Sigma J Eng Nat Sci, Vol. 44, No. 5, pp. 3231–3245, October, 2026

Sigma Journal of Engineering and Natural Sciences
Web page info: https://sigma.yildiz.edu.tr
DOI: 10.14744/sigma.2026.2156

Research Article

Matrix geometric method for Markovian queueing model with breakdown, backup server, working vacation and customer impatience

Indumathi P.[1], **Karthikeyan K.**[1,*]

[1]*Department of Mathematics, School of Advanced Sciences, Vellore Institute of Technology, Tamil Nadu, 632 014, India*

ARTICLE INFO

Article history
Received: 18 January 2025
Revised: 31 May 2025
Accepted: 21 July 2025

Keywords:
ANFIS; Balking; Cost Analysis; Heterogeneous Servers; Matrix Geometric Method; Reneging; Working Vacation

ABSTRACT

The analysis presented here considers a queueing system of Markovian nature that contains several important and complicated features like heterogeneous servers, impatience of customers, working vacation, retrial queues, server failures with repairs and a backup server. Such features are important in practice as many services use queueing systems of this type, for instance in medical and IT systems. Infinite size of the queue is assumed with random service rates. Necessary stability conditions are developed, steady-state probabilities are found using the matrix geometric approach and a cost function is introduced to obtain efficient tradeoff between efficiency and service performance. Data-driven predictions and conclusions are produced using the Adaptive Neuro Fuzzy Inference System. Results indicate that the increase in arrival rate λ from 0.5 to 1.1 (assuming that $\beta = 0.6$) increases the mean queue length from 1.0 to 1.45 customers, while increasing the service rate in working vacation from 0.5 to 2.0 decreases it from 4.0 to 2.1 customers, obtaining accuracies of 99% and 99.8%, correspondingly. The impact of other important parameters is investigated numerically. The originality of this approach is in inclusion of a number of real life features together with data driven predictions which makes the approach more realistic compared to other techniques available in the literature.

Cite this article as: Indumathi P, Karthikeyan K. Matrix geometric method for Markovian queueing model with breakdown, backup server, working vacation and customer impatiencen. Sigma J Eng Nat Sci 2026;44(5):3231–3245.

INTRODUCTION

The queueing systems, a ubiquitous experience, represent the act of queuing to satisfy some needs in various environments and scenarios faced daily. Queueing systems are extremely useful tools that have been applied in the manufacture industry, transport, communication networks models and operating systems. Queuing theory is going to help you utilize resources efficiently, decrease the time spent in the waiting lines and make your systems more productive. This branch of science helps to optimize processes, provide quality services and make good decisions in complex systems. Queuing theory deals with analysis of waiting lines and servicing, considering aspects of customers, computer tasks, or data waiting for servicing. The analysis

***Corresponding author.**
*E-mail address: k.karthikeyan@vit.ac.in

This paper was recommended for publication in revised form by Editor-in-Chief Ahmet Selim Dalkilic

Published by Yıldız Technical University Press, İstanbul, Turkey
© Author. This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

of the arrival rate of people or things, their service period, and queues size allows improving efficiency of functioning of systems.

In this paper, a novel queueing model has been developed. It reflects the process of additional servers' introduction in case of unexpected service interruptions. In the proposed model, an unlimited number of customers can enter the system and become impatient, which means they may balk, refusing to enter the system because of a large queue and renege leaving the system, having waited for some time. For example, consider the queue of people that want to have a vaccine. When they see a very long queue, they may decide not to wait and come again or to another place.

When a customer joins a queue and finds all the servers busy, they are sent to a waiting area called a retrial orbit. From there, they keep checking if a server becomes free. If a customer in this waiting area decides to leave without getting service because they expect to wait too long, this is called reneging. When all bank clerks are busy, a customer who leaves the bank queue due to a long wait is said to have reneged.

As a result, the consideration includes the use of two separate servers with varying service rates to serve incoming clients. Each of the servers in a real-time scenario is subject to interruptions; there might be service disruptions at some point due to server difficulties or system failures. Breakdowns can occur as a result of technical faults, maintenance requirement or any outside disturbance causing disruption in the smooth functioning of the system. Therefore, to cope with such adverse conditions, the immediate repair facility is required.

In the supermarket, there is a technological problem in one of the checkout counters and malfunction occurs. Such malfunction can occur as a result of a computer failure, a broken scanner or any other technical problem. The clients service is provided after the server is repaired. If there is a spike in the number of consumers, an additional server will be introduced until the situation is normalized. At this point, the introduction of the third server aims to provide the possibility of serving all the requests without loss. The goal of the introduction of the server is to avoid long lineups and decrease the waiting time in the system.

Server vacations may be idle due to no work, failure of the server, or diversion of work to some other server. Such an occurrence is seen in many applications like in priority queues, preventive maintenance work in production systems (such as testing and repairing of computers). Suppose there is a call center and has several lines to attend customer service. If a couple of lines gets an error or it is temporarily shutdown for maintenance work, then the system can operate but takes more time to service. Applying the aforementioned components, this research framework has been created as a Markovian queueing model.

Literature Survey

Gumbel [1] proved that there is no equivalent system in heterogeneous servers. Krishnamoorthi [2] managed heterogeneous servers on Poisson queues where the waiting time distribution and their means, considering multiple vacation scenarios were calculated. Kumar and Madheswari [3] employed the matrix geometric technique for this purpose. Ke et al. [4] prepared a concise note of the latest study on vacation queueing frameworks. Jain and Jain [5] conducted research on a single-server vacation queueing framework subjected under various cases of server breakdowns. The stationary queue length is attained within the M/M/2 queueing system that employs two heterogeneous servers. This study utilizes the matrix geometric method developed by Krishnamoorthy and Sreenivasan [6]. Yang and Wu [7] examined matrix-form expressions for the steady-state probability distribution of the system's customer count and performance metrics. Arivudainambi et al. [8] identified the general decomposition law using the supplementary variable technique for a sole server retrial queue with working vacation. Arivudainambi and Godhandaraman [9] examined a sole-server retrial queue with various features, determining job counts using supplementary variable techniques.

Bouchentouf and Messabihi [10] analyzed a queueing framework with dual servers, incorporating feedback, reverse balking, reneging, and the retention of reneged customers. A detailed investigation of the queueing model MAP/PH/1 was performed by Chakravarthy and Kulshrestha [11]. The model investigated server dynamics during vacations, external shocks, maintenance, and the support offered by backup servers. Gao and Zhang [12] examined the sojourn time distribution using the laplace stieltjes transform (LST) method and demonstrated the validity of little's law in the single server retrial queue, considering server vacation. Keerthika et al. [13] found that tandem queues with vacation policies reduced wait times and optimized resource allocation, improving logistics efficiency. Melikov et al. [14] explained the operation of queueing models in systems with diverse servers, specifically focusing on scenarios involving requests, including high and low priority cases. They are also analyzed matrix formulas for stationary probabilities in a diverse server queueing system. Taramin and Klimenok [15] studied a two-phase system with a Markovian batch arrival process and retrial calls. The system has two modes, each with a different retrial strategy.

For the M/M/N queueing system with identical servers and exponential service times, Semenova and Dudin's [16] research provided the expected duration between client departures and the estimated number of clients lost per unit time due to disasters. Wu and yang [17] applied the canonical particle swarm optimization approach to optimize an M/M/(c + 1) queueing framework with a dual-stage heterogeneous service scheme. They also developed a bi-objective optimization model to minimize both the expected cost and waiting time for consumers. Thakur et

al. [18] investigated the adaptive neuro-fuzzy inference system (ANFIS) and presented quantitative outcomes using both matrix-geometric and ANFIS approaches. Sharma et al. [19] paper talks about different waiting line models with impatient customers, first in order of time and then by grouping them based on impatience.

Upadhyaya and colleagues [20] used the particle swarm optimization (PSO) algorithm with the matrix method to find the best system cost. Majid et al. [21] conducted research using a Markovian queue regarding customers who dislike waiting. Their model considered work vacations that were implemented through Bernoulli distributions and generating functions approach was utilized for the model. Indumathi and Karthikeyan [22] addressed the problem of disasters and recovery of the M/M/2 queue. They focused on different server systems, customers abandoning service, and presence of customers in queue.

Vinitha et al. [23] applied recursion to find performance measures of the M/M/2 Markovian retrial queue model, since customers do not wish to wait and retrial queue systems with vacations and additional server are present. Mytalas et al. [24] studied Bernoulli feedback in multiple vacation policy of applying the batch arrivals queueing model. Kumar and Jain [25] introduced novel dual phase service process under hybrid vacation policy [26-28]. Khamehchi and Mahdiani [29–31] researched the optimization in gas lift systems based on constraints and algorithms. The methods they applied are really good in terms of allocation and allocation of resources. These are useful approaches supporting the optimization methods used in our model. ANFIS has been successful in modeling complex system from various fields. Oguanobi et al. [32] applied it to optimize Congo Red dye removal, while Ertunc [33] used it for accurate bearing fault classification based on vibration signals. This research is driven by the aim to help service providers optimize their operations and meet customer demands more effectively by adding extra servers. The control systems for networks, communications, and various complicated frameworks are enhanced through our methodology.

The Key Contributions

- It successfully conveys that the real-word validation of the proposed model was achieved via the design of a queueing situation, with the use of system parameters for clarity.
- The model description outlines the creation and transition diagram structure of our model.
- In the following sections, the matrix geometric solution is explained, and the stability criteria are outlined to enhance understanding of the model.
- The following section presents performance measures for the systems.
- In the next section, delve into the process of deriving the total cost accumulation across activities and cost elements.
- In the next section, an adaptive neuro-fuzzy inference system (ANFIS) was constructed to compare results with analytical findings, as clearly stated in the sensitivity analysis.
- The final section summarizes the outcomes of the proposed structure.

Application Areas of the Model

The model we're proposing can be used in many real-world situations, such as:

- **Healthcare:** It can help lower how long patients wait and make staff scheduling in hospitals better.
- **Cloud Services:** It helps websites and apps run smoothly, even if servers fail.
- **Call Centers:** It improves how quickly calls are answered and makes customers happier by managing resources well.
- **Factories:** It keeps production going without stops by considering machine repairs and maintenance.
- **Transport & Warehouses:** It helps goods and people move efficiently, even when there are delays.
- **ANFIS-Based Control:** It allows systems to be adjusted in real-time and perform better in changing service environments.

These examples prove that we have a model that can represent the idiosyncrasies of actual service systems. When combined with breakdowns, vacations, customer impatience and ANFIS control, our model can help keep the show running in all sorts of industries. We will explore this further in the next section.

Real-World Validation for the Proposed Model

This study looks at how the proposed model can be used in real life, especially in healthcare, within a typical queueing system.

- **Heterogeneous servers**: Different hospital departments, like emergency rooms and specialty clinics, work at different paces to give patients the best care.
- **Balking and Reneging Behavior**: If people with insurance have to wait too long or aren't satisfied, they might go to another hospital. This pushes healthcare providers to reduce wait times and improve the patient experience.
- **Working Vacation**: Hospitals plan staff breaks and shift changes, and do similar maintenance on equipment to keep things running smoothly.
- **Breakdown and Repair Dynamics**: When hospital equipment breaks, quick repairs and regular check-ups keep services going, making sure patients get treated on time.
- **Backup Server Integration**: Hospitals keep medical information safe by storing it in multiple places. Hospitals record information across servers in order to ensure the whereabouts of this information in the event that one server fails, another automatically takes over so that patient records remain available in emergencies.

Research Novelty

There are few attempts and models in the literature to study a mix of heterogenous servers, working vacations, customer impatience factors like reneging and balking, server breakdowns with repair, backup assistance. On top of that, quite a large number of literature do not consider all the above factors in a single model. Only a few models consider the same service rates and do not consider server unavailability, while dealing with the other above-mentioned factors. To fill this void, we have adeveloped a two server Markov retrial queueing system with probabilistic arrivals, flexible server allocation, incorporated repair process and the vacation mode. The general framework of matrix-geometric analysis along with rule-based system structure of an adaptive neuro-fuzzy inference system (ANFIS) is adopted to build up the system. The model is further enriched with cost optimization and backup server handling. It is designated for dependable use in communications, medicine and other similar domain services.

Model Description

Handling queues from the moment customers arrive, how servers are allocated, how it is served and when people take their holidays are essential to the smooth functioning of systems and achieving reliability. These, their interplay, equipment failures and selection of players, and the circumstances of retirement from the system, determine system performance and customer satisfaction. These ideas are elaborated below.

1. **Arrival Process:** In the queueing system, customers arrive following a Poisson process at a rate of λ. When they arrive, they can either opt to enter the queue, represented by a probability of β or decide not to join the queue (known as "balking") with probability of $\bar{\beta} = 1 - \beta$.
2. **Server Assignment Policy for Customer Queue Management:** When both servers 1 and 2 are free, a Bernoulli allocation model might be used. This means customers are sent to different tasks or people based on a series of random choices, and then they wait in an orbit. If server 1 is busy and server 2 is idle, arriving clients will choose server 2, and vice versa. When both server 1 and server 2 are busy, consumers will join a queue to retry the service from either server, depending on which of them completes the operation first.
3. **Service process:** Upon customer arrival, the service time adheres to an exponential distribution with rates μ_1 for the first Server-1, μ_2 for the second Server-2, and μ_B for the additional server, μ_v for service rate of server in working vacation. If any of the servers are unoccupied upon customer arrival, the customer receives service from an available server and exits the system.
4. **Retrial process:** Retrial times are exponentially distributed with a rate of γ. Customers in the orbit then try to get service one by one with a chance of α. If they don't get service, they give up with a probability of 1 - α.
5. **Breakdown and Repair rule:** The server providing service might break down, following a Poisson distribution with a rate of ζ. If it breaks down, it gets repaired immediately, which takes an exponential amount of time with a rate of θ.

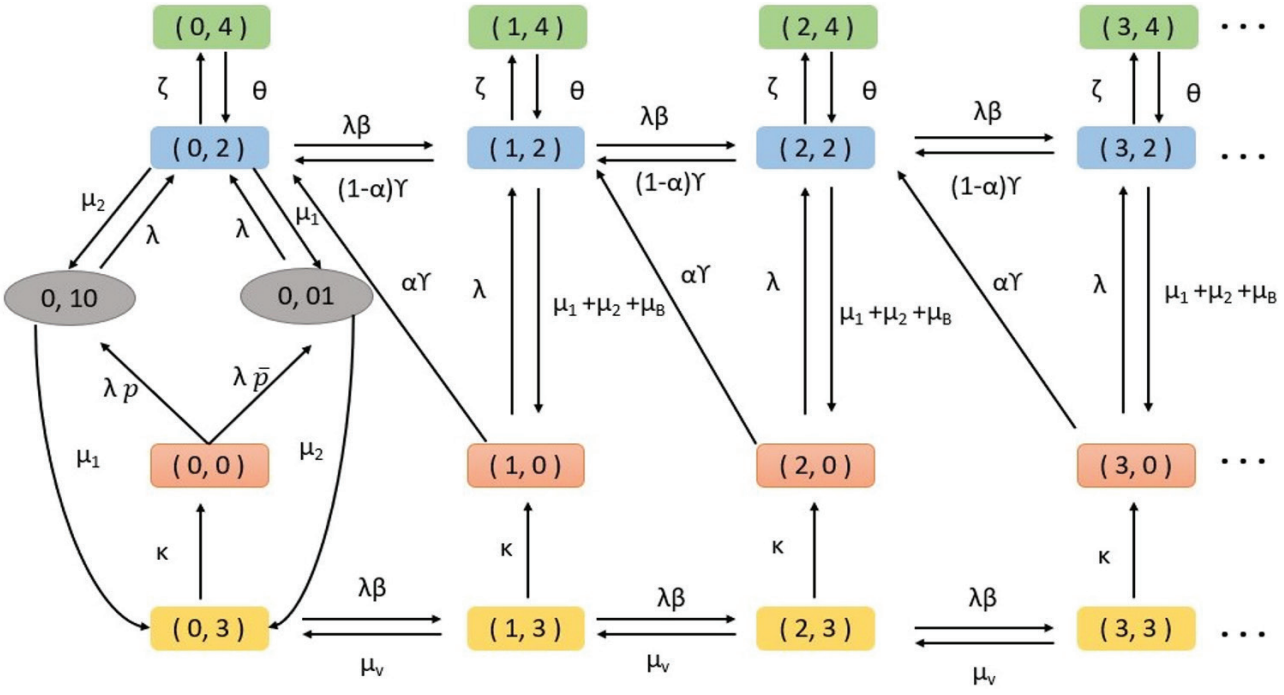


Figure 1. Transition diagram of the model.

6. **Vacation process:** When all consmers in the queue are served and the orbit is empty, the server takes a break of random length. After the break, it starts working again at a rate of $\boldsymbol{\kappa}$, ready to serve the next consumer. All the random processes within the system operate independently of each other. Figure 1 depicts the transition diagram of the proposed Markovian queueing model. Each state is denoted by (*i, j*), where *i* represents the number of customers and *j* indicates the server phase (idle, normal, vacation, or breakdown). Subsection 2.1 provides the notations used in the mathematical formulation of the transition diagram (Fig. 1).

Notations

- λ: Arrival rate
- μ_1: Service rate for server-1
- μ_2: Service rate for server-2
- μ_B: Service rate for an additional server
- μ_v: service rate for server in working vacation
- β: Customers joining the system
- $(1-\beta)$: Balking probability
- γ: Retrial rate
- α: Customers joining in the retrial orbit
- 1-α: Reneging probability
- κ: Retrival rate
- θ: Repair rate
- ζ: Break down rate

ANALYSIS OF THE MODEL

The model's mathematical expression is presented below, accompanied by a visual representation in the form of a transition diagram. At time t, let φ(t) represent the server's state, defined as follows:

$$\varphi(t)=\begin{cases}0, & \text{if the servers are idle,}\\ 1, & \text{if either one of the server is engaged,}\\ 2, & \text{during both the servers are engaged,}\\ 3, & \text{if either one of the servers is on working vacation,}\\ 4, & \text{if either one of the servers is on breakdown.}\end{cases}$$

Additionally, let ψ(t) denote the number of customers in the system and φ(t) represent the server state. The combined stochastic process

$$\{(\psi(t),\varphi(t)),t\geq 0\}$$

operates within the state space

$$\Theta(\mathrm{t})=\{(k,l)\colon k\geq 0,\ l=0,1,2,3,4\}.$$

Here the notation Θ represents the probability of having *k* consumers in the queue when the server is under state 1. The state space of the corresponding Markov process is arranged in lexicographical order as

$$\Omega=\{(0,0)\}\cup\{(k,l)\colon k\geq 1,\ l=0,1,2,3,4,\dots\}$$

To achieve accurate solutions for this problem, we utilize the matrix-geometric method outlined in the subsequent section. This method proves highly efficient in calculating steady-state probabilities, articularly in situations where the state space grows rapidly.

Matrix-Geometric Solution

In this model, the Markov chain is represented by the transition rate matrix Q, which is constructed in the form of a block tridiagonal matrix. It is used to derive the steady state probabilities for the above mentioned queueing structure system. The matrix Q is partitioned in to sub matrices.

$$Q=\begin{bmatrix}A_0 & A_1 & 0 & 0 & 0 & 0 & \cdots\\ A_2 & A_3 & A_4 & 0 & 0 & 0 & \cdots\\ 0 & B_1 & B_2 & B_3 & 0 & 0 & \cdots\\ 0 & 0 & B_1 & B_2 & B_3 & 0 & \cdots\\ 0 & 0 & 0 & B_1 & B_2 & B_3 & \cdots\\ 0 & 0 & 0 & 0 & B_1 & B_2 & \ddots\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots\end{bmatrix}$$

Where

$$A_0=\begin{bmatrix}-(\lambda P+\lambda\bar{P}+\kappa) & \lambda\beta\\ 0 & -(\lambda+\mu_1)\\ 0 & 0\end{bmatrix};$$

$$A_1=\begin{bmatrix}\lambda\bar{\beta} & 0 & \kappa & 0\\ 0 & \lambda & \mu_1 & 0\\ -(\lambda+\mu_2) & \lambda & \mu_2 & 0\end{bmatrix};\quad A_2=\begin{bmatrix}0 & \mu_2\\ \kappa & 0\\ 0 & 0\end{bmatrix};$$

$$A_3=\begin{bmatrix}\mu_1 & -(\mu_2+\zeta+\nu\alpha+\lambda\beta) & 0 & \zeta\\ 0 & 0 & -(\lambda\beta+\kappa) & 0\\ 0 & \theta & 0 & -\theta\end{bmatrix};$$

$$A_4=\begin{bmatrix}\nu\alpha & \lambda\beta & 0 & 0\\ 0 & 0 & \lambda\beta & 0\\ 0 & 0 & 0 & 0\end{bmatrix};$$

$$B_1=\begin{bmatrix}0 & \nu\alpha & 0 & 0\\ 0 & (1-\alpha)\nu & 0 & 0\\ 0 & 0 & \mu_v & 0\\ 0 & 0 & 0 & 0\end{bmatrix};$$

$$B_3=\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & \lambda\beta & 0 & 0\\ 0 & 0 & \lambda\beta & 0\\ 0 & 0 & 0 & 0\end{bmatrix};$$

$$B_2=\begin{bmatrix}-(\lambda+\nu\alpha) & \lambda & 0 & 0\\ \mu_1+\mu_2+\mu_B & -(\lambda\beta+\mu_1+\mu_2+\mu_B+\zeta+(1-\alpha)\nu) & 0 & \zeta\\ \kappa & 0 & -(\lambda\beta+\kappa+\mu_v) & 0\\ 0 & \theta & 0 & -\theta\end{bmatrix};$$

The steady-state probability vector Δ for Q is partitioned as $\Delta=(\Delta_0,\Delta_1,\Delta_2,\dots)$, where the sub-vectors $\Delta_0=\{\delta_{0,0}\}$for $k=0$, $\Delta_k=\{\delta_{k,0},\delta_{k,1},\delta_{k,2},\delta_{k,3},\delta_{k,4}\}$, $k\geq 1$ The static probability matrix Δ is obtained by solving the equation ΔQ =0.

$$\Delta_0A_0+\Delta_1A_2=0 \tag{1}$$

$$\Delta_0A_1+\Delta_1A_3+\Delta_2B_1=0 \tag{2}$$

$$\Delta_1A_4+\Delta_2B_2+\Delta_3B_1=0 \tag{3}$$

$$\Delta_2 B_3 + \Delta_3 B_2 + \Delta_4 B_1 = 0 \quad (4)$$

$$\Delta_3 B_3 + \Delta_4 B_2 + \Delta_5 B_1 = 0 \quad (5)$$

$$\Delta_k B_3 + \Delta_{k+1} B_2 + \Delta_{k+2} B_1 = 0 \quad (6)$$

$$\text{and } \Delta_k = \Delta_2 R^{k-2}, \text{ where } k \geq 3 \quad (7)$$

Let assume matrix R be rate matrix. From eqns. (1) to (6) using (7) we get

$$\Delta_0 A_0 + \Delta_1 A_2 = 0 \quad (8)$$

$$\Delta_0 A_1 + \Delta_1 A_3 + \Delta_2 B_1 = 0 \quad (9)$$

$$\Delta_1 A_4 + \Delta_2 [B_2 + R B_1] = 0 \quad (10)$$

$$\Delta_2 [B_3 + R B_2 + R^2 B_1] = 0 \quad (11)$$

$$\Delta_2 R [B_3 + R B_2 + R^2 B_1] = 0 \quad (12)$$

$$\Delta_2 R^{k-1} [B_3 + R B_2 + R^2 B_1] = 0 \;\; k \geq 3. \quad (13)$$

The normalizing eqn. is stated below,

$$\Delta_0 + \Delta_1 + \Delta_2 (I - R)^{-1} e = 1. \quad (14)$$

In this case, e is a column vector where every elements are 1's in the corresponding column. The matrix R is the quadratic matrix equation's least non-negative solution.

$$B_3 + R B_2 + R^2 B_1 = 0 \quad (15)$$

$$R = -B_2^{-1} (R^2 B_1 + B_3) \quad (16)$$

where in $R \geq 0$ and is an irreducible non-negative matrix with radius less than one. The matrix R can be estimated by subsequent substitution in the recurrence relationship,

$$R_0 = 0 \quad (17)$$

$$R_{m+1} = -B_2^{-1} (R_m^2 B_1 + B_3) \; m \geq 0 \quad (18)$$

Every value of R will rise monotonically, and R will eventually converge to $-B_2^{-1}(R_m^2 B_1 + B_3)$, which is non-negative.

Computation Algorithm for Matrix R and Probability Vectors

The computation algorithm for Matrix R and probability vectors is essential for determining steady-state probabilities and analyzing system behavior in stochastic models. These algorithms systematically solve for these probabilities, offering deep insights into the systems behaviour and overall reliability.

Algorithm 1. An algorithm to obtain the rate matrix

INPUT: B_1, B_1, B_3, e is a column vector of 1's and ϵ_0
OUTPUT: R
Step 1: Set $R_0 = 0$.
Step 2: while $|(B_3 + R_{(m)} B_2 + R_{(m)}^2 B_1)e| \geq \epsilon_0$ $m = 0,1,..$, perform steps 3 and 4.
Step 3: Set $R(m+1) = -B_3 B_2^{-1} - R_{(m)}^2 B_1 B_2^{-1}$, m = 0,1,...
Step 4: Continue **Step 3** until $R_{(m+1)}$ is close to $R_{(m)}$.
Step 5: set $R = R_{(m+1)}$
Step 6: OUTPUT

Algorithm. 2 An algorithm for steady-state probability vectors

INPUT : A_0, A_1, A_2, A_3, A_4, B_1, B_2, R, I e is a column vector of 1's
OUTPUT : Δ_0, Δ_1, Δ_2, Δ_3, ...

Step 1: $A \leftarrow [\Delta_0 \quad \Delta_1 \quad \Delta_2] \begin{bmatrix} A_0 & A_1 & 0 \\ A_2 & A_3 & A_4 \\ 0 & B_1 & B_2 + R * B_1 \end{bmatrix}$
Step 2: Solve A = 0
Step 3: $B \leftarrow \Delta_0 + \Delta_1 + \Delta_2 (I - R)^{-1} e$
Step 4: Solve B = 1
Step 5: Δ_0, Δ_1, $\Delta_2 \leftarrow$ Solve (A,B)
Step 6: Compute the steady-state probability vectors If $m \geq 3$ compute $\Delta_m = \Delta_2 R^{m-2}$
Step 7: OUTPUT

Stability Condition

The Neuts mean drift condition, as detailed by Neuts (1981) theorem, asserts that a Markov process Q exhibits stability if and only if...

$$\Delta B_3 e < \Delta B_1 e \quad (19)$$

The system achieves a steady state using the MGM technique. The row vector $\Delta = (\Delta_0, \Delta_1, \Delta_2, \ldots)$ is derived from the infinitesimal generator matrix $B = B_1 + B_2 + B_3$, which is presented below.

$$B = \begin{bmatrix} -(\lambda + \nu\alpha) & \lambda & 0 & 0 \\ \mu_1 + \mu_2 + \mu_B & -(\mu_1 + \mu_2 + \mu_B + \zeta) & 0 & \zeta \\ \kappa & 0 & -\kappa & 0 \\ 0 & \theta & 0 & -\theta \end{bmatrix} \quad (20)$$

Within a Markov process, the matrix B serves as the minimal generator and is irreducible. Furthermore, the matrix Δ satisfies a set of unique equations. A stationary probability $\Delta = [\Delta_0, \Delta_1, \Delta_2, \Delta_3]$ within B satisfies the equation

$$\Delta B = 0 \text{ and } \Delta e = 1 \quad (21)$$

where $e = [1, 1, 1, 1]^T$. According to Neuts [26], the condition essential and sufficient for system stability is as follows:

$$\Delta B_3 e < \Delta B_1 e \tag{22}$$

Solving eqns. (21) and (22), we get The expressions for Δ_0, Δ_1, Δ_2, and Δ_3 are given by:

$$\Delta_0 = (\mu_1\theta + \mu_2\theta + \mu_B\theta)/(\lambda\theta + \mu_1\theta + \mu_2\theta + \mu_B\theta + \lambda\zeta + \alpha\gamma\theta + \alpha\gamma\zeta) \tag{23}$$

$$\Delta_1 = (\lambda\theta + \alpha\gamma\theta)/(\lambda\theta + \mu_1\theta + \mu_2\theta + \mu_B\theta + \lambda\zeta + \alpha\gamma\theta + \alpha\gamma\zeta) \tag{24}$$

$$\Delta_2 = 0 \tag{25}$$

$$\Delta_3 = (\lambda\zeta + \alpha\gamma\zeta)/(\lambda\theta + \mu_1\theta + \mu_2\theta + \mu_B\theta + \lambda\zeta + \alpha\gamma\theta + \alpha\gamma\zeta) \tag{26}$$

Applying the stability condition, $\Delta B_3 e < \Delta B_1 e$, in the QBD procedure within matrix Q, results in:

$$(\beta\lambda(\lambda\theta + \alpha\gamma\theta))/(\lambda\theta + \mu_1\theta + \mu_2\theta + \mu_B\theta + \lambda\zeta + \alpha\gamma\theta + \alpha\gamma\zeta) < 1 \tag{27}$$

$$(\alpha\gamma(\mu_1\theta + \mu_2\theta + \mu_B\theta))(\lambda\theta + \mu_1\theta + \mu_2\theta + \mu_B\theta + \lambda\zeta + \alpha\gamma\theta + \alpha\gamma\zeta)$$

$$— (\gamma(\lambda\theta + \alpha\gamma\theta)(\alpha - 1))(\lambda\theta + \mu_1\theta + \mu_2\theta + \mu_B\theta + \lambda\zeta + \alpha\gamma\theta + \alpha\gamma\zeta)$$

Special Cases

Case (i): With no breakdown ($\zeta = 0$) and no repair time ($\theta = 0$), the MGM method yields a streamlined model that synthesizes earlier work on retrial, impatience, vacations, and an additional server, following the framework established by [23].

Case (ii): In the absence of breakdowns ($\zeta = 0$), repair time ($\theta = 0$), balking ($\beta = 0$), and reneging ($\alpha = 0$), and with the additional server inactive ($\mu_B = 0$), our model becomes a concise integration of earlier studies on an *M/M/2* queue with heterogeneous servers that incorporates a working vacation, as described by [6].

Performance Measures

Numerous performance metrics for the evaluated model under consideration are presented based on steady-sate probabilities

1. The probability of servers being in an idle state

$$P_{idle} = \delta_{0,0}$$

2. The probability of either server tuning busy

$$P_B{}^1 = \delta_{0,10} + \delta_{0,01}$$

3. The probability of both servers getting into a busy situation.

$$P_B{}^2 = \sum_{k=0}^{\infty} \delta_{k,2}$$

4. When the probability of any of the servers taking a working vacation occurs.

$$P_v = \sum_{k=0}^{\infty} \delta_{k,3}$$

5. In the event of the probability that either one of the servers experiances a breakdown

$$P_{Bd} = \sum_{k=0}^{\infty} \delta_{k,4}$$

6. The average number of customers within the system

$$L_s = \sum_{k=0}^{\infty} k\delta_{k,0} + \sum_{k=0}^{\infty} k\delta_{k,1} + \sum_{k=0}^{\infty} k\delta_{k,2} + \sum_{k=0}^{\infty} k\delta_{k,3} + \sum_{k=0}^{\infty} k\delta_{k,4}$$

7. The average number of customers within the queue

$$L_q = \sum_{k=0}^{\infty} (k-1)\delta_{k,0} + \sum_{k=0}^{\infty} (k-1)\delta_{k,1} + \sum_{k=0}^{\infty} (k-1)\delta_{k,2} + \sum_{k=0}^{\infty} (k-1)\delta_{k,3} + \sum_{k=0}^{\infty} (k-1)\delta_{k,4}$$

8. The avearge waiting time for consumers within the system

$$W_s = \frac{L_s}{\lambda}$$

9. The average waiting time for consumers within the queue

$$W_q = \frac{L_q}{\lambda}$$

Cost Analysis

This part manipulates the cost function to assure the model's economic sustainability. The study of industrial systems usually focuses on reducing total operating costs in order to increase efficiency and profitability. This part manipulates the cost function to assure the model's

economic sustainability. The goal is to develop a cost model that computes the entire cost per unit of time, designated as *TC*.

The current section examines the design of a Markovian queueing model, taking into consideration server service rates, working vacation, customer impatience, breakdowns, and backup servers. Let's define

$$TC = C_H * L_q + C_F * P_{idle} + C_E * P_B{}^1 + C_B * P_B{}^2 + C_V * P_v + C_{Bd} * P_{Bd} + C_1 * \mu_1 + C_2 * \mu_2 \quad (28)$$

In the equation, the variables are outlined as follows:

- C_H signifies the cost borne by each consumer for every unit of time spent within the queue.
- C_F signifies the cost incurred per unit of time for servers that are free.
- C_E indicates the cost per unit of time for either of the servers that is currently busy.
- C_B denotes the cost per unit of time for servers that are in a engaged state.
- C_V indicates the cost incurred for each consumer served while the server is in a working state.
- C_{Bd} signifies the cost when either one of the servers is experiencing a breakdown.
- C_1 indicates the cost incurred for serving each customer while server 1 is operational.
- C_2 reveals the cost of serving each customer when server 2 is operational.

For the given set of cost elements and other parameters, namely: C_H = \$10, C_F = \$150, C_E = \$200, C_B = \$300, C_V = \$100, C_{Bd} = \$250, C_1 = \$20, C_2 = \$25. $\lambda = 1.5$, $\mu_1 = 3$, $\mu_2 = 2.5$, $\mu_B = 1$, $\mu_v = 1.5$, $\theta = 2$, $\kappa = 0.2$, $\beta = 0.6$, $\zeta = 0.7$, $\gamma = 1$, $P = 0.8$, $\alpha = 0.7$.

The calculated total expected cost per unit of time is \$357.65. Furthermore, an analysis is executed

on the expected cost function's behavior in response to gradually changing cost parameter values. Table 1 clearly demonstrate the impact of (C_H, C_F), (C_V, C_{Bd}), (C_1, C_2) on the predicted cost function.

Sensitivity Analysis

In this context, numerical examples are presented to illustrate how different system configurations, impacting five distinct efficiency metrics λ, μ_1, μ_2, μ_B, μ_v affect the system performance measures. These effects are documented in Tables 2 to 6.

From our numerical study, every the values assigned to the system parameters adhere to the stability criterion indicated by equation (27). The curves in Figure 2 illustrate the performance measures against these parameters.

- Table 2: For taking λ values from 0.5 to 1.5, $\mu_1 = 3$, $\mu_2 = 2.5$, $\mu_B = 1$ and $\mu_v = 1.5$
- Table 3: For taking values λ = 1, $\mu_1 = 3$ to 5, $\mu_2 = 2.5$, $\mu_B = 1$, $\mu_v = 1.5$
- Table 4: For taking values $\mu_2 = 2.5$ to 4.5 and λ = 1, $\mu_1 = 6$, $\mu_B = 1$, $\mu_v = 1.5$
- Table 5: For taking values $\mu_B = 1$ to 3 and λ = 1.5, $\mu_1 = 5$ $\mu_2 = 4$, $\mu_v = 1.5$
- Table 6: For taking values $\mu_v = 0.5$ to 2.5 and λ = 1.5, $\mu_1 = 5$, $\mu_2 = 4$, $\mu_B = 3$

Table 2 presents numerical results, indicating that a rise in the arrival rate (λ) corresponds to an elevation in both system size and queue size it shows in 2D Figure 2(a) and also it is purely mentioned in 3D Figure 3(a). As depicted in Figure

Table 1. Effects of (CH, CF), (CV, CBd), and (C1, C2) on TC

(CH, CF)	**(10,150)**	**(10,200)**	**(10,250)**	**(20,150)**	**(30,150)**
TC	357.65	361.505	365.36	370.10	382.55
(CV, CBd)	(100,300)	(100,350)	(100,400)	(150,250)	(200,250)
TC	358.945	360.24	361.535	371.805	385.96
(C1, C2)	(20,30)	(20,35)	(20,40)	(25,30)	(30,30)
TC	370.15	382.65	395.15	385.15	400.15

Table 2. Effects of λ on Ls and Lq

λ	**μ_1**	**μ_2**	**μ_B**	**μ_v**	**Pidle**	**P1S**	**P2S**	**PV**	**PBd**	**Ls**	**Lq**
0.5	3	2.5	1	1.5	0.2295	0.2536	0.6072	0.1373	0.0019	0.9874	0.2169
0.7	3	2.5	1	1.5	0.1816	0.2354	0.5719	0.1884	0.0040	1.1688	0.3507
0.9	3	2.5	1	1.5	0.1464	0.2366	0.5247	0.2311	0.0074	1.3676	0.5142
1.1	3	2.5	1	1.5	0.1188	0.3787	0.4673	0.2313	0.0099	1.4510	0.4826
1.5	3	2.5	1	1.5	0.0771	0.3083	0.3823	0.2831	0.0259	2.1675	1.2450

Table 3. Effects of μ_1 on Ls and Lq

λ	μ_1	μ_2	μ_B	μ_v	Pidle	P1S	P2S	PV	PBd	Ls	Lq
1.0	3.0	2.5	1.0	1.5	0.1317	0.2426	0.4991	0.2482	0.0094	1.4766	0.6090
1.0	3.5	2.5	1.0	1.5	0.1303	0.2396	0.4999	0.2518	0.0086	1.4746	0.6050
1.0	4.0	2.5	1.0	1.5	0.1295	0.2371	0.5003	0.2543	0.0079	1.4694	0.5993
1.0	4.5	2.5	1.0	1.5	0.1288	0.2349	0.5007	0.2565	0.0072	1.4636	0.5931
1.0	5.0	2.5	1.0	1.5	0.1284	0.2331	0.5011	0.2585	0.0067	1.4608	0.5898

Table 4. Effects of μ_2 on Ls and Lq

λ	μ1	μ2	μB	μv	Pidle	P1S	P2S	PV	PBd	Ls	Lq
1.0	6.0	2.5	1.0	1.5	0.1252	0.2294	0.5055	0.2595	0.0056	1.4598	0.5850
1.0	6.0	3.0	1.0	1.5	0.1232	0.2283	0.5083	0.2580	0.0053	1.4603	0.5836
1.0	6.0	3.5	1.0	1.5	0.1220	0.2277	0.5102	0.2569	0.0049	1.4593	0.5816
1.0	6.0	4.0	1.0	1.5	0.1211	0.2268	0.5121	0.2563	0.0046	1.4589	0.5802
1.0	6.0	4.5	1.0	1.5	0.1204	0.2261	0.5135	0.2558	0.0044	1.4580	0.5786

Table 5. Effects of μ_B on Ls and Lq

λ	μ1	μ2	μB	μv	Pidle	P1S	P2S	PV	PBd	Ls	Lq
1.5	5.0	4.0	1.0	1.5	0.0726	0.2839	0.3841	0.3164	0.0150	2.0936	1.1668
1.5	5.0	4.0	1.5	1.5	0.0730	0.2833	0.3838	0.3181	0.0141	2.0820	1.1557
1.5	5.0	4.0	2.0	1.5	0.0735	0.2829	0.3835	0.3198	0.0135	2.0739	1.1477
1.5	5.0	4.0	2.5	1.5	0.0737	0.2824	0.3831	0.3211	0.0127	2.0656	1.1400
1.5	5.0	4.0	3.0	1.5	0.0741	0.2822	0.3829	0.3226	0.0121	2.0600	1.1343

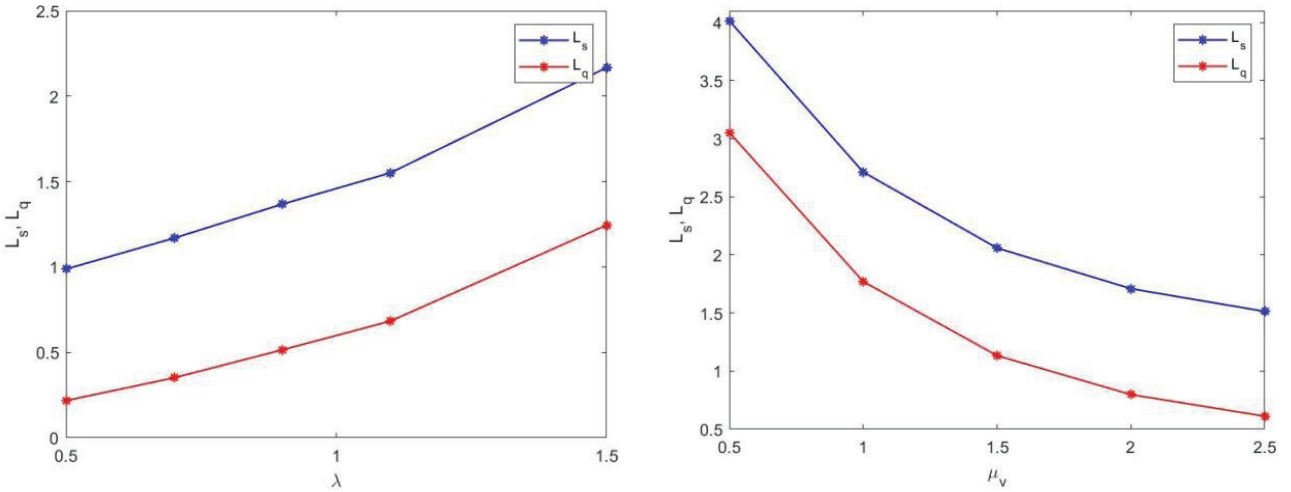
Table 6. Effects of μ_v on Ls and Lq

λ	μ_1	μ_2	μ_B	μ_v	Pidle	P1S	P2S	PV	PBd	Ls	Lq
1.5	5.0	4.0	3.0	0.5	0.0358	0.4443	0.1977	0.3308	0.0247	4.0135	3.0518
1.5	5.0	4.0	3.0	1.0	0.0550	0.3484	0.2833	0.3480	0.0174	2.7120	1.7699
1.5	5.0	4.0	3.0	1.5	0.0741	0.2822	0.3828	0.3225	0.0121	2.0581	1.1326
1.5	5.0	4.0	3.0	2.0	0.0883	0.2422	0.4645	0.2836	0.0092	1.7086	0.7974
1.5	5.0	4.0	3.0	2.5	0.0985	0.2183	0.5257	0.2485	0.0073	1.5126	0.6113

2(b), an increase in the service rates of servers during working vacation time (μv), declines the average no. of consumers in the system and as well as the queue which is also shown in Table 6. An increase in the service rates of the servers reduces the average number of consumers in the system and in the queue, as clearly illustrated in 3D Figures 3(b) to 3(e).

Numerical Illustration

A numerical experiment is performed to highlight the feasibility of the matrix geometric method. Numerical analysis serves as a method for substantiating the precision and dependability of queueing models. The implementation of MATLAB is employed to articulate the requisite code for this purpose, also this approach is used to calculate probabilities, rate matrices, and sub-matrix representations. Let consider a scenario with given parameters: $\lambda = 1.5$; $\mu_1 = 3$; $\mu_2 = 2.5$; $\mu_B = 1$; $\mu_v = 1.5$; $\kappa = 0.2$; $\beta = 0.6$; $\zeta = 0.7$; $\gamma = 1$; $\alpha = 0.7$; $\theta = 2$; $P = 0.8$.



(a) System and queue length with λ (b) System and queue length with μv

Figure 2. Two dimensional variation in system and queue length for different arrival rates (λ) and working vacation rates (μ_v).

$$A_0 = \begin{bmatrix} -0.3000 & 0.9000 \\ 0 & -4.5000 \\ 0 & 0 \end{bmatrix};$$

$$A_1 = \begin{bmatrix} 0.3000 & 0 & 0.2000 & 0 \\ 0 & 1.5000 & 3.0000 & 0 \\ -4.0000 & 1.5000 & 2.5000 & 0 \end{bmatrix};$$

$$A_2 = \begin{bmatrix} 0 & 2.5000 \\ 0.2000 & 0 \\ 0 & 0 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 3.000 & -7.8000 & 0 & 0.7000 \\ 0 & 0 & -1.1000 & 0 \\ 0 & 2.0000 & 0 & -2.0000 \end{bmatrix};$$

$$A_4 = \begin{bmatrix} 0.7000 & 0.9000 & 0 & 0 \\ 0 & 0 & 0.9000 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix};$$

$$B_1 = \begin{bmatrix} 0 & 0.7000 & 0 & 0 \\ 0 & 0.3000 & 0 & 0 \\ 0 & 0 & 1.5000 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix};$$

$$B_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.9000 & 0 & 0 \\ 0 & 0 & 0.9000 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix};$$

$$B_2 = \begin{bmatrix} -2.2000 & 1.5000 & 0 & 0 \\ 6.5000 & -8.4000 & 0 & 0.7000 \\ 0.2000 & 0 & -2.6000 & 0 \\ 0 & 2.0000 & 0 & -2.0000 \end{bmatrix};$$

$$e = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix};$$

By applying Algorithem 1, the rate matrix R is derived through the soliution of Equation (16)

$$R = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1.1228 & 0.3800 & 0 & 0.1330 \\ 0.2339 & 0.0645 & 0.4779 & 0.0226 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Utilizing the rate matrix R and equations (8), (9), (10) and (13), the probability vectors Δ_0, Δ_1, Δ_2 and Δ_3 were determined. The sum of all components, including Δ_0, Δ_1, Δ_2, and Δ_3, and so on, was calculated. In Table 7, the probabilities for each state are presented, confirming a total probability value of approximately one.

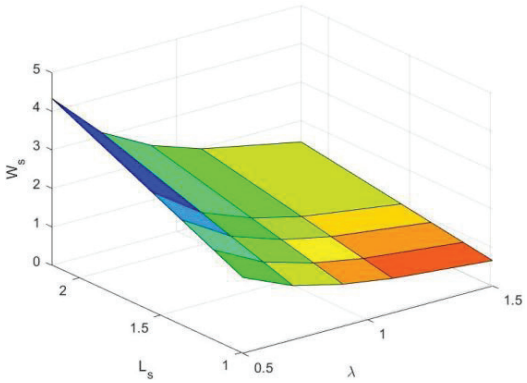
Adaptive Neuro-Fuzzy Inference System Computation

The Adaptive Neuro-Fuzzy Inference System (ANFIS) is a way to estimate analytical results using soft computing. It can handle different kinds of data and deal with the uncertainties found in real-world situations like waiting lines.

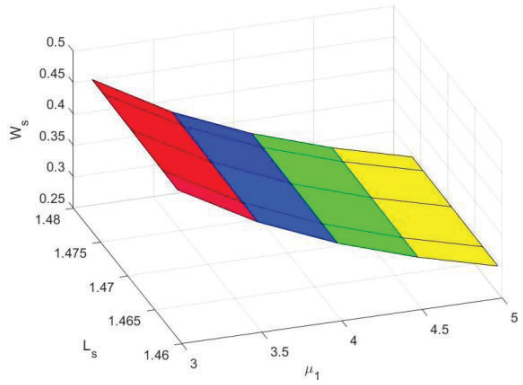
Neuro-fuzzy systems incorporate elements of artificial neural networks and fuzzy systems. The network component learns patterns from data and tunes up the fuzzy rules; in this way, it combines logical reasoning and learning. An example of a neuro-fuzzy system is ANFIS. Another characteristic of ANFIS is that it can model the nonlinear relationship between inputs and outputs.

Zadeh [27] at UC Berkeley in the early 1980s. It integrates neural network and fuzzy logic theories, and it can be applied to a lot of different fields, particularly control systems.

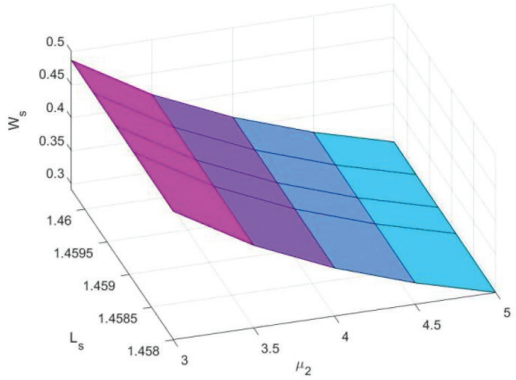
It enables to develop self-adapting controllers and self-predicting models using historical data. This combined fuzzy logic and neural network approach holds significant importance as it brings intelligence and prediction abilities to various systems across diverse domains. ANFIS offers a wide range of applications. This intelligent system is used



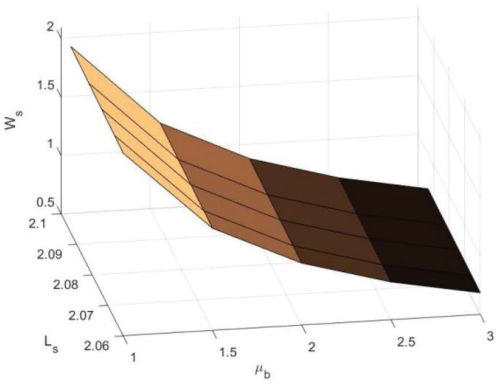
(a) λ with *Ls* and *Ws*



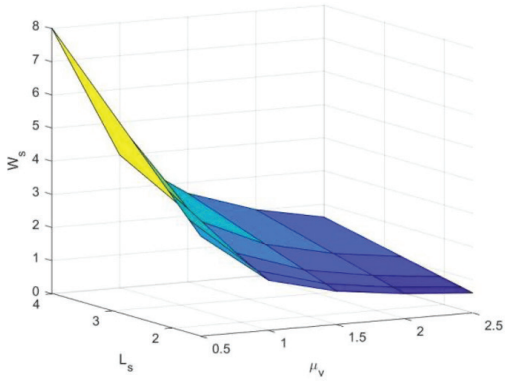
(b) $\mu 1$ with *Ls* and *Ws*



(c) $\mu 2$ with *Ls* and *Ws*



(d) μB with *Ls* and *Ws*



(e) μv with *Ls* and *Ws*

Figure 3. Impact of certain parameters on the 3D representation of performance measures.

Table 7. Steady state probability vector

Δk	δk,0	δk,1	δk,2	δk,3	Total
Δ0	0.0406	0.0189	0.0176		0.0771
Δ1	0.0194	0.2894	0.0068		0.3156
Δ2	0.1061	0.0296	0.1359	0.0104	0.2820
Δ3	0.0650	0.0200	0.0649	0.0070	0.1569
Δ4	0.0377	0.0118	0.0310	0.0041	0.0846
Δ5	0.0205	0.0065	0.0148	0.0023	0.0441
Δ6	0.0107	0.0034	0.0071	0.0012	0.0224
Δ7	0.0055	0.0018	0.0034	0.0006	0.0113
Δ8	0.0028	0.0009	0.0016	0.0003	0.0056
Total	0.3083	0.3823	0.2831	0.0259	0.9996

to develop intelligent robotic control systems and industrial control systems. It is used to predict market trends and optimize the financial systems. It also helps to monitor diseases for doctors by processing complex data for better understanding. Furthermore, it enhances engineering systems and enables outcome predictions.

The numerical calculations for various performance system parameters, of many analytical and numerical methods for predicting the system were observed to be time-consuming due to system complexity. Soft computing methods have proved to be helpful tools in such cases in recent years. The numerical results obtained are compared with the adaptive neuro-fuzzy inference system (ANFIS).

Behind the ANFIS model you will find the following structure comprising of five layers with each working in conjunction for the accurate approximation of the data and its corresponding input variable. The functioning of these five layers is elaborated as under:

- The nodes within the initial layer generate fuzzy sets based on the input variables. This layer, commonly referred to as the fuzzification layer, determines the degree to which a variable is associated with a particular fuzzy set.
- The second-layer nodes multiply input signals using the AND operation, and the output is the minimum value of the two inputs.
- The outputs of the second layer are normalized in the third layer.
- Each adaptive node in the fourth layer performs a particular purpose.
- The fifth layer generates the output by summing all of the input signals.
- The article [28] has provided a detailed explanation of the essential structures of the aforementioned concept, as illustrated in the above diagrammatic representation in Figure 4.

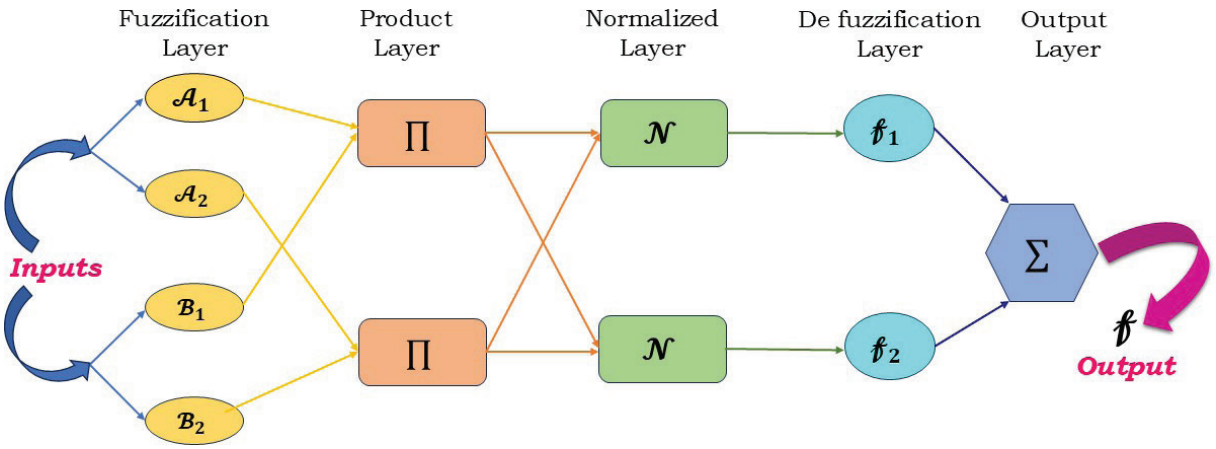
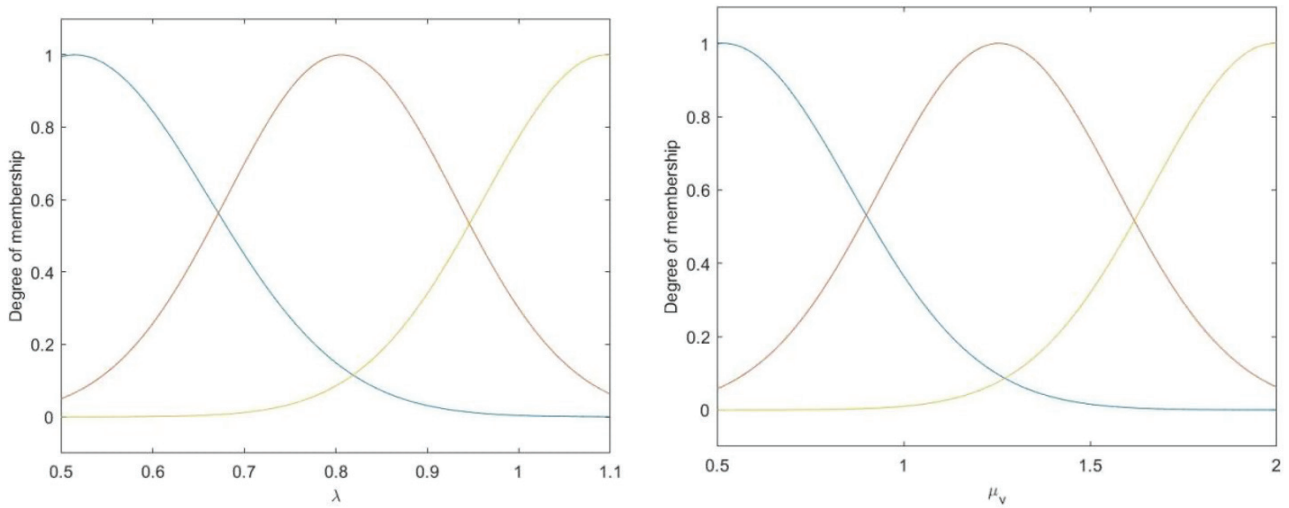


Figure 4. The schematic diagram of ANFIS model- Visualizing fuzzy ınference over sequential layers.



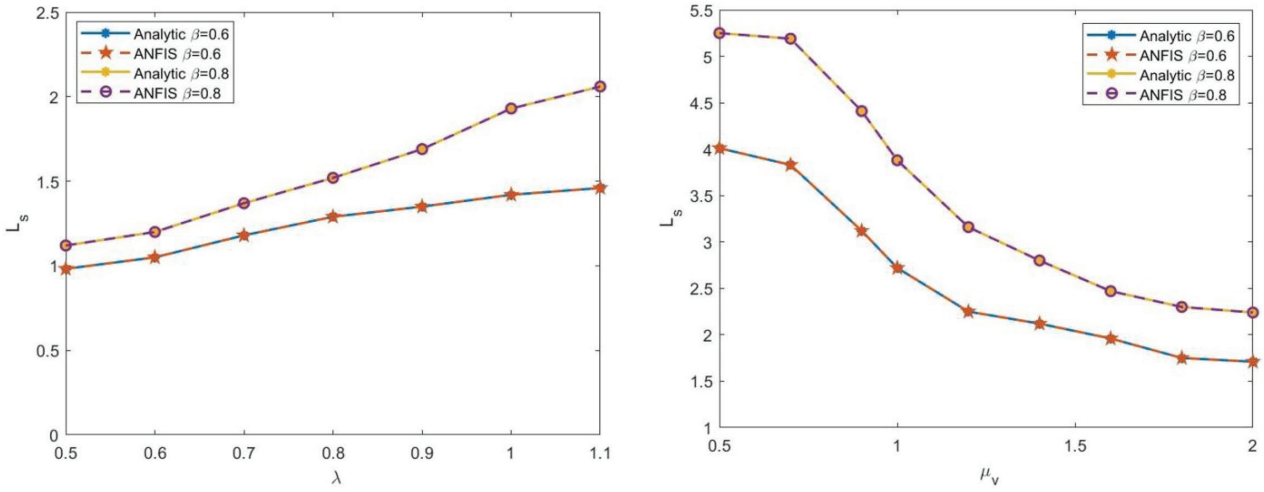
(a)Membership function for λ (b) Membership function for μv

Figure 5. Membership function for λ and μ_v.

Table 8, along with Figures 5(a) and 5(b), highlights the influence of the membership function on λ, μ_v. In Figures 6(a) and 6(b), the highlighted markings show the ANFIS results for the estimate, whereas the continuous lines indicate analytical findings. These visuals show tick marks almost completely overlapping the curved lines, indicating the success of both approaches.

Error Calculations in ANFIS

Determining the Absolute Percentage Error (APE) is done by comparing the accurate value (L_s), found through



a) λ with *Ls* b) μv with *Ls*

Figure 6. Comparison of analytic and ANFIS results.

Table 8. Values of the MF for the linguistics based on input parameters

Input Parameters	No. of Membership Function	Linguistic Values
β, λ	3	Low, Medium, High
β, μv	3	Low, Medium, High

Table 9. Values of the εr, Ls, and Ls* by Varying Input Parameters β, λ, and μv

β	λ	Ls	Ls*	εr	β	μv	Ls	Ls*	Er
0.6	0.5	0.9874	0.982	0.0054	**0.6**	0.5	4.0135	4.01	0.0008
	0.6	1.0510	1.05	0.0009		0.7	3.8345	3.83	0.0011
	0.7	1.1688	1.18	0.0095		0.9	3.1234	3.12	0.0010
	0.8	1.2932	1.29	0.0024		1.0	2.7120	2.72	0.0029
	0.9	1.3676	1.35	0.0128		1.2	2.2467	2.25	0.0014
	1.0	1.4156	1.42	0.0031		1.4	2.1157	2.12	0.0020
	1.1	1.4510	1.46	0.0062		1.6	1.7086	1.75	0.0242
0.8	0.5	1.1688	1.12	0.0417	**0.8**	0.5	5.2523	5.25	0.0004
	0.6	1.3676	1.20	0.1225		0.7	5.1856	5.19	0.0008
	0.7	1.4510	1.37	0.0558		0.9	4.3748	4.41	0.0080
	0.8	1.1204	1.52	0.3566		1.0	3.88411	3.88	0.0010
	0.9	1.3802	1.69	0.2244		1.5	2.6432	2.80	0.0593
	1.0	1.6830	1.93	0.1467		1.8	2.4674	2.47	0.1027
	1.1	2.0579	2.06	0.0010		2.0	2.2399	2.24	0.2062
Mean of εr				0.9896	**Mean of εr**				0.4125
Accuracy in Predicted Value				99.0103	**Accuracy in Predicted Value**				99.7937

analytical methods, with the estimated expected number of customers in the system (L_s^*) obtained using the Adaptive Neuro-Fuzzy Inference System (ANFIS) technique.

The absolute Percentage Error (ϵ_r) is determined by using the following formula

$$\epsilon_r = \frac{|L_s - L_s^*|}{L_s} 100\%$$

as calculated for the various values of λ, μ_v with different values of β. The results were recorded in Table 9, providing a summary of the APE (ϵ_r) and the accuracy of the estimated values. The accuracy is expressed as a percentage of the APE (ϵ_r).

There is a negligible difference between ANFIS results and analytical model results, as the two graphs exhibit almost perfect alignment showed in Figure 6(a), and 6(b).

CONCLUSION

The paper examines the *M/M/2* Markovian retrial queue model, taking into account client impatience, queue retrials, working vacations, breakdown and repair times, and an additional server. This study presented that how the interaction among these parts benefits the system efficiency, reliability, and quality of service. The performance measures were established following the analysis of system steady states. Numerical results were performed to demonstrate the effects of various system settings on the measures and their corresponding probabilities of customers in system and customers in queue. Cost analysis was proposed to facilitate the system. The validation of the results was achieved using an efficient predictive model for Matrix-Geometric Method in queueing systems, which showed good consistency. The novelty of this research consisted in combining all these intricate dynamics in one analytical-intelligent framework. In the future, this model can be extended by incorporating features such as phase type arrivals, Markovian arrival processes, multiple service options, customer priorities, batch arrivals or services, and control policies for system optimization.

AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

DATA AVAILABILITY STATEMENT

The authors confirm that the data that supports the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

CONFLICT OF INTEREST

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ETHICS

There are no ethical issues with the publication of this manuscript.

STATEMENT ON THE USE OF ARTIFICIAL INTELLIGENCE

Artificial intelligence was not used in the preparation of the article.

REFERENCES

[1] Gumbel H. Waiting lines with heterogeneous servers. Oper Res 1960;8:504–511. [CrossRef]

[2] Krishnamoorthi B. On Poisson queue with two heterogeneous servers. Oper Res 1963;11:321–330. [CrossRef]

[3] Kumar BK, Madheswari SP. An M/M/2 queueing system with heterogeneous servers and multiple vacations. Math Comput Model 2005;41:1415–1429. [CrossRef]

[4] Ke JC, Wu CH, Zhang ZG. Recent developments in vacation queueing models: a short survey. Int J Oper Res 2010;7:3–8.

[5] Jain M, Jain A. Working vacations queueing model with multiple types of server breakdowns. Appl Math Model 2010;34:1–13. [CrossRef]

[6] Krishnamoorthy A, Sreenivasan C. An M/M/2 queueing system with heterogeneous servers including one with working vacation. Int J Stoch Anal 2012;2012:145867. [CrossRef]

[7] Yang DY, Wu CH. Cost-minimization analysis of a working vacation queue with N-policy and server breakdowns. Comput Ind Eng 2015;82:151–158. [CrossRef]

[8] Arivudainambi D, Godhandaraman P, Rajadurai P. Performance analysis of a single server retrial queue with working vacation. Opsearch 2014;51:434–462. [CrossRef]

[9] Arivudainambi D, Godhandaraman P. Retrial queueing system with balking, optional service and vacation. Ann Oper Res 2015;229:67–84. [CrossRef]

[10] Bouchentouf AA, Messabihi A. Heterogeneous two-server queueing system with reverse balking and reneging. Opsearch 2018;55:251–267. [CrossRef]

[11] Chakravarthy SR, Kulshrestha R. A queueing model with server breakdowns, repairs, vacations, and backup server. Oper Res Perspect 2020;7:100131. [CrossRef]

[12] Gao S, Zhang D. Performance and sensitivity analysis of an M/G/1 queue with retrial customers due to server vacation. Ain Shams Eng J 2020;11:795–803. [CrossRef]

[13] Keerthika R, Niranjan S. Advancing logistics and resource management quality through two-stage tandem queues and single vacation techniques. Sigma J Eng Nat Sci 2025;43:505–512. [CrossRef]

[14] Melikov AZ, Ponomarenko LA, Mekhbaliyeva EV. Analyzing the models of systems with heterogeneous servers. Cybern Syst Anal 2020;56:89–99. [CrossRef]

[15] Taramin OS, Klimenok VI. Two-phase queuing system with a threshold strategy of retrial calls. Autom Control Comput Sci 2009;43:285–294. [CrossRef]

[16] Semenova OV, Dudin AN. M/M/N queueing system with controlled service mode and disaster. Autom Control Comput Sci 2007;41:350–357. [CrossRef]

[17] Wu CH, Yang DY. Bi-objective optimization of a queueing model with two-phase heterogeneous service. Comput Oper Res 2021;130:105230. [CrossRef]

[18] Thakur S, Jain A, Jain M. ANFIS and cost optimization for Markovian queue with operational vacation. Int J Math Eng Manag Sci 2021;6:894–910. [CrossRef]

[19] Sharma S, Kumar R, Soodan BS, Singh P. Queueing models with customers impatience: a survey. Int J Math Oper Res 2023;26:523–547. [CrossRef]

[20] Upadhyaya S, Agarwal D, Agarwal R, Ahmad I, Ibrahim AH, Al-Homidan S. Queueing and reliability analysis of unreliable multi-server retrial queue with Bernoulli feedback. Carpathian J Math 2023;39:745–758. [CrossRef]

[21] Majid S, Vijaya LP, Angelika BA, Aijaz M. Performance analysis of vacation interruption queue with balking and reneging. Res Sq [Preprint]. 2023. [CrossRef]

[22] Indumathi P, Karthikeyan K. ANFIS-enhanced M/M/2 queueing model investigation in heterogeneous server systems with catastrophe and restoration. Contemp Math 2024;5:2482–2502. [CrossRef]

[23] Vinitha G, Godhandaraman P, Poongothai V. Performance analysis of a Markovian model for two heterogeneous servers accompanied by retrial, impatience, vacation and additional server. Math Stat 2023;11:617–624. [CrossRef]

[24] Mytalas GC, Zazanis MA. Performance analysis for Bernoulli feedback queues subject to disasters: a system with batch Poisson arrivals under a multiple vacation policy. Qual Technol Quant Manag 2022;20:113–146. [CrossRef]

[25] Kumar A, Jain M. Cost optimization of an unreliable server queue with two stage service process under hybrid vacation policy. Math Comput Simul 2023;204:259–281. [CrossRef]

[26] Neuts MF. Matrix-geometric solutions in stochastic models. Johns Hopkins Series in the Mathematical Sciences. 1981;2.

[27] Zadeh LA. The theory and applications of fuzzy logic and fuzzy systems. Comput Surv 1981;13:563608.

[28] Stojčić M, Pamučar D, Mahmutagić E, Stević Ž. Development of an ANFIS model for the optimization of a queuing system in warehouses. Information 2018;9:240. [CrossRef]

[29] Khamehchi E, Mahdiani MR. Constraint optimization. In: Gas allocation optimization methods in artificial gas lift. 2017. p. 25–34. [CrossRef]

[30] Khamehchi E, Mahdiani MR. Optimization algorithms. In: Gas allocation optimization methods in artificial gas lift. 2017. p. 35–46. [CrossRef]

[31] Mahdiani MR, Khamehchi E. Using mapping for increasing the speed and quality of meta heuristic optimization (case study: petroleum engineering). In: 11th International Industrial Conference; 2015; Tehran.

[32] Oguanobi NC, Onu CE, Onukwuli OD, Anike EN, Ude CN. Modeling and kinetics investigation of adsorptive properties and regeneration of modified clay on azo dyes removal from aqueous solution using artificial intelligence (ANN, ANFIS) and RSM. Sigma J Eng Nat Sci 2025;43:316–339. [CrossRef]

[33] Ertunc HM. A combined decision algorithm for diagnosing bearing faults using artificial intelligent techniques. Sigma J Eng Nat Sci 2018;36:1235–1253.