RAIRO-Oper. Res. 58 (2024) 1257–1279
https://doi.org/10.1051/ro/2024044

RAIRO Operations Research
www.rairo-ro.org

PERFORMANCE ANALYSIS AND ANFIS COMPUTING OF A MARKOVIAN QUEUING MODEL WITH INTERMITTENTLY ACCESSIBLE SERVER UNDER A HYBRID VACATION POLICY

K. Divya and K. Indhira*

Abstract. In this study, we investigate a heterogeneous queueing model with intermittent server availability, server catastrophes, and a hybrid vacation policy. Our focus is on a specific scenario: server 1 is always available, while server 2 may experience breakdowns or vacations, making it intermittently accessible. Using the matrix-geometric approach (MGA), we derive matrix-based expressions for the stationary probability distribution of the number of customers in the system and various system performance measures. Additionally, we evaluate the cost function per unit of time to determine optimal values for the system's decision variables. Furthermore, we employ an adaptive neural fuzzy inference system (ANFIS) based on soft computing technology to compare and analyze the numerical results obtained. Through this comprehensive analysis, our study contributes to the understanding and optimization of this complex queueing system, attracting the attention of researchers in the field and offering practical insights for real-world applications.

Mathematics Subject Classification. 60K20, 60K25, 60K30, 60K35, 90B22, 03B52, 68M20.

Received February 3, 2023. Accepted February 13, 2024.

1. Introduction

The study of queueing models with server vacations is becoming more and more popular nowadays. Multi-server queues are a significant type of queuing system with a wide variety of practical applications. Banking, telephone companies, healthcare admission services, cafes and retail, telesales, transportation hubs and other industries use these systems. In general, homogeneous servers are the focus of current multi-server queueing technique research. That is, the individual service rates for each server are identical. This assumption is only achievable if the servicing mechanism is programmatically or electronically controlled. However, the aforesaid assumption about this queueing method with humanistic servers is just unreasonable. Servers often serve at varying rates while offering the same service. Numerous research was impacted as a result of the multi-server queueing method that had different servers. An examination of a heterogeneous queuing model with an intermittently available server and a hybrid vacation policy is carried out with the help of a matrix geometric technique.

Keywords. Heterogeneous, intermittently obtainable server, hybrid vacation, breakdown, disaster, MGA, ANFIS.

Department of Mathematics, School of Advanced Sciences, Vellore Institute of Technology, Vellore 632014, Tamil Nadu, India.
*Corresponding author: kindhira@vit.ac.in

The novelty of this paper stems from several distinct contributions. Firstly, it explores a hybrid vacation policy in a heterogeneous, unreliable server queueing model with intermittently accessible service. This approach combines operational vacation and working vacation, providing a unique perspective on managing service interruptions. Secondly, the paper employs a matrix-geometric approach to efficiently evaluate steady-state probabilities and performance indicators, offering more accurate insights compared to traditional methods. Thirdly, the integration of the Adaptive Network-Based Fuzzy Interference System (ANFIS), a soft computing technique, allows for the calculation of performance indices and the cost function while accommodating diverse industrial and technological constraints. Together, these contributions advance the understanding and practicality of queueing systems in dynamic environments.

The remaining sections of this paper are outlined as follows: Section 2 presents the formulation of the model and discusses the mathematical assumptions underlying it. In Section 3, the development of assumptions and governing equations is explained. The matrix-geometric approach, along with its implementation and stability conditions, is described in Section 4. Performance measurements, such as the expected number of customers in the system or queue based on state probabilities, are presented in Section 5. Furthermore, Section 6 discusses the overall cost associated with the system. Section 7 provides an overview of the preliminary architecture of the ANFIS and discusses training strategies. The computational findings are illustrated in Section 8 through numerical experiments. Sensitivity analysis, examining the impact of various factors on performance measures, is performed in Section 9. Finally, the conclusion and directions for future research are outlined in Section 10.

1.1. Survey of literature

There is a growing interest in the $M/M/2$ queueing model, in which server 1 is available all the time but server 2 is only rarely available. Service interruption occurs for an undetermined amount of time after the service under consideration is completed. When the queue length is greater than or equal to zero, server 2 goes to assist with a few extremely quick and unusual operations. This is referred to as an intermittently accessible service. Agarwal [1] first proposed the idea of an intermittently obtainable server. Sharda [32] studied a queuing issue using intermittently available servers that accept entries and exits in groups of variable sizes. Two heterogeneous server models, one of which is only available occasionally, were investigated by Seenivasan *et al.* [27] using the matrix geometric approach (MGA).

In recent years, heterogeneous server-based queuing systems have gained prominence. Here, each server in the system offers service at a distinct rate. Indra and Vijay Rajan [14] performed an investigation of the queuing behavior of a Markovian queue using a MGA. This model had heterogeneous servers and was subject to catastrophes. Seenivasan *et al.* [29] investigated a $M/M/2$ heterogeneous queueing system with unreliable servers, failures, and recovery. The balking Markovian queueing system with two heterogeneous servers was investigated by Singh and Vijendra [34].

In a practical situation, it is impossible for a server to operate properly without having a failure or system problem at some point. Bouchentouf *et al.* [8] investigated an unreliable multi-station machine model with a working vacation policy. Jain *et al.* [18] investigated a single-server F-policy Markov model with breakdown and retrial orbits. A Markovian working vacation queueing model with a dependent arrival rate and part breakdown was analyzed by Kalyanaraman and Sundaramoorthy [19]. The occurrence of catastrophes will result in the instantaneous destruction of all arrivals and the system's deactivation. Then it will take time for the system to adjust. Some studies investigate the restoration model and catastrophic events (see [2,23]). Wang *et al.* [37] work focus is on enhancing network performance and reducing energy consumption in mobile peer-to-peer (MP2P) networks by employing a vacation concept. The study derives expressions for system performance indicators using mathematical methods such as quasi-birth-and-death processes (QBD), matrix-geometric solutions.

Analysis of continuous-time Markov chains, including QBD processes, may be performed using the matrix geometric approach, which involves the use of transition rate matrices with recurrent block structures. It is useful for dealing with more complex Markovian queueing issues. Neuts [24] was the first to suggest the matrix-geometric technique in stochastic models. Aniyeri *et al.* [3] studied a multiphase queuing system with assorted servers by using matrix geometric method.

Fuzzy logic and neuro-computing form neuro-fuzzy systems, the most evident soft computing combo. Fuzzy logic rules are derived from observations using such systems. ANFIS [15] is a deep learning method that combines flexible techniques, artificial neural networks, and fuzzy inference systems. Thus, for truly nonlinear systems like queueing and energy systems, it should provide the most reliable estimates. A few works on ANFIS that addressed queueing issues have been published in the literature [18, 26, 31]. Ahuja *et al.* [2] suggested a single-server Markovian queue with working vacations and multiple-stage service, in which analytical results are compared with results from ANFIS computing. Deora *et al.* [12] investigated the cost analysis and ANFIS optimization of a machine repair model with a working vacation and feedback policy. Singh *et al.* [35] investigate a queuing system with server breakdowns, incorporating two customer types and their balking behavior. The study utilizes the ANFIS technique to evaluate simulation results, including fuzzy descriptors for realistic queuing scenarios. Jain *et al.* conducted a research study on the queueing analysis and strategic behavior of users in a cognitive radio system [16]. The study considered factors such as retrial orbits, vacation periods, and interruptions.

Queuing models with server vacations have received a lot of attention. The concept of server vacations has been used to a wide variety of real-world situations, including those involving computer and communication networks, industrial production, inventory management, and many more. Servi and Finn [30] suggested a vacation strategy for an $M/M/1$ queue in which the server goes on vacation whenever the system is empty. Bouchentouf *et al.* [6] investigated vacation queues with impatience and neglected customer retention. Jain *et al* [17] explored a single server working vacation queueing model with various server failures to determine the stationary queue length distribution. A queueing model with subject to breakdown and N-Policy vacations was studied by Renisagaya *et al.* [25]. Shekhar *et al.* [33] analyzed Bernoulli-scheduled modified vacation multi-server queueing systems. Wang *et al.* [37] examined a queueing system within the MP2P network that integrated fault repairable and spare servers, while also considering vacation periods. Pankaj Kumar *et al.* [21] introduced a novel double retrial orbit queueing model for fault-tolerant machining systems (FTMS) equipped with redundancy and repair capabilities. By incorporating a threshold policy and working vacation. Bouchentouf *et al.* [5] introduced a multi-server Markovian queueing model and incorporating feedback, multiple vacation policy, and customer impatience. Bouchentouf *et al.* [9] conducted a study on a variant of multiple vacations in a multi-server queue with customers' impatience and Bernoulli feedback. Bouchentouf *et al.* [7] examined a single server Markovian feedback queue with a variant of multiple vacation policy, balking, server's states-dependent reneging, and retention of reneged customers. Cherfaoui *et al.* [11] investigated a feedback queueing system with a variant of multiple vacation policy, balking, and server's states-dependent reneging. Afroun *et al.* [4] studied an impatient customers queuing system with multiple vacations, Bernoulli feedback, and the possibility of a server breakdown and repair.

We looked at vacation queueing models in a more comprehensive framework by introducing some extra moral characteristics since there is comparatively less background in the queueing literature on the combo (hybrid) of working vacation and complete vacation. Anshul Kumar *et al.* [20] research focused on a two-stage service procedure with a hybrid vacation policy. Thakur *et al.* [36] studied a $M/M/1/N$ single server finite capacity Markovian queueing model with operational vacation. Bourennane [10] investigated Markovian Bernoulli queues with operational server vacations and Bernoulli's weak and strong disasters.

There are so many studies of the matrix geometric method under various vacation policies. Nonetheless, no research has been conducted on the matrix geometric technique in a heterogeneous queueing system with a hybrid vacation schedule. Anshul [20] and Bourennane [10] research inspires us to build a heterogeneous Markovian model with an intermittently accessible server and a hybrid vacation policy that combines working vacation and operational vacation. When the queue is empty and the server takes an operational vacation (OV), this research investigates two kinds of disasters. In the first disaster, a strong Bernoulli disaster forces all customers to quit the system, putting the server on vacation. In a weak disaster, the server begins to repair quickly. If the queueing system does not have enough capacity to handle the incoming demand during the operational server vacation period or incorrect vacation schedule, it can result in disasters. Customer service continues during the server vacation after the repair. Repairs may help attract customers. After its operational vacation, the server stays in this state or stops and resumes regular service during busy periods.

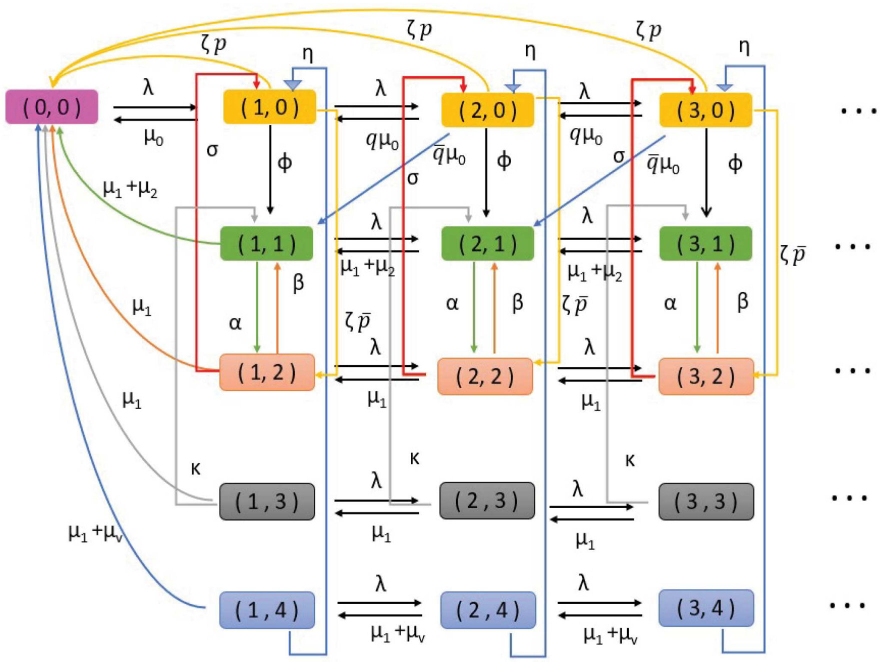


Figure 1. Transition diagram of the model.

2. Model formation and notations

Under the hybrid vacation policy, we propose an unreliable Markovian queueing model with an intermittently accessible server. Our model is explained in the following way:

Arrival process. A customer enters at a rate of λ according to a Poisson process.

Service process. The First-In-First-Out policy governs the service discipline. We consider the queueing model with two heterogeneous servers: server 1 is constantly available and entirely reliable, while server 2 is only occasionally accessible and unreliable. In the operating state, the rates for the services provided by servers 1 and 2 are μ_1 and μ_2, respectively, and follow an exponential distribution. When the line length seems to be higher than or equal to zero, server 2 starts doing certain uncommon and crucial tasks. Prior to that, server 2 must complete the task at hand. Server 2 retrieval capacity rate is κ, it follows an exponential distribution. It should be observed that no service is being offered while server 2 is in the repaired and intermittently accessible state.

Breakdown and repair rule. When the server 2 experiences high workload, it may lead to a potential breakdown caused by server 2's malfunction. In the event of a failure, server 2 is promptly taken out of service and sent for repairs. Once the necessary repairs are completed, server 2 resumes its operation in a working state to serve customers. Throughout the breakdown period, the server remains inactive and does not accept new customers. However, existing customers within the system are retained and patiently wait for service as soon as the server becomes available again after the repair process. The breakdown durations and repair times of the server follow exponential distribution with failure rate α and repair rate β, respectively.

Hybrid vacation process. Server 2 initiates exponential WV with a μ_v service rate during vacation. At the end of the WV period, server 2 must switch to OV with a probability η. While customers enter the system at OV, server 2 has an exponential distribution with a rate of μ_0 for serving customers, and sometimes a disaster occurs according to a Poisson process with a rate of ζ.

Disaster rule. When the system is not empty during the OV time, the strong disaster compels all present customers to quit with a probability of p; otherwise, the weak disaster forces all customers to remain in the system, and a repair period starts immediately with a probability of $\overline{p}$. The repair time follows an exponential distribution of rate σ. When the repair time is over, the customer in service resumes service during the OV period. If server 2 terminates a service while in an OV period, it will begin a busy period with the probability $\overline{q}$ if the queue is not empty; otherwise, server 2 will continue in the same period with the probability q. The duration of operational vacation rate is ϕ. All stochastic processes in the system are independent of one another. The structure of this model's transition diagram is depicted in the Figure 1. In the following table, we provide the notations used in the mathematical formulation.

λ	Arrival rate
μ_1	Server 1 service rate
μ_2	Server 2 service rate
μ_v	Service rate for server 2 during working vacation state
μ_0	Service rate for server 2 during operational vacation state
ϕ	Mean of operational vacation duration
κ	The retrieval capacity rate from intermittently obtainable state
η	The retrieval capacity rate from working vacation state
$\alpha(\beta)$	Mean breakdown (repair) rate
ζ	Disaster rate
$p(\overline{p})$	Probability of strong (weak) disaster
$q(\overline{q})$	After completing OV period server 2 starts a busy (same) period probabilities
$\Delta(t)$	Two-state Markov process at time t
$\mathcal{N}(t)$	Customer numbers at time t
$\mathcal{S}(t)$	Current status of server 2 at t
G	Infinitesimal generator matrix
$D_0, D_1,$ $E_0, E_1,$ V_0, V_1, V_2	Sub-matrices of G
Δ	Steady-state probability vector
P_{ij}	probability of i customers in the system while the server is in j state
Δ_i	The system's steady-state probability of having i customers
R	Rate matrix
V	Least generator matrix
e	Column vector of 1's

3. Steady state equations

For the mathematical analysis of the proposed model, a bi-variate continuous time Markov process $\Delta(t) = \{(\mathcal{N}(t), \mathcal{S}(t)); t \geq 0\}$ be a Markov process (MP) with the system state space at time t. we denote by $\mathcal{N}(t)$ represents the number of customers in the system, and $\mathcal{S}(t)$ the state of the server. The state space of MP is organised in lexicographical manner as follows.

$$\Omega = (0,0) \bigcup (1,0) \bigcup (i,j); \qquad i \geq 1, \quad j = 0,1,2,3,4$$

and the server's states are

$$\mathcal{S}(t) = \begin{cases} 0, & \text{if server 2 is in } \textit{Operational vacation} \\ 1, & \text{if server 2 is in } \textit{Working} \\ 2, & \text{if server 2 is in } \textit{Breakdown} \\ 3, & \text{if server 2 is in } \textit{Intermittently obtainable} \\ 4, & \text{if server 2 is in } \textit{Working vacation.} \end{cases}$$

We denote $\delta_{i,j}$ as the steady state probability of "i" number of customers in the system at jth state of the server $(j = 0, 1, 2, 3, 4)$.

By using QBD process, Governing equations can be formulated as follows:

– **Server is in idle state**

$$\lambda\delta_{0,0} = \mu_0\delta_{1,0} + (\mu_1 + \mu_2)\delta_{1,1} + \mu_1\delta_{1,2} + \mu_1\delta_{1,3} + (\mu_1 + \mu_v)\delta_{1,4} + \zeta p\sum_{n=1}^{\infty}\delta_{n,0}. \tag{1}$$

– **Server is in operational vacation state**

$$[\lambda + \mu_0 + \zeta + \phi]\delta_{1,0} = \lambda\delta_{0,0} + q\mu_0\delta_{2,0} + \sigma\delta_{1,2} + \eta\delta_{1,4} \tag{2}$$

$$[\lambda + \mu_0 + \zeta + \phi]\delta_{n,0} = \lambda\delta_{n-1,0} + q\mu_0\delta_{n+1,0} + \sigma\delta_{n,2} + \eta\delta_{n,4}, \quad n = 2, 3, 4, \ldots \tag{3}$$

– **Server is in busy state**

$$[\mu_1 + \mu_2 + \alpha + \lambda]\delta_{1,1} = \phi\delta_{1,0} + \beta\delta_{1,2} + \kappa\delta_{1,3} + \overline{q}\mu_0\delta_{2,0} + (\mu_1 + \mu_2)\delta_{2,1} \tag{4}$$

$$[\mu_1 + \mu_2 + \alpha + \lambda]\delta_{n,1} = \phi\delta_{n,0} + \beta\delta_{n,2} + \kappa\delta_{n,3} + \overline{q}\mu_0\delta_{n+1,0} \tag{5}$$

$$+ (\mu_1 + \mu_2)\delta_{n+1,1} + \lambda\delta_{n-1,1}, \quad n = 2, 3, 4, \ldots \tag{6}$$

– **Server is in repair state**

$$[\lambda + \mu_1 + \beta + \sigma]\delta_{1,2} = \zeta\overline{p}\delta_{1,0} + \alpha\delta_{1,1} + \mu_1\delta_{2,2} \tag{7}$$

$$[\lambda + \mu_1 + \beta + \sigma]\delta_{n,2} = \zeta\overline{p}\delta_{n,0} + \alpha\delta_{n,1} + \mu_1\delta_{n+1,2} + \lambda\delta_{n-1,2}, \quad n = 2, 3, 4\ldots \tag{8}$$

– **Server is in intermittently obtainable state**

$$[\mu_1 + \kappa + \lambda]\delta_{1,3} = \mu_1\delta_{2,3} \tag{9}$$

$$[\lambda + \mu_1 + \kappa]\delta_{n,3} = \mu_1\delta_{n+1,3} + \lambda\delta_{n-1,3}, \quad n = 2, 3, 4, \ldots \tag{10}$$

– **Server is in working vacation state**

$$[\mu_1 + \mu_v + \eta + \lambda]\delta_{1,4} = (\mu_1 + \mu_v)\delta_{2,4} \tag{11}$$

$$[\mu_1 + \mu_v + \eta + \lambda]\delta_{n,4} = (\mu_1 + \mu_v)\delta_{n+1,4} + \lambda\delta_{n-1,4}, \quad n = 2, 3, 4\ldots \tag{12}$$

Here, $\delta_{i,j}$ represents the status of i no.of customers in the jth state.

4. Matrix geometric method

The system state is symbolized by $\varpi(t)$ and $\tau(t)$. Let $\{(\varpi(t), (\tau(t)); t \geq 0\}$, with the state space organised in Infinitesimal generator matrix G is expressed as follows

$$G = \begin{bmatrix} D_0 & D_1 & 0 & 0 & 0 & 0 & \cdots \\ E_0 & V_1 & V_2 & 0 & 0 & 0 & \cdots \\ E_1 & V_0 & V_1 & V_2 & 0 & 0 & \cdots \\ E_1 & 0 & V_0 & V_1 & V_2 & 0 & \cdots \\ E_1 & 0 & 0 & V_0 & V_1 & V_2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

where

$$D_0 = \begin{bmatrix}-\lambda\end{bmatrix}; \quad D_1 = \begin{bmatrix}\lambda & 0 & 0 & 0 & 0\end{bmatrix}; \quad E_0 = \begin{bmatrix}\mu_0 + \zeta p \\ \mu_1 + \mu_2 \\ \mu_1 \\ \mu_1 \\ \mu_1 + \mu_v\end{bmatrix}; \quad E_1 = \begin{bmatrix}\zeta p \\ 0 \\ 0 \\ 0 \\ 0\end{bmatrix}$$

$$V_2 = \begin{bmatrix}\lambda & 0 & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 & 0 \\ 0 & 0 & \lambda & 0 & 0 \\ 0 & 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & 0 & \lambda\end{bmatrix}; \quad V_0 = \begin{bmatrix}q\mu_0 & \overline{q}\mu_0 & 0 & 0 & 0 \\ 0 & \mu_1 + \mu_2 & 0 & 0 & 0 \\ 0 & 0 & \mu_1 & 0 & 0 \\ 0 & 0 & 0 & \mu_1 & 0 \\ 0 & 0 & 0 & 0 & \mu_1 + \mu_v\end{bmatrix} \text{ and}$$

$$V_1 = \begin{bmatrix}-(\lambda + \mu_0 + \phi + \zeta) & \phi & \zeta\overline{p} & 0 & 0 \\ 0 & -(\lambda + \alpha + \mu_1 + \mu_2) & \alpha & 0 & 0 \\ \sigma & \beta & -(\lambda + \mu_1 + \sigma + \beta) & 0 & 0 \\ 0 & \kappa & 0 & -(\lambda + \mu_1 + \kappa) & 0 \\ \eta & 0 & 0 & 0 & -(\lambda + \mu_1 + \mu_v + \eta)\end{bmatrix}.$$

4.1. Matrix geometric solution

We define $P_{ij} = \{(\mathcal{N}(t) = i, \mathcal{S}(t) = j)\}$ where i indicates the total number of customers in the queue and j reflects the server state. Under the stability condition $\rho < 1$. The prob. vector is described as follows: $\Delta = [\Delta_0, \Delta_1, \Delta_2, \ldots]$, where $\Delta_0 = [\delta_{0,0}]$, $n = 0$; $\Delta_1 = [\delta_{1,0}\delta_{1,1}, \delta_{1,2}, \delta_{1,3}, \delta_{1,4}]$.

We generalize $\Delta_n = [\delta_{n,0}\delta_{n,1}, \delta_{n,2}, \delta_{n,3}, \delta_{n,4}]$ $n = 1, 2, 3, 4, \ldots$ Since the steady-state criterion is achieved, then the sub-prob.vectors Δ_i satisfies $\Delta G = 0$ as follows:

$$\Delta_0 D_0 + \Delta_1 E_0 + \Delta_2 E_1 + \Delta_3 E_1 + \Delta_4 E_1 + = \ldots = 0 \tag{13}$$

$$\Delta_0 D_1 + \Delta_1 V_1 + \Delta_2 V_0 + 0 + \ldots = 0 \tag{14}$$

$$\Delta_1 V_2 + \Delta_2 V_1 + \Delta_3 V_0 + 0 + \ldots = 0 \tag{15}$$

$$\Delta_2 V_2 + \Delta_3 V_1 + \Delta_4 V_0 + 0 + \ldots = 0 \tag{16}$$

$$\vdots$$

$$\Delta_i V_2 + \Delta_{i+1} V_1 + \Delta_{i+2} V_0 + \ldots = 0 \text{ where } i \geq 2 \tag{17}$$

$$\Delta_k = \Delta_1 R^{k-1}, \text{ where } k \geq 2. \tag{18}$$

Let the matrix R represents the rate matrix. By substituting equation (18) into the equations (13)–(17), we get

$$\Delta_0 D_0 + \Delta_1 \left[E_0 + R(I - R)^{-1} E_1\right] = 0 \tag{19}$$

$$\Delta_0 D_1 + \Delta_1 [V_1 + R V_0] = 0 \tag{20}$$

$$\Delta_1 \left[V_2 + R V_1 + R^2 V_0\right] = 0 \tag{21}$$

$$\Delta_1 R \left[V_2 + R V_1 + R^2 V_0\right] = 0 \tag{22}$$

$$\Delta_1 R^{n-1} \left[V_2 + R V_1 + R^2 V_0\right] = 0 \quad n \geq 2. \tag{23}$$

The normalizing equation is provided by

$$\left[\Delta_0 + \Delta_1 [I - R]^{-1}\right] e = 1. \tag{24}$$

Here e is a column vector in which all elements are 1's in the corresponding column. By using methodologies from Neuts [24] and Latouche and Ramaswami [22], we have estimated the rate matrix R.

Thus, the matrix quadratic equations have a minimum non-negative solution in R.

$$V_2 + RV_1 + R^2V_0 = 0 \tag{25}$$
$$R = -V_2V_1^{-1} - R^2V_0V_1^{-1} \tag{26}$$

where $R \geq 0$ and it's an irreducible non-negative matrix of spectral radius less than unit [24]. Matrix R can be calculated using an iterative approach as shown below.

$$R_0 = 0 \tag{27}$$
$$R_{k+1} = -V_2V_1^{-1} - R_k^2V_0V_1^{-1} \qquad k \geq 1. \tag{28}$$

All values of R will expand monotonically, and non-negative matrix R is converging to $-V_1^{-1}$ and $[V_2 + R^2V_0]$. The steady state is attained *via* the Matrix-Geometric Method (MGM).

4.2. Stability condition

Theorem 4.1. *The inequality* $\rho = \frac{(\lambda-\mu_1)((\alpha+\beta)\eta-\alpha\sigma)}{\beta\lambda\mu_2-\alpha\sigma\mu_v} < 1$ *is the necessary and sufficient condition for the system to be stable.*

Proof. Let us define the matrix $V = V_0 + V_1 + V_2$ given by

$$V = \begin{bmatrix} -(\phi+\zeta+\overline{q}\mu_0) & (\overline{q}\mu_0+\phi) & \zeta\overline{p} & 0 & 0 \\ 0 & -\alpha & \alpha & 0 & 0 \\ \sigma & \beta & -(\sigma+\beta) & 0 & 0 \\ 0 & \kappa & 0 & -\kappa & 0 \\ \eta & 0 & 0 & 0 & -\eta \end{bmatrix}. \tag{29}$$

There exists a stationary probability $\Delta = [\Delta_0, \Delta_1, \Delta_2, \Delta_3, \Delta_4]$ of F such that

$$\Delta V = 0; \quad \Delta e = 1 \tag{30}$$

where $e = [1,1,1,1,1]^T$. Using Theorem 3.1.1 of Netus [24], the necessary and sufficient condition for the stability of the system is as follows:

$$\Delta V_2 e < \Delta V_0 e. \tag{31}$$

Solving equations (29) and (30), we get

$$\lambda[\Delta_0 + \Delta_1 + \Delta_2 + \Delta_3 + \Delta_4] < \mu_0\Delta_0 + (\mu_1+\mu_2)\Delta_1 + \mu_1(\Delta_2+\Delta_3) + (\mu_1+\mu_v)\Delta_4 \tag{32}$$
$$\frac{(\lambda-\mu_1)((\alpha+\beta)\eta-\alpha\sigma)}{\beta\lambda\mu_2-\alpha\sigma\mu_v} < 1 \tag{33}$$

where

$$\Delta_0 = 0, \tag{34}$$
$$\Delta_1 = \frac{\beta\eta}{\alpha\eta+\beta\eta-\alpha\sigma}, \tag{35}$$
$$\Delta_2 = \frac{\alpha\eta}{\alpha\eta+\beta\eta-\alpha\sigma}, \tag{36}$$
$$\Delta_3 = 0, \tag{37}$$
$$\Delta_4 = \frac{-\alpha\sigma}{\alpha\eta+\beta\eta-\alpha\sigma}. \tag{38}$$

□

4.3. Methodology comparison: Highlighting the matrix geometric method

In this study, a comprehensive comparison of methods for solving Markovian queueing models is presented. The table provided below illustrates the advantages of each method, with a specific focus on the MGM. The MGM stands out due to its computational efficiency, enhanced understanding of system behavior, sensitivity analysis capabilities, and the ability to facilitate comparisons and trade-offs. Furthermore, it offers scalability and flexibility, making it a preferred choice for solving Markovian queueing models. Other methods, such as Difference differential equation method, Discrete-Time Markov Chain (DTMC) Analysis, Simulation, and Exact Solution Techniques, demonstrate strengths in certain aspects but do not match the comprehensive advantages offered by the MGM.

5. Performance indices

Utilizing performance estimations may be of benefit in the process of designing, developing, implementing, and adjusting a queueing system that is more realistic. Using the probability from the previous part, we construct numerous performance indices.

5.1. System state probabilities

By calculating the probabilities of each system state, we can gain insights into the server's utilization and availability. This information helps us understand how often the system is in different states, such as working, on vacation, or busy in each phase. It is an essential performance measure for evaluating system reliability and efficiency.

- P [servers are in idle state]

$$P_S^{\text{Idle}} = \delta_{0,0}.$$

- P [server 2 being on a operational vacation state]

$$P_{S2}^{\text{OV}} = \sum_{j=1}^{\infty} \delta_{j,0}.$$

- P [server 2 being on a working state]

$$P_{S2}^{W} = \sum_{j=1}^{\infty} \delta_{j,1}.$$

- P [server 2 being on a breakdown state]

$$P_{S2}^{\text{BD}} = \sum_{j=1}^{\infty} \delta_{j,2}.$$

- P [server 2 being on a intermittently obtainable state]

$$P_{S2}^{\text{IO}} = \sum_{j=1}^{\infty} \delta_{j,3}.$$

- P [server 2 being on a working vacation state]

$$P_{S2}^{\text{WV}} = \sum_{j=1}^{\infty} \delta_{j,4}.$$

5.2. Expected numbers of customers in the system and queue

The expected number of customers in the system and queue helps assess capacity and identify congestion issues. Understanding the number of customers waiting enables resource allocation improvements and enhances customer satisfaction. Waiting time evaluation reflects system responsiveness and overall service quality, directly impacting customer experience, queue management, and resource planning. These performance measures provide insights for optimizing operational efficiency and delivering excellent customer service.

– Mean system length when the server 2 being on a working state:

$$E(S)_W = \sum_{j=1}^{\infty} j\delta_{j,1}.$$

– Mean system length when the server 2 being on a breakdown state:

$$E(S)_{\mathrm{BD}} = \sum_{j=1}^{\infty} j\delta_{j,2}.$$

– Mean system length when the server 2 being on a intermittently obtainable state:

$$E(S)_{\mathrm{IO}} = \sum_{j=1}^{\infty} j\delta_{j,3}.$$

– Mean system length when the server 2 being on a vacation state:

$$E(S)_V = E(S)_{\mathrm{OV}} + E(S)_{\mathrm{WV}} = \sum_{j=1}^{\infty} j\delta_{j,0} + \sum_{j=1}^{\infty} j\delta_{j,4}.$$

– Mean queue length when the server 2 being on a working state:

$$E(Q)_W = \sum_{j=1}^{\infty} (j-1)\delta_{j,1}.$$

– Mean queue length when the server 2 being on a breakdown state:

$$E(Q)_{\mathrm{BD}} = \sum_{j=1}^{\infty} (j-1)\delta_{j,2}.$$

– Mean queue length when the server 2 being on a intermittently obtainable state:

$$E(Q)_{\mathrm{IO}} = \sum_{j=1}^{\infty} (j-1)\delta_{j,3}.$$

– Mean queue length when the server 2 being on a vacation state:

$$E(Q)_V = E(Q)_{\mathrm{OV}} + E(Q)_{\mathrm{WV}} = \sum_{j=1}^{\infty} (j-1)\delta_{j,0} + \sum_{j=1}^{\infty} (j-1)\delta_{j,4}.$$

– The expected system length is:

$$L_s = E(S)_{\mathrm{OV}} + E(S)_W + E(S)_{\mathrm{BD}} + E(S)_{\mathrm{IO}} + E(S)_{\mathrm{WV}}.$$

– The expected queue length is:

$$L_q = E(Q)_{\mathrm{OV}} + E(Q)_W + E(Q)_{\mathrm{BD}} + E(Q)_{\mathrm{IO}} + E(Q)_{\mathrm{WV}}.$$

– The expected waiting time in the system:

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

– The expected waiting time in the queue:

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

6. Cost model

Creating a cost function in terms of time units allows us to create an optimum control strategy for the model. As a result, we are aiming to reduce costs as much as possible. Assumptions are made about the following components of certain activities' costs:

$$\begin{aligned} C_H &= \text{Holding cost per unit time.} \\ C_W &= \text{Cost per unit time in the server's working state.} \\ C_V &= \text{Cost per unit time in the server's vacation state.} \\ C_{\mathrm{IO}} &= \text{Cost per unit time in the intermittently obtainable state.} \\ C_{\mathrm{BD}} &= \text{Cost spent per customer while a server is being repaired.} \\ C_1 &= \text{Cost per customer served in server 1.} \\ C_2 &= \text{Cost per customer served in server 2.} \end{aligned}$$

The total cost (T_{cost}) is defined as:

$$T_{\mathrm{cost}} = C_H L_s + C_W P_{S2}^{W} + C_V \left(P_{S2}^{\mathrm{WV}} + P_{S2}^{\mathrm{OV}}\right) + C_{\mathrm{BD}} P_{S2}^{\mathrm{BD}} + C_{\mathrm{IO}} P_{S2}^{\mathrm{IO}} + C_1\mu_1 + C_2\mu_2.$$

In our calculations, $C_H = \$85$; $C_W = \$200$; $C_v = \$70$; $C_V = \$100$; $C_{\mathrm{BD}} = \$100$; $C_{\mathrm{IO}} = \$50$; $C_1 = \$10$; $C_2 = \$20$, which we use to determine total costs. All calculations have been rounded to four places after the decimal. The total expected cost per unit of time or T_{cost} is found to be **\$154.4780** for the following values of the other parameters, such as: $\lambda = 0.3, \mu_1 = 2, \mu_2 = 2.5, \mu_0 = 1, \mu_v = 1.5, \phi = 1.6, \zeta = 0.5, \alpha = 2.5, \beta = 4, \sigma = 0.5, \eta = 1.2, \kappa = 0.5, p = 0.7, \overline{p} = 0.3, q = 0.6, \overline{q} = 0.4$. Furthermore, we scrutinize how the estimated cost function responds when the cost parameters are changed. The impacts of (C_H, C_W), (C_V, C_{BD}), and (C_1, C_2) on the predicted cost function are depicted in Tables 1, 2 and 3 respectively. As can be observed, the estimated cost function exhibits a linearly increasing pattern of rising cost parameters.

TABLE 1. Influence of (C_H, C_W) on the expected cost function T_{cost} with $C_V = \$70$, $C_{\mathrm{BD}} = \$100$, $C_{\mathrm{IO}} = \$50$, $C_1 = \$10$ and $C_2 = \$20$.

(C_H, C_W)	$(85, 100)$	$(85, 150)$	$(85, 200)$	$(90, 100)$	$(95, 100)$
T_{cost}	154.4780	156.1080	157.7380	155.2460	156.0140

TABLE 2. Influence of (C_V, C_{BD}) on the expected cost function T_{cost} with $C_H = \$85$, $C_W = \$300$, $C_{\mathrm{IO}} = \$50$, $C_1 = \$10$ and $C_2 = \$20$.

(C_V, C_{BD})	$(70, 100)$	$(70, 150)$	$(70, 200)$	$(80, 100)$	$(90, 100)$
T_{cost}	157.738	158.513	159.288	163.994	173.5100

TABLE 3. Influence of (C_1, C_2) on the expected cost function T_{cost} with $C_H = \$85$, $C_W = \$200$, $C_V = \$70$, $C_{\mathrm{BD}} = \$100$, $C_{\mathrm{IO}} = \$50$.

(C_1, C_2)	$(10, 20)$	$(10, 25)$	$(10, 30)$	$(15, 20)$	$(20, 20)$
T_{cost}	157.7380	170	183	167.738	177.738

6.1. Practical application

An example of our queueing model with heterogeneous servers that is intermittently available under hybrid vacations is found in online stores, hospitals, and many banking systems. At a bank, there are two or more servers available, with one always available to handle customer requests and the other intermittently available to handle more complex requests. Depending on the situation, the intermittently accessible server may be in a working vacation state, providing services at a slower rate than usual, or an operational vacation state, providing services at a different rate. In the event of a large number of requests or a surge in customer inquiries, both servers may become available to assist in providing services. This allows for greater flexibility in handling customer requests and helps reduce customer wait times.

6.2. Special cases

Case i. No vacation.

Let $\mu_0 = 0$, $\mu_v = 0$, ϕ=0, Our model is reduced to a Markovian Queueing Model with Intermittently Accessible Server and Restoration, which is coincides with [28, 29].

Case ii. No working vacation and intermittently obtainable server.

Let $\mu_v = 0$, $\mu_2 = 0$, Our model is Markovian Bernoulli queues with operational server vacation, Bernoulli's weak and strong disasters. which is coincides with [10].

7. ANFIS

Complexities in computing performance indices might make it appear that the analytical approaches used to analyze and estimate the behavior of complicated systems are very difficult to put into practice. Soft computing methods are applicable in these scenarios. To deal with the inherent uncertainty of real-world circumstances, soft computing offers adaptable data processing methods. It makes use of people's willingness to deal with fuzziness and ambiguity in order to make things more manageable, cheaper, and easier to talk about.

ANFIS is a hybrid soft-computing system that incorporates artificial neural networks (ANNs) and fuzzy systems (FS). In Fig. 2, Node and directed connections make up the network structure of an adaptive network. Each node in an adaptive network processes incoming data according to its own settings. Some or all nodes are adaptive, meaning that their outputs depend on their parameters, and the training algorithm determines which parameters should be updated to maximize some objective function. These parameters are adjusted depending on new training data and a gradient-based learning technique to produce the appropriate input-output mapping.

An input-output mapping may be generated by ANFIS based not just on human knowledge, which is represented by fuzzy if-then rules, but also on a collection of input-output data pairings that have already been

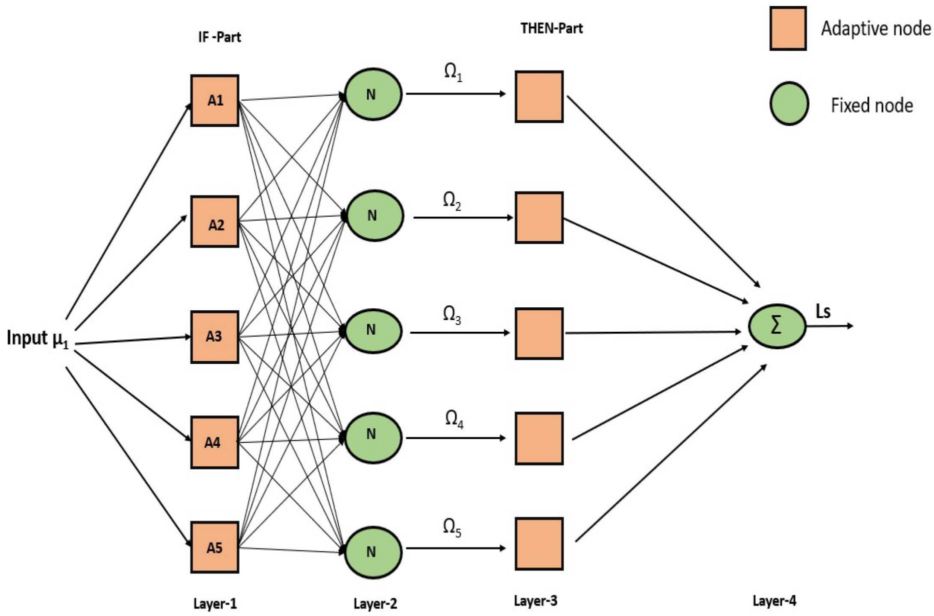


FIGURE 2. ANFIS architecture.

identified. We have a fuzzy if-then rule when both A and B name for fuzzy sets that have membership functions that are clearly specified. In a situation when there is a lot of congestion, for instance, it is an undeniable fact that the length of the line will be relatively short if the arrival rate is low (high), which is an illustration of a fuzzy if-then rule (high). The pace of arrival, as well as the length of the line, are both examples of linguistic variables, whereas low and high are examples of linguistic values that are determined by the membership function.

A fuzzy inference system is comprised of three primary components:

- A rule base that contains a collection of fuzzy rules.
- A database that contains definitions of the membership functions used by the fuzzy rules.
- A reasoning mechanism that executes the inference procedure in accordance with the rules and the given condition in order to obtain a reasonable result.

Each of these components is described in more detail below. Variations in the arrival rate (λ), service rates (μ_1, μ_2), and vacation rate (μ_0) are used in the neuro-fuzzy technique to calculate the anticipated number of customers in the system L_s. This is accomplished by changing the vacation rate. In the context of fuzzy systems, these factors are regarded as linguistic variables, and within ANFIS networks, they are used as input variables. The Gaussian function provides the membership functions for all of these input variables.

The number of membership functions and their matching linguistic input parameter values are provided in Table 4. Figure 3 depicts the forms of the related membership functions. To do this, we first calculate the analytical performance indices and then evaluate how they stack up against the output from ANFIS. We think about a fuzzy inference system that takes in four inputs (λ, μ_1, μ_2 and μ_0) and produces a single output (L_s).

8. Numerical illustration

We conducted a numerical experiment to show that the matrix geometric method is feasible. MATLAB is used to write the necessary step for this purpose. The value of $\lambda, \mu_1, \mu_2, \mu_0, \mu_v$, now varies whereas the remaining variables remain constant. First, the rate matrix was computed by using Algorithm 1. The rate matrix R was

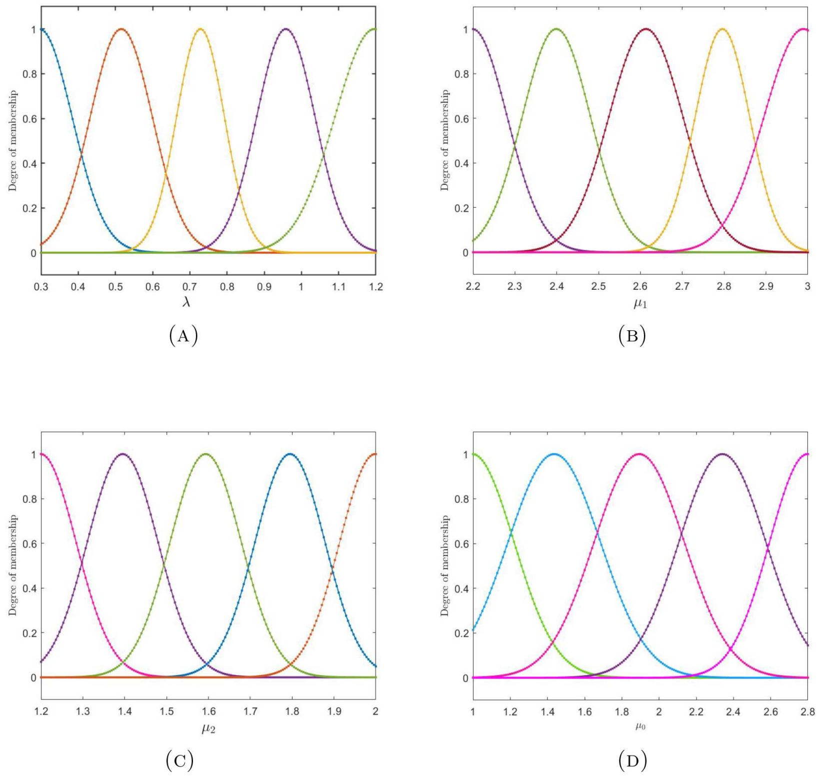


FIGURE 3. Membership function for (A) λ, (B) μ_1, (C) μ_2, (D) μ_0 input variables.

TABLE 4. Values of the membership function that correspond to linguistics for the input parameters.

Input parameters	No. of membership function	Linguistic values
λ, μ_1, μ_2, μ_0	5	Very Low Low Moderate High Very High

applied to equations (19)–(21) and solved to find the probability vectors Δ_0 and Δ_1. Then, by substituting Δ_1 into equation (18), we can obtain the remaining vectors $\Delta_i's$ (by Algorithm 2). Finally, we calculated the sum of the five components (*i.e.*, Δ_0, Δ_1, Δ_2, Δ_3, and Δ_4) and confirmed that the aggregate probability value is approximately 1.

As an example, we provide numerical results. Additional parameters are as follows: $\lambda = 0.3$, $\mu_1 = 2$, $\mu_2 = 2.5$, $\mu_0 = 1$, $\mu_v = 1.5$, $\phi = 1.6$, $\gamma = 0.5$, $\alpha = 2.5$, $\beta = 4$, $\sigma = 0.5$, $\eta = 1.2$, $\kappa = 0.5$, $p = 0.7$, $q = 0.6$. Probabilities,

TABLE 5. Steady state probability vectors.

	0	1	2	3	4	Total
Δ_0	0.8640					0.8640
Δ_1	0.0794	0.0275	0.0126	0.0000	0.0000	0.1195
Δ_2	0.0075	0.0045	0.0025	0.0000	0.0000	0.0145
Δ_3	0.0007	0.0006	0.0004	0.0000	0.0000	0.0017
Total						0.9997

rate matrices, and submatrix representations are derived in this way.

$$D_0 = \begin{bmatrix}-0.3000\end{bmatrix}; \quad D_1 = \begin{bmatrix}0.3000 & 0 & 0 & 0 & 0\end{bmatrix}; \quad E_0 = \begin{bmatrix}1.3500\\4.5000\\2.0000\\2.0000\\3.5000\end{bmatrix};$$

$$E_1 = \begin{bmatrix}0.3500\\0\\0\\0\\0\end{bmatrix}; \quad V_0 = \begin{bmatrix}0.6000 & 0.4000 & 0 & 0 & 0\\0 & 4.5000 & 0 & 0 & 0\\0 & 0 & 2.0000 & 0 & 0\\0 & 0 & 0 & 2.0000 & 0\\0 & 0 & 0 & 0 & 3.5000\end{bmatrix};$$

$$V_1 = \begin{bmatrix}-3.4000 & 1.6000 & 0.1500 & 0 & 0\\0 & -7.3000 & 2.5000 & 0 & 0\\0.5000 & 4.0000 & -6.8000 & 0 & 0\\0 & 0.5000 & 0 & -2.8000 & 0\\1.2000 & 0 & 0 & 0 & -5.0000\end{bmatrix}; \text{ and}$$

$$V_2 = \begin{bmatrix}0.3000 & 0 & 0 & 0 & 0\\0 & 0.3000 & 0 & 0 & 0\\0 & 0 & 0.3000 & 0 & 0\\0 & 0 & 0 & 0.3000 & 0\\0 & 0 & 0 & 0 & 0.3000\end{bmatrix}.$$

The rate matrix is obtained by solving equation (14)

$$R = \begin{bmatrix}0.0919 & 0.0319 & 0.0146 & 0 & 0\\0.0033 & 0.0561 & 0.0214 & 0 & 0\\0.0090 & 0.0374 & 0.0594 & 0 & 0\\0.0008 & 0.0123 & 0.0049 & 0.1169 & 0\\0.0234 & 0.0087 & 0.0040 & 0 & 0.0628\end{bmatrix}.$$

State probabilities are shown in Table 5.

Other performance indices are $P_S^{\text{Idle}} = 0.8640, P_{S2}^{\text{OV}} = 0.9516, P_{S2}^{W} = 0.0326, P_{S2}^{\text{BD}} = 0.0155, P_{S2}^{\text{IO}} = 0, E(S)_W = 0.0383, E(S)_{\text{BD}} = 0.0188, E(S)_{\text{IO}} = 0, E(S)_V = 0.0965, L_s = 0.1536, E(Q)_W = 0.0057, E(Q)_{\text{BD}} = 0.0033, E(Q)_{\text{IO}} = 0.0000, E(Q)_V = 0.0089, L_q = 0.0179$. Cost factors used for cost function analysis are: $C_H = 85, C_W = 200, C_V = 70, C_{\text{BD}} = 100, C_{\text{IO}} = 50, C_1 = 10, C_2 = 20$, The corresponding optimal cost T_{cost} is **$157.7380**.

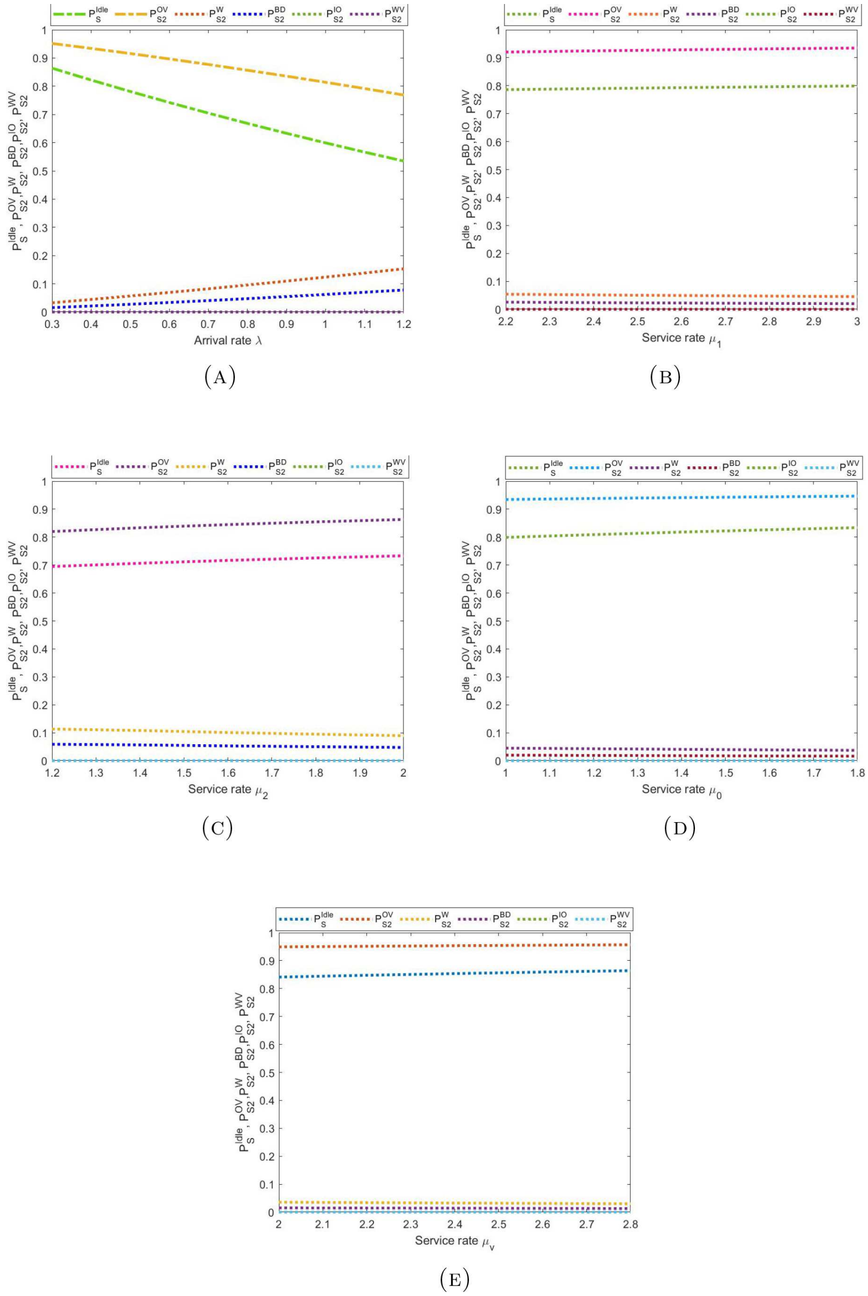


FIGURE 4. System state probabilities *vs.* λ, μ_1, μ_2, μ_0. (A) λ *vs.* System state probabilities. (B) μ_1 *vs.* System state probabilities. (C) μ_2 *vs.* System state probabilities. (D) μ_0 *vs.* System state probabilities. (E) μ_v *vs.* System state probabilities.

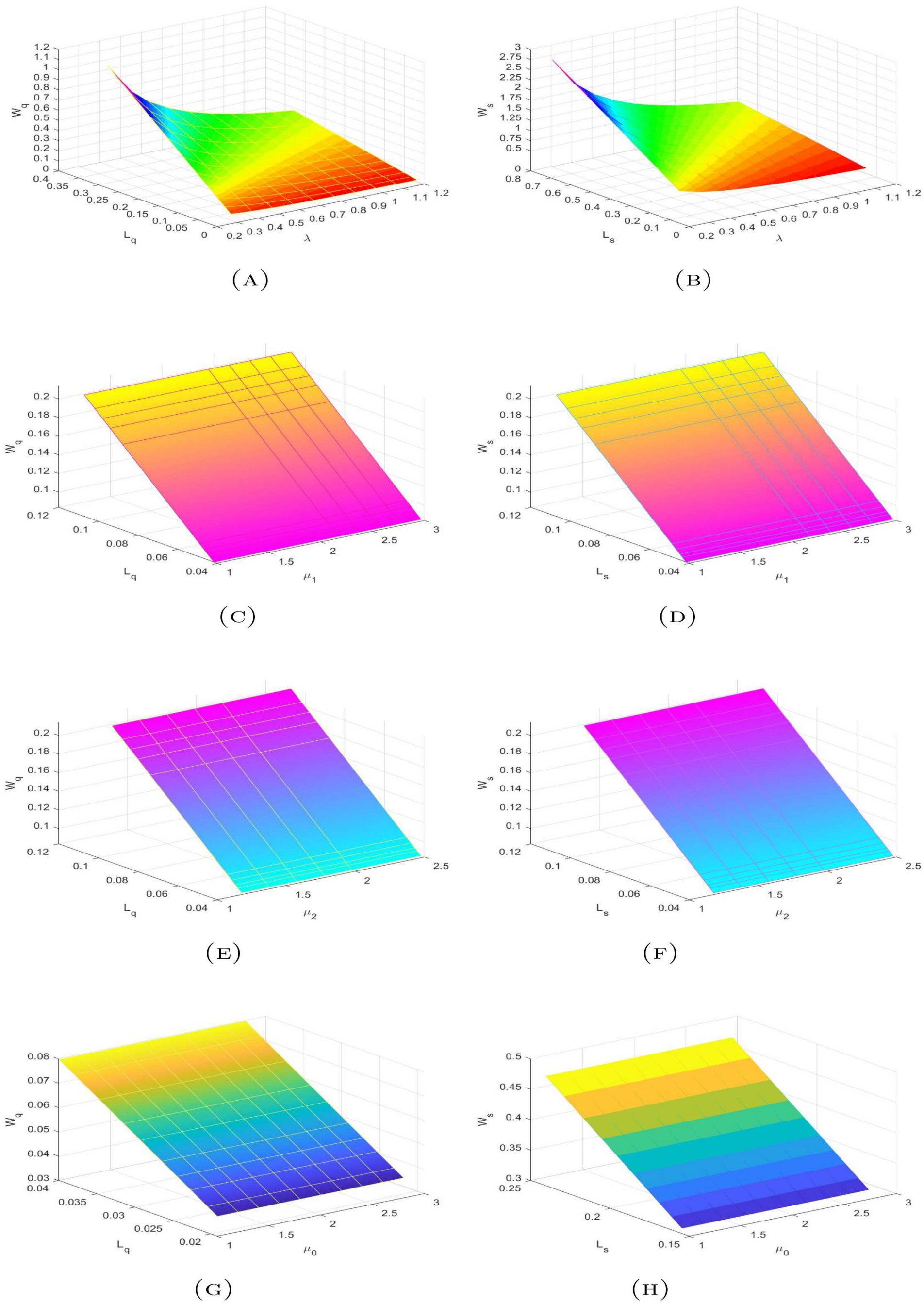


FIGURE 5. Effects of a few parameters on 3D representation. (A) λ *vs.* L_q, W_q. (B) λ *vs.* L_s, W_s. (C) μ_1 *vs.* L_q, W_q. (D) μ_1 *vs.* L_s, W_s. (E) μ_2 *vs.* L_q, W_q. (F) μ_2 *vs.* L_s, W_s. (G) μ_0 *vs.* L_q, W_q. (H) μ_0 *vs.* L_s, W_s.

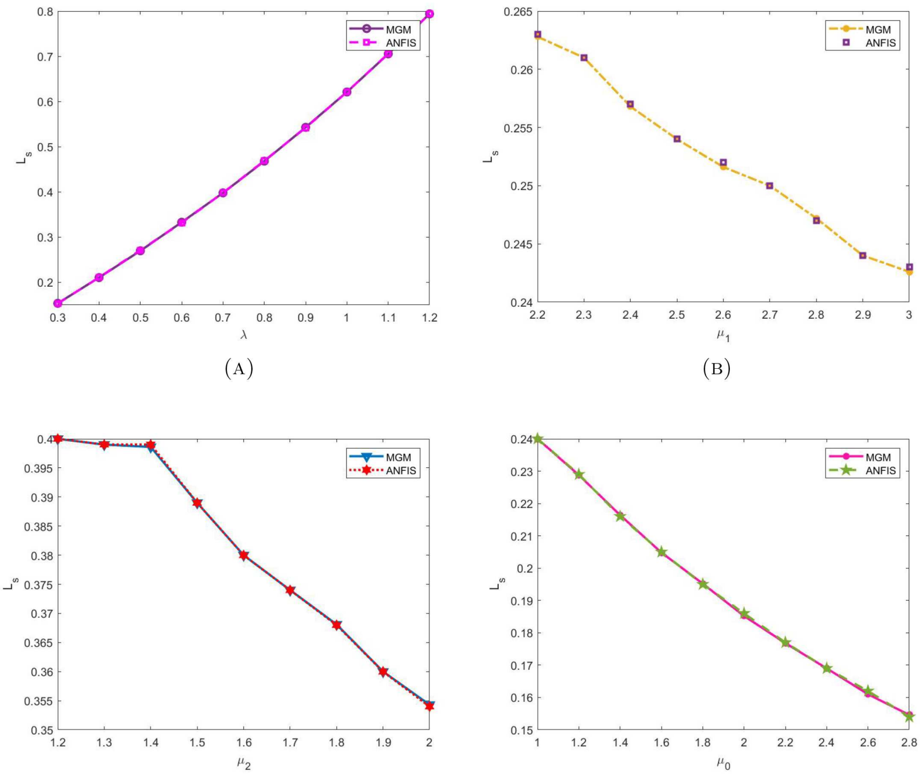
TABLE 6. Effect of λ, (μ_1, μ_2), (μ_0, μ_v) on performace measures.

λ	$E(S)_V$	$E(S)_W$	$E(S)_{\text{BD}}$	$E(S)_{\text{IO}}$	L_s	$E(Q)_V$	$E(Q)_W$	$E(Q)_{\text{BD}}$	$E(Q)_{\text{IO}}$	L_q	T_{cost}
0.3	0.0965	0.0383	0.0188	0.0000	0.1536	0.0089	0.0057	0.0033	0.0000	0.0179	157.7380
0.4	0.1278	0.0554	0.0274	0.0000	0.2106	0.0157	0.0109	0.0061	0.0000	0.0327	164.3180
0.5	0.1582	0.0744	0.0372	0.0000	0.2698	0.0238	0.0177	0.0099	0.0000	0.0514	171.1160
0.6	0.1878	0.0961	0.0491	0.0000	0.3330	0.0334	0.0268	0.0153	0.0000	0.0755	178.3210
0.7	0.2163	0.1197	0.0619	0.0000	0.3979	0.0439	0.0375	0.0215	0.0000	0.1029	185.6985
0.8	0.2438	0.1474	0.0769	0.0000	0.4681	0.0555	0.0517	0.0296	0.0000	0.1368	193.6205
0.9	0.2712	0.1775	0.0942	0.0000	0.5429	0.0687	0.068	0.0396	0.0000	0.1763	202.0055
1	0.2965	0.2113	0.1133	0.0000	0.6211	0.0818	0.0876	0.0511	0.0000	0.2205	210.7335
1.1	0.3221	0.2483	0.1350	0.0000	0.7054	0.0968	0.1102	0.0650	0.0000	0.2720	220.0050
1.2	0.3450	0.2903	0.1586	0.0000	0.7939	0.1109	0.1372	0.0806	0.0000	0.3287	229.7245
(μ_1, μ_2)	$E(S)_V$	$E(S)_W$	$E(S)_{\text{BD}}$	$E(S)_{\text{IO}}$	L_s	$E(Q)_V$	$E(Q)_W$	$E(Q)_{\text{BD}}$	$E(Q)_{\text{IO}}$	L_q	T_{cost}
$(1, 1.2)$	0.1494	0.1624	0.0872	0.0000	0.3990	0.0242	0.0491	0.0284	0.0000	0.1017	153.8550
$(1, 1.4)$	0.1530	0.1590	0.0866	0.0000	0.3986	0.0259	0.0510	0.0301	0.0000	0.1070	157.4830
$(1, 1.6)$	0.1535	0.1461	0.0800	0.0000	0.3796	0.0254	0.0453	0.0271	0.0000	0.0978	158.8520
$(1, 1.8)$	0.1548	0.1373	0.0760	0.0000	0.3681	0.0257	0.0424	0.0259	0.0000	0.0940	161.1005
$(1, 2.0)$	0.1554	0.1281	0.0708	0.0000	0.3543	0.0255	0.0387	0.0235	0.0000	0.0877	163.1495
$(2.2, 2.5)$	0.1581	0.0700	0.0347	0.0000	0.2628	0.0235	0.0162	0.0092	0.0000	0.0489	172.0760
$(2.4.2.5)$	0.1585	0.0666	0.0317	0.0000	0.2568	0.0235	0.0152	0.0080	0.0000	0.0467	173.2000
$(2.6, 2.5)$	0.1589	0.0630	0.0297	0.0000	0.2516	0.0235	0.0139	0.0074	0.0000	0.0448	174.4240
$(2.8, 2.5)$	0.1592	0.0603	0.0279	0.0000	0.2474	0.0235	0.0132	0.0069	0.0000	0.0436	175.7750
$(3, 2.5)$	0.1591	0.0573	0.0262	0.0000	0.2426	0.0232	0.0122	0.0063	0.0000	0.0417	177.0740
(μ_0, μ_v)	$E(S)_V$	$E(S)_W$	$E(S)_{\text{BD}}$	$E(S)_{\text{IO}}$	L_s	$E(Q)_V$	$E(Q)_W$	$E(Q)_{\text{BD}}$	$E(Q)_{\text{IO}}$	L_q	T_{cost}
$(1, 1.0)$	0.1579	0.0565	0.0258	0.0000	0.2402	0.0223	0.0116	0.006	0.0000	0.0399	176.7990
$(1.2, 1)$	0.1502	0.0540	0.0246	0.0000	0.2288	0.0209	0.0112	0.0058	0.0000	0.0379	175.5690
$(1.4, 1)$	0.1422	0.0511	0.0231	0.0000	0.2164	0.0189	0.0104	0.0053	0.0000	0.0346	174.2120
$(1.6, 1)$	0.1348	0.0482	0.0218	0.0000	0.2048	0.017	0.0095	0.0049	0.0000	0.0314	172.9320
$(1.8, 1)$	0.1284	0.0459	0.0209	0.0000	0.1952	0.0156	0.0089	0.0047	0.0000	0.0292	171.8880
$(2, 1.0)$	0.1223	0.0436	0.0193	0.0000	0.1852	0.0142	0.0082	0.004	0.0000	0.0264	170.7960
$(2.2, 1)$	0.1167	0.0417	0.0184	0.0000	0.1768	0.0129	0.0077	0.0038	0.0000	0.0244	169.8790
$(2.4, 1)$	0.1116	0.0396	0.0178	0.0000	0.1690	0.0118	0.0071	0.0037	0.0000	0.0226	169.0130
$(2.6, 1)$	0.1066	0.0377	0.0167	0.0000	0.1610	0.0106	0.0065	0.0033	0.0000	0.0204	168.1220
$(2.8, 1)$	0.1023	0.0361	0.0162	0.0000	0.1546	0.0098	0.0061	0.0032	0.0000	0.0191	167.4100

8.1. Algorithm for the rate matrix and steady-state probability computation

Algorithm 1. An algorithm for computing rate matrix.

INPUT: V_0, V_1, V_2, e is a column vector of 1's and ϵ
OUTPUT: R
Step 1: Set $R_0 = 0$.
Step 2: while $|(V_2 + R_{(k)}V_1 + R_{(k)}^2 V_0)e)| \geq \epsilon \quad k = 0, 1, \ldots$
do steps 3 and 4
Step 3: set $R_{(k+1)} = -V_2 V_1^{-1} - R_{(k)}^2 V_0 V_1^{-1}. \quad k = 0, 1, \ldots$
Step 4: Continue **Step 3** untill $R_{(k+1)}$ is close to $R_{(n)}$.
Step 5: set $R = R_{(k+1)}$
Step 6: OUTPUT



(A) (B) (C) (D)

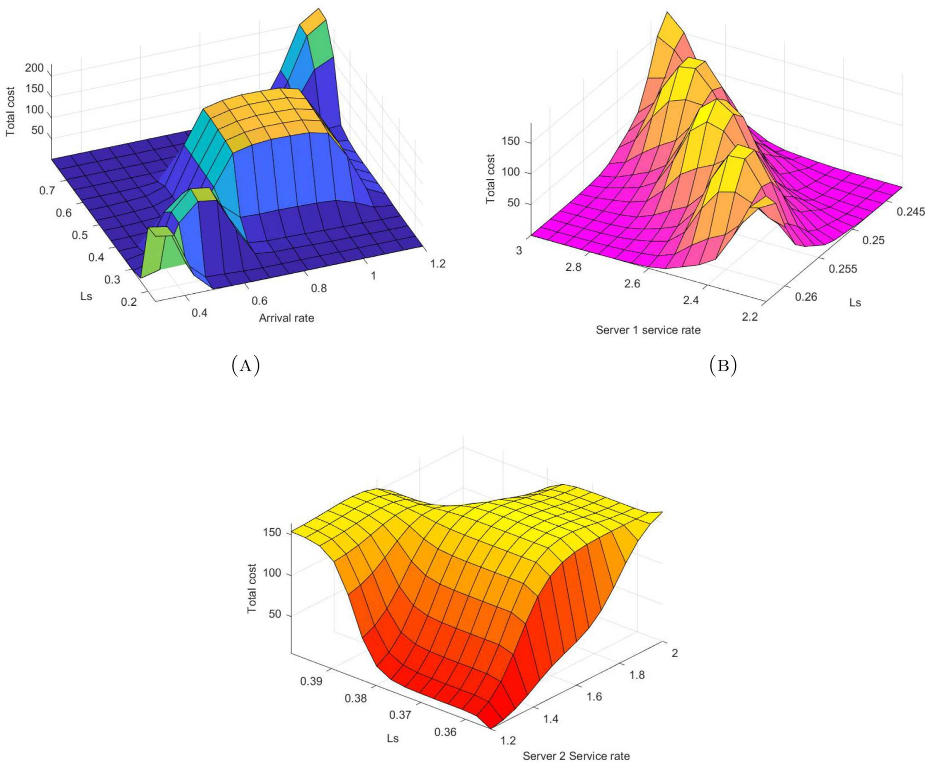
FIGURE 6. (A) λ *vs.* L_s. (B) μ_1 *vs.* L_s. (C) μ_2 *vs.* L_s. (D) μ_0 *vs.* L_s.

Algorithm 2. An algorithm for steady-state probability vectors.

INPUT: D_0,D_1, E_0, E_1 V_0, V_1, V_2, R, I, e is a column vector of 1's
OUTPUT: $\Delta_0, \Delta_1, \Delta_2, \Delta_3, \ldots$
Step 1: $P \leftarrow \begin{bmatrix}\Delta_0 \ \Delta_1\end{bmatrix} \begin{bmatrix} D_0 & E_0 + R(I-R)^{-1}E_1 \\ D_1 & V_1 + RV_0 \end{bmatrix}$
Step 2: Solve $P = 0$
Step 3: $Q \leftarrow \Delta_0 e + \Delta_1 (I-R)^{-1} e$
Step 4: Solve $Q = 1$
Step 5: $\Delta_0, \Delta_1 \leftarrow$ Solve (P, Q)
Step 6: Compute the steady-state probability vectors
If $k \geq 2$ compute $\Delta_k = \Delta_1 R^{k-1}$
Step 6: OUTPUT

9. Sensitivity analysis

In this section, we obtain some estimation findings from the probability vectors, such as the probability of servers being idle, the probability of server 2 being on operational vacation, the probability of server 2 being in working mode, the probability of server 2 being in breakdown mode, the probability of server 2 being intermittently obtainable, the probability of server 2 being on working vacation, and the mean number of customers in the system or queue when server 2 is in the OV, W, BD, IO, and WV states. By varying the



(A) (B) (C)

FIGURE 7. Effects of total cost on 3D representation. (A) λ, L_s *vs.* T_{cost}. (B) μ_1, L_s *vs.* T_{cost}. (C) μ_2, L_s *vs.* T_{cost}.

parameters $\lambda = 0.3$ to 1.2, $\mu_1 = 2.2$ to 3, $\mu_2 = 1.2$ to 2, and $\mu_0 = 1$ to 1.8, the system performance measures are shown below in Table 6.

- λ escalates, L_q, L_s also escalates for the value of $\mu_0 = 1$, $\mu_1 = 2$, $\mu_2 = 2.5$, $\mu_v = 1.5$, $\phi = 1.6$, $\gamma = 0.5$, $\alpha = 2.5$, $\beta = 4$, $\sigma = 0.5$, $\eta = 1.2$, $\kappa = 0.5$, $p = 0.7$, $q = 0.6$.
- μ_1 escalates, L_q, L_s also diminish for the value of $\lambda = 0.5$, $\mu_0 = 1$, $\mu_v = 1.5$, $\phi = 1.6$, $\gamma = 0.5$, $\alpha = 2.5$, $\beta = 4$, $\sigma = 0.5$, $\eta = 1.2$, $\kappa = 0.5$, $p = 0.7$, $q = 0.6$.
- μ_2 escalates, L_q, L_s also diminish for the value of $\lambda = 0.5$, $\mu_0 = 1$, $\mu_v = 1.5$, $\phi = 1.6$, $\gamma = 0.5$, $\alpha = 2.5$, $\beta = 4$, $\sigma = 0.5$, $\eta = 1.2$, $\kappa = 0.5$, $p = 0.7$, $q = 0.6$.
- μ_0 escalates, L_q, L_s also diminish for the value of $\lambda = 0.5$, $\mu_1 = 3$, $\mu_2 = 2.5$, $\mu_v = 1$, $\phi = 1.6$, $\gamma = 0.5$, $\alpha = 2.5$, $\beta = 4$, $\sigma = 0.5$, $\eta = 1.2$, $\kappa = 0.5$, $p = 0.7$, $q = 0.6$.

Figure 4 reveals that, if the arrival rate rises, the probability of servers being idle, the probability of server 2 being in the OV state diminishes, and the probability of server 2 being in the W, BD, IO, and WV states elevates. If the service rates of servers 1 and 2, vacation rates are rises, the probability of servers being idle, probability of server 2 being in OV state is slightly elevated, and the probability of server 2 being in W, BD, IO, and WV state slightly diminishes.

Figures 5A–5H depicts a three-dimensional graph of system performance measures. In Figure 5A, the surface displays the escalation of the arrival rate (λ), expected queue length (L_q), expected waiting time (W_q) as they increase. In Figure 5C, we found that the expected queue length (L_q), and expected waiting time (W_q) diminish while increasing the service rate μ_1. Figure 5E we found that the expected queue length (L_q), and expected waiting time (W_q) diminish while increasing the service rate μ_2. Figure 5G we found that the expected queue length (L_q), and expected waiting time (W_q) diminish while increasing the service rate μ_0.

The expected number of clients in the system, L_s, is shown graphically in Figures 6A–6D by changing the arrival rate λ, the service rate μ_1, the vacation rate μ_2, and the vacation rate μ_0. The thick (dotted) lines indicate the MGM (ANFIS) outcomes. To explain, in Figure 6A, we observe that the expected number of people in the system rises significantly as more customers join the system, *i.e.*, as λ increases. The predicted number of system users is substantially reduced when the service rate μ_1 rises, as shown in Figure 6B. Figures 6C and 6D illustrates the effect of μ_2 and μ_0 on the expected number of customers in the system. We found a similar pattern for μ_1 on L_s as noticed in Figure 6B. *i.e.*, on increasing μ_2 and μ_0 initially, there is a gradual decrement in L_s.

It is clear from all the figures that the predicted number of clients in the system drops dramatically as the number of servers rises, as is to be expected in actual congestion circumstances. Additionally, we discovered that in every situation, the ANFIS findings are far more accurate than the numerical results.

Figures 7A–7C displays the total expected cost per unit time for various cost factors by modifying arrival rate λ, server service rates $\mu 1$, and $\mu 2$ simultaneously.

10. Conclusion

In this study, we have examined a heterogeneous, unreliable server queueing model with intermittently accessible service under a hybrid vacation policy. Our investigation focuses on a Markovian queueing model that incorporates a hybrid vacation approach, comprising both operational vacation and working vacation. Through numerical examples, we have demonstrated the computational efficiency of the matrix-geometric approach in evaluating steady-state probabilities and other performance indicators. By utilizing the matrix geometric technique and employing a soft computing approach, specifically the ANFIS, we have successfully calculated performance indices and the cost function, affirming the practical applicability of our model within real-time systems. Moreover, our comparison between the ANFIS and numerical results (MGM) highlights the potential and convenience of the neuro-fuzzy tool in predicting the performance of queueing systems operating under diverse industrial and technological constraints. Overall, this study showcases the effectiveness of incorporating phase-type arrival, impatient customers, and multi-stage service concepts, making our method applicable in realistic and close-to-real-life scenarios.

Acknowledgements

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

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] N.N. Agarwal, *Some problems in the theory of reliability and queues*. Ph.D. thesis, Kurukshetra university, Kurukshetra (1965).

[2] A. Ahuja, A. Jain and M. Jain, Transient analysis and ANFIS computing of unreliable single server queueing model with multiple stage service and functioning vacation. *Math. Comput. Simul.* **192** (2022) 464–490.

[3] R. Aniyeri and C.R. Nadar, A multiphase queuing system with assorted servers by using matrix geometric method. *Int. J. Appl. Eng.* **382** (2017) 12052–12059.

[4] F. Afroun, D. Aïssani, D. Hamadouche and M. Boualem, Q-matrix method for the analysis and performance evaluation of unreliable $M/M/1/N$ queueing model. *Math. Methods Appl. Sci.* **41** (2018) 9152–9163.

[5] A.A. Bouchentouf and M. Boualem, Analysis and performance evaluation of Markovian feedback multi-server queueing model with vacation and impatience. *AJMMS* **40** (2021) 261–282.

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

[7] A.A. Bouchentouf, M. Boualem, M. Cherfaoui and L. Medjahri, Variant vacation queueing system with Bernoulli feedback, balking and server's states-dependent reneging. *Yugoslav J. Oper. Res.* **31** (2021) 557–575.

[8] A.A. Bouchentouf, M. Boualem, L. Yahiaoui and H. Ahmad, A multi-station unreliable machine model with working vacation policy and customers' impatience. *Qual. Technol. Quant. Manage.* **19** (2022) 766–796.

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

[10] E.B. Bourennane, M. Ourbih-Tari, H. Saggou and Z. Ezzourgui, The Markovian Bernoulli queues with operational server vacation, Bernoulli's weak and strong disasters, and linear impatient customers. *Commun. Math. Sci.* (2022) 31.

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

[12] P. Deora, U. Kumari and D.C. Sharma, Cost analysis and optimization of machine repair model with working vacation and feedback-policy. *Int. Comput. Appl. Math.* **7** (2021) 1–14.

[13] A. Goel, A. Chauhan and A.K. Malik, Applications of Advanced Optimization Techniques in Industrial Engineering. CRC Press (2022).

[14] Indra and Vijay Rajan, Queuing analysis of markovian queue having two heterogeneous servers with catastrophes using matrix geometric technique. *IJSS* **12** (2017) 205–212.

[15] J.S.R. Jang and N. Gulley, Fuzzy Logic Toolbox User'S Guide. Vol. 1. The Mathworks Inc. (1995) 19.

[16] M. Jain and S. Dhibar, ANFIS and metaheuristic optimization for strategic joining policy with re-attempt and vacation. *Math. Comput. Simul.* **211** (2023) 57–84.

[17] M. Jain and A. Jain, Working vacations queueing model with multiple types of server breakdowns. *Appl. Math. Model.* **34** (2010) 1–13.

[18] M. Jain, S.S. Sanga and R.K. Meena, Control F-policy for Markovian retrial queue with server breakdowns, in 2016 IEEE 1st International Conference on Power Electronics, ICS. IEEE (2016) 1–5.

[19] R. Kalyanaraman and A. Sundaramoorthy, A Markovian working vacation queue with server state-dependent arrival rate and partial breakdown. *Int. J. Recent Technol. Eng.* **7** (2019) 664–668.

[20] A. Kumar and M. Jain, Cost optimization of an unreliable server queue with two stage service process under hybrid vacation policy. *Math. Comput. Simul.* **204** (2023) 259–281.

[21] P. Kumar, M. Jain and R.K. Meena, Transient analysis and reliability modeling of fault-tolerant system operating under admission control policy with double retrial features and working vacation. *ISA Trans.* **1** (2023) 183–199.

[22] G. Latouche and V. Ramaswami, Introduction to Matrix Analytic Methods in Stochastic Modeling. SIAM (1999).

[23] G.C. Mytalas and M.A. Zazanis, Performance analysis for Bernoulli feedback queues subject to disasters: a system with batch Poisson arrivals under a multiple vacation policy. *Qual. Technol. Quant. Manage.* **20** (2023) 113–146.

[24] M.F. Neuts, Matrix-Geometric Solutions in Stochastic Models. Vol. 2. Johns Hopkins Univ. Baltimore, USA (1981).

[25] M.R. Raj and B. Chandrasekar, Matrix-geometric method for queueing model with subject to breakdown and N-policy vacations. *Int. J. Math. Aeterna* **5** (2015) 917–926.

[26] S.S. Sanga and M. Jain, Cost optimization and ANFIS computing for admission control of $M/M/1/K$ queue with general retrial times and discouragement. *Appl. Math. Comput.* **363** (2019) 124624.

[27] M. Seenivasan and M. Indumathi, Performance analysis of two heterogeneous server queuing model with intermittently obtainable server using matrix geometric method. *Am. J. Phys.: Conf. Ser.* **1724** (2021) 012001.

[28] M. Seenivasan, M. Kameswari and M. Indumathi, Markovian Queueing Model with Intermittently Accessible Server and Restoration Using Matrix Geometric Method. IEEE (2022) 1–7.

[29] M. Seenivasan, R. Senthilkumar and K.S. Subasri, $M/M/2$ heterogeneous queueing system having unreliable server with catastrophes and restoration. *Mater. Today: Proc.* **51** (2022) 2332–2338.

[30] L.D. Servi and S.G. Finn, $M/M/1$ queues with working vacations ($M/M/1/WV$), perform. *Evaluation* **50** (2002) 41–52.

[31] R. Sethi, M. Jain, R.K. Meena and D. Garg, Cost optimization and ANFIS computing of an unreliable $M/M/1$ queueing system with customers impatience under N-policy. *Int. J. Appl. Comput. Math.* **6** (2020) 1–14.

[32] Sharda, A queuing problem with intermittently available server and arrivals and departures in batches of variable size. *ZAMM* **48** (1968) 471–476.

[33] C. Shekhar, S. Varshney and A. Kumar, Matrix-geometric solution of multi-server queueing systems with Bernoulli scheduled modified vacation and retention of reneged customers: a meta-heuristic approach. *Qual. Technol. Quant. Manage.* **18** (2021) 39–66.

[34] V.P. Singh, Two-server Markovian queues with balking: heterogeneous vs. homogeneous servers. *Oper. Res.* **18** (1970) 145–159.

[35] N. Singh, M. Jain and S. Dhibar, ANFIS computing for $M/M/\infty$ queue with two types of service interruption and balking. *Proc. Nat. Acad. Sci. India Sect. A: Phys. Sci.* **94** (2024) 63–73.

[36] S. Thakur, A. Jain and M. Jain, ANFIS and cost optimization for markovian queue with operational vacation. *Int. J. Math. Eng. Manage. Sci.* **6** (2021) 894.

[37] S. Wang, Z. Ma, X. Niu and Y. Liu, Performance analysis of a Queueing system based on vacation with fault repairable and spare servers in the MP2P network. *Wireless Networks* **29** (2023) 2321–2336.