the server is broken down

μ_T: Threshold value at which the breakdown server will be repaired

$(1-\xi)$: Balking probability of the waiting customers

ψ: Parameter associated with the service pressure coefficient

K: Capacity of the system

The present study develops a finite capacity service system with numerous realistic queueing notions like customer impatience, service pressure coefficient, partial server breakdown, and threshold-based recovery policy. The capacity of the studied service system is proposed as K. The prospective customer joins the service system for intended service following the Poisson process with parameter $\lambda\,(> 0)$. If the service facility is idle at the arrival epoch, the customer gets the intended service instantly; otherwise, arrived customer queues in the waiting line. The server selects the customer to serve from the queue following the *First-Come-First-Serve* (FCFS) queue discipline. It is assumed that the service times to serve the customers follow an exponential distribution with parameter μ_b during the normal busy state. The server is deteriorated (partially broken down) due to some technical issues that occur following the Poisson process with parameter α. In this state, the server continues service uninterruptedly to waiting customers at a slower rate instead of complete termination. The service times during the partial breakdown period of the server also follow an independent and identically (iid) exponentially distributed with rate parameter μ_d. The notion of the threshold recovery policy is employed to mitigate the mean cost of the service system due to customers in waiting. According to this, the partial breakdown server is not recovered until the number of customers in the system attains a pre-specified threshold value $T\,(1 \leq T \leq K)$. The recover times of the breakdown server follow an iid exponential distribution with rate parameter β. After accomplishing the recovery action, the server is ready to furnish the service to the waiting customers immediately at a normal efficiency. When the server is busy or malfunctioning, the customers who intended to join the service system tend to become impatient, causing them to depart the system with a probability of $1-\xi$. These customers may remain in the system with the complimentary probability ξ. If the number of customers in the system is T or more, the concept of the service pressure coefficient is considered. The pressure factor is assumed to be dependent on number of customers in the system and parameter ψ. Additionally, we assume that all continuous random variables, namely, inter-arrival times, breakdown times, and service/repair times, are mutually independent. The events arrival, service, repair, recovery, and balking are independent to each other.

Let us define the following terms $N(t)$ = number of customers in the service system at time instant t, and $J(t)$ = the server's state at the time t, where

$$J(t) = \begin{cases} 0, & \text{if the server is} \\ & \text{in normal working attribute} \\ 1, & \text{if the server is} \\ & \text{in working breakdown state} \end{cases}$$

Thus, the process $\{(J(t), N(t)); t \geq 0\}$ constitutes a continuous-time Markov chain (CTMC) defined in a two-tuple irreducible form, with the state-space $\Omega = \{\{(0, n); n = 0, 1, 2, \cdots, K\} \cup \{(1, n); n = 1, 2, \cdots, K\}\}$. Hence, at time instant t ($t \geq 0$), all the system-state probabilities are outlined as follows

$$\begin{aligned} P_{0,n}(t) =& \text{Prob}\{J(t) = 0, N(t) = n\}, \\ & n = 0, 1, 2, \cdots, K \\ P_{1,n}(t) =& \text{Prob}\{J(t) = 1, N(t) = n\}, \\ & n = 1, 2, \cdots, K \end{aligned}$$

Assuming all the considerations, the state-dependent mean service rate of the server is defined as

$$\mu_b^{(n)} = \begin{cases} \mu_b, & 1 \leq n \leq T - 1 \\ \left(\frac{2n}{n+1}\right)^{\psi} \mu_b, & T \leq n \leq K \end{cases}$$

$$\mu_d < \mu_b^{(n)}, \ \forall n$$

Now, using the theoretical concepts and axioms of the QBD (quasi birth and death) process, the system of Chapman-Kolmogorov forward differential-difference equations, that governs the proposed model, is delineated to exhibit the transient-state probabilities representing the likelihood of distinguished states of the service system. Following the different system states, we have

When the server is Idle

$$P'_{0,0}(t) = -\lambda P_{0,0}(t) + \mu_b^{(1)} P_{0,1}(t) \quad (1)$$

When the server is in the regular working attribute

$$\begin{aligned} P'_{0,1}(t) =& -\left(\lambda\xi + \mu_b^{(1)} + \alpha\right) P_{0,1}(t) \\ & + \lambda P_{0,0}(t) + \mu_b^{(2)} P_{0,2}(t) \quad (2) \\ P'_{0,n}(t) =& -\left(\lambda\xi + \mu_b^{(n)} + \alpha\right) P_{0,n}(t) \\ & + \lambda\xi P_{0,n-1}(t) \mu_b^{(n+1)} P_{0,n+1}(t), \\ & 2 \leq n \leq T - 1 \quad (3) \\ P'_{0,T}(t) =& -\left(\lambda\xi + \mu_b^{(T)} + \alpha\right) P_{0,T}(t) \\ & + \lambda\xi P_{0,T-1}(t) + \mu_b^{(T+1)} P_{0,T+1}(t) \\ & + \beta P_{1,T}(t) \quad (4) \\ P'_{0,n}(t) =& -\left(\lambda\xi + \mu_b^{(n)} + \alpha\right) P_{0,n}(t) \\ & + \lambda\xi P_{0,n-1}(t) + \mu_b^{(n+1)} P_{0,n+1}(t) \\ & + \beta P_{1,n}(t), \\ & T + 1 \leq n \leq K - 1 \quad (5) \\ P'_{0,K}(t) =& -\left(\mu_b^{(K)} + \alpha\right) P_{0,K}(t) \\ & + \lambda\xi P_{0,K-1}(t) + \beta P_{1,K}(t) \quad (6) \end{aligned}$$

When the server is in working breakdown state

$$\begin{aligned} P'_{1,0}(t) =& -\lambda P_{1,0}(t) + \alpha P_{0,0}(t) + \mu_d P_{1,1}(t) \quad (7) \\ P'_{1,1}(t) =& -\left(\lambda\xi + \mu_d\right) P_{1,1}(t) + \lambda P_{1,0}(t) \\ & + \alpha P_{0,1}(t) + \mu_d P_{1,2}(t) \quad (8) \\ P'_{1,n}(t) =& -\left(\lambda\xi + \mu_d\right) P_{1,n}(t) + \lambda\xi P_{1,n-1}(t) \\ & + \alpha P_{0,n}(t) + \mu_d P_{1,n+1}(t), \\ & 2 \leq n \leq T - 1 \quad (9) \end{aligned}$$

$$\begin{aligned} P'_{1,n}(t) = &-\left(\lambda\xi + \mu_d + \beta\right)P_{1,n}(t) \\ &+ \lambda\xi P_{1,n-1}(t) + \alpha P_{0,n}(t) \\ &+ \mu_d P_{1,n+1}(t), \\ &\qquad\qquad T \le n \le K-1 \end{aligned} \tag{10}$$

$$\begin{aligned} P'_{1,K}(t) = &-\left(\mu_d + \beta\right)P_{1,K}(t) + \lambda\xi P_{1,K-1}(t) \\ &+ \alpha P_{0,K}(t) \end{aligned} \tag{11}$$

At $t = 0$, the initial condition is

$$\begin{cases} P_{0,0}(0) = 1 \\ P_{0,n}(0) = 0, \quad n = 1, 2, \cdots, K \\ P_{1,n}(0) = 0, \quad n = 1, 2, \cdots, K \end{cases} \tag{12}$$

3. Matrix Analytic Method

In equilibrium condition, $i.e.$, $t \to \infty$, the following are the state probabilities for the analysis of the service system, which are depicted as

for $n = 0, 1, 2, \cdots, K$, $\lim_{t\to\infty} P_{0,n}(t) = P_{0,n}$

and $\lim_{t\to\infty} P'_{0,n}(t) = 0$

for $n = 1, 2, \cdots, K$, $\lim_{t\to\infty} P_{1,n}(t) = P_{1,n}$

and $\lim_{t\to\infty} P'_{1,n}(t) = 0$

Now, to derive the state probability distribution, we adopt the matrix analytic method as the system of equations is highly complicated, making it challenging to calculate the closed/vector-form of expression of the state probabilities because of intricate constraints like multi-equation, multi-variable, and multiple parameters. The matrix analytic method, pioneered by Neuts (1981), leverages the concept of embedded Markov chains to handle numerous realistic queue-based service systems. For the matrix approach, we characterize the probability vector $\tilde{\mathbf{P}}_n; n = 0, 1, 2, \cdots, K$ as row vector having steady-state probabilities as elements, i.e., $\tilde{\mathbf{P}}_0 = [P_{0,0}]$ and $\tilde{\mathbf{P}}_n = [P_{0,n}, P_{1,n}]; n = 1, 2, \cdots, K$. The transition rate matrix of the Markov chain can equivalently be defined using the QBD process. Hence, by balancing the incoming and outgoing transitions, the tridiagonal generator matrix $\mathbf{Q}$ of the studied CTMC is defined as follows

$$\mathbf{Q} = \begin{bmatrix} \mathbf{A}_0 & \mathbf{B}_0 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_0 & \mathbf{A}_1 & \mathbf{B}_1 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_1 & \mathbf{A}_2 & \mathbf{B}_1 & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_2 & \mathbf{A}_3 & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{K-2} & \mathbf{B}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C}_{K-2} & \mathbf{A}_{K-1} & \mathbf{B}_1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{C}_{K-1} & \mathbf{A}_K \end{bmatrix}$$

The elements of the transition rate matrix $\mathbf{Q}$ as block submatrices are represented as follows.

$$\mathbf{A}_0 = \begin{bmatrix} -\lambda \end{bmatrix}, \qquad \mathbf{B}_0 = \begin{bmatrix} -\lambda & 0 \end{bmatrix},$$

$$\mathbf{B}_1 = \begin{bmatrix} \lambda\xi & 0 \\ 0 & \lambda\xi \end{bmatrix}, \qquad \mathbf{C}_0 = \begin{bmatrix} \mu_b^1 \\ 0 \end{bmatrix},$$

$$\mathbf{A}_n = \begin{bmatrix} a_{11}^{(n)} & a_{12}^{(n)} \\ a_{21}^{(n)} & a_{22}^{(n)} \end{bmatrix}$$

We depict each element of the block submatrix $\mathbf{A}_n; n = 1, 2, \cdots, K$ as the scalar $a_{ij}^{(n)}$ whose closed form structure is defined as follows

$$a_{ij}^{(n)} = \begin{cases} -\left(\lambda\xi + \alpha + \mu_b^{(n)}\right), & i = j = 1\,;\; 1 \le n \le K-1 \\ -\left(\alpha + \mu_b^{(n)}\right), & i = j = 1\,;\; n = K \\ \quad\alpha, & i < j\,;\; 1 \le n \le K \\ \quad\beta, & i > j\,;\; 1 \le n \le K \\ -\left(\lambda\xi\right), & i = j = 2\,;\; n = 1 \\ -\left(\lambda\xi + \mu_d\right), & i = j = 2\,;\; 2 \le n \le T-1 \\ -\left(\lambda\xi + \beta + \mu_d\right), & i = j = 2\,;\; T \le n \le K-1 \\ -\left(\beta + \mu_d\right), & i = j = 2\,;\; n = K \\ \quad 0, & \text{otherwise} \end{cases}$$

Similarly, we define the block submatrix $\mathbf{C}_n; n = 1, 2, \cdots, K-1$ as

$$\mathbf{C}_n = \begin{bmatrix} c_{11}^{(n)} & 0 \\ 0 & c_{22}^{(n)} \end{bmatrix}$$

where, element of the matrix $\mathbf{C}_n$ for $n = 1, 2, \cdots, K-1$ is the scalar $c_{ii}^{(n)}$ outlined as

$$c_{ii}^{(n)} = \begin{cases} \mu_b^{(n+1)}, & i = 1\ ;\ 1 \le n \le K-1 \\ \mu_d, & i = 2\ ;\ 1 \le n \le K-1 \\ 0, & \text{otherwise} \end{cases}$$

Let $\tilde{\mathbf{P}} = \left[\tilde{\mathbf{P}}_0, \tilde{\mathbf{P}}_1, \cdots, \tilde{\mathbf{P}}_{K-1}, \tilde{\mathbf{P}}_K\right]$ be the probability vector in equilibrium associated to the predefined generator matrix $\mathbf{Q}$. Considering the partition of the probability vector $\tilde{\mathbf{P}}$, we represent governing system of equations in matrix form as

$$\tilde{\mathbf{P}}\mathbf{Q} = \mathbf{0} \tag{13}$$

The homogeneous governing system of equations 13 can straightforwardly be represented in the form of pre-defined block submatrices as

$$\tilde{\mathbf{P}}_0\mathbf{A}_0 + \tilde{\mathbf{P}}_1\mathbf{C}_0 = \mathbf{0} \tag{14}$$

$$\tilde{\mathbf{P}}_0\mathbf{B}_0 + \tilde{\mathbf{P}}_1\mathbf{A}_1 + \tilde{\mathbf{P}}_2\mathbf{C}_1 = \mathbf{0} \tag{15}$$

$$\tilde{\mathbf{P}}_{n-1}\mathbf{B}_1 + \tilde{\mathbf{P}}_n\mathbf{A}_n + \tilde{\mathbf{P}}_{n+1}\mathbf{C}_n = \mathbf{0}, \quad n = 2, 3, \cdots, K-1 \tag{16}$$

$$\tilde{\mathbf{P}}_{K-1}\mathbf{B}_1 + \tilde{\mathbf{P}}_K\mathbf{A}_K = \mathbf{0} \tag{17}$$

Now, after appropriate matrix operation and recursive substitution of each element, we obtain

$$\tilde{\mathbf{P}}_0 = \tilde{\mathbf{P}}_1\mathbf{C}_0\left(-\mathbf{A}_0^{-1}\right) = \tilde{\mathbf{P}}_1\mathbf{\Xi}_0$$

$$\tilde{\mathbf{P}}_1 = \tilde{\mathbf{P}}_2\mathbf{C}_1\left[-\left(\mathbf{\Xi}_0\mathbf{B}_0 + \mathbf{A}_1\right)^{-1}\right] = \tilde{\mathbf{P}}_2\mathbf{\Xi}_1$$

$$\tilde{\mathbf{P}}_n = \tilde{\mathbf{P}}_{n+1}\mathbf{C}_n\left[-\left(\mathbf{\Xi}_{n-1}\mathbf{B}_1 + \mathbf{A}_n\right)^{-1}\right] = \tilde{\mathbf{P}}_{n+1}\mathbf{\Xi}_n, \quad n = 2, 3, \cdots, K-1$$

where,

$$\mathbf{\Xi}_n = \begin{cases} -\mathbf{C}_0\mathbf{A}_0^{-1}, & n = 0 \\ -\mathbf{C}_1\left(\mathbf{\Xi}_0\mathbf{B}_0 + \mathbf{A}_1\right)^{-1}, & n = 1 \\ -\mathbf{C}_n\left(\mathbf{\Xi}_{n-1}\mathbf{B}_1 + \mathbf{A}_n\right)^{-1}, & 2 \le n \le K-1 \end{cases}$$

Again by the recursive back substitution, we redefine each of the state probability vector $\tilde{\mathbf{P}}_n$ in the closed product form of $\mathbf{\Xi}_n;\ n = 0, 1, 2, \cdots, K-1$ as

$$\tilde{\mathbf{P}}_n = \tilde{\mathbf{P}}_K\{\mathbf{\Xi}_{K-1}\mathbf{\Xi}_{K-2}\mathbf{\Xi}_{K-3}\cdots\mathbf{\Xi}_{n+2}\mathbf{\Xi}_{n+1}\mathbf{\Xi}_n\}, \quad n = 0, 1, 2, \cdots, K-1$$

$$\tilde{\mathbf{P}}_n = \tilde{\mathbf{P}}_K\left(\prod_{i=n}^{K-1}\mathbf{\Xi}_i\right) = \tilde{\mathbf{P}}_K\mathbf{\Phi}_n, n = 0, 1, 2, \cdots, K-1 \tag{18}$$

Following the total probability rule, we define the normalization condition for the state-probability distribution as $\tilde{\mathbf{P}}\mathbf{e} = 1$, which can be equivalently rewritten using the partition of the probability vector as

$$\left[\tilde{\mathbf{P}}_0\mathbf{e}_1 + \tilde{\mathbf{P}}_1\mathbf{e}_2 + \tilde{\mathbf{P}}_2\mathbf{e}_2 + \cdots + \tilde{\mathbf{P}}_{K-1}\mathbf{e}_2 + \tilde{\mathbf{P}}_K\mathbf{e}_2\right] = 1 \tag{19}$$

where $\mathbf{e}_1$ and $\mathbf{e}_2$ are column vectors having order one and two respectively such that each element of both the vectors is unity. Now using the Eq.(18), the Eq.(19) can be redefined as

$$\tilde{\mathbf{P}}_K\mathbf{\Phi}_0\mathbf{e}_1 + \left[\tilde{\mathbf{P}}_1 + \tilde{\mathbf{P}}_2 + \cdots + \tilde{\mathbf{P}}_{K-1} + \tilde{\mathbf{P}}_K\right]\mathbf{e}_2 = 1$$

$$\tilde{\mathbf{P}}_K\mathbf{\Phi}_0\mathbf{e}_1 + \left[\tilde{\mathbf{P}}_K\mathbf{\Phi}_1 + \tilde{\mathbf{P}}_K\mathbf{\Phi}_2 + \cdots + \tilde{\mathbf{P}}_K\mathbf{\Phi}_{K-1} + \tilde{\mathbf{P}}_K\right]\mathbf{e}_2 = 1$$

$$\tilde{\mathbf{P}}_K\mathbf{\Phi}_0\mathbf{e}_1 + \tilde{\mathbf{P}}_K\left[\mathbf{\Phi}_1 + \mathbf{\Phi}_2 + \cdots + \mathbf{\Phi}_{K-1} + \mathbf{I}\right]\mathbf{e}_2 = 1$$

$$\implies \tilde{\mathbf{P}}_K\left[\mathbf{\Phi}_0\mathbf{e}_1 + \left(\prod_{n=1}^{K-1}\mathbf{\Phi}_n + \mathbf{I}\right)\mathbf{e}_2\right] = 1 \tag{20}$$

The state probability vector $\tilde{\mathbf{P}}_K$ is evaluated from Eq.(17) and Eq.(20), henceforth, all the other steady-state probabilities $\tilde{\mathbf{P}}_0, \tilde{\mathbf{P}}_1, \cdots, \tilde{\mathbf{P}}_{K-1}$ are evaluated from the Eq.(18).

Following the computation of state probabilities, we define various performance indices in the next section to tract the modeling and analyze the efficiency of the service system.

4. System Performance Measures

In general, there are many standard system performance indicators that can effectively illustrate the quality performance of the service systems. This paper also introduces several queueing-based system performance indices for finite capacity service systems with service pressure coefficient, threshold-based recovery policy, and working breakdown to outline the modeling and methodology used. These system performance measures prove valuable for conducting the parametric investigation to achieve the objective of decision-making. Moreover, all the system performance indicators defined in this section are correlated and recognized as prime importance in a specific situation. Next, we characterize these system performance indicators in the closed/vector form in terms of governing state probabilities.

- Expected number of customers in the queueing system

$$L_S = \mathbf{\Pi}_K \left(\sum_{n=1}^{K-1} n \mathbf{\Phi}_n \mathbf{e}_2 + K \mathbf{e}_2 \right) \tag{21}$$

  Queue length can influence balking behavior, where customers might opt not to join a queue if it appears too lengthy, impacting potential revenue.

- Expected number of customers in the waiting queue

$$L_Q = \mathbf{\Pi}_K \left(\sum_{n=1}^{K-1} (n-1) \mathbf{\Phi}_n \mathbf{e}_2 + (K-1) \mathbf{e}_2 \right) \tag{22}$$

- Probability that server is in working breakdown state

$$P_{WD} = \mathbf{\Pi}_K \left(\sum_{n=1}^{K-1} \mathbf{\Phi}_n \mathbf{e}_3 + \mathbf{e}_3 \right) \tag{23}$$

  where, $\mathbf{e}_3 = [0\ 1]^{\mathrm{T}}$

- Probability that the server is in a busy state

$$P_B = \mathbf{\Pi}_K \left(\sum_{n=1}^{K-1} \mathbf{\Phi}_n \mathbf{e}_4 + \mathbf{e}_4 \right) \tag{24}$$

  where, $\mathbf{e}_4 = [1\ 0]^{\mathrm{T}}$

- Probability that server is idle

$$P_I = \mathbf{\Pi}_0 \mathbf{e}_1 \tag{25}$$

- Throughput of the service system

$$\begin{aligned} \tau_p = & \mathbf{\Pi}_K \left(\sum_{n=1}^{K-1} \mu_b^{(n)} \mathbf{\Phi}_n \mathbf{e}_4 + \mu_b^{(K)} \mathbf{e}_4 \right. \\ & \left. + \sum_{n=1}^{K-1} \mu_d \mathbf{\Phi}_n \mathbf{e}_3 + \mu_d \mathbf{e}_3 \right) \end{aligned} \tag{26}$$

  Throughput refers to the rate at which customers successfully pass through a service system. It signifies the system's capacity to handle and process incoming customers. This is a critical performance measure as it offers insights into the efficiency and effectiveness of the service system. Throughput can be influenced by various factors, including arrival rates, service rates, the number of servers, and the queueing discipline. Analyzing throughput assists in understanding the system's ability to meet demand and process customers efficiently, which in turn aids in system optimization and performance evaluation.

- Average balking rate

$$\mathrm{ABR} = \mathbf{\Pi}_K \left(\sum_{n=1}^{K-1} (1-\xi) \lambda \mathbf{\Phi}_n \mathbf{e}_2 \right) \tag{27}$$

The average balking rate is a crucial metric in queueing theory, indicating how often customers choose not to join a queue due to various factors. It provides insights into customer behavior, helps optimize service resources, and guides improvements in service quality. High balking rates can signal issues with wait times and service quality, while addressing balking behavior can lead to better resource allocation and enhanced customer experiences.

- Effective arrival rate

$$\lambda_{\text{eff}} = \lambda \mathbf{\Pi}_0 \mathbf{e}_1 + \mathbf{\Pi}_K \left(\sum_{n=1}^{K-1} \xi \lambda \mathbf{\Phi}_n \mathbf{e}_2 \right) \tag{28}$$

- Expected waiting time in the service system

$$W_S = \frac{\mathbf{\Pi}_K \left(\sum_{n=1}^{K-1} n \mathbf{\Phi}_n \mathbf{e}_2 + K\mathbf{e}_2 \right)}{\lambda \mathbf{\Pi}_0 \mathbf{e}_1 + \mathbf{\Pi}_K \left(\sum_{n=1}^{K-1} \xi \lambda \mathbf{\Phi}_n \mathbf{e}_2 \right)} \tag{29}$$

Waiting time reflects the duration customers spend in the queue before being served, directly affecting customer satisfaction and system efficiency.

Using above defined performance indices, we develop the cost optimization problem in the next section with pertinent decision parameters and design parameters.

5. Cost Analysis

For the economical analysis of the studied Markovian single-server finite capacity service system, our focus is on the formulation of the mean cost function utilizing different cost factors incurred. The parameters μ_d and μ_b are considered as decision variables. The core purpose of the study is to use the joint stationary probability distribution and system performance characteristics of the developed model to optimize the long-run mean cost at the optimal value of the decision parameters. The key cost elements unified to the different system states of the queueing model are defined as follows.

C_h ≡ The unit cost associated with customers in the service system

C_d ≡ The unit cost associated with the partial breakdown of the server

C_b ≡ The unit cost associated with the busy state of the server

C_i ≡ The unit cost associated with the idle server

C_{μ_b} ≡ Unit cost for providing the service with rate μ_b

C_{μ_d} ≡ Unit cost for providing the service with rate μ_d

C_w ≡ cost associated with each waiting customer present in the system

We use the above-defined components related to the mean cost and performance indices defined in the previous section to formulate the cost function as follows

$$\begin{aligned} TC(\mu_b, \mu_d) =& C_h L_S + C_d P_{WD} + C_b P_B + C_i P_I \\ & + C_{\mu_b} \mu_b + C_{\mu_d} \mu_d + C_w W_S \end{aligned} \tag{30}$$

We examine two service modes for waiting customers. The first employs a normal service policy where the server operates at its full capacity. Conversely, the second mode features a reduced service rate due to a partially downed state. Underlying these assumptions is the consistent premise that the service rate during the impaired state remains lower than that of the server's normal operational mode. The notion of multiplying service rates by their state probabilities is relevant under the Bernoulli service regime, it should be noted that this specific aspect has not been explored in the present investigation. The cost optimization (minimization) problem is framed mathematically as an optimal control problem.

$$TC(\mu_b^*, \mu_d^*) = \min_{\mu_d < \mu_b} \{TC(\mu_b, \mu_d)\} \tag{31}$$

where μ_b^* and μ_d^* are the optimized values of decision variables that minimize the mean cost. It is realistic to consider minimizing the mean cost by adjusting the service rate of a server in different states, including when the server is partially broken. In certain real-world situations, service rates can indeed be adjusted based on the server's operational condition. This concept is particularly relevant in systems where equipment can operate at different levels of functionality or efficiency. For example, in manufacturing plants, if a machine experiences partial breakdown or reduced efficiency, its production rate might be adjusted to avoid further damage and maintain a certain level of output. In service industries, such as call centers, if a certain number of operators are unavailable due to technical issues, the service rate could be adjusted to manage incoming calls efficiently. Similarly, in computer systems or cloud computing environments, if certain processing nodes are temporarily unavailable or experiencing issues, the overall computational capacity might be adjusted dynamically to ensure continued operation while minimizing disruption. While the specific mechanisms for adjusting service rates may vary across industries and scenarios, the fundamental concept of adapting service rates based on the server's operational state is indeed applicable in real-world situations. It allows organizations to balance operational efficiency, resource utilization, and customer satisfaction while minimizing costs. We employ the classical and meta-heuristic optimization techniques to determine the optimal mean cost. The details and results are discussed in the forthcoming sections.

6. Special Cases

In this section, for the validity and applicability of developed model, the comparative study with several existing research articles is provided by aligning or relaxing one or more assumptions. It proves that, the results of the governing model resemble with the actual findings in the queueing literature. The presentation demonstrates the versatility and accuracy of our model.

Case 1: For $\xi = 1$, $\mu_b^{(n)} = \mu$, and $\alpha = 0$, the studied model analogous to classical $M/M/1/K$ queueing model (Kleinrock 1975).

Case 2: For $\xi \neq 1$, $\mu_b^{(n)} = \mu$, and $\alpha = 0$, our model and findings align with the outcomes of a queueing system with balking proposed by Haight (1957).

Case 3: By substituting $\xi \neq 1$, $\mu_b^{(n)} = \mu$, $\alpha \neq 0$, and $\beta = 0$, the governing model converts to the queueing problem with working breakdown and customer impatience investigated by Liou (2015).

Case 4: By taking $\xi = 1$, and $\alpha = 0$, the studied model deduces to a queueing model with service pressure coefficient proposed by Hillier (2012).

Case 5: In the case when $\xi = 1$, $\mu_b^{(n)} = \mu$, $\alpha \neq 0$, and $\beta = 0$, the current model resembles with the single server service system with working breakdown of the server proposed by Kalidass and Kasturi (2012).

Case 6: By setting the combination of parameters as $\xi = 1$, $\mu_b^{(n)} = \mu$, $\alpha \neq 0$, and $\beta \neq 0$, the model becomes a finite capacity queue-based service system with working breakdown and threshold-based recovery policy which was examined by Efrosinin and Semenova (2010) in the literature.

Through these comparative cases, we showcase the alignment of our model with diverse scenarios from the queueing literature, validating its comprehensiveness and applicability.

7. Optimization Techniques

In the pursuit of optimizing the system's operational efficiency and minimizing mean cost, we employ the semi-classical and meta-heuristic techniques to determine the optimal value of decision parameters. The results of each technique are compared to others to validate the newly evolved meta-heuristic techniques. The results are compiled in the next section. In the subsequent subsection, we give detail, algorithm, procedure, and pseudo-code of semi-classical method: quasi-Newton method (QN), and meta-heuristic optimization techniques: particle swarm optimization (PSO), cuckoo search (CS).

7.1 Quasi-Newton Method

The literature on optimization algorithms shows that gradient-based optimization algorithms have errors ($i.e.$ zigzagging) when dealing with ill-conditioned optimization problems. As a solution, the quasi-Newton technique of order two is gaining interest as it uses curvature information and efficacy in dealing with ill-conditioned cost optimization problems. Second-order techniques offer advantageous over the first-order methods, including a high rate of local converging simulations (usually super-linear) and preserving invariance (non-sensitiveness to the choice of coordinates) due to its quicker estimation of Jacobian matrices, particularly when dealing with extensive solution space ranges. Inspiring by this fact, we have incorporated the semi-classical optimizer: the QN method, for the governing multi-objective problem. The advantage of the QN method for multi-objective and multi-constraint optimization is that the estimation of Jacobian matrices is reasonably faster than their actual estimation. This change is significantly more apparent when the range of the problem's solution space is extensive.

The QN method provides the optimal results in two steps. First, we compute a search direction p^t, which indicates the direction of the input space (vector including initial values of system design parameters) at iteration t. The second step determines how far we have to move in this direction by computing a step length $\alpha^t \in \mathbb{R}_+$. Therefore, it is an optimization method that searches for optimality with a descent direction.

$$p^t = -(H^t)\nabla f(x^t) \tag{32}$$

We then obtain the next iterate as

$$x^{t+1} = x^t + \alpha^t p^t \tag{33}$$

Here, the Hessian approximation $B^t \simeq (H^t)^{-1}$ must satisfy the quasi-Newton condition called secant equation.

$$B^t(x^{t+1} - x^t) = y^t \tag{34}$$

where, $y^t = \nabla f(x^{t+1}) - \nabla f(x^t)$ in which $f : D \to \mathbb{R}$ is continuously differentiable function on the domain D and $\nabla f(x^t) \in \mathbb{R}^n$ denotes the gradient of f at x^t. Further, instead of computing the actual Hessian in the quasi-Newton method, we approximate the Hessian with the help of a positive definite symmetric matrix $B^t \in \mathbb{R}^{n\times n}$, which is updated at every iteration as

$$B^{t+1} = B^t + U$$

Now, we utilize the concept of the popular BFGS-method to compute the matrix U as

$$U = \frac{y^t(y^t)^T}{(y^t)^T(s^t)} - \frac{(B^t s^t)(B^t s^t)^T}{(s^t)^T B^t s^t} \tag{35}$$

where, $s^t = \alpha^t p^t$ and $(y^t)^T$ represents transpose of y^t.

The distinction between the QN method and the original Newton's method lies in the utilization of the analytically computed Jacobian matrix $J(x^t)$ as a replacement for B^t. The primary difference between Newton and the QN method is that Newton method uses the exact Jacobian matrix while the QN method uses approximated results. Therefore, the QN method is more famous for feasible superlinear convergence and is not calculated the Jacobian if some of the involved functions are twice continuously differentiable and strongly non-convex or convex (Zhou 2020). The procedure for utilizing the quasi-Newton method is outlined below:

Procedure for QN method

Initialization:

1. Choose an initial guess x^0 for the solution.
2. Initialize the iteration counter.
3. Set a positive definite matrix B^0 as an approximation to the Hessian.

Iteration:

1. Calculate the search direction by solving $B^t p^t = -\nabla f(x^t)$.
2. Choose a step size α^t (line search or other methods).
3. Update the solution $x^{t+1} = x^t + s^t$.
4. Compute gradient at new point $\nabla f(x^{t+1})$.

Updating Approximation:

1. Calculate changes $s^t = x^{t+1} - x^t$ and $y^t = \nabla f(x^{t+1}) - \nabla f(x^t)$.
2. Update B^{t+1} using a formula (eg., BFGS update).

Termination:

1. Check convergence criteria (iterations, gradient, changes in x).

Repeat:

1. If not converged, increment as per step-size and repeat iteration.

Final Result:

1. Once terminated, x^{new} is the approximate optimal solution.

The pseudo-code for quasi-Newton method

```
**Initialize:** starting point $x^0$, $B^0$, and $t_{max}$;
**for**($t < t_{max}$)
    solve $B^t p^t = -\nabla f(x^t)$;
    step size $s^t = \alpha^t p^t$ (line search along $p^t$);
    update iteration $x^{t+1} = x^t + s^t$
    update $B^{t+1} = B^t + U$, where $U$ is given by Eq.(35)
**end for**
**Output:** $B^{new}$ and $x^{new}$
```

7.2 Particle Swarm Optimization

Particle swarm optimization (PSO) algorithm is an agent-based optimization technique, which was firstly introduced by Kennedy and Eberhart in 1995 (Kennedy and Eberhart 1995) having been inspired by swarm intelligence and its collective movement. When birds (particles) fly in a flock (swarm) to search for food randomly, they share information about what they find among themselves and help the entire flock get the best hunt. Similarly, particles in PSO share information to collectively improve their search for the best solution. The roaming nature of birds in the flock will inspire the exploration phase of the optimization procedure, which aims to avoid being stuck in the local region.

PSO is a bio-inspired process that searches for an optimal solution in the solution space globally. PSO algorithm excels at tackling non-linear, non-convex, and multi-modal optimization problems. Multiple local and global optimal are present, and we need to obtain the global optimum of the problem. In PSO, we use both global best (p^t_{gb}) and the individual (particle) best ($p^{t^*}_i$) concurrently at the iteration t. Using certain individuals best aims to escalate the diversity in the promising solutions; however, this diversity may be mimicked by employing randomization. As a result, if the optimization problem of interest is substantially non-linear and multi-modal, there is no convincing justification for choosing the individual best (Yang 2020). An initial set of locations (solutions) (p^0_i) and velocities (v^0_i) are generated randomly for each particle (bird) in the swarm (flock). Each particle's speed is stochastically accelerated towards its prior best position (individual best) and the global best solution across iterations in the search space (Lindfield and Penny 2017).

$$v^{t+1}_i = v^t_i + c_1 r_1 (p^{t^*}_i - p^t_i) + c_2 r_2 (p^t_{gb} - p^t_i) \quad (36)$$

where c_1 and c_2 are positive constants chosen at the initiation of the process. The vector $p^{t^*}_i$ is the finest position (best solution) for the particles till time instant t, determined using the objective function $f(p_i)$ in the local search region. The vector p^t_{gb} is defined as the universally best ($i.e.$, global best) position vector for all the particles. At each iteration, the solution vector is updated to provide the terminating optimum position. The vectors p^t_i and v^t_i are the current values of the position and velocity vector respectively. Furthermore, r_1 and r_2 are the random vectors chosen from the uniformly distributed random variate r_u in the continuous range [0, 1], re-selected at each iteration of the algorithm. Here, randomness shows a significant role in avoiding getting trapped at a local optimum and fostering exploration of the solution space.

The second term of Eq.(36) assures complete exploitation of the local area in the search space to pinpoint an accurate value of the local optimum. Similarly, the third term of Eq.(36) prompts that the entire search space is explored to find a global optimum and escape getting confined at a local optimum. Thus, the choice of c_1 and c_2 is critical in confirming compatibility, and hence their selection should be made sensibly.

Concurrently, each particle's position is updated according to its velocity. The position updating formula is defined as

$$p^{t+1}_i = p^t_i + v^{t+1}_i \quad (37)$$

The equation above describes a global exploration process, ensuring that each new point is evaluated for potential improvement across the solution space. Further, we adopt the concept of inertia function $\Omega(t)$ (Shi and Eberhart 1998) to stabilize the exploration of the particles. This function prevents particles to be stuck in a local region or overshoot from optimum value. Henceforth, the velocity formula is restructured as follows

$$v_i^{t+1} = \Omega v_i^t + c_1 r_1 (p_i^{t^*} - p_i^t) + c_2 r_2 (p_{gb}^t - p_i^t) \tag{38}$$

The appropriate value of the inertia function $\Omega(t)$ is chosen within the range $[0.5, 0.9]$. Here is the procedure for employing the particle swarm optimization algorithim:

Procedure for PSO algorithm

Initialization:

1. Initialize a population of particles with random positions and velocities.
2. Set personal best positions $(p_i^{t^*})$ of each particle to its initial position.
3. Set global best position (p_{gb}^t) to the position of the best particle.

Iteration:

1. For each particle, calculate its fitness value.
2. For each particle, update personal best positions $(p_i^{t^*})$ if the current fitness is better.
3. For each particle, update global best position (p_{gb}^t) if the current fitness is better than the overall best.
4. For each particle, update particles velocity and position based on its previous velocity, position, $(p_i^{t^*})$, and (p_{gb}^t).

Termination:

1. Check convergence criteria (iterations, fitness, solution improvement).
2. If convergence is met, terminate; otherwise, continue to the next iteration.

Repeat:

1. If not converged, increment the iteration count and repeat the iteration process.

Final Result:

1. Once terminated, the position of (p_{gb}^t) represents the optimal solution found by the algorithm.

The pseudo-code for Particle Swarm Optimization algorithm

```
Input: Objective function, population size, $r_1$, $r_2$, $c_1$, $c_2$, starting particle position, $t_{max}$;
Initialize population: find position of $n$ particles;
while($t < t_{max}$ or convergence criterion)
for loop over all $n$ particles and all $d$ dimensions
    update new velocity $v_i^{t+1}$ according to Eq.(38);
    update new position of particle $p_i^{t+1}$ according to Eq.(37);
    evaluate objective function at new position;
    find the current best position $(p_i)$ for each particle;
end for
    update global best $p_{gb}$;
    $t \to t+1$;
end while
```

Output: optimal objective value TC^* at p^*

7.3 Cuckoo Search

Cuckoos are captivating birds, known not just for their melodious sounds but also for their peculiar reproduction method. Most of the cuckoo species lay their eggs in communal nests, and they may throw down the eggs of others to maximize the chances of their eggs hatching. Nevertheless, some species practice obligatory brood parasitism, which involves laying their eggs in the nests of other host birds. Some cuckoo species have evolved due to genetic variation where female parasitic cuckoos are capable of imitating the color and pattern of eggs of certain host species. The behavior lessens the likelihood of their eggs forsaking, increasing their reproductive potential. The competitiveness dynamics between cuckoos and host species forms a combat system where cuckoos' eggs can be exposed and thrown down with a probability of P^*.

In the algorithm, the resemblance of two eggs (solutions) p_i and p_j can be roughly evaluated by their difference $(p_j - p_i)$. Thus, the location at iteration t can be modified by

$$p_i^{t+1} = p_i^t + \alpha s \otimes H(P_a - \epsilon) \otimes (p_j^t - p_k^t) \quad (39)$$

where s is step-size, which is ranged by a variable α that takes positive values, H is a Heaviside step-function used to simulate the discovery probability with the help of random number ϵ taken from a uniform distributed range $[0, 1]$. Furthermore, the product notation $\otimes$ of two vectors means entry-wise multiplications. Now, for generating new solution p_i^{t+1} for the i^{th} cuckoo, a Lévy flight is performed as

$$p_i^{t+1} = p_i^t + \alpha L(s, \lambda) \quad (40)$$

where the Lévy flights are random walks with phases being taken from

$$L(s, \lambda) \sim \frac{1}{s^{1+\lambda}} \left(\frac{\lambda \Gamma(\lambda) \sin(\pi\lambda)/2}{\pi} \right) \quad (41)$$

to approximate a Lévy probability distribution with an exponent $0 \leq \lambda \leq 2$. Here, the gamma function is defined as

$$\Gamma(\lambda) = \int_0^\infty z^{\lambda-1} e^{-u} du \quad (42)$$

The procedure for applying the cuckoo search algorithm is as follows:

Procedure for CS algorithm

Initialization:

1. Initialize a population of cuckoos (solution candidates).
2. Assign random solutions (nest positions) to the cuckoos.

Iteration:

1. For each cuckoo, generate a new solution by performing random walk or Lévy flights.
2. For each cuckoo, evaluate the fitness of the new solution.
3. For each cuckoo, replace the old solution with the new one if it's better.
4. For each cuckoo, some solutions are randomly abandoned (cuckoos lay eggs in other nests).

Lévy Flights:

1. Used for exporing solution space efficiently.
2. Lévy flights introduce randomness while taking larger steps in random directions.

Lévy Flight Equation:

1. Generate step size using the Lévy distribution.

2. Update position using the current position and step size.

Termination:

1. Check convergence criteria (iterations, fitness, solution improvement).
2. If convergence is met, terminate; otherwise, continue to the next iteration.

Repeat:

1. If not converged, increment the iteration count and repeat the iteration process.

Final Result:

1. Once terminated, the best solution found among the cuckoos represents the optimal solution according to the algorithm.

The pseudo-code for Cuckoo Search algorithm

```
Objective function: $TC(x)$, $x = \{x_1^0, x_2^0, \cdots, x_d^0\}$, population size, $t_{max}$;
Initialize: population of $n$ host nests $x_i (1 \leq i \leq n)$;
while($t < t_{max}$ or convergence criterion)
    Get a cuckoo randomly (say, $i$) by Lévy distribution;
    Evaluate its fitness value $F_i$;
    Choose a nest among $n$ (say, $j$) randomly;
    Evaluate its fitness value $F_j$;
    if ($F_i > F_j$)
        replace $j$ by the new solution;
    end if
    Abandon a fraction ($P_a$) of worse nests and built new ones;
    Keep the best solutions/nests;
    Rank the solutions/nests and find the current best;
    Rank the solutions/nests and find the current best;
end while
Output: If the stopping criterion is met, then $p^*$ is the best global solution found so far.
```

8. Results and Discussion

In this section, several numerical examples are given to perform the sensitivity analysis of the stationary system performance indices of the proposed single server finite capacity service system for various intricate system parameters. The numerical results and illustrations are outlined in Figs. 1–4, which showcase the influence of several system parameters on the key system performance indices, namely, the mean number of customers in the service system (L_S), and throughput of the service system (τ_p).

For illustrations, we standardize the capacity of the service system as $K = 15$ and threshold $T = 7$. The other system parameters are fixed as follows: $\lambda = 1.5$, $\xi = 0.7$, $\mu_b = 3.0$, $\mu_d = 1.5$, $\psi = 1.0$, $\alpha = 0.01$, $\beta = 8.0$. In Figs. 1 and 2, we illustrate the line graphs for the mean number of customers in the service system w.r.t. λ and μ_b, respectively, for the varied parametric values of design parameters T, ξ, α, and ψ. It is easy to observe that L_S shows a growing trend on increasing values of λ and the reverse effect on increased values of μ_b as intuitively expected. For the fixed value of λ, an increasing trend is observed for the higher values of T, ξ, and α as in Fig. 1. Nevertheless, at the same time, the reverse trend is observed in the case of ψ. Similarly, in Fig. 2-(iv), it is observed that for the definite values

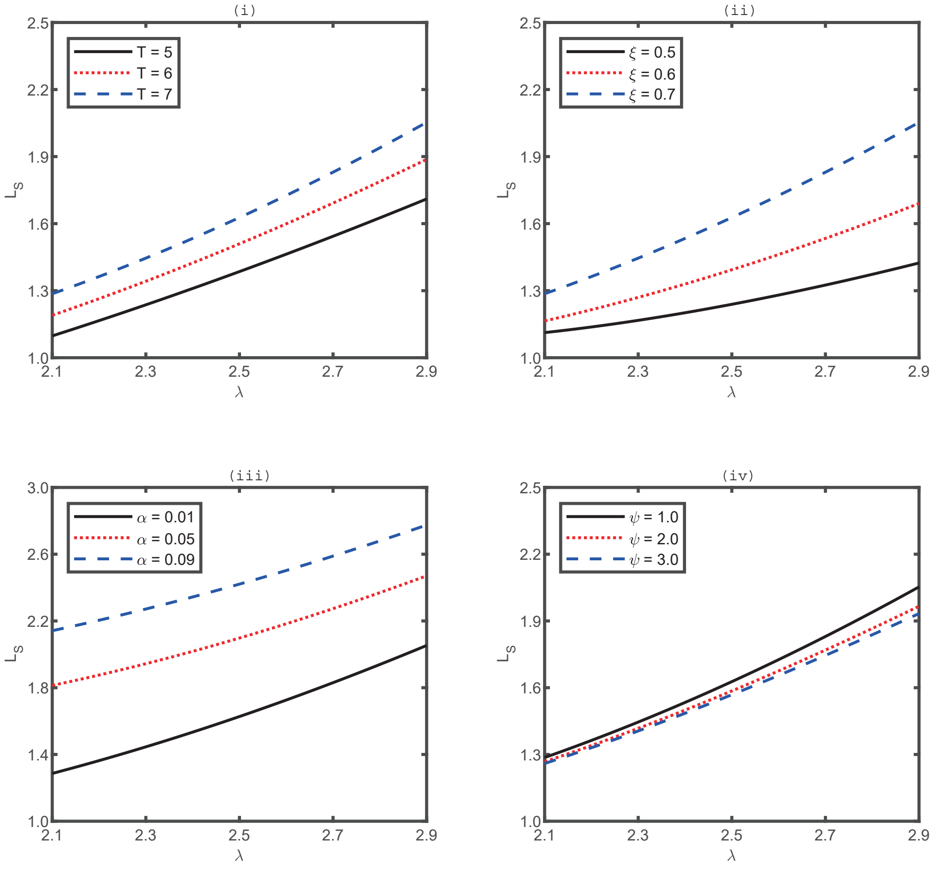

Figure 1 Effect of Varied (i) T, (ii) ξ, (iii) α, and (iv) ψ w.r.t. λ on Mean Number of Customers in the Service System

of μ_b and increasing ψ, L_S is decreasing. It is apparent from the fact that as the pressure factor increases, the active servers' service rate increases, which results in a decreasing trend in L_S.

The influence of system parameters λ and μ_b on the throughput (τ_p) of the service system is depicted in Figs. 3 and 4, respectively, as bar graphs. These figures provide a better and more important understanding to the system analysts on distinguishing the variations of throughput of the service system w.r.t. to various system parameters value. Throughput gives the mean number of customers served by the server either in normal mode or partial breakdown state; subsequently, it increases when the number of arrivals in the service system increases and the service rate increases. The parameter ξ positively affects throughput, as shown in Figs. 3(ii) & 4(ii), whereas T and α negatively that can be observed in Figs. 3(i) & (iii) and Figs. 4(i) & (iii). Moreover, τ_p is the least sensitive w.r.t. ψ, which results in a minor change with higher values of ψ, as presented in Figs. 3(iv) & 4(iv).

Besides the earlier fixed default value of system parameters, the default values of several cost elements are also considered as C_h =

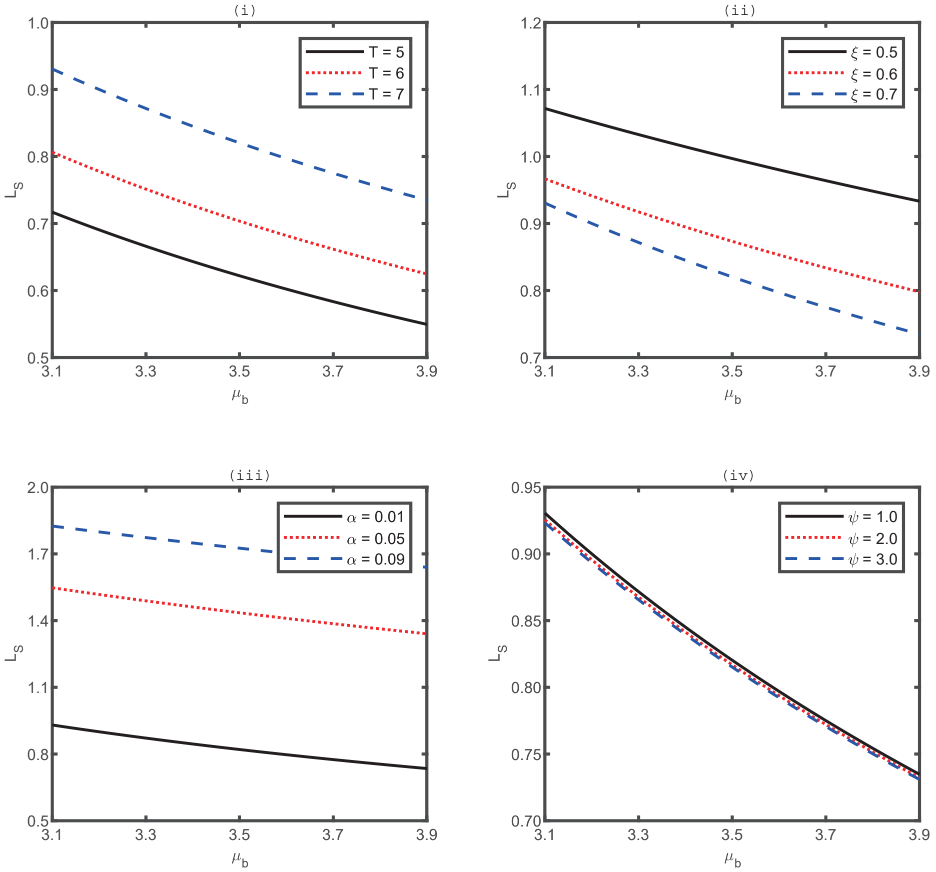

Figure 2 Effect of Varied (i) T, (ii) ξ, (iii) α, and (iv) ψ w.r.t. μ_b on Mean Number of Customers in the Service System

5, $C_d = 60$, $C_b = 250$, $C_i = 170$; $C_{\mu_b} = 2$, $C_{\mu_d} = 17$, and $C_w = 100$ to analyze studied service system economically. For the various combinations of default parameters, Figs. 5 and 6 depict the variation in the value of the mean cost (TC) of the system, given in Eq. (30). Fig. 5(i) characterizes the variation on TC for increasing values of T and λ, revealing that the mean cost (TC) enhances as intuitively expected. From Figs. 5(ii) & (iv), we notice that for higher values of combinations (λ, ξ) and (λ, ψ), the mean cost TC is deduced rapidly in comparison to Fig. 5(iii). This trend can be attributed to the decrease in the expected number of customers in the service system caused by balking and the pressure coefficient. Correspondingly, TC significantly raises with the higher values of parameters μ_b and T as in Fig. 6(i). In Fig. 6(ii), it is noticeable that, first, the TC increases more rapidly w.r.t. positively varied (ξ, μ_b) and remains almost constant later. Similar findings are exhibited for the remaining figures as well. Our findings highlight the intricate interplay of factors like arrival rates, service rates, breakdown probabilities, and recovery policies on the system's efficiency and cost-effectiveness.

Therefore, all of these statistics incite that

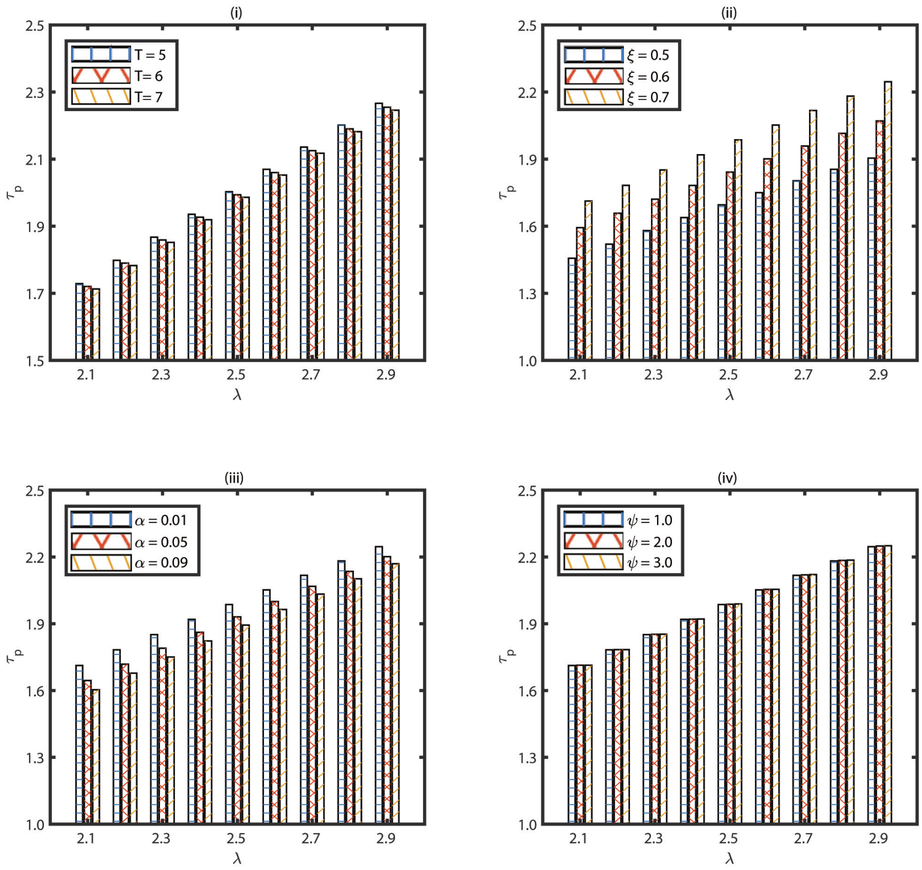

Figure 3 Effect of Varied (i) T, (ii) ξ, (iii) α, and (iv) ψ w.r.t. λ on the Throughput of the Service System

the default parametric values used here are praiseworthy in decision making, planning, and designing the service system, which plays a significant role in the development of the governing model. From the results provided in the above Figs. 1–6, it is perceived that there is a strategic need to estimate the optimal operating policy to minimize the mean cost incurred in the service system. Generally, it is highly typical to evaluate the analytical and closed-form of μ_b^* and μ_d^*, because of the high order complexity and non-linearity involved in the cost optimization problem. The trend for incurred TC w.r.t. to the system design parameters μ_b and μ_d respectively, have been calculated numerically with the help of Figs. 7–9. In this context, the values of different default system parameters and performance associated unit cost, are considered as follows: $K = 20$; $T = 10$; $\lambda = 4.0$, $\xi = 0.3$, $\psi = 1.0$, $\alpha = 0.2$, $\beta = 3.0$, $C_h = 130$, $C_d = 60$, $C_b = 100$, $C_i = 350$, $C_{\mu_b} = 5$, $C_{\mu_d} = 35$, and $C_w = 100$. The lower and upper limits of the decision/system design parameters μ_b and μ_d are taken as [2 20] and [1 7] respectively. From Fig. 7, the conclusion be inferred that the mean cost $TC(\mu_b, \mu_d)$ is convex in nature as intuitively anticipated. Optimizing service

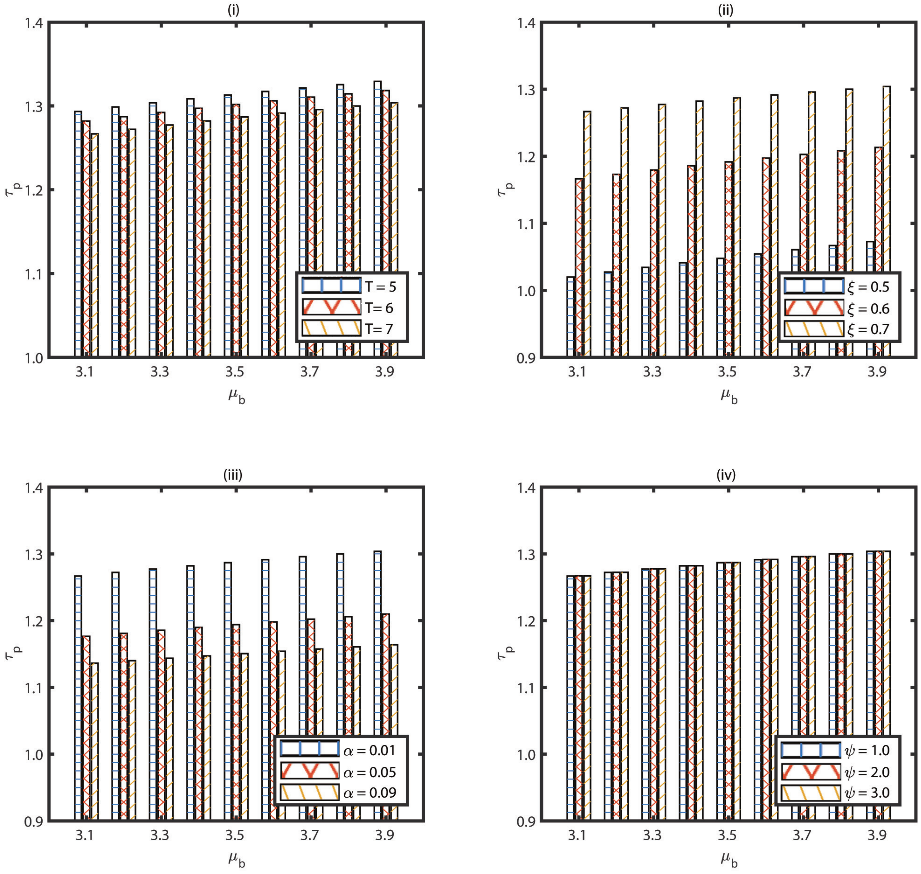

Figure 4 Effect of Varied (i) T, (ii) ξ, (iii) α, and (iv) ψ w.r.t. μ_b on the Throughput of the Service System

system performance hinges on effective cost allocation. Numerical experiments and graphs reveal cost variations, refining parameters for convex cost function graphs. Costs directly impact outcomes, as shown in the graphs. Nature-inspired optimization computes optimal costs and design parameters, confirmed in tables. A careful cost consideration, combined with experimentation, offers a comprehensive understanding of their impact. Our approach draws from prior research articles.

To calculate the optimal combinations of the design decision parameters μ_b and μ_d, the nature-inspired optimization technique: particle swarm optimization (PSO), and cuckoo search (CS) algorithm are utilized. The results are compared with the results of the quasi-Newton method. The results delineated in Figs. 8–10 infer the convex nature of cost function w.r.t. to decision parameters. Several generations of the PSO algorithm have also been depicted in Fig 11 to display the robustness and working nature of the PSO algorithm. These results show that the mean cost of the service system w.r.t. combined values of continuous system design parameters, μ_b and μ_d, is optimal, and the used algorithm plays an essential role in providing converging results.

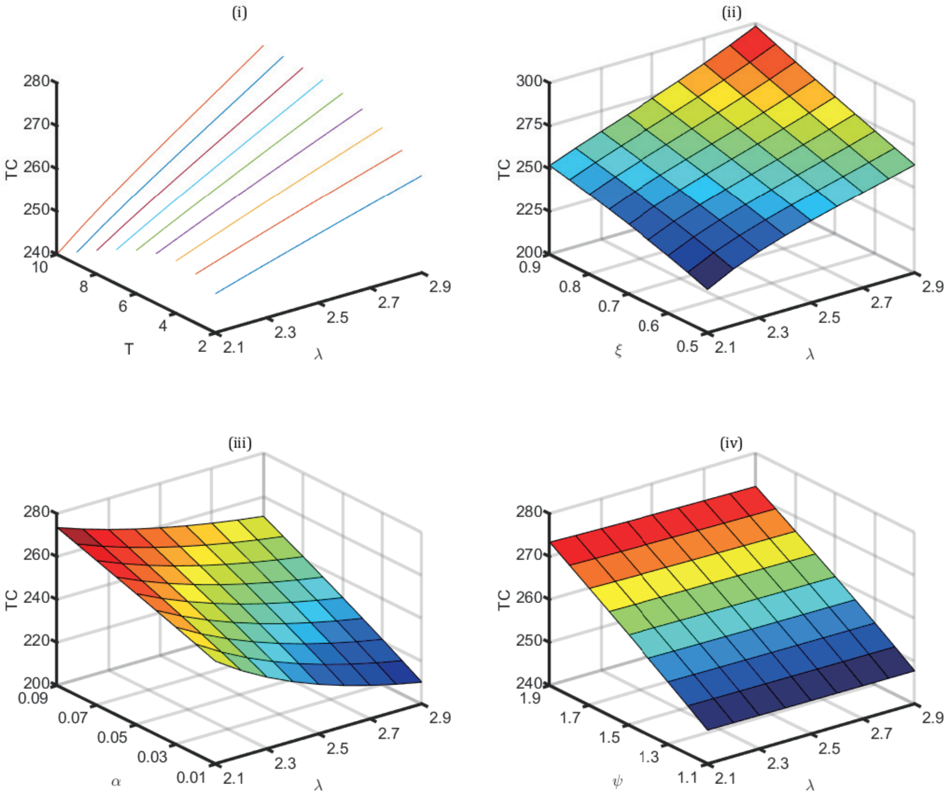

Figure 5 Mean Cost (TC) w.r.t. Varied (i) (T, λ), (ii) (ξ, λ), (iii) (λ, α), and (iv) (ψ, λ)

Table 1 Iterations of QN method in Finding the Optimal Values of μ_b and μ_d

Number of iterations	μ_d	μ_b	$TC(\mu_b, \mu_d)$
0	3.000000	12.000000	549.252374
1	2.005001	11.651540	530.572348
2	2.307443	10.644663	527.526168
3	2.212510	10.519532	526.369986
4	2.175001	10.475921	526.289346
5	2.181594	10.460087	526.285852
6	2.181004	10.457057	526.285811
7	2.180910	10.456292	526.285810
8	2.180910	10.456295	526.285810
9	2.180910	10.456296	526.285810
10	2.180910	10.456297	526.285810
11	**2.180910**	**10.456297**	**526.285810**

Next, we also provide numerous simulations w.r.t. several combinations of system parameters to validate the converging results and the convexity of the formulated cost func-

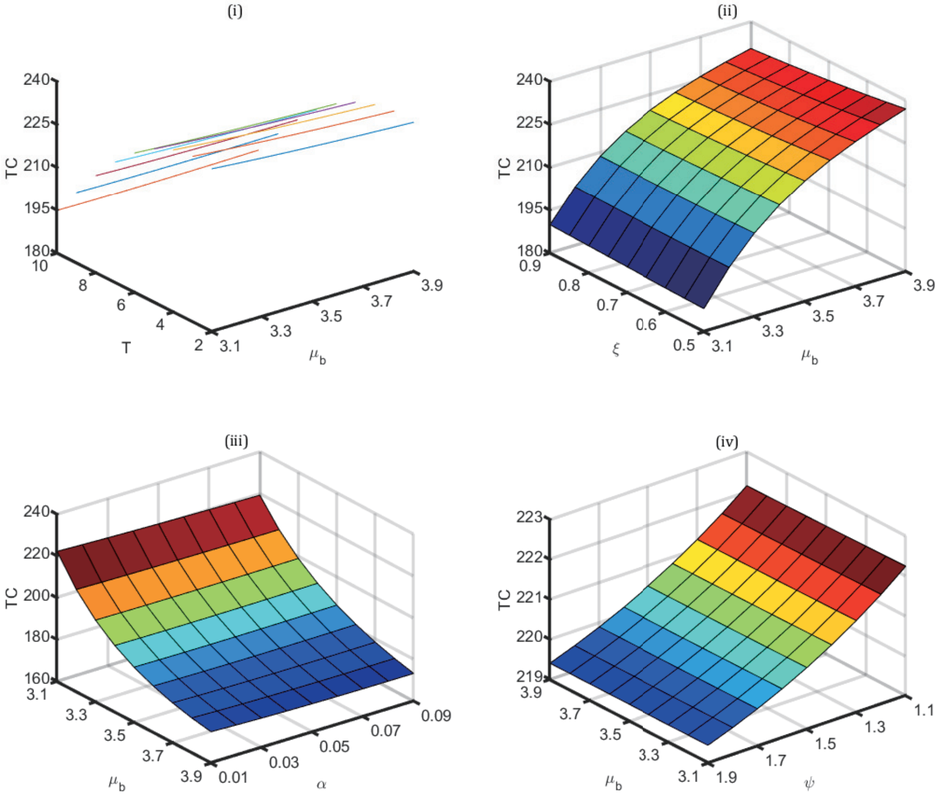

Figure 6 Mean Cost (TC) w.r.t. Varied (i) (T, μ_b), (ii) (ξ, μ_b), (iii) (μ_b, α), and (iv) (μ_b, ψ)

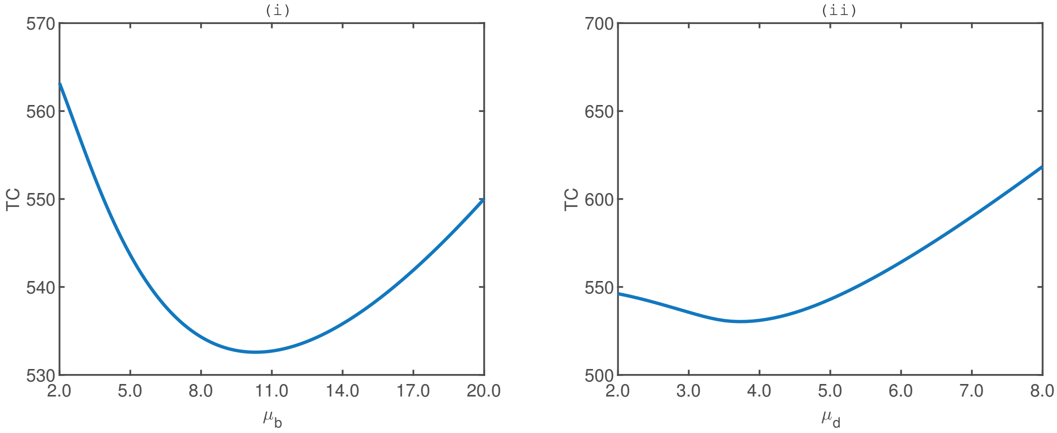

Figure 7 Mean Cost (TC) w.r.t. Decision Variables μ_b and μ_d

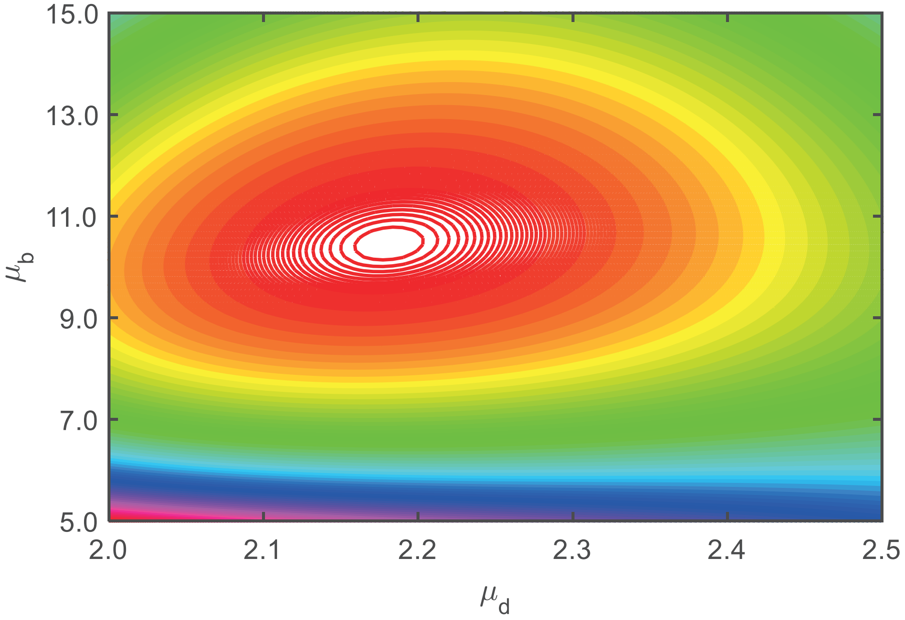

Figure 8 Contour Plot for TC w.r.t. Varied μ_b and μ_d

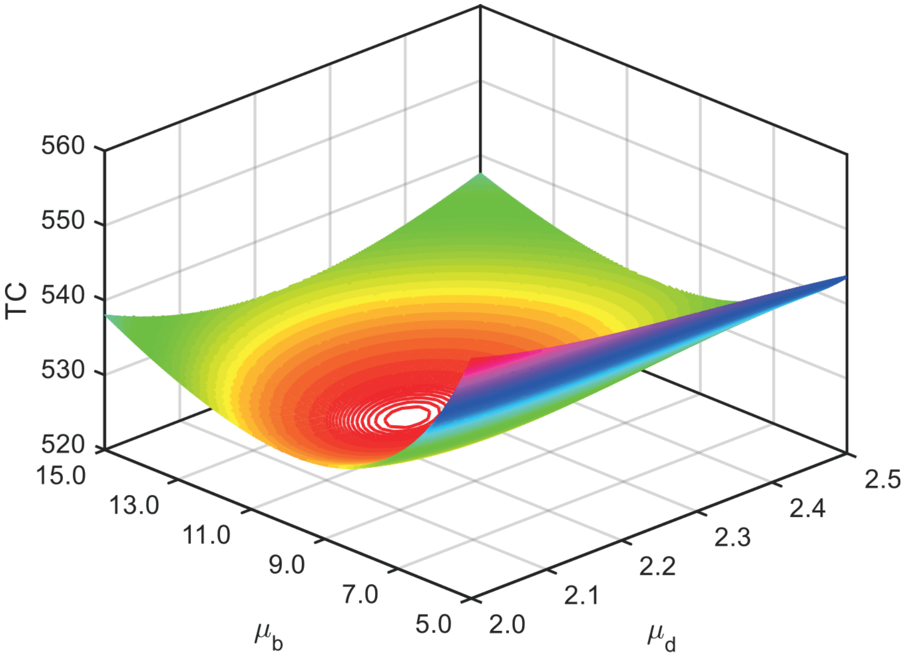

Figure 9 Three Dimensional Contour Plot for Mean Cost (TC) w.r.t. Varied μ_b and μ_d

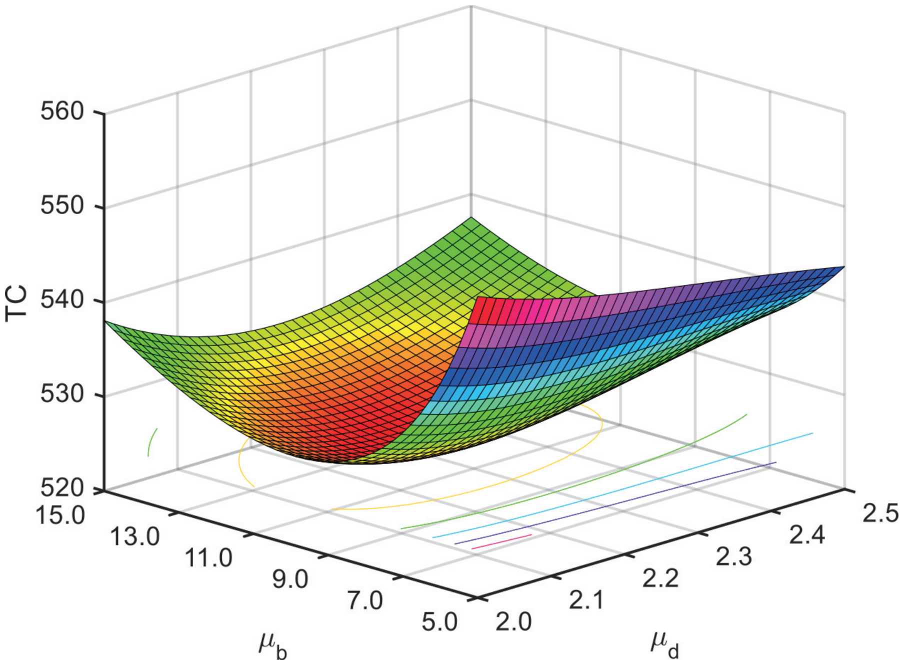

Figure 10 Surface Plot for TC w.r.t. Varied (μ_b, μ_d)

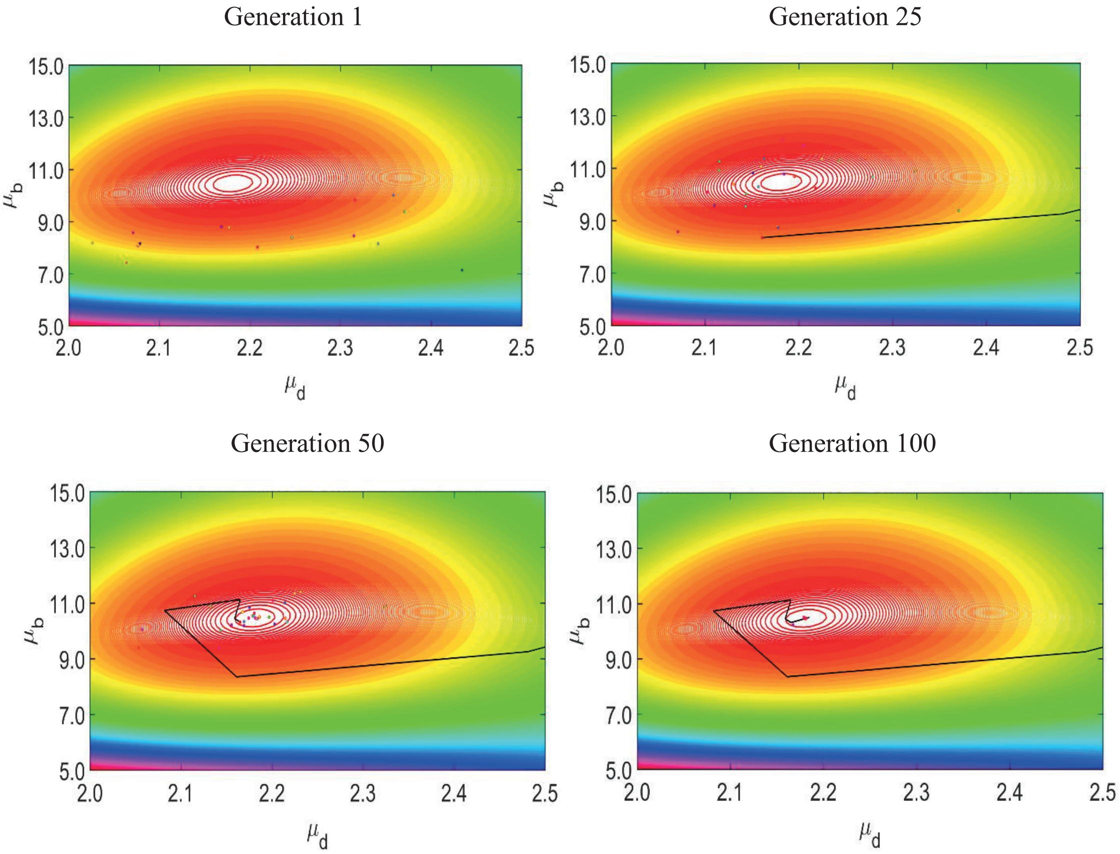

Figure 11 PSO Algorithm's Different Generations

Table 2 Optimal Values of μ_b and μ_d with Optimal Expected Cost Value TC^* using QN Method

$(\lambda, \xi, \alpha, \beta)$	**(3.8,0.3,0.2,3.0)**	**(4.0,0.3,0.2,3.0)**	**(4.2,0.3,0.2,3.0)**	**(4.0,0.4,0.2,3.0)**	**(4.0,0.5,0.2,3.0)**
(μ_d^0, μ_b^0)	(3,12)	(3,12)	(3,12)	(3,12)	(3,12)
Total Iterations	10	11	14	12	15
μ_d^*	2.062034	2.18090	2.300244	2.982308	3.808015
μ_b^*	10.642130	10.456297	10.278914	9.384290	8.486856
$TC(\mu_d^*, \mu_b^*)$	524.706586	526.28581	528.034607	540.489366	558.81171
$\frac{\partial TC}{\partial \mu_d}$	–0.957199	–0.707379	–0.695862	–5.273421	–9.221684
$\frac{\partial TC}{\partial \mu_b}$	–0.264034	–0.421598	–0.565635	–0.959683	–1.510192

Table 3 Optimal Values of μ_b and μ_d with Optimal Expected Cost Value TC^* using QN Method

$(\lambda, \xi, \alpha, \beta)$	**(4.0,0.3,0.1,3.0)**	**(4.0,0.3,0.15,3.0)**	**(4.0,0.3,0.1,4.0)**	**4.0,0.3,0.1,3.5)**	**(4.0,0.3,0.1,2.5)**
(μ_d^0, μ_b^0)	(3,12)	(3,12)	(3,12)	(3,12)	(3,12)
Total Iterations	14	12	11	13	14
μ_d^*	2.246358	2.205677	2.254851	2.251055	2.240368
μ_b^*	8.623722	9.647259	8.539841	8.576771	8.685731
$TC(\mu_d^*, \mu_b^*)$	497.738489	513.662107	495.710162	496.603728	499.234312
$\frac{\partial TC}{\partial \mu_d}$	2.743551	1.027515	2.787818	2.771251	2.695925
$\frac{\partial TC}{\partial \mu_b}$	–0.565639	–0.513366	–0.571368	–0.569079	–0.560221

Table 4 Optimal Values of μ_b^* and μ_d^* with Minimal Expected Cost TC^* using PSO Algorithm

$(K,T,\lambda,\xi,\alpha,\beta)$	μ_d^*	μ_b^*	$TC^*(\mu_b^*, \mu_d^*)$	**mean**$\left\{\frac{TC_i}{TC^*}\right\}$	**max**$\left\{\frac{TC_i}{TC^*}\right\}$	**CPU time**
(20, 10, 4.0, 0.3, 0.2, 3.0)	2.180950	10.456491	526.285810	1.0000000020	1.0000000046	294.88
(25, 10, 4.0, 0.3, 0.2, 3.0)	2.205506	10.035226	517.335965	1.0000000033	1.0000000093	475.34
(30, 10, 4.0, 0.3, 0.2, 3.0)	2.226941	9.698844	510.423085	1.0000000115	1.0000000332	438.36
(20, 8, 4.0, 0.3, 0.2, 3.0)	2.368139	7.890944	471.088229	1.0000000090	1.0000000021	290.35
(20, 10, 3.8, 0.3, 0.2, 3.0)	2.062034	10.642108	524.706586	1.0000000005	1.0000000014	288.39
(20, 10, 4.2, 0.3, 0.2, 3.0)	2.300265	10.278137	528.034607	1.0000000041	1.0000000117	287.64
(20, 10, 4.0, 0.4, 0.2, 3.0)	2.982303	9.384467	540.489366	1.0000000177	1.0000000486	287.64
(20, 10, 4.0, 0.5, 0.2, 3.0)	3.808010	8.486571	558.811711	1.0000000071	1.0000000209	288.38
(20, 10, 4.0, 0.3, 0.10, 3.0)	2.246466	8.622392	497.738492	1.0000000056	1.0000000147	287.85
(20, 10, 4.0, 0.3, 0.15, 3.0)	2.205811	9.646706	513.662109	1.0000000007	1.0000000014	287.74
(20, 10, 4.0, 0.3, 0.1, 2.5)	2.240297	8.685781	499.234313	1.0000000029	1.0000000072	287.47
(20, 10, 4.0, 0.3, 0.1, 3.5)	2.251017	8.578444	496.603730	1.0000000041	1.0000000069	287.44
(20, 10, 4.0, 0.3, 0.1, 4.0)	2.254923	8.538948	495.710164	1.0000000026	1.0000000056	287.26

tion (30) in Tables 1–5. We have incorporated the semi-classical optimizer: QN method and meta-heuristics like PSO and CS algorithm. Because the PSO algorithm does not involve the computation of gradients, it is an appropriate technique to calculate the optimum of single/multi-modal optimization problems. The advantage of the meta-heuristics like PSO and CS algorithms is that these can be employed to examine the optimal values of deci-

Table 5 Optimal Values of μ_b^* and μ_d^* with Minimal Expected Cost TC^* using CS Algorithm

$(K,T,\lambda,\xi,\alpha,\beta)$	μ_d^*	μ_b^*	$TC^*(\mu_b^*,\mu_d^*)$	mean$\left\{\frac{TC_i}{TC^*}\right\}$	max$\left\{\frac{TC_i}{TC^*}\right\}$	CPU time
(20, 10, 4.0, 0.3, 0.2, 3.0)	2.180949	10.456489	526.285810	1.0000000139	1.0000000251	320.71
(25, 10, 4.0, 0.3, 0.2, 3.0)	2.205506	10.035226	517.335965	1.0000000105	1.0000000263	512.84
(30, 10, 4.0, 0.3, 0.2, 3.0)	2.226941	9.698844	510.423086	1.0000000451	1.0000000659	649.73
(20, 8, 4.0, 0.3, 0.2, 3.0)	2.368204	7.890913	471.088229	1.0000000223	1.0000000247	329.51
(20, 10, 3.8, 0.3, 0.2, 3.0)	2.062093	10.642086	524.706586	1.0000000045	1.0000000074	338.13
(20, 10, 4.2, 0.3, 0.2, 3.0)	2.300295	10.278109	528.034608	1.0000000087	1.0000000135	350.62
(20, 10, 4.0, 0.4, 0.2, 3.0)	2.982291	9.384449	540.489367	1.0000000197	1.0000000502	309.67
(20, 10, 4.0, 0.5, 0.2, 3.0)	3.808075	8.486598	558.811711	1.0000000098	1.0000000179	310.88
(20, 10, 4.0, 0.3, 0.1, 3.0)	2.246501	8.622378	497.738493	1.0000000129	1.0000000213	325.11
(20, 10, 4.0, 0.3, 0.15, 3.0)	2.205852	9.646717	513.662111	1.0000000012	1.0000000025	314.54
(20, 10, 4.0, 0.3, 0.1, 2.5)	2.240315	8.685793	499.234313	1.0000000054	1.0000000096	299.38
(20, 10, 4.0, 0.3, 0.1, 3.5)	2.251009	8.578429	496.603730	1.0000000076	1.0000000104	305.15
(20, 10, 4.0, 0.3, 0.1, 4.0)	2.254927	8.538941	495.710164	1.0000000045	1.0000000062	307.69

sion variables whether discrete or continuous. The parametric values of the system components are taken as the same as in the previous simulation to demonstrate the converging results. The PSO algorithm pertinent parameters are fixed as $c_1 = 2$, $c_2 = 2$ and $\Omega = 0.5$. We conventionally fix the lower and upper bounds for μ_b and μ_d as [5.0 15.0] and [2.0 5.0] respectively, and obtained the optimal operating decision parameters in Table 4 up to the tenth place of decimal. The numerical results in Table 4 are depicted by considering 20 independent runs with 100 generations in each run and 50 particles generated randomly for each PSO simulation. For the validity purpose, we have also used the notion of statistical characteristics: mean-ratio and maximum-ratio of the optimal mean cost for all independent runs, to show the robustness of the proposed PSO algorithm.

For a better understanding of the research findings, a comparative study between the QN, PSO, and CS algorithm is accomplished for several combinations of system parameters in Tables 1-5. The computation time among all the iterations and optimal results are fundamental aspects for comparing the efficacy and effectiveness of an algorithm. So inspired by this, we have used both in each table with each combination of system parameters. It is observed that the calculated optimal values of design parameters and mean cost by the proposed algorithms QN, PSO, and CS algorithm are almost equivalent. The CPU time (in seconds) for the PSO algorithm is slightly less than the CS algorithm in each iteration. The associated mean cost enlisted by the PSO algorithm meets the optimality considerably and efficiently for all considered test instances. The results of the PSO algorithm are also superior to the QN method, for each numerical example. Newton's method involves the computation of gradient for calculating the Hessian. We obtain gradient numerically due to the high non-linearity and complexity of the optimization problem. It includes the high-scale estimation, which minimizes the efficacy of the algorithm.

While both the PSO algorithm and the CS method are nature-inspired optimization techniques, several reasons explain why the cuckoo search method often performs less efficiently than the PSO technique. Firstly, the cuckoo search approach could result in a slower rate of convergence and additional computational/processing costs due to its significant reliance on the random walk concept and the random selection of host nests. Additionally, the lack of social interaction between particles in the cuckoo search method, which allows particles to communicate information about their optimal placements and facilitates a more effective search of the search space, can be considered a disadvantage compared to particle swarm optimization. Lastly, PSO integrates velocity updates resulting from both personal and communal understanding, enabling particles to continuously adjust their trajectory, while the cuckoo search approach relies mostly on stochastic perturbations, thereby limiting its capacity to effectively converge to optimal solutions. Hence, from the above examples/numerical experiments and deliberations, it can be concluded that the PSO algorithm effectively provides optimal results compared to the CS algorithm and the quasi-Newton method. It is also noted that the optimum setup of system design parameters is essential to reduce the mean cost required in rendering service to potential customers.

9. Conclusion and Scope of the Future

The uniqueness of the current work is to observe the interplay among several queueing characteristics, viz customer impatience, threshold recovery policy, and partial server breakdown under the pressure condition, on the operational capability and performance of the service system. The Chapman-Kolmogorov differential-difference equations have been provided for modeling purposes by capturing the system dynamics. The steady-state probability distribution has been derived using the matrix method analytically. The analytical foundation enables to show the quality performance of the service system. Henceforth, numerous system performance indices have been provided. A pivotal aspect of the research involved formulating the mean cost function and the associated cost optimization problem, enabling economic analysis of the system. To address this, the nature-inspired optimizer, PSO, and CS algorithm have been used for the numerical illustration of cost analysis. Comparative analyses were conducted, not only against the semi-classical Quasi-Newton (QN) optimizer but also between the CS and PSO algorithms to depict the optimal operating combination ($i.e.$, optimal service rates μ_b^* and μ_d^*) with the minimal mean cost TC^* of the service system. The present study is mainly based on efficient resource utilization of real-life queueing-based service systems. This research provides essential theoretical and practical contributions to service systems that can be replicated in an organization with limited resources facing the challenge of queues. As a practical aspect, the insights derived from this study can help decision-makers take the necessary actions to reduce the overall cost of the service systems. Looking forward, this study serves as a stepping stone for future research endeavors. In addition to the substantial insights in this research, there are many other queueing notions, such as MRP, working vacation, general arrival/service pattern, etc., that present future research opportunities for academics, managers, and policymakers expanding our understanding of service system optimization and resource management.

Acknowledgments

Authors acknowledge the sincere thanks to anonymous reviewers to improve the quality of the paper and member of editorial board for considering our manuscript for possible publication.

Data Availability

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Conflicts of Interest

The authors declare no conflict of interest.

Funding

The third author (MD) extends his sincere thanks to the funding agency CSIR-UGC, India, for the financial support (SRF/NET 1081/(CSIR-UGC NET DEC. 2018)).

References

Abou El Ata M O, Shawky A I (1992). The single-server Markovian overflow queue with balking, reneging, and an additional server for longer queues. *Microelectronics Reliability* 32(10): 1389-1394.

Abou El Ata M O, Hariri A M A (1992). The $M/M/c/N$ queue with balking and reneging. *Computers & Operations Research* 19(8): 713-716.

Drekic S, Woolford D G (2005). A preemptive priority queue with balking. *European Journal of Operational Research* 164(2): 387-401.

Efrosinin D V, Semenova O V(2010). An $M/M/1$ system with an unreliable device and a threshold recovery policy. *Journal of Communications Technology and Electronics* 55(12): 1526-1531.

Haight F A (1957). Queueing with balking. *Biometrika* 44(3/4): 360-369.

Haight F A (1959). Queueing with reneging. *Metrika* 2(1): 186-197.

Hillier F S (2012). *Introduction to Operations Research. Tata McGraw-Hill Education*.

Kalidass K and Kasturi R (2012). A queue with working breakdowns. *Computers & Industrial Engineering* 63(4): 779-783.

Kennedy J, Eberhart R (1995). Particle swarm optimization. *Proceedings of ICNN'95- International Conference on Neural Networks*. Perth, WA, Australia, 27 November - 01 December, 1995.

Kleinrock L (1975). *Queueing Systems, 1*, Wiley Inter Science.

Krishnamoorthy, Pramod P K, Chakravarthy S R (2014). Queues with interruptions: A survey. *TOP* 22(1): 290-320.

Jain M, Bhagat A (2012). Finite population retrial queueing model with threshold recovery, geometric arrivals and impatient customers. *Journal of Information and Operations Management* 3(1): 162.

Li L, Wang J, Zhang F (2013). Equilibrium customer strategies in Markovian queues with partial breakdowns. *Computers & Industrial Engineering* 66(4): 751-757.

Lindfield G, Penny J (2017). *Introduction to Nature-Inspired Optimization*. Academic Press.

Liou D (2015). Markovian queue optimization analysis with an unreliable server subject to working breakdowns and impatient customers. *International Journal of Systems Science* 46(12): 2165-2182.

Liu Z, Song Y (2014). The $M^X/M/1$ queue with working breakdown. *RAIRO-Operations Research* 48(3): 399-413.

Lozano M, Moreno P (2008). A discrete-time single-server queue with balking: Economic applications. *Applied Economics* 40(6): 735-748.

Neuts M F (1981). *Matrix Geometric Solutions in Stochastic Models: An Algorithmic Approach*, Baltimore, MD, USA.

Rajadurai P (2018). Sensitivity analysis of an $M/G/1$ retrial queueing system with disaster under working vacations and working breakdowns. *RAIRO-Operations Research* 52(1): 35-54.

Shekhar C, Kumar N, Gupta A, Kumar A, Varshney, S (2020a). Warm-spare provisioning computing network with switching failure, common cause failure, vacation interruption, and synchronized reneging. *Reliability Engineering & System Safety* 199: 106910.

Shekhar C, Varshney S, Kumar, A (2020b). Optimal and sensitivity analysis of vacation queueing system with F-policy and vacation interruption *Arabian Journal for Science and Engineering* 45(8): 7091-7107.

Shekhar C, Varshney S, Kumar A (2021a). Matrix-geometric solution of multi-server queueing systems with Bernoulli scheduled modified vacation and retention of reneged customers: A meta-heuristic approach. *Quality Technology & Quantitative Management* 18(1): 39-66.

Shekhar C, Varshney S, Kumar A (2021b). Standbys provisioning in machine repair problem with unreliable service and vacation interruption. *The Handbook of Reliability, Maintenance, and System Safety through Mathematical Modeling.* Academic Press.

Shi Y, Eberhart R (1998). A modified particle swarm optimizer, *1998 IEEE International Conference on Evolutionary Computation Proceedings*, Anchorage, AK, USA, May 04-09, 1998.

Sridharan V and Jayashree P J (1996). Some characteristics on a finite queue with normal partial and total failures. *Microelectronics Reliability* 36(2): 265-267.

Sun W, Li S, Tian, N (2017). Equilibrium and optimal balking strategies of customers in unobservable queues with double adaptive working vacations *Quality Technology & Quantitative Management* 14(1): 94-113.

Wang K H, Lin Y H (2020). Profit analysis of a repairable system with imperfect coverage and service pressure coefficient. *IEEE Conference on Computational Science and Optimization* 127-131.

Wang K H, Liou C D, Lin Y H (2013). Comparative analysis of the machine repair problem with imperfect coverage and service pressure condition. *Applied Mathematical Modelling* 37(5): 2870-2880.

Yang D Y, Chiang Y C, Tsou C S (2013). Cost analysis of a finite capacity queue with server breakdowns and threshold-based recovery policy. *Journal of Manufacturing Systems* 32(1): 174-179.

Yang D Y, Chiang Y C (2014). An evolutionary algorithm for optimizing the machine repair problem under a threshold recovery policy. *Journal of the Chinese Institute of Engineers* 37(2): 224-231.

Yang X S (2020). *Nature-Inspired Optimization Algorithms*. Academic Press.

Yang Y, Chen Y H (2018). Computation and optimization of a working breakdown queue with second optional service. *Journal of Industrial and Production Engineering* 35(3): 181-188.

Yen T C, Wu C H, Wang K H, Lai W P (2022). Optimization analysis of the F-policy retrial machine repair problem with working breakdowns. *International Journal of Industrial and Systems Engineering* 40(2): 200-227.

Zhou W (2020). A modified BFGS type quasi-Newton method with line search for symmetric nonlinear equations problems. *Journal of Computational and Applied Mathematics* 367: 112454.

Shreekant Varshney is working as assistant professor in the Department of Mathematics, School of Technology (SoT) at PDEU (Formerly PDPU) since October 10, 2022. Prior joining to PDEU, he has worked for IFHE, Hyderabad. He has completed the Doctor of Philosophy (Ph.D.) from the Department of Mathematics, BITS Pilani, Pilani Campus. Also, he has cleared CSIR JRF/NET twice with AIR 18 and 67, respectively. In the year 2017, he was appointed as a co-instructor by the Practice School Division, BITS Pilani, Pilani Campus and mentored students at IIRS (ISRO), WIHG (DST), and CSIR-IIP in Dehradun. He has published more than 16 research articles in several journal of repute with high impact factor like RESS (Elsevier), QTQM (Taylor & Francis), JCAM (Elsevier), AJSE (Springer), etc. He has presented many research papers at national and international conferences of repute. Moreover, in October 2019, he has been awarded with second prize in a technical writing competition organized by SIAM journal publishing.

Suman Kaswan received her Bachelor of Science, B.Sc. (Hons.) in Mathematics, from the University of Delhi in 2017. In 2019, she received her M.Sc. in mathematics from the Indian Institute of Technology, Patna. Currently, she is working towards a Ph.D. degree from the Birla Institute of Technology and Science, Pilani. Her research interests lie primarily in the areas of development of queueing models and reliability theory incorporating several features like retrials, vacations, impatience, service regimes, arrival control policies, etc.; implementation of optimization techniques on the cost minimization problem of the system; a matrix-analytic method for solving stationary distributions; and probability generating functions techniques. She has published two research articles in peer-reviewed journal to her credit.

Mahendra Devanda is an assistant professor in the Department of Mathematics at MSBU, Bharatpur. He has submitted his Ph.D. thesis in the Department of Mathematics at the Birla Institute of Technology and Science, Pilani. He achieved an all-India rank of 122 in the CSIR-UGC NET exam (December 2018) and also qualified for the GATE exam in 2019. Dr. Devanda earned his Master of Science in mathematics from the University of Rajasthan in 2017. He has published four research articles in peer-reviewed journals and has submitted additional work to international journals. His research interests include the development of queuing models, stochastic models, and reliability theory.

Chandra Shekhar is the professor and ex-HoD in the Department of Mathematics in BITS Pilani, India, is actively involved in research and teaching in the area of queueing theory, computer and communication systems, machine repair problems, reliability and maintainability, stochastic process, evolutionary computation, statistical analysis, fuzzy set, and logic. Besides attending, presenting scientific papers, and delivering invited talks in national/international conferences and FDPs, he has organized a number of conferences, workshops, and symposiums as convener and organizing secretary. The best research paper award has been bestowed at the international conference. He has more than 50 research articles in these fields in journals of high repute and has supervised three Ph.D. theses. Besides some book chapters in an edited book published by the publisher of international reputation, authorship of the textbook entitled Differential Equations, Calculus of Variations and Special Functions and the edited book entitled Mathematical Modeling and Computation of Real-Time Problems: An Interdisciplinary Approach is also to his credit. He is also a member of the editorial board and reviewer of many reputed journals, academic societies and doctoral research committee, advisory board, faculty selection committee, the examination board of many governments and private universities, institutions, or research labs. As a professional, he has visited IIRS (ISRO), CSIR-IIP, NIH, WIHG, CPWD, Bank of Maharashtra, APS Lifetech.