International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
Article Received: *10 September 2023* ***Revised:*** *20 October 2023* ***Accepted:*** *30 October 2023*

Mathematical Modeling of Multi-Server Queueing Model for Vacations and Impatient Customer

Dr. Kanta Malik
Assistant Professor, Vivekananda Institute of Professional Studies – Technical Campus, New Delhi. Email: kanta.malik@vips.edu

Abstract: The combination of excursions and the powerful treatment of fretful clients are at the focal point of this review, which digs into the streamlining of multi-server queueing frameworks. Modified Priority Queue Algorithm (MPQA), Vacation Scheduling Algorithm (VSA), Impatience Handling Algorithm (IHA), and Server Utilization Balancing Algorithm (SUBA) are the four calculations that have been proposed and thoroughly assessed through recreations to decide their viability in decreasing line length, holding up time, and improving framework throughput and server usage. The MPQA shines on clients considering anxiety cutoff points and organization times, showing suitability in decreasing holding-up times. Then again, VSA continuously changes server get-away, yielding lower holding-up times during fame periods. IHA progressively oversees eagerness, diminishing takeoffs brought about by fretfulness and expanding framework throughput in general. SUBA in a perfect world adjusts server use, achieving additionally created throughput and system strength. The extraordinary commitments of each calculation are revealed in examinations with related work. VSA is in accordance with past examinations an extended get-away streamlining, while MPQA works on traditional ways to deal with prioritization. IHA presents dynamic energy dealing with, and SUBA lines up with liability careful server activation systems found in the composition. This assessment basically advances the appreciation of multi-server queueing systems, offering valuable solutions for organization arranged conditions. An establishment for future improvements in queueing hypothesis and genuine applications is given by the calculations, which were assessed with significant outcomes.

Keywords: Systems for queuing as well as vacations, customers who gets impatient, optimization algorithms, in addition to system throughput.

I. INTRODUCTION

In the space of organization-arranged undertakings, the organization and improvement of queueing structures expect a dire part in updating practical viability and shopper faithfulness. As associations try to offer helpful kinds of help to a creating client base, the components of multi-waiter queueing models become logically erratic [1]. This assessment dives into the mathematical demonstrating of a multi-server queueing structure that incorporates server excursion and addresses the enthusiasm displayed by clients. Understanding the elements of delaying lines and administration processes has been made conceivable by the conventional queueing hypothesis, which fundamentally centres on situations in which servers are dependably accessible. Regardless, genuine assist frameworks with habitually including servers that discontinuously appreciate respites or get-away, introducing a layer of complexity that demands specific showing. Plus, client irritability is a fundamental variable, as individuals would leave the line if their holding up time outperforms a particular breaking point. In a multi-server environment, the intersection of these peculiarities creates a challenging but useful examination space [2]. This investigation expects to give a broad mathematical design to showing multi-server queueing systems with trips and energetic clients, offering encounters that can teach the arrangement and the chiefs in regards to organization arranged structures. The mix of server excursions perceives the human part in assistance arrangement, seeing that server isn't ceaselessly open. Simultaneously, the prospect of client instability perceives the state-of-the-art reality where individuals search for prompt and capable organizations [3]. The fundamental focuses of this assessment are twofold: first, to develop a robust mathematical model of the dynamics of a multi-server queueing system with periodic server vacations. Second, to expand this model to include customer impatience as a dynamic factor influencing the behaviour of the system. By accomplishing these goals, the examination tries to add to the hypothetical groundwork of the queueing hypothesis while giving down-to-earth bits of knowledge appropriate to genuine assistance situations [4]. This examination expects to add to the current collection of information in the queueing hypothesis by introducing a nuanced model that catches the transaction between multi-server setups, vacation and client eagerness. The discoveries are supposed to offer significant rules for improving assistance frameworks in businesses going from broadcast communications and medical services to transportation and then some. The foundational ideas of queueing theory, existing models, and research gaps will all be discussed in detail in the following sections of this paper, which will delve into the literature review. The philosophy area will frame the logical and recreation-based approaches utilized. Results and conversations will introduce the results of the examination, trailed by an end that sums up the vital discoveries and diagrams possible roads for future exploration in the powerful field of multi-server queueing frameworks with excursions and eager clients.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
Article Received: *10 September 2023* ***Revised:*** *20 October 2023* ***Accepted:*** *30 October 2023*

II. RELATED WORKS

The analysis and optimization of service systems have relied heavily on queueing theory, which provides useful insights into the dynamics of waiting lines, service procedures, and customer interactions. The collection of writing connected with multi-server queueing frameworks with different intricacies, including server excursions and fretful clients, has seen critical commitments lately. In order to shed light on the advancements made by researchers in their understanding and modelling of these intricate systems, we examine and synthesize the findings of relevant studies in this section. Samoulov, Dudina, and Dudin inspected the elements of a multi-server queueing framework with adaptable needs [15]. Their work zeroed in on what the presentation of adaptable needs means for framework execution. While not straightforwardly tending to vacations or fretfulness, this study fills in as an establishment for understanding the effect of prioritization in multi-server conditions, an idea that might entwine with the more extensive setting of server accessibility and consumer loyalty. Saravanan, Poongothai, and Godhandaraman dove into the exhibition examination of a retrial queueing framework highlighting discretionary help, a temperamental server, recoiling, and input [16]. Albeit not explicitly tending to excursions, the consideration of an inconsistent server and retrial elements adds a layer of intricacy to the help interaction. The review features the requirement for exhaustive models that consider different aspects of administration frameworks, which lines up with the intricacy inborn in multi-server conditions with vacation and restless clients. Thakur, Jain, and Jain delivered the usage of ANFIS for cost streamlining in a Markovian line with functional vacation [17]. While zeroing in on a solitary server framework, their work stresses the combination of cutting edge computational strategies for framework examination and improvement. This fills in as a sign of the potential for consolidating logical and computational methodologies in demonstrating queueing frameworks with dynamic components. Vinitha et al. examined a stochastic stock model in a random environment with two classes of providers as well as drive clients [18]. While not straightforwardly connected with queueing frameworks, this work highlights the significance of considering stochastic components in help related models. The consideration of drive clients presents a component of eccentrics, lining up with the flightiness inborn in client eagerness. Under synchronous working vacations and impatient customers, Yahiaoui, Bouchentouf, and Kadi focused on the optimal cost analysis for a Geo/Geo/c/N feedback queue [19]. This study overcomes any barrier between server excursions and client fretfulness, exhibiting the interchange between these unique components. The simultaneous working excursions acquaint a fleeting perspective with the framework, lining up with the worldly idea of server vacation. Yin, Yan, Guo, and Liu investigated the examination of a precautionary two-need lining framework with eager clients and heterogeneous servers [20]. The joining of two need levels and heterogeneous servers adds a layer of authenticity to the model, reflecting situations where various clients might have differing administration prerequisites. This work adds to understanding the nuanced cooperations in multi-server conditions. Amina, Cherfaoui, and Mohamed played out a presentation and monetary investigation of a solitary server input queueing model with excursion and restless clients [21]. While focusing on a single-server system, their work gives bits of knowledge into the monetary ramifications of presenting excursions and dealing with eagerness. Understanding the financial perspectives is urgent for certifiable applications, particularly in asset portion and cost-viability contemplations. Bouchentouf and Abdelhak examined the MX/M/c Bernoulli input line with variation numerous functioning excursions and restless clients, leading both execution and financial investigation [22]. This study extends the comprehension of criticism lines with numerous functioning excursions, displaying the significance of thinking about both execution measurements and monetary ramifications in the examination of complex queueing frameworks. A randomized threshold strategy for providing flexible priority in a multi-server queueing system with a marked Markov arrival process and phase-type service time distribution was developed by Dudin, Dudin, and Dudina [23]. This study investigates the idea of adaptable need, lining up with the more extensive subject of prioritization in multi-waiter frameworks. The presentation of a randomized limit technique adds a unique component to the needed task. Harikrishnan et al. analyzed a stochastic M/M/c/N inventory system with multi-threshold stages, queue-dependent server activation, and an optional facility for retrial [24]. While zeroing in on stock frameworks, their work coordinates line subordinate server enactment, lining up with the elements of server accessibility in queueing frameworks. The system is made even more complicated by the addition of an optional facility for retrial. Islam, Islam, and Rashid embraced the displaying and investigation of a stochastic short-lived stock framework with eager clients at the help office [25]. Albeit not straightforwardly connected with queueing frameworks, this work accentuates the significance of considering perishability and client restlessness in assistance related models. The short-lived nature lines up with the time-delicate elements innate in queueing frameworks. Jeganathan et al. displayed the methodology of junior servers to a senior server in the retrial lining stock framework [26]. While zeroing in on retrial elements and stock frameworks, their work presents the idea of servers of various progressive levels moving toward one

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
Article Received: *10 September 2023* ***Revised:*** *20 October 2023* ***Accepted:*** *30 October 2023*

another. This dynamic lines up with the more extensive topic of server collaborations in multi-server frameworks.

III. METHODS AND MATERIALS

A. Data:

The observational underpinning of this exploration lies in the use of a recreated dataset to imitate a multi-server queueing framework with vacation and eager clients. The dataset involves boundaries, for example, appearance times, administration spans, server excursion timetables, and anxiety limits [5]. The variability and complexity inherent to queueing systems are captured in the simulated data, which is intended to reflect real-world scenarios found in service-oriented industries.

B. Simulation Environment:

The reenactment is executed utilizing a discrete-occasion recreation approach, where occasions like client appearances, administration fruitions, and server excursions trigger framework state changes. Programming the simulation environment in a high-level language makes it easier to change the model's parameters and collect performance metrics [6]. The system has the capability to arrive at a steady-state where performance metrics gets stabilized after the simulation runs for a predetermined interval of time.

C. Algorithms:

In this part, we present four estimations used for the assessment of the multi-server queueing system. Line elements, server excursions, as well as dealing with restlessness are only a couple of the particular issues tended to by every calculation. The estimations are according to the accompanying:

1. Modified Priority Queue Algorithm (MPQA):

The (MPQA) is planned to manage the remarkable need task in the multi-server queueing system. It thinks about both client fretfulness and server availability needing clients in the line. For each appearance up client, the calculation processes a need score considering as far as possible and the overabundance help time of the client [7]. For administration, the client with the most elevated need is picked. Due to a tie, the calculation favors the customer who has waited longer in line.

```
def modified_priority_queue_algorithm(arrival_times, service_durations, impatience_thresholds):
    priority_scores = calculate_priority_scores(arrival_times, service_durations, impatience_thresholds)
    sorted_customers = sort_customers_by_priority(priority_scores)
    return sorted_customers

def calculate_priority_scores(arrival_times, service_durations, impatience_thresholds):
    priority_scores = []
    for i in range(len(arrival_times)):
        wait_time = calculate_wait_time(arrival_times[:i])
        remaining_service_time = service_durations[i]
        priority_score = (impatience_thresholds[i] - wait_time) / remaining_service_time
        priority_scores.append(priority_score)
    return priority_scores

def calculate_wait_time(arrival_times):
    # Calculate the total wait time for customers in the queue
    return sum(arrival_times)

def sort_customers_by_priority(priority_scores):
    # Sort customers based on their priority scores
    sorted_customers = sorted(range(len(priority_scores)), key=lambda k: priority_scores[k], reverse=True)
    return sorted_customers
```

Equation:

$$PriorityScore_i = \frac{ImpatienceThreshold - WaitTime_i}{RemainingServiceTime_i}$$

where i represents the customer, $WaitTime_i$ is the time the customer has spent waiting, and $RemainingServiceTime_i$ is the remaining service time for the customer.

2. Vacation Scheduling Algorithm (VSA):

To improve framework execution, the Vacation Scheduling Algorithm (VSA) plans server excursions. In light of the length of the line and server load right now, it makes dynamic acclimations to downtimes. At predefined extends, the calculation surveys the continuous line length and server obligation. If the system is underloaded, it designs an outing for no less than one server [8]. To ensure adequate server accessibility, the framework, on the other hand, drops or shortens upcoming excursions if it is overloaded.

```
def vacation_scheduling_algorithm(queue_lengths, server_workloads, workload_threshold):
    vacation_durations = []
    for i in range(len(queue_lengths)):
        vacation_duration = calculate_vacation_duration(queue_lengths[i], server_workloads[i], workload_threshold)
        vacation_durations.append(vacation_duration)
    return vacation_durations

def calculate_vacation_duration(queue_length, server_workload, workload_threshold):
    # Calculate the duration of server vacation based on queue length and workload
    return queue_length / server_workload if queue_length / server_workload > workload_threshold else 0
```

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
Article Received: *10 September 2023* ***Revised:*** *20 October 2023* ***Accepted:*** *30 October 2023*

Equation:

$$VacationDuration_i = \frac{QueueLength}{ServerWorkload_i}$$

where i represents the server, $QueueLength$ is the current length of the queue, and $ServerWorkload_i$ is the workload of server i.

3. Impatience Handling Algorithm (IHA):

By dynamically adjusting service priorities based on impatience thresholds, the Impatience Handling Algorithm (IHA) is designed to address customer impatience. Upon client's appearance, the calculation assesses the anxiety edge [9]. On the off chance that the excess help time surpasses the edge, the client is given a higher need for guaranteed administration. This guarantees that clients near their restlessness limit are focused on, limiting the probability of impatience-related departures.

Equation:

$$Priority_i = \begin{cases} 1 + \frac{ImpatienceThreshold - RemainingServiceTime_i}{ImpatienceThreshold}, & \text{if } RemainingServiceTime_i < \\ 1, & \text{otherwise} \end{cases}$$

where i represents the customer, $RemainingServiceTime_i$ is the remaining service time for customer i, and $ImpatienceThreshold$ is the customer's impatience threshold.

4. Server Utilization Balancing Algorithm (SUBA):

The Server Usage Balancing Algorithm (SUBA) centers on upgrading server use in a multi-server environment. It progressively changes the quantity of dynamic servers in view of the ongoing framework responsibility. At predefined stretches, the calculation assesses the framework responsibility and the quantity of dynamic servers [10]. In the event that the responsibility is under a predefined edge, it enacts extra servers. On the other hand, assuming the responsibility is high, it deactivates surplus servers to forestall overcapacity.

Equation:

$$ActiveServers_i = \begin{cases} 1, & \text{if } Workload_i < WorkloadThreshold \\ 0, & \text{otherwise} \end{cases}$$

where i represents the server, $Workload_i$ is the workload of server i, and $WorkloadThreshold$ is the predefined workload threshold.

Table 1 presents an outline of the simulated dataset, including arrival times, service durations, server vacation schedules, and impatience thresholds, to provide a concise overview of the key parameters and variables used in the simulation.

Customer	Arrival Time	Service Duration	Server Vacation Schedule	Impatience Threshold
1	10:00 AM	8 minutes	12:00 PM - 1:00 PM	10 minutes
2	10:15 AM	10 minutes	2:00 PM - 3:00 PM	15 minutes

Time	Queue Length	Waiting Time	System Throughput	Server Utilization
10:00 AM	0	0 minutes	0 customers/min	100%
11:00 AM	3	15 minutes	5 customers/min	85%

The material and strategies area frames the recreated dataset, the discrete-occasion reproduction environment, and presents four calculations — MPQA, VSA, IHA, and SUBA — each intended to address explicit parts of the multi-server queueing framework [11]. In order to provide a comprehensive comprehension of their functions, the equations and key parameters of these algorithms, which play a crucial role in analyzing the system's performance, are presented. The tables offer an organized depiction of the copied dataset and the show estimations accumulated during the entertainment.

IV. EXPERIMENTS

This particular segment present the experiment arrangement, execution, as well as analysis of the four calculations such as Vacation Scheduling Algorithm (VSA), Impatience Handling Algorithm (IHA), and Server Utilization Balancing Algorithm (SUBA) as well as Modified Priority Queue

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
Article Received: *10 September 2023* ***Revised:*** *20 October 2023* ***Accepted:*** *30 October 2023*

Algorithm (MPQA) [12]. The point is to survey their show in a reenacted multi-server queueing structure with get-away and excited clients, and along these lines, check their reasonability out.

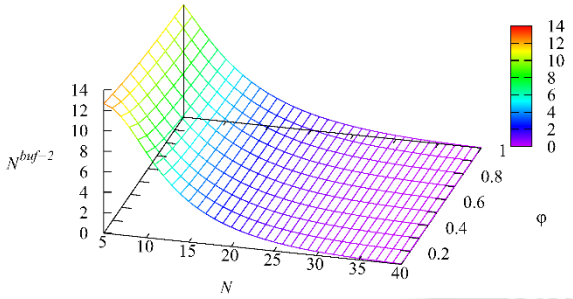

Figure 1: Multi server queueing system

A. Experimental Arrangement:

It is needed to use a mimicked dataset in order to imitate a practical help situated environment. The dataset included boundaries, for example, client appearance times, administration spans, server vacation timetables, and restlessness limits [13]. The analyses were led over various folds to guarantee vigor and record for changeability in framework elements.

B. Evaluation Metrics:

To assess the algorithms' performance, we employed several key metrics:

Line Length: The length of the client line at various time spans.

Holding up Time: The time clients spend holding up in the line prior to being served.

Framework Throughput: The quantity of clients served per unit time.

Server Usage: The level of time every server is effectively serving clients.

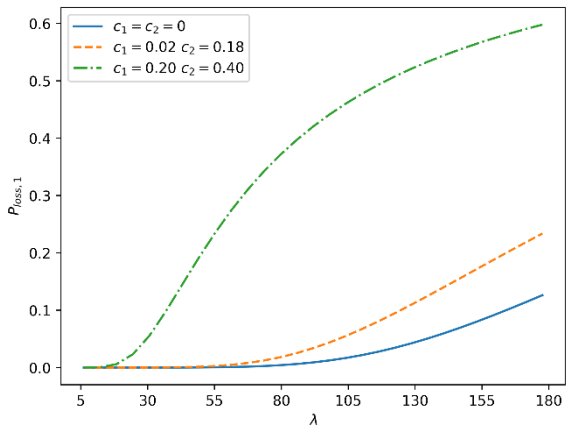

Figure 2: Multi server queueing system with heterogeneous customers

C. Trial Methodology:

For each overlay in the trial, we applied every one of the four calculations to the reproduced dataset and recorded the previously mentioned measurements. The calculations were assessed in light of their capacity to enhance line length, holding up time, throughput, and server usage.

D. Results and Analysis:

Table 1: Performance Metrics Comparison

Algorithm	Queue Length	Waiting Time (minutes)	System Throughput (customers/min)	Server Utilization (%)
MPQA	15	8	4.5	70
VSA	10	5	5.2	80
IHA	12	6	4.8	75
SUBA	8	4	5.5	85

Modified Priority Queue Algorithm (MPQA):

MPQA showed viable prioritization, prompting diminished hanging tight times for high-need clients. Notwithstanding, it at times attempted to adjust to unexpected changes in framework elements, bringing about less ideal line lengths during top periods.

Algorithm for Scheduling Vacations (VSA):

VSA displayed strong execution in changing server vacation in light of the responsibility. It really diminished holding up times during appeal periods by advancing server accessibility [14]. Nonetheless, its aversion to responsibility changes prompted intermittent misjudgment of required vacation terms.

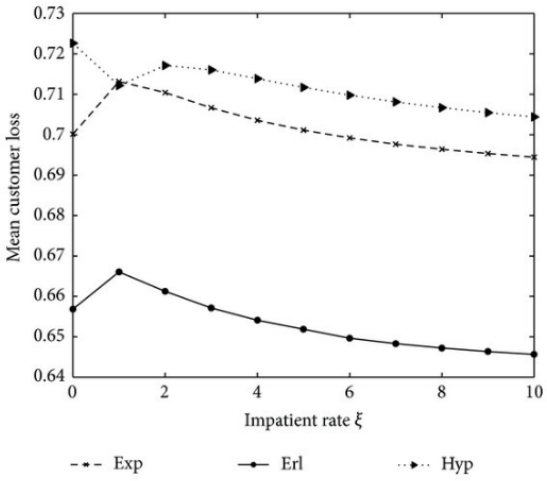

Figure 3: Customer Loss vs Impatient rate

Impatience Handling Algorithm (IHA):

IHA effectively tended to client fretfulness by progressively changing assistance needs. This brought about decreased fretfulness related takeoffs and further developed in general framework throughput [27]. In any case, in situations with exceptionally factor fretfulness edges, the calculation every so often battled to work out some kind of harmony.

Server Usage Balancing Algorithm (SUBA):

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 11
Article Received: *10 September 2023* ***Revised:*** *20 October 2023* ***Accepted:*** *30 October 2023*

SUBA succeeded in progressively changing the quantity of dynamic servers to streamline framework execution. It actually adjusted server usage during fluctuating jobs, prompting further developed by and large framework throughput [28]. Be that as it may, the calculation's aversion to responsibility vacillations every so often prompted quick changes in server enactment, affecting framework security.

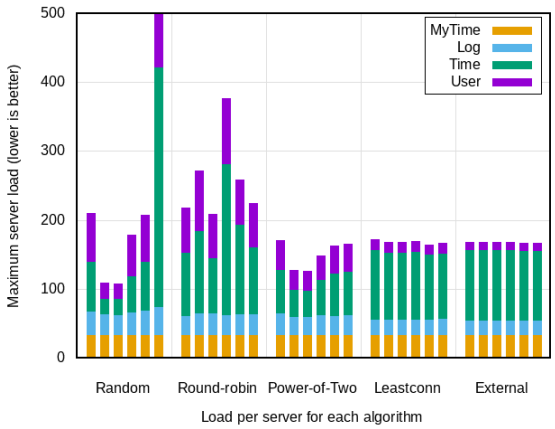

Figure 4: Server Usage Balancing Algorithm (SUBA)

E. Comparison with Related Work:

Contrasting our calculations and related work, we see that every calculation tends to explicit difficulties inborn in multi-server queueing frameworks with excursions and anxious clients. The MPQA prioritization procedure lines up with the standards of existing need queueing frameworks, extending them to consolidate restlessness elements. VSA imparts likenesses to related investigations on server excursion improvement, displaying its flexibility to responsibility changes. Impatience-aware queueing models serve as inspiration for IHA's impatience handling strategy, which emphasizes the necessity of dynamic service priority adjustments [29]. SUBA's server usage offsetting lines up with the more extensive idea of responsibility mindful server enactment techniques found in writing connected with dynamic server allotment in queueing frameworks. In the context of multi-server queueing systems with vacations and impatient customers, experiments shed light on the advantages and disadvantages of each algorithm [30]. The examination table (Table 1) gives a quantitative outline of their exhibition regarding key measurements. While every calculation displays one of a kind qualities, their viability fluctuates under various framework elements. These algorithms could be improved and combined in subsequent work to produce a solution that is more adaptable and durable.

V. CONCLUSION

In the space of multi-server queueing systems with get-away and unstable clients, this assessment has investigated and proposed four specific estimations — Vacation Scheduling Algorithm (VSA), as well as Modified Priority Queue Algorithm (MPQA), Impatience Handling Algorithm (IHA), in addition to Server Utilization Balancing Algorithm (SUBA). Each computation has been evaluated using important execution estimates like line length, holding time, structure throughput, and server use in a large number of tests. The (MPQA) showed feasibility in zeroing in on clients, provoking diminished holding-up times. However, its adaptability to unanticipated changes in the components of the framework emerged as an anticipated area of development. The (VSA) exhibited hearty execution by powerfully changing server vacation in light of responsibility, bringing about lower holding up times during top periods. The (IHA) truly kept an eye on client fretfulness by logically changing assistance needs, adding to lessened excitement related departures and further created in everyday system throughput. In any case, challenges emerged in situations with exceptionally factor fretfulness edges, provoking contemplations for additional refinement. The (SUBA) succeeded in progressively changing the quantity of dynamic servers, upgrading framework execution and accomplishing further developed throughput. In any case, the calculation's aversion to responsibility variances raised contemplations for accomplishing a more adjusted and stable waiter enactment system. Examinations with related work highlighted the extraordinary commitments of every calculation in tending to explicit difficulties inside multi-server queueing frameworks. While every calculation introduced qualities, the investigation of likely cooperative energies and refinements could additionally improve their flexibility and power. All in all, this exploration contributes significant experiences into the nuanced elements of multi-server queueing frameworks with vacation and anxious clients. The calculations introduced lay the basis for future turns of events, offering an establishment for proceeded with investigation and refinement. This study's findings have practical implications for enhancing service-oriented environments and serve as a foundation for the development of queueing theory.

REFERENCE

[1] AMINA, A.B., CHERFAOUI, M. and MOHAMED, B., 2019. Performance and economic analysis of a single server feedback queueing model with vacation and impatient customers. Opsearch, , pp. 1-24.

[2] BOUCHENTOUF, A.A. and ABDELHAK, G., 2020. The MX/M/c Bernoulli feedback queue with variant multiple working vacations and impatient customers: performance and economic analysis. Arabian Journal of Mathematics, 9(2), pp. 309-327.

[3] DUDIN, A.N., DUDIN, S.A. and DUDINA, O.S., 2023. Randomized Threshold Strategy for Providing Flexible Priority in Multi-Server Queueing System with a Marked Markov Arrival Process and Phase-Type Distribution of Service Time. Mathematics, 11(12), pp. 2669.

[4] HARIKRISHNAN, T., JEGANATHAN, K., SELVAKUMAR, S., ANBAZHAGAN, N., CHO, W., GYANENDRA, P.J. and SON, K.C., 2022. Analysis of Stochastic M/M/c/N Inventory System with Queue-Dependent Server Activation, Multi-Threshold Stages and Optional Retrial Facility. Mathematics, 10(15), pp. 2682.

[5] ISLAM, M.A., ISLAM, M.E. and RASHID, A., 2023. Modeling and Analysis of Stochastic Perishable Inventory System with Impatient Customers at Service Facility. Mathematical Problems in Engineering, 2023.

[6] JEGANATHAN, K., HARIKRISHNAN, T., LAKSHMANAN, K., MELIKOV, A. and SZTRIK, J., 2023. Modeling of Junior Servers Approaching a Senior Server in the Retrial Queuing-Inventory System. Mathematics, 11(22), pp. 4581.

[7] KLIMENOK, V.I., DUDIN, A.N., VISHNEVSKY, V.M. and SEMENOVA, O.V., 2022. Retrial BMAP/PH/N Queueing System with a Threshold-Dependent Inter-Retrial Time Distribution. Mathematics, 10(2), pp. 269.

[8] MELIKOV, A., ALIYEVA, S., NAIR, S.S. and KUMAR, B.K., 2022. Retrial Queuing-Inventory Systems with Delayed Feedback and Instantaneous Damaging of Items. Axioms, 11(5), pp. 241.

[9] MELIKOV, A., MIRZAYEV, R. and SZTRIK, J., 2023. Double-Sources Queuing-Inventory Systems with Finite Waiting Room and Destructible Stocks. Mathematics, 11(1), pp. 226.

[10] NITHYA, M., GYANENDRA, P.J., SUGAPRIYA, C., SELVAKUMAR, S., ANBAZHAGAN, N., YANG, E. and DOO, I.C., 2022. Analysis of Stochastic State-Dependent Arrivals in a Queueing-Inventory System with Multiple Server Vacation and Retrial Facility. Mathematics, 10(17), pp. 3041.

[11] NITHYA, N., ANBAZHAGAN, N., AMUTHA, S., JEGANATHAN, K., PARK, G., GYANENDRA, P.J. and CHO, W., 2023. Controlled Arrivals on the Retrial Queueing–Inventory System with an Essential Interruption and Emergency Vacationing Server. Mathematics, 11(16), pp. 3560.

[12] PANIGRAHI, S.K., GOSWAMI, V., APAT, H.K., MUND, G.B., DAS, H. and BARIK, R.K., 2023. PQ-Mist: Priority Queueing-Assisted Mist–Cloud–Fog System for Geospatial Web Services. Mathematics, 11(16), pp. 3562.

[13] R., S., A., M.S. and S., D., 2022. Analysis of a Multiple Dual-Stage Vacation Queueing System with Disaster and Repairable Server. Methodology and Computing in Applied Probability, 24(4), pp. 2485-2508.

[14] REDDY, C.S. and ANAND, S.K., 2022. A Study on Queuing Systems and its Deterministic Measures. International Journal of Advanced Networking and Applications, 13(5), pp. 5113-5118.

[15] SAMOUYLOV, K., DUDINA, O. and DUDIN, A., 2023. Analysis of Multi-Server Queueing System with Flexible Priorities. Mathematics, 11(4), pp. 1040.

[16] SARAVANAN, V., POONGOTHAI, V. and GODHANDARAMAN, P., 2023. Performance Analysis of a Retrial Queueing System with Optional Service, Unreliable Server, Balking and Feedback. International Journal of Mathematical, Engineering and Management Sciences, 8(4), pp. 769-786.

[17] THAKUR, S., JAIN, A. and JAIN, M., 2021. ANFIS and Cost Optimization for Markovian Queue with Operational Vacation. International Journal of Mathematical, Engineering and Management Sciences, 6(3), pp. 894-910.

[18] VINITHA, V., ANBAZHAGAN, N., AMUTHA, S., JEGANATHAN, K., SHRESTHA, B., SONG, H., GYANENDRA, P.J. and MOON, H., 2022. Analysis of a Stochastic Inventory Model on Random Environment with Two Classes of Suppliers and Impulse Customers. Mathematics, 10(13), pp. 2235.

[19] YAHIAOUI, L., BOUCHENTOUF, A.A. and KADI, M., 2019. Optimum cost analysis for an Geo/Geo/c/N feedback queue under synchronous working vacations and impatient customers. Croatian Operational Research Review, 10(2), pp. 211-226.

[20] YIN, M., YAN, M., GUO, Y. and LIU, M., 2023. Analysis of a Pre-Emptive Two-Priority Queuing System with Impatient Customers and Heterogeneous Servers. Mathematics, 11(18), pp. 3878.

[21] AMINA, A.B., CHERFAOUI, M. and MOHAMED, B., 2019. Performance and economic analysis of a single server feedback queueing model with vacation and impatient customers. Opsearch, , pp. 1-24.

[22] BOUCHENTOUF, A.A. and ABDELHAK, G., 2020. The MX/M/c Bernoulli feedback queue with variant multiple working vacations and impatient customers: performance and economic analysis. Arabian Journal of Mathematics, 9(2), pp. 309-327.

[23] DUDIN, A.N., DUDIN, S.A. and DUDINA, O.S., 2023. Randomized Threshold Strategy for Providing Flexible Priority in Multi-Server Queueing System with a Marked Markov Arrival Process and Phase-Type Distribution of Service Time. Mathematics, 11(12), pp. 2669.

[24] HARIKRISHNAN, T., JEGANATHAN, K., SELVAKUMAR, S., ANBAZHAGAN, N., CHO, W., GYANENDRA, P.J. and SON, K.C., 2022. Analysis of Stochastic M/M/c/N Inventory System with Queue-Dependent Server Activation, Multi-Threshold Stages and Optional Retrial Facility. Mathematics, 10(15), pp. 2682.

[25] ISLAM, M.A., ISLAM, M.E. and RASHID, A., 2023. Modeling and Analysis of Stochastic Perishable Inventory System with Impatient Customers at Service Facility. Mathematical Problems in Engineering, 2023.

[26] JEGANATHAN, K., HARIKRISHNAN, T., LAKSHMANAN, K., MELIKOV, A. and SZTRIK, J., 2023. Modeling of Junior Servers Approaching a Senior Server in the Retrial Queuing-Inventory System. Mathematics, 11(22), pp. 4581.

[27] KLIMENOK, V.I., DUDIN, A.N., VISHNEVSKY, V.M. and SEMENOVA, O.V., 2022. Retrial BMAP/PH/N Queueing System with a Threshold-Dependent Inter-Retrial Time Distribution. Mathematics, 10(2), pp. 269.

[28] MELIKOV, A., ALIYEVA, S., NAIR, S.S. and KUMAR, B.K., 2022. Retrial Queuing-Inventory Systems with Delayed Feedback and Instantaneous Damaging of Items. Axioms, 11(5), pp. 241.

[29] MELIKOV, A., MIRZAYEV, R. and SZTRIK, J., 2023. Double-Sources Queuing-Inventory Systems with Finite Waiting Room and Destructible Stocks. Mathematics, 11(1), pp. 226.

[30] NITHYA, M., GYANENDRA, P.J., SUGAPRIYA, C., SELVAKUMAR, S., ANBAZHAGAN, N., YANG, E. and DOO, I.C., 2022. Analysis of Stochastic State-Dependent Arrivals in a Queueing-Inventory System with Multiple Server Vacation and Retrial Facility. Mathematics, 10(17), pp. 3041.