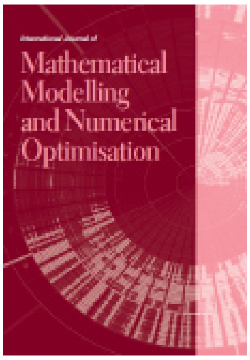


International Journal of Mathematical Modelling and Numerical Optimisation

ISSN online: 2040-3615 - ISSN print: 2040-3607
https://www.inderscience.com/ijmmno

Optimising communication and performance in IoT with RabbitMQ: a bulk arrival single server retrial queueing model with multi-phase service

Vivesini Devaraj, J. Ebenesar Anna Bagyam, T. Poongodi

Article History:

Received:	04 October 2023
Last revised:	05 October 2023
Accepted:	07 October 2023
Published online:	08 April 2025

Optimising communication and performance in IoT with RabbitMQ: a bulk arrival single server retrial queueing model with multi-phase service

Vivesini Devaraj*

Department of Mathematics,
Avinashilingam Institute for Home
Science and Higher Education for Women,
Coimbatore, Tamil Nadu–641043, India
Email: viveshpriya@gmail.com

J. Ebenesar Anna Bagyam

Department of Mathematics,
Karpagam Academy of Higher Education,
Coimbatore, Tamil Nadu – 641043, India
Email: ebenesar.j@gmail.com

T. Poongodi

Department of Mathematics,
Avinashilingam Institute for Home
Science and Higher Education for Women,
Coimbatore, Tamil Nadu – 641043, India
Email: poongodi.math@gmail.com

Abstract: RabbitMQ is an advanced messaging technology in which the communication layer interfaces with a message broker and a message queue. This system has opted for a specific message queue, enabling asynchronous communication by implementing message-oriented middleware. This approach encourages applications to be loosely connected, potentially enhancing system performance and optimising resource utilisation. In this study, RabbitMQ serves as an intermediary for sending and receiving user messages. Queueing models can be employed to determine the necessary resources for providing high-quality service. For RabbitMQ (IoT) based on the computing system, we propose a bulk arrival single server retrial queueing model with a mandatory multi-phase second service phase for theoretical modelling. We solve steady-state equations and derive various performance metrics. Subsequently, we develop a cost model. To compute the total expected cost of the service rate, we utilise the particle swarm optimisation algorithm. Finally, we perform a numerical analysis on our model to gain insights into its behaviour.

Keywords: cloud computing; bulk arrival; retrial queue; two phase service; RabbitMQ; particle swarm optimisation; PSO.

Reference to this paper should be made as follows: Devaraj, V., Bagyam, J.E.A. and Poongodi, T. (2025) 'Optimising communication and performance in IoT with RabbitMQ: a bulk arrival single server retrial queueing model with multi-phase service', *Int. J. Mathematical Modelling and Numerical Optimisation*, Vol. 15, No. 1, pp.6–26.

Biographical notes: Vivesini Devaraj is a Research Fellow in the Department of Mathematics at the Avinashilingam Institute for Home Science and Higher Education for Women, located in Coimbatore. Her research interests lie in cloud computing and queueing theory. She has presented research papers on queueing theory and cloud computation, both based on queueing theory, at various international conferences. As a burgeoning researcher, her recent works focus on applications that yield results and ideas with potential benefits for society. Simultaneously, she is dedicated to learning and cultivating new ideas.

J. Ebenesar Anna Bagyam is currently serving as an Assistant Professor at the Karpagam Academy of Higher Education in Coimbatore. With over 15 years of experience, she possesses a strong background in both teaching and research. Her research interests encompass queueing theory, fuzzy queues, optimisation techniques, and machine learning. She has an impressive publication record, with more than 25 papers published in esteemed national and international journals. Additionally, she has presented her research findings at various international conferences. A life member of IMS and ISCA, she actively engages in guiding research scholars in the domains of queueing theory, machine learning, and optimisation techniques.

T. Poongodi is currently serving as an Assistant Professor at the Avinashilingam Institute for Home Science and Higher Education for Women in Coimbatore. With over 23 years of experience, she possesses a strong background in both teaching and research. Her research interests encompass queueing theory, fuzzy control chart and machine learning. She has an impressive publication record, with more than 12 papers published in esteemed national and international journals. Additionally, she has presented her research findings at various international conferences. A life member of ISTE and Indian Science Congress, she actively engages in guiding research scholars in the domains of queueing theory, machine learning, and optimisation techniques.

1 Introduction

RabbitMQ is portable and easy to set up both locally and in the cloud. It supports various messaging protocols and can be deployed in distributed and federated configurations to meet scalability and availability demands. Initially designed to facilitate communication between processes within an operating system, RabbitMQ's message queues are now widely used across various application domains. In distributed environments, RabbitMQ connects different system components, enabling asynchronous data exchange. The message queue of RabbitMQ is compatible with open-source RabbitMQ clients.

Sadooghi et al. (2014) considered message queues in cloud and high-performance computing environments, emphasising effective work scheduling systems for achieving high throughput and system utilisation.

Piorkowski and Werewka (2010) analysed the performance of a scheduling algorithm that prioritises sending the shortest messages first, a strategy deemed suitable for the proposed model.

Sachs et al. (2013) employed a comprehensive modelling methodology to analyse message-oriented, event-driven systems using a supply chain management case study. The case study was evaluated using the SPECjms2007 standard benchmark on a leading commercial middleware platform.

Vilaplana et al. (2014) created a model for cloud computing infrastructures with quality of service (QoS) descriptions. They utilised open Jackson's networks and queueing theory to ensure specific performance levels in terms of queuing and response times.

Rao et al. (2017) investigated single and multi-server systems within cloud data centres based on queueing systems. They evaluated different measures using MATLAB for the proposed system.

Vaquero et al. (2008) conducted a comprehensive analysis of the cloud computing concept to establish a complete definition of cloud computing.

Tang (2022) examined edge computing systems powered by the internet of things, focusing on a finite-capacity single-server queueing model with bulk arrivals and steady-state probabilities.

Lazidis et al. (2022) provided an in-depth analysis of design elements, technology, and taxonomy in Publish-Subscribe systems for cloud and IoT applications, covering publish-subscribe protocols and message queuing.

Rajadurai et al. (2015) studied a batch arrival feedback retrial queueing system with two service phases, incorporating bad customers, breakdowns, repairs, and a BVS. The model's applications span various domains, including SMTP, computer networking, manufacturing, and communication systems.

Wang and Zhou (2010) researched a batch arrival retrial queue with initial failures, feedback, and admission control, deriving performance measures in a steady state.

Sangeetha and Udaya Chandrika (2022) analysed the M[x]/G/1 retry G-queue with feedback, randomised J vacation, and orbital search, investigating server availability, failure frequency, and parameter effects.

Maragathasundari et al. (2022) addressed queueing issues on a WhatsApp server, using supplementary variable techniques to solve performance measures. They explored an M[x]/G/1 retrial queueing model with mandatory and optional service phases, probabilistic feedback, and admission based on server status.

Abdollahi and Salehi Rad (2022) examined a batch arrival retrial queueing model with M stages of service, verifying the Bernoulli vacation stochastic decomposition law.

Nila and Sumitha (2022) are analysed complex queueing model involving batch arrival retrial queues with impatient customers, Bernoulli vacation policies, feedback mechanisms, and server failures. The server provides two phases of service: essential phase service (FEPS) and optional phase service (SOPS). Customers may choose to balk or renege at specific times. After SOPS completion, the server can either take a vacation or remain idle with a certain probability. Additionally, customers can provide feedback after both service phases. The paper derives steady-state solutions and investigates the impact of system parameters through numerical analysis.

Singh et al. (2011) studied a realistic bulk arrival queueing model, determining queue length distribution with a second optional service. They provided an explicit formula for queue size distribution by maximising Shannon's entropy function.

Bagyam et al. (2015) considered a bulk arrival retrial queueing model with M stages of service, validating the Bernoulli vacation stochastic decomposition law.

Upadhyaya (2015) analysed a bulk retrial queueing system with Bernoulli feedback under Bernoulli vacation.

Micheal and Indhira (2023) are analysed delves into the intrinsic retrial phenomenon in various queueing systems, addressing a notable gap concerning vacation periods. Focusing on retrial queueing systems with Bernoulli vacations, it offers a detailed analysis and reviews existing literature. The application scope spans computer networking, manufacturing, inventory management, and service sectors. This study serves as a valuable resource for researchers, administrators, and practitioners in congestion simulation through queuing theory, offering insights into appropriate models and highlighting potential future research directions.

Ghimire et al. (2014) investigated a bulk queueing model with a fixed batch size ‘*b*’, obtaining numerical illustrations using MATLAB.

Begum and Choudhury (2021) analysed a single-server queueing system with unreliable bulk arrivals and two heterogeneous optional repeated services (THORS) incorporating delayed repair, all operating under a Bernoulli vacation schedule (BVS) and an *N*-policy. They delve into the complexities of this model by deriving the joint distribution of the server’s state and queue length, considering both elapsed and remaining times.

Jain et al. (2013) explored queuing and reliability indices of an unreliable server M/G/1 queue under realistic assumptions, including various service, maintenance, and vacation choices.

Choudhury et al. (2010) examined the steady-state behaviour of an MX/G/1 retrial queue with a second optional service and interruptions due to server breakdowns.

Arivudainambi and Godhandaraman (2012) analysed a single server retrial queueing system with repeated tries, batch arrivals, two service phases, feedback, and K optional vacations.

Jain et al. (2012) investigated the M/G/1 retrial queue with a second optional service phase, server breakdowns, and various additional concepts.

Kumar and Arumuganathan (2013) studied the stability and reliability of a single server batch arrival retrial queue with active breakdowns, second optional service, and impatient behaviour.

Varalakshmi et al. (2016) analysed a single server retrial queueing system with two service phases, immediate Bernoulli feedback, a single vacation, and starting failures, providing numerical results.

Bouchentout and Guendouzi (2019) analysed the MX/M/C Bernoulli feedback queuing system, obtaining performance measures using steady state equations and cost values through the quartic fit method.

Yang and Chen (2018) observed the M/M/1 queue with a second optional service and working breakdown, calculating various system performance measures and using the Genetic algorithm to find total cost for service rates.

Upadhyaya et al (2023) analysed the system with retrials, assessing various performance measures and reliability indices. The results are presented visually through tables and graphs, offering a comprehensive view of the system’s behaviour. Additionally, a cost analysis of the model is conducted, aiming to find the optimal cost for the system. This optimisation process is achieved using the particle swarm optimisation (PSO) algorithm for the matrix method.

Sanga and Charan (2023) observed the study introduces the parametric non-linear programming (PNLP) technique to adapt the MRP to a fuzzy environment, utilising trapezoidal and sigmoidal fuzzy numbers. Leveraging Zadeh's extension principle and PNLP, they extract the α-cuts of the developed performance predictors, aiding in cost analysis using PSO algorithm.

Li (2022) analysed the significant impact of electrical engineering on people's lives, emphasising the role of the PSO algorithm in enhancing convenience and living standards. It mentions how advanced electrical engineering automation, aided by the PSO algorithm, reduces worker workload in fields like subway engineering, ensuring efficient project execution. The PSO algorithm is commended for improving automation performance and enhancing operational efficiency in electrical engineering. This study delves into the concept and workflow of the PSO algorithm, providing insights into the intelligent principles and internal mechanisms of electrical engineering. In summary, the integration of PSO algorithm with electrical engineering greatly enhances automation intelligence and efficiency.

Roy and Das (2021) are analysed the significance of DSM programs in smart grids and their benefits to customers. The paper proposes a load shifting approach based on a hierarchical smart grid structure, integrating renewable and conventional energy sources. The multi-objective problem of ensuring benefits to all participants of the electricity market is addressed using a hybrid GA-PSO algorithm, optimising cost and load allocation for a day-ahead market. The fusion factor is introduced to balance exploration and exploitation, improving the algorithm's performance. The simulation results demonstrate the effectiveness of the proposed approach in reducing the PAR, improving smart grid efficiency, and providing financial benefits to all participants. The comparison with existing optimisation techniques confirms its superior performance.

Martinez-Rico et al. (2020) have analysed the key role of batteries in compensating for the intermittent nature of renewable energy sources and increasing the benefits of renewable power plants. Energy arbitrage, i.e., storing energy at low electricity price moments and selling it when electricity price is high, is identified as a promising strategy. A hybrid renewable energy system consisting of wind and solar power with batteries is studied, and an optimisation process is conducted to maximise profits using a multi-objective cost function. Results show the importance of including the energy efficiency of the battery in the cost function to optimise profits and extend battery lifetime. However, low profits are observed in the analysed Iberian electricity market when batteries are used solely for arbitrage purposes using PSO algorithm.

Shekhar et al. (2020) analysed a single server with emergency vacation, comparing bat algorithm, Particle Swarm algorithm, and Quasi-Newton method for cost optimisation.

Bagyam et al. (2023) are analysed RabbitMQ emerges as a formidable messaging software, boasting a robust communication layer leading to efficient message brokerage and queue management. Widely adopted as a middleman for message exchange, its integration with cloud computing enhances its theoretical underpinnings through queueing models. Particularly suitable for the single-server general service retrial queueing system with a mandatory two-phase service model, RabbitMQ optimises system activity through evaluated performance measures. The study culminates in a thorough comparative analysis, shedding light on both computational and simulated assessments of the devised metrics.

The proposed system selected RabbitMQ as an application, solving it theoretically using supplementary variable techniques and experimentally with MATLAB software.

2 Proposed system

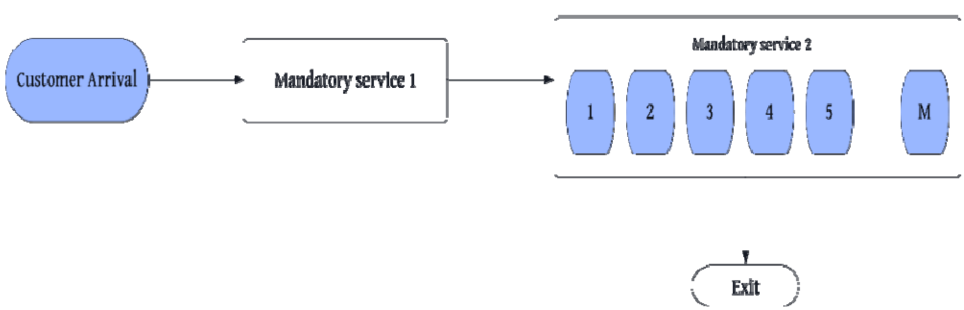
Contemporary event-driven systems often employ RabbitMQ, a messaging queue technology, to facilitate loosely connected communication among distributed software components. This method involves asynchronous message passing rather than direct requests. Acting as an intermediary broker, RabbitMQ receives messages from multiple producers and delivers them to potential consumers.

RabbitMQ offers four exchange types: direct, topic, fanout, and headers. These routes ensure targeted information delivery. Users can either create custom exchanges or utilise predefined defaults. Prominent companies such as Verizon, JP Morgan Chase, XPO logistics, and travellers have integrated RabbitMQ into their operations.

Yang validator also uses RabbitMQ and Nameko to establish a process for validating JSON messages against open config standards. The job scheduling process within the message queue resembles a queue, requiring users to wait their turn for service. Messaging queues find applicability across diverse use cases. They play a pivotal role in 'Multi-Point Broadcast' applications, enabling messages to be broadcast to multiple destinations. RabbitMQ functions as a messaging broker, facilitating message transmission and retrieval, offering a secure storage space for messages.

A message queue serves as a sequential service-to-service transmission method commonly used in serverless and micro-services architectures. Messages wait in a queue until they are processed. This paper introduces a model featuring a bulk arrival single server queue with a mandatory second multi-phase service. This model encompasses various application scenarios, similar to the previously mentioned RabbitMQ model.

Figure 1 Structural outline of RabbitMQ (see online version for colours)



3 Description of queueing model for Rabbit MQ

Consider a single server retrial queueing system with two phase service. Customer arrives in batches according to Poisson process with rate α. Let Y be the batch size which is a random variable with $P(Y = K) = C_K$, $K = 1,2,3\ldots$, $\sum_{K=1}^{\infty} C_K = 1$ and $C(z)$ is the probability generating function (PGF) with first two moments B_1 and B_2.

Blocked customers enter the retrial group termed as orbit. The retrial time is generally distributed with distribution function $A(s)$, density function $a(s)$, Laplace Stieltje's transform $A^*(\alpha)$ and the and conditional completion rate

$$\tau(x) = \frac{a(x)}{1 - A(x)}$$

The arriving customer takes service immediately when the server is in ideal state. The first service will serve for the entire arriving customer and it is generally distributed with distribution function $B_1(x)$, density function $b_1(x)$, Laplace transform $B_1^*(x)$, two moments b_1, b_2. As soon as the first service is completed, the customer will automatically perform the compulsory multi-service with probability (σ_m). The compulsory second phase service time follows an arbitrary distribution with distribution function $B_m(x)$, density function $b_m(x)$, Laplace transform $B_m^*(x)$, two moments $\overline{c}_1$ and $\overline{c}_2$ and conditional complete rate are

$$\varepsilon_1 = \frac{b_1(x)}{1 - B_1(x)} \; \varepsilon_m = \frac{b_m(x)}{1 - B_m(x)}$$

After the second service, the customer will leave the entire system.

3.1 Steady – state distributions

The system of equations that governs the model under steady state, by supplementary variable method are

$$\alpha R_0^F = \int_0^{\infty} \sum_{m=1}^{k} R_{m,0}^Q(x)\varepsilon_m(x)dx \tag{1}$$

$$\frac{dR_n^F(x)}{dx} = -(\alpha + \tau(x))R_n^F(x) \quad n \geq 1 \tag{2}$$

$$\frac{dR_0^E(x)}{dx} = -(\alpha + \varepsilon_1(x))R_0^E(x) \tag{3}$$

$$\frac{dR_n^E(x)}{dx} = --\left(\alpha + \varepsilon_1(x)\right) R_n^E(x) \sum_{k=1}^{n} c_k \alpha R_{n-1}^E(x) +, \; n \geq 1 \tag{4}$$

$$\frac{dR_{m,0}^Q(x)}{dx} = -(\alpha + \varepsilon_m(x))R_{m,0}^Q(x) \quad m = 1,2,\ldots k \tag{5}$$

$$\frac{dR_{m,n}^Q(x)}{dx} = --\left(\alpha + \varepsilon_m(x)\right) R_{m,n}^Q(x) \sum_{k=1}^{n} c_k \alpha R_{m,n-1}^Q, \; n \geq 1, \tag{6}$$

Boundary conditions

$$R_0^F(0) = \int_0^\infty \sum_{m=1}^{k} R_{m,n}^Q(x)\varepsilon_m(x)dx, \quad n \geq 1 \tag{7}$$

$$R_0^E(0) = c_1\alpha R_0^F + \int_0^\infty R_1^F(x)\tau(x)dx \quad m = 1, 2\ldots k \tag{8}$$

$$R_n^E(0) = \sum_{k=1}^{n} c_k\alpha \int_0^\infty R_n^F(x) + \int_0^\infty R_{n+1}^E(x)\tau(x)dx + c_{n+1}\alpha R_0^F \quad n \geq 1 \tag{9}$$

$$R_{m,n}^Q(0) = \sigma_m \int_0^\infty R_n^E(x)\varepsilon_1(x)dx \qquad n \geq 0 \qquad m = 1, 2\ldots k \tag{10}$$

The normalising condition is

$$R_0^F + \sum_{n=1}^{\infty}\int_0^\infty R_n^F(x)dx + \sum_{n=0}^{\infty}\int_0^\infty R_n^E(x)dx + \sum_{n=0}^{\infty}\int_0^\infty R_{m,n}^Q(x)dx = 1$$

3.2 Steady state solution

We use the PGF corresponding to different states of the server to solve the set of supplementary variable methods, enabling us to obtain the steady-state solution of the retrial queuing model. We define the PGF corresponding to the various states as

$$R^F(x,z) = \sum_{n=1}^{\infty} R_n^F(x)z^n$$

$$R^E(x,z) = \sum_{n=0}^{\infty} R_n^E(x)z^n$$

$$R_m^Q(x,z) = \sum_{n=0}^{\infty} R_{m,n}^Q(x)z^n$$

By solving the partial differential equations from equation (2) to equation (6), we obtain

$$R^F(x,z) = R^F(0,z)e^{-\alpha x}\left(1 - A(x)\right) \tag{12}$$

$$R^E(0,z)e^{-\alpha(1-c(z))x}\left(1 - B_1(x)\right) = R^F(x,z) = R^F(0,z)e^{-\alpha x}(1 - A(x)) \tag{13}$$

$$R_m^Q(x,z) = R_m^Q(0,z)e^{-\alpha(1-c(z))x}\left(1 - B_m(x)\right) \tag{14}$$

By multiplying the steady-state equation and steady-state boundary conditions from equations (7) to (10) by z^n and summing over n, where $n = 0, 1, 2, \ldots$, we get

$$R^F(0,z) = \sum\nolimits_{m=1}^{k} R_m^Q(0,z)B_m^*(\alpha(1 - c(z)) - \alpha R_0^F \tag{15}$$

$$R^E(0,z) = \frac{\alpha R_0^F}{z} c(z) + + \frac{R^F(0,z)}{z}[c(z) + A^*(\alpha)(1 - c(z)] \tag{16}$$

$$R_m^Q(0,z) = \sum_{m=1}^{k} \sigma_m R^E(0,z) B_1^*(\alpha(1 - c(z))) \tag{17}$$

By substituting the expression provided in equation (15) into equation (16), we obtain

$$R^E(0,z) = \frac{\alpha R_0^F}{z} c(z) + \frac{\sum_{m=1}^{k} R_m^Q(0,z) B_m^*(\alpha(1 - c(z))) - \alpha R_0^F}{z}[c(z) + A^*(\alpha)(1 - c(z] \tag{18}$$

By utilising the equation $R_m^Q(0,z)$ provided in (17) within equation (18) and simplifying, we get

$$R^E(0,z) = \frac{\alpha R_0^F[(A^*(\alpha)(1 - c(z))]}{D(z)} \tag{19}$$

where

$$D(z) = \sum_{m=1}^{k} \sigma_m B_1^*\left(\alpha(1 - c(z))\right) B_m^*\left(\alpha(1 - c(z))\right)[c(z) + A^*(\alpha)(1 - c(z)] - z$$

By substituting the expression for $R^E(0, z)$ given in equation (19) into equation (17), we obtain

$$R_m^Q(0,z) = \frac{\sum_{m=1}^{k} \sigma_m \alpha R_0^F A^*(\alpha)(1 - c(z)) B_1^*(\alpha(1 - c(z)))}{D(z)} \tag{20}$$

By substituting the equation for $R_m^Q(0,z)$ provided in (20) into equation (15)

$$R^F(0,z) = \frac{\alpha R_0^F}{D(z)}\left[z - \sum_{m=1}^{k} \sigma_m B_1^*(\alpha(1 - c(z))) B_m^*(\alpha(1 - c(z)))\right] \tag{21}$$

By inserting the $R^F(0, z)$ equation provided in (21) into equation (12)

$$R^F(x,z) = \frac{\alpha R_0^F}{D(z)}\left[z - \sum_{m=1}^{k} c(z) \sigma_m B_1^*(\alpha(1 - c(z))) B_m^*(\alpha(1 - c(z)))\right] e^{-\alpha}\left(1 - A(x)\right)$$

Substituting the expression for $R^E(0, z)$, $R_m^Q(0,z)$ given into equations (19) and (20) in the equations (13) and (14), we get

$$R^E(x,z) = \frac{\alpha R_0^F[(A^*(\alpha)(1 - c(z))]}{D(z)} e^{-\alpha(1 - c(z))x}[1 - B_1(x)]$$

$$R_m^Q(x,z) = \frac{\sum_{m=1}^{k} \sigma_m \alpha R_0^F A^*(\alpha)(1-c(z)) B_1^*(\alpha(1-c(z)))}{D(z)} e^{-\alpha(1-c(z))x}[1-B_m(x)]$$

3.3 Steady state analysis

The PGF of the orbit size when the server is in idle is

$$R^F(z) = \int_0^\infty R^F(x,z)dx.$$

$$R^F(z) = \frac{R_0^F}{D(z)}\left[z - \sum_{m=1}^{k} \sigma_m B_1^*\left(\alpha(1-c(z))\right) B_m^*(\alpha(1-c(z)))\right]\left[1-A^*(\alpha)\right]$$

The probability generation function of the orbit size when the server is busy is

$$R^E(z) = \int_0^\infty R^E(x,z)dx. R^E(z) \frac{R_0^F[(A^*(\alpha))]}{D(z)}\left[1-B_1^*(\alpha(1-c(z))\right]$$

The probability generation function of the orbit size when the server in multi-service is

$$R_m^Q(z) = \int_0^\infty R_m^Q(x,z)dx.$$

$$R_m^Q(z) = \frac{\sum_{m=1}^{k} \sigma_m R_0^F A^*(\alpha) B_1^*(\alpha(1-c(z)))}{D(z)}[1-B_m(x)(1-c(z))]$$

Using the normalising condition $R_0^F + R^F(1) + R^E(1) + \sum_{m=1}^{k} R_M^Q(1) = 1,$ we get

$$R_0^F(z) = \frac{D'(1)}{\sum_{m=1}^{k} \sigma_m A^*(\alpha)\alpha \bar{c}_1 \mu_1 - A^*(\alpha)\left[1+\alpha \bar{c}_1 \mu_1\right]}$$

4 Performance measures

In this section, we will outline the process of deriving performance measures for the proposed model.

- The steady state probability that the server is idle during the retrial time is

$$R^F(1) = \frac{R_0^F\left[1-\sum_{m=1}^{k}\sigma_m\left[\alpha\overline{c}_1\mu_2+\alpha\overline{c}_1\mu_1\right]-\sum_{m=1}^{k}\sigma_m\overline{c}\right][1-A^*(\alpha)]}{D'(1)}$$

- The steady state probability that the server is busy in the first phase service is

$$R^E(1) = \frac{R_0^F A^*(\alpha)\alpha\overline{c}_1\mu_1}{D'(1)}$$

- The steady state probability that the server is busy in second phase of multi-service is

$$R_m^Q(1) = \frac{R_0^F\sum_{m=1}^{k}\sigma_m A^*(\alpha)\alpha\overline{c}_1\mu_2}{D'(1)}$$

- The PGF of the number of customers in the queue is given by:

$$\begin{aligned} P_q(z) &= R_0^F + R^F(z) + R^E(z) + R_m^Q(z) \\ &= \frac{R_0^F}{D(z)}\left[A^*(\alpha)\left(B_1^*(\alpha(1-c(z))\right)+\sum_{m=1}^{k}\sigma_m A^*(\alpha)B_1^*\left(\alpha(1-c(z))\right)-zA^*(\alpha)\right] \end{aligned} \quad (22)$$

$$N(Z) == R_0^F\left[A^*(\alpha)\left(B_1^*\left(\alpha(1-c(z))\right)\right)+\sum_{m=1}^{k}\sigma_m A^*(\alpha)B_1^*(\alpha(1-c(z)))-zA^*(\alpha)\right]$$

- The mean number of customers in the queue:

$$L_q = \frac{D'N'' - N''D''}{2D'^2}$$

where

$$D' = \sum_{m=1}^{k}\sigma_m\left(\overline{c}_1(1-A^*(\alpha)\right)+\sum_{m=1}^{k}\sigma_m\left(\alpha\overline{c}_1\mu_2+\alpha\overline{c}_1\mu_1\right)-1$$

$$D'' = \sum_{m=1}^{k}\sigma_m\left(\alpha\overline{c}_1\mu_2+\alpha^2\overline{c}_1^2\mu_2+\alpha\overline{c}_1\mu_1+\alpha^2\overline{c}_1^2\mu_1\right)\left(\overline{c}_2(1-A^*(\alpha))\right)$$

$$N' = R_0^F\left[\sum_{m=1}^{k}\sigma_m A^*(\alpha)\alpha\overline{c}_1\mu_1 - A^*(\alpha)\alpha\overline{c}_1\mu_1\right]$$

$$N'' = R_0^F\left[\sum_{m=1}^{k}\sigma_m A^*(\alpha)\alpha\overline{c}_1\mu_1+\alpha^2\overline{c}_1^2\mu_1 - A^*(\alpha)\alpha\overline{c}_1\mu_1+\alpha^2\overline{c}_1^2\mu_1\right]$$

- The PGF of the number of customers in the system is given by

$$P_s(z) = R_0^F + R^F(z) + zR^E(z) + zR_m^Q(z)$$

$$T(z)=\frac{R_0^F}{D(z)}\left[\begin{array}{l}\left(\begin{array}{l}\sum_{m=1}^{k}\sigma_m B_1^*(\alpha(1-c(z)))B_m^*(\alpha(1-c(z)))A^*(\alpha)\\ -zA^*(\alpha)B_1^*(\alpha(1-c(z)))\end{array}\right)\\ +\left(\sum_{m=1}^{k}\sigma_m A^*(\alpha)B_1^*(\alpha(1-c(z)))[1-B_m^*(\alpha(1-c(z)))]\right)\end{array}\right] \tag{23}$$

- The mean number of customers in the system

  $$L_s=\frac{D'T''-T'D''}{2D'^2}$$

  where

  $$T'=R_0^F\left[\sum_{m=1}^{k}\sigma_m A^*(\alpha)\alpha\bar{c}_1\mu_1-A^*(\alpha)\alpha\bar{c}_1\mu_1-A^*(\alpha)\right]$$

The busy period is a crucial performance measure in retrial situations. The system's busy period, denoted as B, is defined as the duration starting from an epoch when an arriving customer finds an empty system and ending at the next departure epoch when the system is empty again.

$$E(B)=\frac{1}{\alpha}\left(R_0^{F^{-1}}-1\right)$$

$$=\frac{\left[\sum_{m=1}^{k}\sigma_m A^*(\alpha)\alpha\bar{c}_1\mu_1-A^*(\alpha)[1+\alpha\bar{c}_1\mu_1]-\sum_{m=1}^{k}\sigma_m\left(\bar{c}(1-A^*(\alpha)\right)+\sum_{m=1}^{k}\sigma_m\left(\alpha\bar{c}_1\mu_2+\alpha B_1\lambda_{11}\right)-1\right]}{\alpha\sum_{m=1}^{k}\sigma_m\left(\bar{c}(1-A^*(\alpha)\right)+\sum_{m=1}^{k}\sigma_m\left(\alpha\bar{c}_1\mu_2+\alpha\bar{c}_1\mu_1\right)-1}$$

4.1 Special case

If $C(z) = z$, and arrivals occur one by one according to a Poisson process with rate λ, then the model simplifies to a single-server general-service retrial queuing system with a mandatory two-phase service model. (7)

5 Numerical illustration

In this section, we present numerical examples aimed at studying the impact of various parameters on the primary system performance. The system parameters include the arrival rate (α), the retrial rate (τ), the rate of the first-phase mandatory service (ε_1), the probability of a customer opting for multi-phase mandatory service (σ_m), and the batch size, which follows a geometric distribution with a mean of $1/\mu$, where $\mu \in$ (0, 1]. We analyse the following performance measures through these numerical examples.

- L_s expected number of customers in the system.
- L_q expected number of customers in the queue.

- $E(B)$ expected number of busy period

Table 1 illustrates the impact of the retrial rate and the rate of the second phase of multi-service on various parameters. As the rate of the second phase of multi-service increases, both the system length and the queue size also increase.

- Figure 2 demonstrates that as the retrial rate increases, the system size also increases.
- Figure 3 shows that as the arrival rate increases, the queue size also increases.
- Figure 4 indicates that as the arrival rate increases, the system size also increases.
- Figure 5 highlights that as the arrival rate increases, the busy period of the system decreases.

Table 1 Values of L_s and L_q by varying σ

	$\sigma = 0.1$		$\sigma = 0.2$		$\sigma = 0.3$	
τ	*Lq*	*Ls*	*Lq*	*Ls*	*Lq*	*Ls*
1.1	0.8459	1.3060	0.01266	0.0224	0.4864	0.4563
1.2	0.8455	1.5327	0.01380	0.0246	0.4865	0.5595
1.3	0.8451	1.7232	0.0149	0.0268	0.4866	0.6463
1.4	0.8446	1.8853	0.0161	0.0291	0.4867	0.7203
1.5	0.8442	2.0248	0.0173	0.0313	0.4868	0.7841
1.6	0.8438	2.1457	0.0184	0.0336	0.4869	0.8395
1.7	0.8434	2.2515	0.0195	0.0359	0.4870	0.8880
1.8	0.8430	2.3447	0.0207	0.0382	0.4871	0.9308
1.9	0.8426	2.4273	0.0219	0.0405	0.4872	0.9688
2	0.8422	2.5007	0.0231	0.0429	0.4873	1.0027

Figure 2 Effect of ε_2 and τ on L_s (see online version for colours)

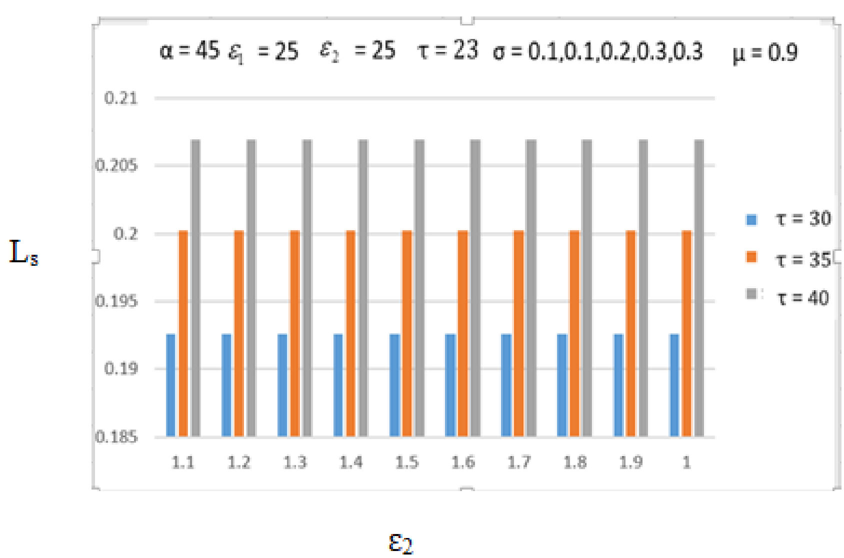


Figure 3 Effect of ε_1 and α on L_q (see online version for colours)

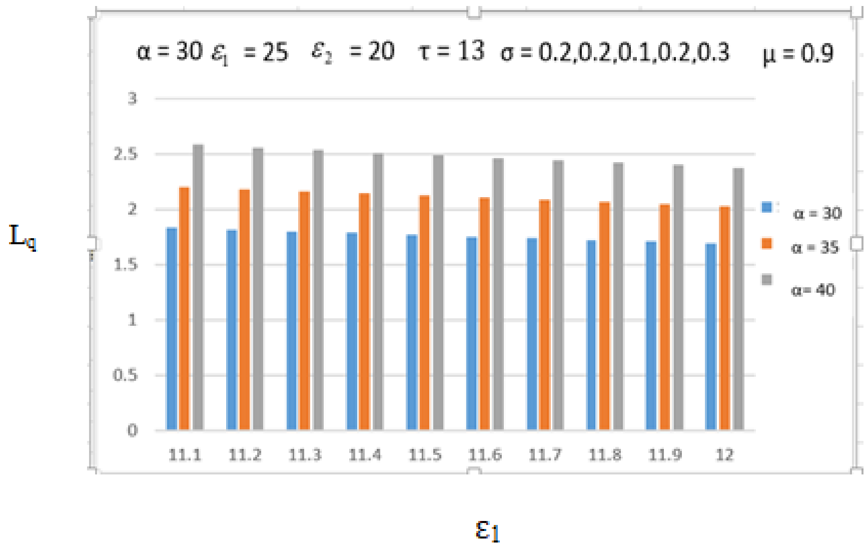


Figure 4 Effect of σ and α on L_s (see online version for colours)

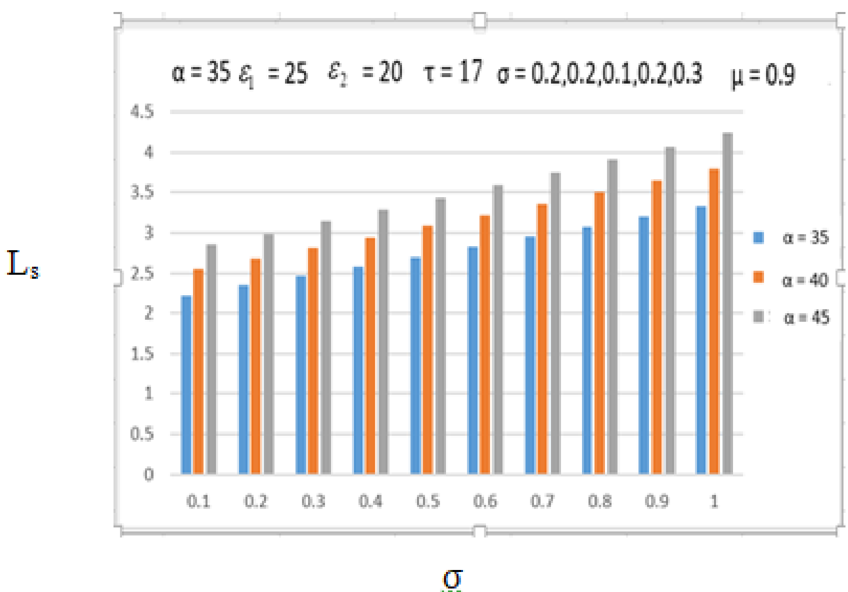
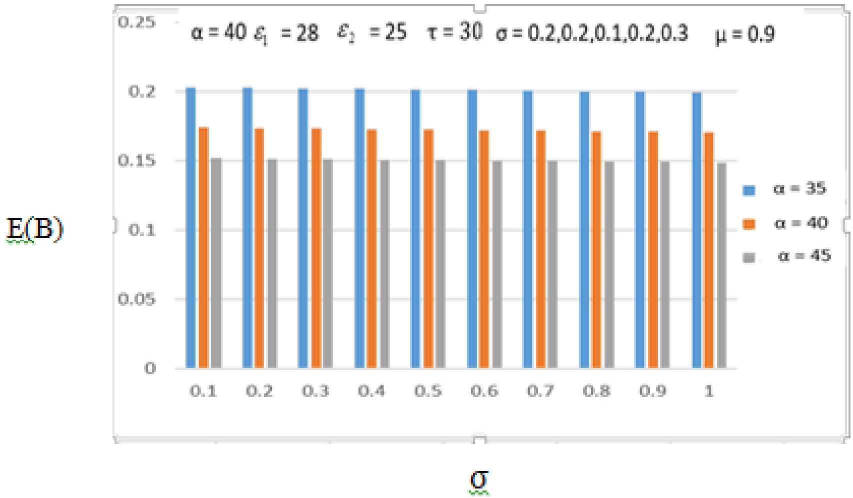


Figure 5 Effect of σ and α on E(b) (see online version for colours)



6 Cost analysis

We formulate an expected cost function per unit for the Markovian Queue with a two-phase multi-service retrial queue. Our objective is to determine the optimal values for the service rates (ε_1, ε_m), denoted as (ε_1^*, ε_m^*), that minimise the proposed cost function.

Let us define the cost elements as follows:

Ch Holding cost per unit time for each customer present in the system.

Pe Probability that the server is idle during retrial time.

Pf Probability that the server is busy in first phase of service.

Pr Probability of that the server is busy in second phase of multi – service.

Pb Probability of busy period.

Utilising the cost element defined above, we construct the cost function as follows:

$$Tc(\varepsilon_1, \varepsilon_m) = C_h L_s + P_e R^F(1) + P_f R^E(1) + P_r R_m Q(1) + P_b E(B).$$

The cost minimisation problem of the proposed model can be mathematically described as an unconstrained problem as follows:

$$Tc(\varepsilon_1^*, \varepsilon_m^*) = \min \; Tc(\varepsilon_1, \; \varepsilon_m)$$

6.1 Optimal analysis

The optimisation problem in service systems is commonly nonlinear and involves various intricate constraints. Even for a single objective, multiple cost factors often conflict, complicating the search for the optimal solution. Classical optimisation techniques may not be suitable for tackling such complexities. An alternative efficient algorithm is required to find a near-optimal solution for such problems. In this paper, the particle swarm algorithm is employed to discover the optimal solution for our model.

6.2 Particle swarm optimisation

In this section, we employ the PSO algorithm to address the formulated cost optimisation problem. The PSO algorithm is a stochastic optimisation technique based on swarm behaviour observed in nature, including insects, birds, and fish. Each particle in the PSO algorithm navigates the search space with its own position and velocity. Particle movement is influenced by both its local best (p-best) and the global best (g-best) positions within the feasible solution space.

Particle swarm optimisation: Pseudo code	
Input	ε_1, ε_m service rate, learning parameter
Output	Approximate the solution vector $[\varepsilon_1^*, \varepsilon_m^*]$ and compute the values of cost function $Tc[\varepsilon_1, \varepsilon_m]$
Step 1	Initialisation: Find location ε_m of m particle.
Step 2	Find G^* (g-best) from (min) form $[T_c(\varepsilon_1), Tc(\varepsilon_2) \dots Tc(\varepsilon_m)]$
Step 3	while (t < max generation) or (stop criterion)
	for loop over all n particle and all dimension.
Step 4	Find new location for i^{th} particle $\varepsilon_1^t + \varepsilon_m^{t+1}$
Step 5	Evaluate objective function at new location ε_m^{t+1}
Step 6	Find the current best (p-best) for each particle ε end for
Step 7	update global for G^*
	t–t + 1
	end while
Step 8	output find results ε_1^*, ε_m^*

6.3 Numerical analysis

This section presents a numerical analysis of the parameter optimisation for the queueing system under consideration.

We consider the parameters $C_h = 8$, $P_e = 9$, $P_f = 4$, $P_r = 9$, $P_b = 8$

Table 2 illustrates the diverse values of system parameters used in the proposed algorithm. In Table 2, we calculate the total expected cost of the first phase of service and the multi-phase service for the RabbitMQ model. As the number of messages requiring service in the two phases increases, the total expected cost also increases.

Figure 6 Total excepted cost with parameters ε_1, and ε_m (see online version for colours)

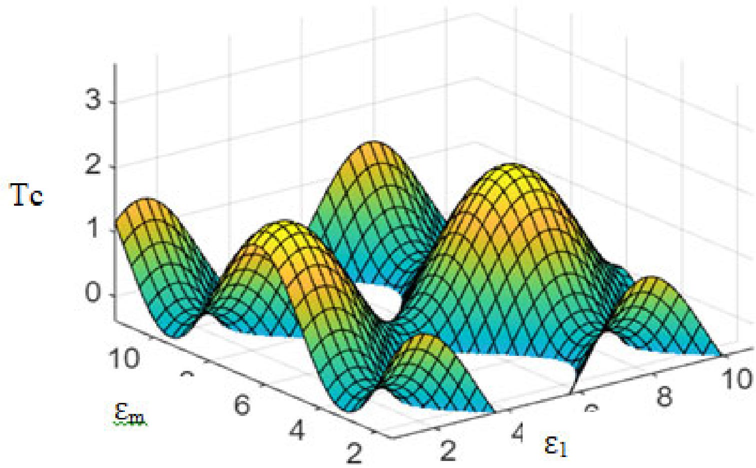


Figure 7 Total excepted cost with parameters ε_m and τ (see online version for colours)

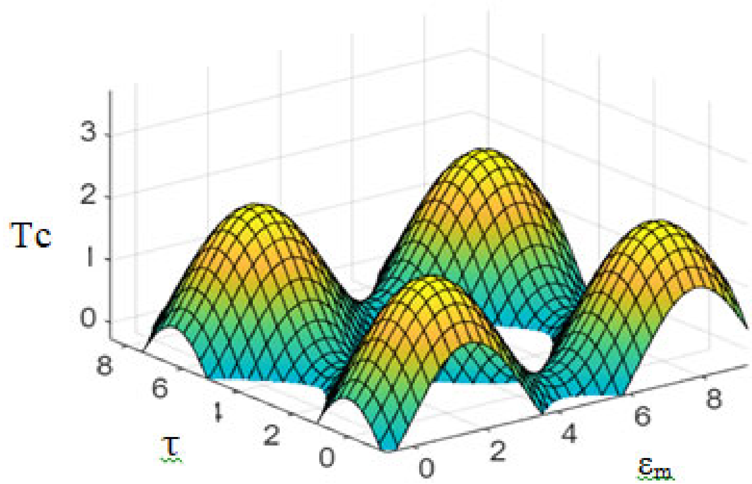


Table 2 Optimal value of ε_1^* and ε_m^* with minimal excepted cost T_c

$(\varepsilon_1, \varepsilon_m)$	ε_1^*	ε_m^*	T_c
(10, 15)	6.9159	3.8406	5880.71
(10, 20)	7.1829	3.9913	6053.99
(10, 25)	7.4180	4.306	6179.28
(15, 15)	7.77542	5.8253	6114.91
(20, 15)	6.3465	7.9006	6308.64
(25, 15)	6.0965	8.3681	6464.25

Figure 8 Total excepted cost with parameters α and τ (see online version for colours)

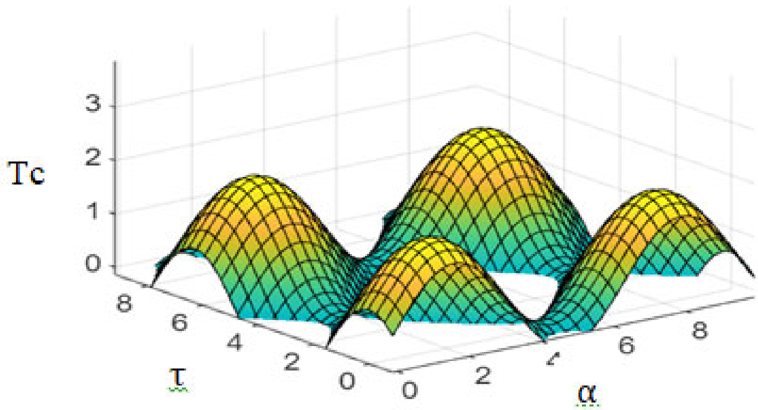
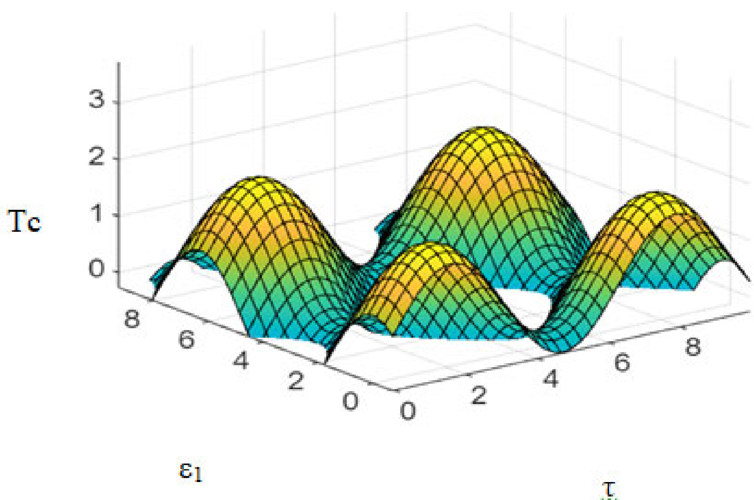


Figure 9 Total excepted cost with parameters τ and ε_1 (see online version for colours)



7 Conclusions

This paper delves into the utilisation of a bulk arrival two-phase retrial multi-service queue using the RabbitMQ model presents a promising approach to efficiently manage and handle complex queuing scenarios in distributed systems. This model offers the advantage of handling various types of services simultaneously while accommodating retrials, making it suitable for real-world applications with diverse service requirements and customer behaviours. Its scalability and fault-tolerant features make it a valuable tool for optimising resource allocation and improving system performance. As we continue to explore innovative queuing mechanisms in the realm of distributed computing, the bulk arrival two-phase retrial multi-service queue with RabbitMQ stands as a robust solution that can enhance the reliability and responsiveness of modern service-oriented. The PGF of the customer count within the system is derived through the supplementary variable technique. The study includes an analysis of performance metrics and special cases. The behaviour of RabbitMQ is explored using numerical analysis. Additionally, a cost model is developed employing the particle swarm algorithm to ascertain total cost optimisation for service rates within the RabbitMQ model. In future work, a multi-server setup with two phases and server breakdowns will be tackled.

References

Abdollahi, S. and Salehi Rad, M.R. (2022) 'Analysis of a batch arrival retrial queue with two-phase services, feedback and admission', *Bulletin of the Iranian Mathematical Society*, Vol. 48, No. 3, pp.791–804, https://link.springer.com/article/10.1007/s41980-021-00547-1.

Arivudainambi, D. and Godhandaraman, P. (2012) 'A batch arrival retrial queue with two phases of service, feedback and K optional vacations', *Applied Mathematical Sciences*, Vol. 6, No. 22, pp.1071–1087.

Ayyappan, G. and Nirmala, M. (2020) 'An M [X]/G (a, b)/1 queue with unreliable server, second optional service, closedown, setup with N-policy and multiple vacation', *International Journal of Mathematics in Operational Research*, Vol. 16, No. 1, pp.53–81, http://dx.doi.org/10.1504/IJMOR.2020.104686.

Bagyam, E.A.J., Vivesini, D. and Preethi, S.K. (2023) 'Adaptive power quality for power management units using smart technologies', in Arti Vaish, Pankaj Kumar Goswami, Surbhi Bhatia and Mokhtar Shouran (Eds.): Chapter 11, CRC Press, https://doi.org/10.1201/9781003436461.

Bagyam, J.E.A., Chandrika, K.U. and Viswanathan, A. (2015) 'Bulk arrival multi-stage retrial queue with vacation', *Applied Mathematical Sciences*, Vol. 9, No. 54, pp.2691–2706, DOI: 10.12988/ams.2015.4121005.

Begum, A. and Choudhury, G. (2022) 'Analysis of a bulk arrival N-policy queue with two-service genre, breakdown, delayed repair under Bernoulli vacation and repeated service policy', *RAIRO-Operations Research*, Vol. 56, No. 2, pp.979–1012, DOI: 10.1051/ro/2021167.

Bouchentouf, A.A. and Guendouzi, A. (2019) 'Cost optimization analysis for an M^ X/M/c MX/M/c vacation queueing system with waiting servers and impatient customers', *SeMA Journal*, Vol. 76, pp.309–341, https://doi.org/10.1007/s40324-018-0180-2.

Choudhury, G., Tadj, L. and Deka, K. (2010) 'A batch arrival retrial queueing system with two phases of service and service interruption', *Computers and Mathematics with Applications*, Vol. 59, No. 1, pp.437–450, DOI: 10.1016/j.camwa.2009.06.021.

Ghimire, S., Ghimire, R.P. and Thapa, G.B. (2014) 'Mathematical models of Mb/M/1 bulk arrival queueing system', *Journal of the Institute of Engineering*, Vol. 10, No. 1, pp.184–191, https://doi.org/10.3126/jie.v10i1.10899.

Jain, M., Sharma, G.C. and Sharma, R. (2012) 'A batch arrival retrial queuing system for essential and optional services with server breakdown and Bernoulli vacation', *International Journal of Internet and Enterprise Management,* Vol. 8, No. 1, pp.16–45, DOI: 10.1504/IJIEM.2012.049868.

Jain, M., Sharma, G.C. and Sharma, R. (2013) 'Unreliable server M/G/1 queue with multi-optional services and multi-optional vacations', *International Journal of Mathematics in Operational Research*, Vol. 5, No. 2, pp.145–169, https://doi.org/10.1504/IJMOR.2013.052458.

Kumar, M.S. and Arumuganathan, R. (2013) 'An MX/G/1 unreliable retrial queue with two phase service and persistence behaviour of customers in service', *RAIRO-Operations Research*, Vol. 47, No. 1, pp.9–32, https://doi.org/10.1051/ro/2012024.

Lazidis, A., Tsakos, K. and Petrakis, E.G. (2022) 'Publish–Subscribe approaches for the IoT and the cloud: Functional and performance evaluation of open-source systems', *Internet of Things*, Vol. 19, p.100538, DOI: 10.1186/s13677-022-00278-6.

Li, Z. (2022) 'Intelligent electrical engineering automation on account of particle swarm optimization (PSO) algorithm', in *Journal of Physics: Conference Series*, October, Vol. 2355, No. 1, p.12013, IOP Publishing, DOI: 10.1088/1742-6596/2355/1/012013.

Maragathasundari, S., Manikandan, P. and Geetha, R. (2022) 'Solving of queuing issue in a WhatsApp server', in *AIP Conference Proceedings*, Vol. 2516, No. 1, p.360014, AIP Publishing LLC, https://doi.org/10.1063/5.0108861.

Martinez-Rico, J., Zulueta, E., de Argandoña, I.R., Fernandez-Gamiz, U. and Armendia, M. (2020) 'Multi-objective optimization of production scheduling using particle swarm optimization algorithm for hybrid renewable power plants with battery energy storage system', *Journal of Modern Power Systems and Clean Energy*, Vol. 9, No. 2, pp.285–294, DOI: 10.35833/MPCE.2019.000021.

Micheal, M.N.G. and Indhira, K. (2023) 'A literature review on retrial queueing system with Bernoulli vacation', *Yugoslav Journal of Operations Research*, p.20 [online] https://doiserbia.nb.rs/img/doi/0354-0243/2023%20OnLine-First/0354-02432300020M.pdf.

Nila, M. and Sumitha, D. (2022) *Two Phase Retrial Queue with Starting Failures, Customer Impatience, Feedback and Bernoulli Vacation*.

Piórkowski, A. and Werewka, J. (2010) 'Minimization of the total completion time for asynchronous transmission in a packet data-transmission system', *Int. J. Appl. Math. Comput. Sci.*, Vol. 20, No. 2, pp.391–400, DOI: 10.2478/v10006-010-0029-z.

Rajadurai, P., Chandrasekaran, V.M. and Saravanarajan, M.C. (2015) 'Steady state analysis of batch arrival feedback retrial queue with two phases of service, negative customers, Bernoulli vacation and server breakdown', *International Journal of Mathematics in Operational Research*, Vol. 7, No. 5, pp.519–546, DOI: 10.1504/IJMOR.2015.071276.

Rao, N.T., Srinivas, P., Rajkumar, C., Bhattacharyya, D. and Kim, H.J. (2017) 'Studies on performance analysis of cloud computing based data centers with queuing models using MATLAB', *International Journal of Grid and Distributed Computing*, Vol. 10, No. 6, pp.55–69, http://dx.doi.org/10.14257/ijgdc.2017.10.6.05.

Roy, C. and Das, D.K. (2021) 'A hybrid genetic algorithm (GA)–particle swarm optimization (PSO) algorithm for demand side management in smart grid considering wind power for cost optimization', *Sādhanā*, Vol. 46, No. 2, p.101, https://www.ias.ac.in/article/fulltext/sadh/046/0101.

Sachs, K., Kounev, S. and Buchmann, A. (2013) 'Performance modeling and analysis of message-oriented event-driven systems', *Software and Systems Modeling*, Vol. 12, No. 4, pp.705–729.

Sadooghi, I., Palur, S., Anthony, A., Kapur, I., Belagodu, K., Purandare, P. and Raicu, I. (2014) 'Achieving efficient distributed scheduling with message queues in the cloud for many-task computing and high-performance computing', in *2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp.404–413, IEEE, DOI: 10.1109/CCGrid.2014.30.

Sanga, S.S. and Charan, G.S. (2023) 'Fuzzy modeling and cost optimization for machine repair problem with retrial under admission control F-policy and feedback', *Mathematics and Computers in Simulation*, Vol. 211, pp.214–240, DOI: 10.1520/JTE20180882.

Sangeetha, N. and Chandrika, K.U. (2022) 'MX/G/1 retrial G-queue with multistage and multi-optional services, feedback, randomized J vacation and orbital search', in *AIP Conference Proceedings*, Vol. 2516, No. 1, p.100003, AIP Publishing LLC [online] https://aip.scitation.org/doi/abs/10.1063/5.0109259.

Shekhar, C., Varshney, S. and Kumar, A. (2020) 'Optimal control of a service system with emergency vacation using bat algorithm', *Journal of Computational and Applied Mathematics*, Vol. 364, p.112332, https://doi.org/10.1016/j.cam.2019.06.048.

Singh, C.J., Jain, M. and Kumar, B. (2011) 'Queueing model with state-dependent bulk arrival and second optional service', *International Journal of Mathematics in Operational Research*, Vol. 3, No. 3, pp.322–340, DOI: 10.1504/IJMOR.2011.040029.

Tang, S. (2022) 'Transient analysis of a finite queueing system with bulk arrivals in IoT-based edge computing systems', *IoT*, Vol. 3, No. 4, pp.435–449, https://doi.org/10.3390/iot3040023.

Upadhyaya, S. (2015) 'Admission control of bulk retrial feedback queue with K-optional vacations', *International Journal of Mathematics in Operational Research*, Vol. 7, No. 2, pp.215–239, https://doi.org/10.1504/IJMOR.2015.068293,

Upadhyaya, S., Agarwal, D., Agarwal, R., Ahmad, I., Ibrahim, A.H. and Al-Homidan, S. (2023) 'Queueing and reliability analysis of unreliable multi-server retrial queue with Bernoulli feedback', *Carpathian Journal of Mathematics*, Vol. 39, No. 3, pp.745–758, DOI: 10.37193/CJM.2023.03.13.

Vaquero, L.M., Rodero-Merino, L., Caceres, J. and Lindner, M. (2008) 'A break in the clouds: towards a cloud definition', *ACM Sigcomm Computer Communication Review*, Vol. 39, No. 1, pp.50–55, https://Doi.Org/10.1145/1496091.149610.

Varalakshmi, M., Rajadurai, P., Saravanarajan, M.C. and Chandrasekaran, V.M. (2016) 'An M/G/1 retrial queueing system with two phases of service, immediate Bernoulli feedbacks, single vacation and starting failures', *International Journal of Mathematics in Operational Research*, Vol. 9, No. 3, pp.302–328, DOI: 10.1504/IJMOR.2016.10000127.

Vilaplana, J., Solsona, F., Teixidó, I., Mateo, J., Abella, F. and Rius, J. (2014) 'A queuing theory model for cloud computing', *The Journal of Supercomputing*, Vol. 69, No. 1, pp.492–507, https://Doi.Org/10.1007/S11227-014-1177-Y.

Wang, J. and Zhou, P.F. (2010) 'A batch arrival retrial queue with starting failures, feedback and admission control', *J. Syst. Sci. Syst. Eng.*, Vol. 19, pp.306–320, https://doi.org/10.1007/s11518-s010-5140-z.

Yang, D.Y. and Chen, Y.H. (2018) 'Computation and optimization of a working breakdown queue with second optional service', *Journal of Industrial and Production Engineering*, Vol. 35, No. 3, pp.181–188, https://doi.org/10.1080/21681015.2018.1439113.